Distribution Deviation-Aware Split Federated Learning in Resource-Limited Wireless Networks

Chunfeng Xie, Zhixiong Chen, *Member, IEEE*, Wenqiang Yi, *Member, IEEE*, Hyundong Shin, *Fellow, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—The escalating complexity of deep neural networks introduces substantial challenges to deploying federated learning (FL) in resource-limited edge environments. To address these limitations, split federated learning (SFL) has emerged as a promising paradigm, alleviating client-side computational and communication burdens via strategic model splitting, and periodically aggregating client-side and server-side models consistent with the principles of FL. Nevertheless, existing SFL frameworks encounter significant performance degradation arising from data heterogeneity and imbalance, client heterogeneity, as well as constrained wireless resources. To overcome these issues, this paper introduces a novel data distribution deviation-aware split federated learning (DA-SFL) framework. DA-SFL dynamically adjusts aggregation weights according to the deviation of clients' data distributions from a global distribution, effectively mitigating biases induced by data imbalance and heterogeneity. Furthermore, we theoretically establish the convergence bound of DA-SFL under a non-convex loss function setting, demonstrating that minimizing the data deviation in each training round enhances learning efficacy. Motivated by this, we formulate a mixed-integer nonlinear programming to optimize learning performance under long-term energy constraints. Leveraging the Lyapunov optimization framework, we decompose the problem into a series of tractable subproblems in each learning round, and propose efficient algorithms to find the client scheduling, adaptive cut layer selection, bandwidth allocation, and aggregation weighting policies. Extensive experimental evaluations conducted on Fashion-MNIST, CIFAR-10, and CINIC-10 datasets across diverse scenarios of data heterogeneity and imbalance demonstrate that DA-SFL significantly outperforms baselines regarding test accuracy, time and energy efficiency, while exhibiting notable robustness and scalability.**

***Index Terms*—Split federated learning, client scheduling, resource allocation, data deviation, data heterogeneity.**

I. Introduction

The rapid proliferation of smartphones, wearables, and connected vehicles has deployed large numbers of edge clients in wireless networks [2], producing unprecedented volumes of edge data [3]. To exploit this data, on-device learning has become essential. Federated learning (FL) [4] enables privacy-preserving training by allowing clients to update local models on private data and upload model updates for server-side aggregation. However, the growing complexity of deep neural networks [5] hinders deployment in FL. For example, the on-device LLM Gemini Nano-2 [6] contains 3.25 billion parameters, which is about 3 GB in 32-bit floats. The resulting computation load and communication overhead for transmitting full models motivate alternatives to standard FL.

Part of this work was presented at the IEEE International Conference on Communication (ICC), 2025 [1].

Split federated learning (SFL) [7], inspired by split learning (SL) [8], partitions training between clients and the edge server, thereby reducing client-side computation and communication, and the client and server models are periodically aggregated in accordance with FL principles [9]. Despite these benefits, SFL in wireless networks faces three key challenges: 1) *Data Heterogeneity and Imbalance*: client data are typically private and label-biased, and the global data is usually imbalanced, leading to non-IID distributions, biased global updates, and degraded performance [10]. 2) *Client Heterogeneity*: devices vary widely in computation and communication capability, creating stragglers and uneven round times [11]. 3) *Wireless Resource Constraints*: limited spectrum restricts client participation, which can bias global models and slow convergence [12]. Addressing these bottlenecks requires robust and scalable solutions within the SFL framework.

A. Related Works

Existing research addressing data heterogeneity primarily concentrates on model adjustment [13]–[16] and client sampling [17], [18]. In the context of model adjustment, a significant portion of studies focuses on local model adjustment. Specifically, FedProx [13] introduces a proximal regularization term penalizing deviations of client weights from the global model, thus enhancing stability of FedAvg in highly non-IID settings. A dynamic linear regularizer is proposed in [14], adjusting each round based on optimality conditions to guide local objectives toward the global optimum without a fixed penalty parameter. In [15], the authors quantify the discrepancy between local and global data and employ this static measure to determine the aggregation strategy throughout training. Global model adjustment approach is explored in [16], the authors propose FedFTG to enable data-free knowledge distillation to mitigate non-IID drift while preserving privacy and communication efficiency. Client sampling methods, such as the probabilistic client sampling algorithm introduced in [17], select participant subsets to minimize global class imbalance. Additionally, data skewness, a critical metric influencing learning outcomes, is leveraged in [18] to develop globally class-balanced client scheduling strategies. Nevertheless, existing studies seldom explore adaptive aggregation weight adjustments based on data distribution characteristics, instead relying solely on dataset size or static weighting schemes.

To facilitate generalized global model training while enabling heterogeneous client-side model adaptations aligned

with varying computational and communication capabilities, model splitting approaches based on SL and SFL have been proposed. Existing split-based inference methods primarily fall into static [19]–[21] and dynamic categories [22]. For instance, the work in [19] selects partition layers dynamically at runtime based on per-layer latency profiling and available bandwidth, integrating early-exit strategies to meet latency and accuracy constraints. In [20], a two-timescale optimization strategy jointly selects cut layers, forms clusters, and allocates radio spectrum, addressing latency minimization amidst device heterogeneity and fluctuating wireless channels. Furthermore, probabilistic methods for optimal cut layer selection within the SFL paradigm are proposed in [21]. Additionally, the work in [22] employs Lyapunov optimization to dynamically manage model segment caching and inference splitting, minimizing long-term latency and transmission costs. Despite these contributions, personalized and adaptive cut layer selections for each round remain underexplored, particularly given the highly variable nature of wireless channel conditions and resource availability in practical scenarios.

To mitigate wireless resource limitations, extensive literature emphasizes client selection [23]–[25] and resource optimization [26], [27]. For instance, an integrated approach coupling mini-batch size, power control, and aggregation frequency to jointly minimize wireless energy consumption and training loss is proposed in [23]. Additionally, a threshold-based policy dependent on queue backlog and channel states is introduced in [24], providing long-term optimal solutions for client selection and bandwidth allocation. An energy-aware activation strategy for over-the-air FL, balancing convergence speed with battery conservation in fading channels, is presented in [25]. Further resource management advancements include joint bandwidth and CPU/GPU workload allocation strategies proposed in [26], aimed at minimizing device-side energy while adhering to round-time limitations. Closed-form scaling laws relating transmit power and aggregation intervals to minimize energy-per-accuracy drop are developed in [27]. Although a variety of resource allocation and client scheduling schemes have been introduced, an integrated optimization of communication and computation in SFL remains unexplored.

B. Motivations and Contributions

While existing approaches in [13]–[18] have effectively mitigated performance degradation caused by data heterogeneity, relatively few studies have explored aggregation policies in SFL specifically informed by class distribution characteristics. Furthermore, current model splitting methods [19]–[22] inadequately address the dynamic adaptation required by time-varying channel conditions and extreme client heterogeneity. In addition, existing client selection schemes and resource optimization strategies [23]–[27] rely on naive averaging of local updates for global aggregation and thus fail to accommodate the pronounced data heterogeneity intrinsic to SFL. To address these limitations, this work jointly optimizes learning mechanisms and wireless network resources to enhance SFL performance. Specifically, we propose a novel data distribution deviation-aware split federated learning (DA-SFL) framework, which dynamically adjusts client aggregation weights based on the deviation level of local data distributions, thereby mitigating global biases induced by data heterogeneity and imbalance. To counteract negative impacts stemming from client heterogeneity and constrained resources, we propose an integrated strategy encompassing client scheduling, adaptive cut layer selection, and bandwidth allocation, significantly enhancing SFL performance in practical wireless environments. The primary contributions of this work are as follows:

- We introduce DA-SFL to mitigate data heterogeneity and imbalance in SFL by computing aggregation weights dynamically from per-round client data distributions. By up-weighting clients with lower data deviation, DA-SFL improves performance across different deviation levels. To our knowledge, it is the first SFL framework that explicitly addresses data heterogeneity and imbalance through data distribution deviation-aware weighting.
- To optimize SFL under wireless resource constraints, we formulate global loss minimization subject to long-term energy budgets and bandwidth limits. Because the global loss is implicit and difficult to minimize directly, we analyze the convergence of DA-SFL for nonconvex losses and introduce a metric that quantifies the deviation level of scheduled clients' data. Minimizing this metric promotes reduction of the global loss.
- We formulate deviation-level minimization as a unified optimization over client scheduling, cut layer selection, bandwidth allocation, and weighting. The ensuing mixed-integer nonlinear programme is tackled with Lyapunov optimization, which converts the long-term stochastic problem into a set of deterministic subproblems. Base on the reformulation, we devise an adaptive scheme for cut layer selection and bandwidth allocation, a set expansion strategy for client scheduling, and a deviation-aware algorithm for determining aggregation weights.
- Experiments show that DA-SFL achieves higher test accuracy than representative baselines across a range of data heterogeneity and imbalance settings. The joint optimization further reduces energy consumption and training time relative to existing methods. Additional simulations across data distributions, discrepancy metrics, and client scales confirm the robustness and scalability of DA-SFL.

C. Organizations

The paper proceeds as follows. Section II introduces the DA-SFL system model and formulates the problem. Section III presents the convergence analysis and recasts the problem. Section IV details algorithms for joint client scheduling, cut layer selection, bandwidth allocation, and aggregation weight optimization. Section V reports comprehensive simulation results. Section VI concludes this work.

II. System Model and Learning Mechanism

In this section, we study a wireless SFL architecture subject to channel noise and limited resources, in which a single server coordinates multiple clients to collaboratively optimize a shared global model.

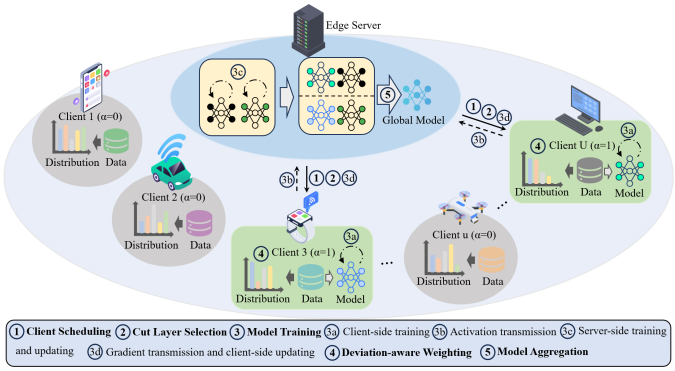


Fig. 1. Illustration of the considered DA-SFL system.

A. Split Federated Learning System

We consider an SFL system with one edge server and U clients, indexed by $\mathcal{U} = \{1, 2, \cdots, U\}$. Client $u \in \mathcal{U}$ owns a private dataset $\mathcal{D}_u$ with $D_u = |\mathcal{D}_u|$ samples. Considering generality, the local datasets are disjoint. Thus the aggregated global dataset is $\mathcal{D} = \bigcup\{\mathcal{D}_u\}_{u=1}^{U}$ with total size $D = \sum_{u=1}^{U} D_u$. Each sample $(\boldsymbol{x}, y) \in \mathcal{D}$ comprises an o-dimensional input vector $\boldsymbol{x} \in \mathbb{R}^o$ and a label $y \in \mathbb{R}$. The global model $\boldsymbol{w}$ is partitioned into a client-side model $\boldsymbol{w}_c$ and a server-side model $\boldsymbol{w}_s$. Let $f(\boldsymbol{x}, y; \boldsymbol{w})$ denote the sample-wise loss. Hence, the local loss for client u is expressed as

$$F_u(\boldsymbol{w}) = \frac{1}{D_u} \sum\nolimits_{\{\boldsymbol{x},y\}\in\mathcal{D}_u} f_u(\boldsymbol{x}, y; \boldsymbol{w}). \tag{1}$$

Consequently, the global loss function aggregating local losses across all clients is defined as $F(\boldsymbol{w}) = \sum_{u=1}^{U} p_u F_u(\boldsymbol{w})$, where $p_u \geq 0$ represents the weight assigned to client u, satisfying $\sum_{u=1}^{U} p_u = 1$. Differing notably from previous studies [26], [28], our framework seeks optimal aggregation weights, detailed in the subsequent subsection. The primary objective of the SFL system is to minimize the global loss $F(\boldsymbol{w})$ across the entire dataset $\mathcal{D}$: $\min_{\boldsymbol{w}} F(\boldsymbol{w})$.

B. SFL with Deviation-Aware Collaboration

To address data heterogeneity and imbalance, we propose DA-SFL, which departs from approaches that primarily adjust the model $\boldsymbol{w}$. DA-SFL dynamically sets aggregation weights p_u based on each client's data distribution deviation. An overview is given in Fig. 1. The iterative process comprises I global rounds. In round i ($i \in \{0, 1, \cdots, I-1\}$), the following steps are executed:

1) **Client Scheduling**: A subset of participants is scheduled by the server. Let $\alpha_{u,i} \in \{0, 1\}$ denote the round-i scheduling indicator for client u, where $\alpha_{u,i} = 1$ indicates that the client is selected to participate. The client subset is represented by $\boldsymbol{M}_i = \{u : \alpha_{u,i} = 1, \forall u \in \mathcal{U}\}$, with cardinality $M = |\boldsymbol{M}_i|$.

2) **Cut Layer Selection**: The server determines the cut layer at the beginning of each round based on the clients' and server's computational capabilities and the prevailing channel conditions. Let $\boldsymbol{s}_i = \{s_{1,i}, s_{2,i}, \cdots, s_{u,i}, \forall u \in \mathcal{U}\}$ denote the round-i cut layer decisions, where $s_{u,i} \in \mathcal{S}$ and $\mathcal{S} = \{1, 2, \cdots, S\}$ is the set of permissible split points in the model. The index $s_{u,i}$ determines the client-server partition used for training. To preserve raw-data privacy, the input layer is excluded from $\mathcal{S}$. When $s_{u,i} = S$, the server-side model is empty and the scheme degrades to conventional FL.

3) **Model Training**: The server disseminates the current client-side model $\boldsymbol{w}_{u,i}^{\mathrm{c}}$ and execute forward propagation (FP), which comprises three sequential steps: client-side computation, smashed activations transmission, and server-side computation. During the client-side phase, each scheduled client u uniformly samples a mini-batch $\mathcal{B}_{u,i,j} \subseteq \mathcal{D}_u$ of size $\beta = |\mathcal{B}_{u,i,j}|$. Let $\boldsymbol{X}_{u,i} \in \mathbb{R}^{\beta\times o}$ and $Y_{u,i} \in \mathbb{R}^{\beta}$ denote the stacked input vectors and their associated labels, respectively, where o is the dimension of the input data sample. Feeding $\boldsymbol{X}_{u,i}$ into the client-side model yields the smashed activations

$$\boldsymbol{A}_{u,i} = g(\boldsymbol{X}_{u,i}; \boldsymbol{w}_{u,i}^{\mathrm{c}}) \in \mathbb{R}^{\beta\times O}, \forall u \in \mathcal{U}, \tag{2}$$

where $g(\cdot; \boldsymbol{w}^{\mathrm{c}})$ denotes the client-side mapping parameterized by $\boldsymbol{w}^{\mathrm{c}}$, and the scalar O specifies the dimensionality of each smashed-activation vector. During the uplink phase, every client transmits its smashed data to the server, where these activations are used to drive server-side training. Upon reception, the server feeds $\boldsymbol{A}_{u,i}$ into the corresponding server-side model and executes its FP, yielding

$$\hat{\boldsymbol{y}}_{u,i} = l(\boldsymbol{A}_{u,i}; \boldsymbol{w}_{u,i}^{\mathrm{s}}) \in \mathbb{R}^{\beta\times O}, \forall u \in \mathcal{U}, \tag{3}$$

where $l(\cdot; \boldsymbol{w}^{\mathrm{s}})$ denotes the server-side mapping parameterized by $\boldsymbol{w}^{\mathrm{s}}$. The FP phase is complete. The loss is subsequently evaluated by comparing $\hat{\boldsymbol{y}}_{u,i}$ with the ground-truth labels $Y_{u,i}$. Backward propagation (BP) then updates server- and client-side parameters. The BP phase comprises three stages: updating the server-side model, transmitting the resulting cut layer gradients to the client, and updating the client-side model. In the first stage, the server performs J steps of stochastic gradient descent (SGD) on its local data, i.e.,

$$\boldsymbol{w}_{u,i,j+1}^{\mathrm{s}} = \boldsymbol{w}_{u,i,j}^{\mathrm{s}} - \eta_s \tilde{\nabla} F_u(\boldsymbol{w}_{u,i,j}^{\mathrm{s}}), \forall j = 0, \cdots, J-1, \tag{4}$$

where η_s denotes the learning rate employed at the server side. Parameters are updated layer-wise from the output layer down to the cut layer via BP (chain rule). In (4), the stochastic gradient is $\tilde{\nabla} F_u(\boldsymbol{w}_{u,i,j}^{\mathrm{s}}) = \frac{1}{\beta} \sum_{\{\boldsymbol{x},y\}\in\mathcal{B}_{u,i,j}} \nabla f_u(\boldsymbol{x}, y; \boldsymbol{w}_{u,i,j}^{\mathrm{s}})$. Second, once BP arrives at the designated cut layer, the gradient of the associated smashed activations for the current mini-batch is sent back to the originating client. Upon receipt, the client refines its local parameters according to

$$\boldsymbol{w}_{u,i,j+1}^{\mathrm{c}} = \boldsymbol{w}_{u,i,j}^{\mathrm{c}} - \eta_c \tilde{\nabla} F_u(\boldsymbol{w}_{u,i,j}^{\mathrm{c}}), \forall j = 0, \cdots, J-1, \tag{5}$$

where $\boldsymbol{w}_{u,i,j}^{\mathrm{c}}$ denotes the client u's client-side model at the j-th local SGD step in round i (initialized as $\boldsymbol{w}_{u,i,0}^{\mathrm{c}} = \boldsymbol{w}_i^{\mathrm{c}}$), and $\eta_c > 0$ is the client learning rate. The stochastic gradient in (5) is computed as $\tilde{\nabla} F_u(\boldsymbol{w}_{u,i,j}^{\mathrm{c}}) = \frac{1}{\beta} \sum_{\{\boldsymbol{x},y\}\in\mathcal{B}_{u,i,j}} \nabla f_u(\boldsymbol{x}, y; \boldsymbol{w}_{u,i,j}^{\mathrm{c}})$. After completing J local iterations, the training proceeds to next stage.

4) **Deviation-aware Weighting**: Each client computes the discrepancy between its local label distribution and an assumed global distribution. We take the global distribution to be uniform to promote class fairness and to enhance model generalization. The deviation level k_u is computed locally

and only a scalar value is reported to the server. This avoids sending raw labels or the full label distribution. However, k_u is still related to the local data distribution and may leak coarse information about the degree of imbalance. If stronger privacy is required, clients can protect k_u using differential privacy or secure aggregation for future extension. Let $\boldsymbol{q}_u$ and $\boldsymbol{G}$ denote the local and global label distributions, respectively, with $\boldsymbol{G}_c = 1/C$ for all classes c. The client's deviation level is $k_u = e(\boldsymbol{q}_u, \boldsymbol{G}) \in \mathbb{R}$, where $e(\cdot)$ is a prescribed metric function such as the L2 distance. Using the L2 distance yields $k_u = \sqrt{\sum_{c=1}^{C}(\boldsymbol{q}_{u,c} - \boldsymbol{G}_c)^2}$. In this work, we set $\boldsymbol{G}$ as the uniform distribution to promote class fairness. This makes DA-SFL give larger weights to clients whose data are more class-balanced (smaller k_u), even when the overall dataset is globally imbalanced. The server then collects $\{k_u\}$ from participating clients as illustrated in Fig. 1. Let $p_{u,i} \geq 0$ denote the aggregation weight for client u in round i, and let $\boldsymbol{p}_i = \{p_{1,i}, p_{2,i}, \cdots, p_{M,i}\}$ be the weight vector. We determine weights by jointly considering the relative dataset size $\hat{D}_u = \frac{D_u}{D}$ and the deviation level k_u

$$p_{u,i} = \frac{a k_u^{-1} + \hat{D}_u + b}{\sum_{u \in \boldsymbol{M}_i}(a k_u^{-1} + \hat{D}_u + b)}, \tag{6}$$

where a controls the trade off between $\hat{D}_u$ and k_u. A larger a gives more weight to clients with smaller k_u, while a smaller a makes the weighting closer to the standard rule dominated by $\hat{D}_u$. The parameter b is an offset that smooths the weights and prevents extreme or noisy k_u values from producing overly concentrated allocations, which improves stability across rounds. Overall, this design gives larger weights to clients with smaller deviations, helping reduce global bias under class imbalance. Although we focus on a uniform reference distribution for fairness, the same mechanism applies to any chosen target distribution that may be imbalanced.

5) **Model Aggregation**: The client-side models transmitted by the corresponding clients and, together with their server-side counterparts, forms the complete local models $\boldsymbol{w}_{u,i} = [\boldsymbol{w}_{u,i}^{\mathrm{c}}, \boldsymbol{w}_{u,i}^{\mathrm{s}}]$. The global model is then updated via a weighted aggregation of the participating clients' models

$$\boldsymbol{w}_{i+1} = \sum\nolimits_{u \in \boldsymbol{M}_i} p_{u,i} \boldsymbol{w}_{u,i}, \tag{7}$$

C. Latency Model

The complete SFL workflow spans multiple global rounds, and the communication and computation latencies in round i are detailed below.

1) **Communication Overhead**: We adopt frequency division multiple access (FDMA) over a total bandwidth B Hz for client-server uplink and downlink. We assume FDMA among scheduled clients and sufficiently accurate synchronization so that inter-user interference is negligible. This assumption keeps the per-round rate, latency, and energy models tractable, which supports the joint optimization of client scheduling, cut layer selection, and bandwidth allocation. In interference-limited networks or non-orthogonal multiple access (NOMA) settings, the achievable rate depends on the signal to interference plus noise ratio and is coupled across simultaneously transmitting clients. This coupling can increase transmission latency and energy when sending smashed activations and gradients. Recent studies on over-the-air computation and over-the-air FL have investigated communication and energy efficient learning under non-orthogonal transmission, and they also point out practical issues such as tight synchronization, channel state compensation, and aggregation distortion that can affect learning accuracy [29]–[31]. Extending DA-SFL to include interference-coupled or over-the-air communication models is an important direction for future work. The bandwidth share allocated to client u in round i is indexed by $\omega_{u,i} \in [0, 1]$ and the vector of allocations is $\boldsymbol{\omega}_i = (\omega_{1,i}, \cdots, \omega_{u,i}, \cdots, \omega_{U,i})$. For uplink transmission, let $\hat{p}_u$ denote the transmit power of client u and let $h_{u,i}$ denote the channel gain between client u and the server in round i. The achievable uplink rate is

$$r_{u,i}(\omega_{u,i}) = \omega_{u,i} B \log_2\left(1 + \frac{\hat{p}_u h_{u,i}}{\omega_{u,i} B N_0}\right), \tag{8}$$

where N_0 denotes the noise power spectral density. Let γ^c, γ^a and γ^g denote the sizes of the client-side model, the smashed activations, and the activation gradients, respectively. The upload and download latencies for the client-side model are $t_u^{\mathrm{cu}} = \frac{\gamma_u^c(s_u)}{r_u(\boldsymbol{\omega}_u)}$ and $t_u^{\mathrm{cd}} = \frac{\sum_{u \in \boldsymbol{M}_i} \gamma_u^c(s_u)}{r_s}$, where r_s is the server transmit rate. And the latencies for transmitting smashed activations and their gradients are $t_u^{\mathrm{su}} = \frac{\beta \gamma_u^a(s_u)}{r_u(\boldsymbol{\omega}_u)}$ and $t_u^{\mathrm{gd}} = \frac{\beta \sum_{u \in \boldsymbol{M}_i} \gamma_u^g(s_u)}{r_s}$. Hence, the communication latency is

$$T_u^{\mathrm{CM}} = \sum\nolimits_{j=1}^{J}(t_u^{\mathrm{su}} + t_u^{\mathrm{gd}}) + t_u^{\mathrm{cu}} + t_u^{\mathrm{cd}}, \forall u \in \mathcal{U}. \tag{9}$$

The corresponding communication energy consumption for client u is

$$E_u^{\mathrm{CM}} = \hat{p}_u(\sum\nolimits_{j=1}^{J} t_u^{\mathrm{su}} + t_u^{\mathrm{cu}}), \forall u \in \mathcal{U}. \tag{10}$$

Note that due to time varying fading, the achievable uplink rate changes across rounds. This variation directly affects the transmission time of smashed data and gradients, and therefore changes the per round latency. It also impacts communication energy, since maintaining a given latency under poorer channel conditions typically requires allocating more bandwidth and or higher transmission power.

2) **Computing Overhead**: In the FP stage, the latency of the client-side computation is $t_{u,j}^{\mathrm{ct}} = \frac{\beta \psi_u^{\mathrm{ct}}(s_u)}{f_u}$, where $\psi_u^{\mathrm{ct}}(s_u)$ denotes the per-sample computational workload of the client-side forward pass and f_u denotes client u's computational capacity, measured in floating-point operations per second (FLOPs). In addition, the latency of the server-side computation is $t_{u,j}^{\mathrm{st}} = \frac{\beta \sum_{u \in \mathcal{M}} \psi_u^{\mathrm{st}}(s_u)}{f_s}$, where $\sum_{u \in \mathcal{M}} \psi_u^{\mathrm{st}}(s_u)$ is the total server workload and f_s is the server computing capability. In the BP stage, let $\psi_u^{\mathrm{cu}}(s_u)$ and $\sum_{u \in \mathcal{M}} \psi_u^{\mathrm{su}}(s_u)$ denote the workloads of the client-side and server-side updates. Hence, the corresponding latencies are $t_{u,j}^{\mathrm{cu}} = \frac{\beta \psi_u^{\mathrm{cu}}(s_u)}{f_c}$ and $t_{u,j}^{\mathrm{cu}} = \frac{\beta \sum_{u \in \mathcal{M}} \psi_u^{\mathrm{su}}(s_u)}{f_s}$. Aggregating them, the total latency of computation in round i is

$$T_u^{\mathrm{CP}} = \sum\nolimits_{j=1}^{J}(t_{u,j}^{\mathrm{ct}} + t_{u,j}^{\mathrm{st}} + t_{u,j}^{\mathrm{su}} + t_{u,j}^{\mathrm{cu}}), \forall u \in \mathcal{U}. \tag{11}$$

The corresponding energy consumption of client u is

$$E_u^{\mathrm{CP}} = \kappa J \beta (\psi_u^{\mathrm{ct}} + \psi_u^{\mathrm{cu}}) f_u^2, \tag{12}$$

where κ is a device-dependent energy coefficient determined by the chip architecture. It worth noting that transmitting k_u adds only a scalar value per participating client, so its communication overhead is negligible compared with split learning transmissions, and computing k_u is a simple vector operation and is negligible compared with model training. Additionally, aggregation latency is negligible given its low computational cost.

D. Problem Formulation

In this subsection, we aim to improve DA-SFL by minimizing the expected global loss after I communication rounds, that is, $\mathbb{E}[F(\boldsymbol{w}_I)]$, where $\boldsymbol{w}_I$ denotes the global model at round I. We co-optimize client scheduling, cut layer selection, bandwidth allocation, and aggregation weighting under latency and energy budgets. The problem is formulated as:

$$\mathcal{P}: \quad \min_{\{\boldsymbol{s}_i, \boldsymbol{M}_i, \boldsymbol{\omega}_i, \boldsymbol{p}_i\}_{i=0}^{I-1}} \mathbb{E}[F(\boldsymbol{w}_I)] \tag{13}$$

$$\text{s. t.} \quad T_{u,i}^{\mathrm{CM}} + T_{u,i}^{\mathrm{CP}} \le T_{\max}, \forall u \in \mathcal{U}, \forall i, \tag{13a}$$

$$\sum\nolimits_{i=0}^{I-1} E_{u,i} \le E_u, \forall u \in \mathcal{U}, \forall i, \tag{13b}$$

$$\sum\nolimits_{u=1}^{U} \omega_{u,i} \le 1, \forall u \in \mathcal{U}, \forall i, \tag{13c}$$

$$0 \le \omega_{u,i} \le 1, \forall u \in \mathcal{U}, \forall i, \tag{13d}$$

$$s_{u,i} \in \mathcal{S}, \forall u \in \mathcal{U}, \forall i, \tag{13e}$$

$$\alpha_{u,i} \in \{0,1\}, \forall u \in \mathcal{U}, \forall i, \tag{13f}$$

$$\sum\nolimits_{u=1}^{U} p_{u,i} = 1, \forall u \in \mathcal{U}, \forall i, \tag{13g}$$

$$p_{u,i} \ge 0, \forall u \in \mathcal{U}, \forall i, \tag{13h}$$

where constraint (13a) enforces a per-round latency limit $T_{\max}$. Constraint (13b) imposes an energy budget, where $E_{u,i} = E_{u,i}^{\mathrm{CM}} + E_{u,i}^{\mathrm{CP}}$. Constraints (13c) and (13d) regulate the total and per-client bandwidth allocations. Constraint (13e) specifies the cut layer choice. Constraint (13f) encodes the client scheduling decision. Constraints (13g) and (13h) define the aggregation weight simplex. Problem $\mathcal{P}$ is challenging because the objective $\mathbb{E}[F(\boldsymbol{w}_I)]$ lacks a closed-form expression. As a closed-form objective is intractable, we derive a tractable upper bound and minimize it in Section III-A. Moreover, an offline optimal solution would require noncausal knowledge of channel states and client energy across all rounds, which is unrealistic under time-varying wireless conditions. Motivated by these challenges, we analyze the convergence of DA-SFL and reformulate $\mathcal{P}$ as an equivalent bound-minimization problem, as detailed in the next section.

III. Convergence Analysis and Problem Transformation

In this section, we derive a convergence bound for DA-SFL and find a deviation-level metric affects learning performance. Inspired by this, we utilize it to guide client scheduling, cut layer selection, bandwidth allocation, and aggregation weighting. The objective is then reformulated to minimize this metric, thereby tightening the gap between the global loss and the optimum. Using Lyapunov optimization, we further convert $\mathcal{P}$ into a deterministic per-round problem.

A. Convergence Analysis

Analyzed in this subsection is the convergence behavior of DA-SFL. To facilitate the analysis, the following assumptions on the loss function $F(\cdot)$ are adopted.

Assumption 1. (L-smooth) The joint loss $F(\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{s}})$ is continuously differentiable with respect to $\boldsymbol{w}^{\mathrm{c}}$ and $\boldsymbol{w}^{\mathrm{s}}$. There exist constants L_c, L_s, L_{cs}, and L_{sc} such that, $\forall s \in \mathcal{S}$ and all feasible $\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{c}'}, \boldsymbol{w}^{\mathrm{s}}, \boldsymbol{w}^{\mathrm{s}'}$, $\nabla_c F(\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{s}})$ is L_c-Lipschitz in $\boldsymbol{w}^{\mathrm{c}}$ and L_{cs}-Lipschitz in $\boldsymbol{w}^{\mathrm{s}}$, that is,

$$\|\nabla_c F(\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{s}}) - \nabla_c F(\boldsymbol{w}^{\mathrm{c}'}, \boldsymbol{w}^{\mathrm{s}})\| \le L_c \|\boldsymbol{w}^{\mathrm{c}} - \boldsymbol{w}^{\mathrm{c}'}\|, \tag{14}$$

and

$$\|\nabla_c F(\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{s}}) - \nabla_c F(\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{s}'})\| \le L_{cs} \|\boldsymbol{w}^{\mathrm{s}} - \boldsymbol{w}^{\mathrm{s}'}\|. \tag{15}$$

Similarly, $\nabla_s F(\boldsymbol{w}^{\mathrm{c}}, \boldsymbol{w}^{\mathrm{s}})$ is L_s-Lipschitz in $\boldsymbol{w}^{\mathrm{s}}$ and L_{sc}-Lipschitz in $\boldsymbol{w}^{\mathrm{c}}$.

Assumption 2. (Unbiased Gradient and Bounded Variance) For each client, the stochastic gradient is unbiased, $\mathbb{E}_{\mathcal{B}}[\tilde{\nabla} F_u(\boldsymbol{w}|\mathcal{B})] = \nabla F_u(\boldsymbol{w})$, and has bounded variance, $E_{\mathcal{B}}[\|\tilde{\nabla} F_u(\boldsymbol{w}|\mathcal{B}) - \nabla F_u(\boldsymbol{w})\|^2] \le \sigma^2$.

Assumption 3. (Bounded Dissimilarity) For each loss function $F_u(\boldsymbol{w})$, there exists a constant $\delta > 0$ such that $\|\nabla F_u(\boldsymbol{w})\|^2 \le \|\nabla F(\boldsymbol{w})\|^2 + \delta k_u$.

Assumption 1 and 2 are widely adopted in the previous works [10], [32], [33]. Specifically, Assumption 1 is satisfied by most deep neural networks. Modern neural networks typically consist of multiple stacked layers, and a network defined as a composition of functions is Lipschitz if each constituent layer is Lipschitz, as discussed in [34]. Prior studies [34], [35] have shown that common building blocks such as convolutional and linear layers, as well as several nonlinear activation functions (e.g., sigmoid and tanh), are Lipschitz mappings. Consequently, many practical deep networks satisfy Lipschitz regularity and admit Lipschitz-continuous gradients under standard conditions. Moreover, for a Lipschitz neural network in which all layers are Lipschitz, both the feature extractor and the predictor, each constructed by composing Lipschitz layers, are also Lipschitz. Therefore, Assumption 1 holds by modeling the overall neural network as Lipschitz continuous. Assumption 3 is a deviation-aware bounded-dissimilarity condition, introduced by adapting the modeling strategy used in [15], which itself follows the bounded-dissimilarity assumption line in heterogeneous federated optimization [36]. Assumption 3 is adopted as a deviation-aware modeling assumption that links the local gradient norm to the deviation level k_u. It is motivated by discrepancy-aware federated learning analyses and bounded-dissimilarity modeling in heterogeneous federated optimization. Section V-F provides empirical support for the plausibility of Assumption 3 in the tested settings. Here k_u is a deviation-level scalar that quantifies the severity of class imbalance: $k_u = 0$ indicates

class-balanced data distribution; $k_u > 0$ refers to data class-imbalanced; while larger k_u reflects more severe imbalance. Using such a scalar heterogeneity indicator is consistent with non-IID FL analyses that parameterize data heterogeneity via a universal measure [33], [37]. We now present an auxiliary lemma that will be used in the analysis.

Lemma 1. *Let Assumption* 1 *holds, we get the relationship for clients' averaged local loss function:*

$$\begin{aligned} &F_u(\boldsymbol{w}_u^{c'}, \boldsymbol{w}_u^{s'}) - F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s) \\ &\leq \langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c \rangle + \frac{1+\chi}{2} L_c \|\boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\|^2 \\ &+ \langle \nabla_s F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s \rangle + \frac{1+\chi}{2} L_s \|\boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s\|^2, \end{aligned} \tag{16}$$

where $\chi = \max\{L_{cs}, L_{sc}\}/\sqrt{L_c L_s}$ *measures relative cross-sensitivity of* $\nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s)$ *with respect to* $\boldsymbol{w}_u^s$ *and of* $\nabla_s F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s)$ *with respect to* $\boldsymbol{w}_u^c$.

Proof: Please see Appendix A. ∎

Based on Lemma 1, we analyze the proposed DA-SFL under the standard smooth non-convex objective setting, and provide a stationarity-type bound that is applicable to deep split learning models. In this respect, the one-round convergence bound of DA-SFL is derived as follows:

Theorem 1. *Let Assumption* 1, 2, 3 *hold, and the learning rate satisfy* $\eta_c \leq \frac{1}{2J(1+\chi)}$, $\eta_s \leq \frac{1}{2J(1+\chi)}$, *we have*

$$\begin{aligned} &F(\boldsymbol{w}_{i+1}^c, \boldsymbol{w}_{i+1}^s) - F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) \\ &\leq \{-\frac{J\eta_c}{2} + J\eta_c \sum\nolimits_{u=1}^{U} \alpha_{u,i} p_{u,i}\} \|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 \\ &+ J\eta_c \delta \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + (1+\chi) J \eta_c^2 \sigma^2 \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2 \\ &+ \{-\frac{J\eta_s}{2} + J\eta_s \sum\nolimits_{u=1}^{U} \alpha_{u,i} p_{u,i}\} \|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 \\ &+ J\eta_s \delta \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + (1+\chi) J \eta_s^2 \sigma^2 \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2, \end{aligned} \tag{17}$$

Proof: Please see Appendix B. ∎

Based on Theorem 1, we further derive the I-rounds convergence bound of DA-SFL:

Corollary 1. *Let the assumptions in Theorem* 1 *hold, the* T*-rounds convergence bound of DA-SFL is*

$$\begin{aligned} &F(\boldsymbol{w}_I) - F(\boldsymbol{w}^*) \leq a_1^I [F(\boldsymbol{w}_0) - F(\boldsymbol{w}^*)] \\ &+ a_2 \sum\nolimits_{i=1}^{I-1} a_1^{I-1-i} \sum\nolimits_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + \frac{1-a_1^I}{1-a_1} a_3, \end{aligned} \tag{18}$$

where $a_1 = 1 - J\eta_c L_c + 2J\eta_c L_c \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} - J\eta_s L_s + 2J\eta_s L_s \sum_{u=1}^{U} \alpha_{u,i} p_{u,i}$, $a_2 = J\delta(\eta_c + \eta_s)$, $a_3 = (1+\chi) J\sigma^2(\eta_c^2 + \eta_s^2) \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2$.

Proof: Please see Appendix C. ∎

From Corollary 1, bounded by three components is the expected gap between the global and optimal losses: (i) the initial gap between $F(\boldsymbol{w}_0)$ and $F(\boldsymbol{w}^*)$, (ii) the cumulative deviation level of clients' data over I rounds, and (iii) a constant that depends on the learning-system hyperparameters. Note that $a_1 < 1$ under the step-size conditions $\eta_c \leq \frac{1}{2J(1+\chi)}$ and $\eta_s \leq \frac{1}{2J(1+\chi)}$. As the number of global rounds increases, the first term vanishes, while the third term approaches a constant. These two terms are independent of the deviation-aware aggregation policy. The second term is explicit and depends on the weighting variables, which motivates the following remark.

Remark 1. According to Theorem 1 and Corollary 1, reducing the deviation level in each round tightens the bound on the optimality gap. We adopt the objective $\sum_{i=0}^{I-1} \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u$, which directly decreases the per-round upper bound and thus promotes minimization of the global loss.

It worth noting that (18) is derived under the smooth and potentially non-convex setting and therefore characterizes convergence in terms of a stationarity measure. An explicit function suboptimality rate such as $O(1/T)$ would require additional convexity or strong convexity assumptions and a stepsize schedule tailored to convex analysis. Since our target application is split federated learning with deep models, we focus on the non convex stationarity type guarantee and use (18) to highlight how deviation aware heterogeneity affects convergence.

B. Problem Transformation

From Corollary 1, the optimality gap can be reduced by minimizing the second term on the right hand side of (18). Direct minimization is intractable because this term depends on unknown quantities such as the Lipschitz constants L_c and L_s. Prior work shows that computing exact Lipschitz constants for deep architectures is intractable even for two-layer neural networks [35]. Following [23], [25], we reformulate problem $\mathcal{P}$ as the minimization of $\sum_{i=0}^{I-1} \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u$ as follows:

$$\begin{aligned} \hat{\mathcal{P}}: \quad & \min_{\{\boldsymbol{s}_i, \boldsymbol{M}_i, \boldsymbol{\omega}_i, \boldsymbol{p}_i\}_{i=0}^{I-1}} \sum\nolimits_{i=0}^{I-1} \sum\nolimits_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u \\ \text{s. t.} \quad & \text{(13a)}, \text{(13b)}, \text{(13c)}, \text{(13d)}, \text{(13e)}, \text{(13f)}, \text{(13g)}, \text{(13h)}. \end{aligned} \tag{19}$$

In general, $\hat{\mathcal{P}}$ is mixed-integer and NP-hard, as it couples discrete and continuous decision variables. Additionally, computing the offline optimum is intractable, as it requires optimally distributing each client's energy across rounds subject to long-term energy budgets. A further difficulty is that solving $\hat{\mathcal{P}}$ directly would require noncausal information of channel condition for all clients across all rounds at the start of training, which is impractical. To achieve online dynamic client sampling while handling temporal coupling, we adopt a Lyapunov optimization framework. For each client u, we construct a virtual queue $\lambda_{u,i}$ that tracks the deviation between energy consumption by round i and the allocated energy budget, and evolves according to

$$\lambda_{u,i+1} = \max\{\lambda_{u,i} + \alpha_{u,i} E_{u,i} - \frac{E_u}{I}, 0\}, \tag{20}$$

Initialize the virtual queues with $\lambda_{u,0} = 0$ for all clients. Given these queues, we embed the long-term energy constraint (13b) and the metric $\sum_{i=0}^{I-1} \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u$ are incorporated

Algorithm 1 Adaptive Bandwidth Allocation and Cut Layer Selection Algorithm

Initialization: Cut layer selection decision $\boldsymbol{s}_i$, client scheduling decision $\boldsymbol{M}_i$, aggregation weight $\boldsymbol{p}_i$, lower bound $z_{lb}=0$, upper bound z_{ub}, bandwidth allocation $\boldsymbol{\omega}_i$, precision requirement $\varepsilon>0$, and iteration $\tau \leftarrow 0$.
repeat
  $\tau \leftarrow \tau+1$
  Bandwidth Allocation Optimization:
  repeat
    Set $z=(z_{lb}+z_{ub})/2$.
    For each client $u \in \boldsymbol{M}_i$, compute required bandwidth allocation $\omega_{u,i}(z)$ by (26).
    Compute total bandwidth $\sum_{u\in\boldsymbol{M}_i}\omega_{u,i}(z)$.
    if $\sum_{u\in\boldsymbol{M}_i}\omega_{u,i}(z)>1$ **then**
      Set $z_{lb}=z$.
    else if $0<\sum_{u\in\boldsymbol{M}_i}\omega_{u,i}(z)<1-\varepsilon$ **then**
      Set $z_{ub}=z$.
    else
      Break the loop.
  until $|z_{ub}-z_{lb}|<\varepsilon$
  Compute optimal bandwidth allocation $\boldsymbol{\omega}_i^*$ from (26).
  Cut Layer Selection Optimization:
  for $u \in \boldsymbol{M}_i$ **do**
    for each cut layer $s_{u,i}\in\mathcal{S}$ **do**
      Compute learning latency $\tilde{T}(s_{u,i})=T_{u,i}^{\mathrm{CM}}+T_{u,i}^{\mathrm{CP}}$.
    Select optimal cut layer $s_{u,i}^*=\arg\min_{s_{u,i}\in\mathcal{S}}\tilde{T}(s_{u,i})$.
  Update objective $\mathcal{L}_\tau$ by substituting $\boldsymbol{\omega}_i^*$ and $\boldsymbol{s}_i^*$ into (22).
until $|\mathcal{L}_\tau-\mathcal{L}_{\tau-1}|<\varepsilon$
return Optimal cut layer decision $\boldsymbol{s}_i^*$ and bandwidth allocation policy $\boldsymbol{\omega}_i^*$.

using the Lyapunov drift-plus-penalty framework [38]. Hence, Problem $\tilde{\mathcal{P}}$ is recast as minimizing the drift-plus-penalty ratio

$$\tilde{\mathcal{P}}: \min_{\{\boldsymbol{s}_i,\boldsymbol{M}_i,\boldsymbol{\omega}_i,\boldsymbol{p}_i\}_{i=0}^{I-1}} -\mathcal{V}\sum_{u=1}^{U}\alpha_{u,i}p_{u,i}k_u+\sum_{u=1}^{U}\lambda_{u,i}\alpha_{u,i}E_{u,i}$$
$$\text{s. t. } (13a),(13b),(13c),(13d),(13e),(13f),(13g),(13h), \quad (21)$$

where $\mathcal{V}\geq 0$ is a hyperparameter that controls the trade-off between data deviation and energy consumption. Note that if the goal is to match the true global distribution rather than enforce class balance, we can set $\boldsymbol{G}$ to the empirical global label distribution instead of a uniform distribution.

IV. Adaptive Client Scheduling, Cut Layer Selection, Resource Allocation and Weighting

In this section, we develop efficient algorithms for client scheduling, cut layer selection, bandwidth allocation, and aggregation weighting to solve problem $\tilde{\mathcal{P}}$. As a mixed-integer nonlinear program, $\tilde{\mathcal{P}}$ is thus computationally demanding, and we therefore decompose it into tractable subproblems and solve them separately.

A. Adaptive Bandwidth Allocation and Cut Layer Selection

Given fixed cut layer decisions $\boldsymbol{s}_i$, client scheduling set $\boldsymbol{M}_i$, and aggregation weights $\boldsymbol{p}_i$, we isolate the bandwidth allocation subproblem of $\tilde{\mathcal{P}}$ as

$$\mathcal{P}_1: \min_{\boldsymbol{\omega}_i} \sum\nolimits_{u\in\boldsymbol{M}_i}\lambda_{u,i}E_{u,i} \quad (22)$$
$$\text{s. t. } (13c),(13d),$$
$$\frac{\gamma_u^c(s_u)+J\beta\gamma_u^a(s_u)}{T_{\max}-T_{u,i}^*}\leq\omega_{u,i}B\log\left(1+\frac{\hat{p}_{u,\max}h_{u,i}}{\omega_{u,i}BN_0}\right), \quad (22a)$$

where $T_{u,i}^*=T_{u,i}^{\mathrm{CP}}+\sum_{j=1}^{J}t_{u,i}^{\mathrm{gd}}+t_{u,i}^{\mathrm{cd}}$. For analytical convenience, we introduce the following auxiliary function for each client u ($u\in\mathcal{U}$).

$$q_u(\omega_{u,i})=\exp\left(\frac{[\gamma_u^c(s_u)+J\beta\gamma_u^a(s_u)]\ln 2}{\omega_{u,i}B(T_{\max}-T_{u,i}^*)}\right)-1. \quad (23)$$

By removing the constant terms in the objective of $\mathcal{P}_1$, the bandwidth allocation objective becomes

$$\vartheta(\boldsymbol{\omega}_i)=\sum_{u\in\boldsymbol{M}_i}\frac{\omega_{u,i}BN_0\lambda_{u,i}(T_{\max}-T_{u,i}^*)}{h_{u,i}}q_u(\omega_{u,i}). \quad (24)$$

Accordingly, the wireless bandwidth allocation subproblem is

$$\tilde{\mathcal{P}}_1: \min_{\boldsymbol{\omega}_i}\ \vartheta(\boldsymbol{\omega}_i) \quad (25)$$
$$\text{s. t. } (13c),(13d),(22a).$$

Problem $\tilde{\mathcal{P}}_1$ is a typical convex program, while the optimum is characterized by the following lemma.

Lemma 2. *The optimal bandwidth allocation for problem $\tilde{\mathcal{P}}_1$ satisfies $\omega_{u,i}^*=\max\{\omega_{u,i}(z),\omega_{u,i}^{\min}\}$, where*

$$\omega_{u,i}(z)=\frac{[\gamma_u^c(s_u)+J\beta\gamma_u^a(s_u)]\ln 2}{B(T_{\max}-T_{u,i}^*)(Z(\frac{zh_{u,i}}{BN_0\lambda_{u,i}(T_{\max}-T_{u,i}^*)e}-\frac{1}{e})+1)}, \quad (26)$$

and $\omega_{u,i}^{\min}$ satisfies constraint (13c)*, z is the Lagrange multiplier chosen to meet $\sum_{u=1}^{U}\omega_{u,i}(z^*)=1$. Here $Z(\cdot)$ denotes the principal branch of the Lambert function defined by $Z(\mathrm{x})\mathrm{e}^{Z(\mathrm{x})}=\mathrm{x}$, and e is Euler's number.*

Proof: Please see Appendix D. ∎

Although Lemma 2 characterizes the optimal bandwidth allocation, the Lagrange multiplier z remains unknown. We determine the optimal z via a bisection procedure. Since $z\geq 0$, it follows that $\frac{zh_{u,i}}{BN_0\lambda_{u,i}(T_{\max}-T_{u,i}^*)e}-\frac{1}{e}\geq-\frac{1}{e}$. The function $Z(\mathrm{x})$ is monotonically increasing for $\mathrm{x}\geq-\frac{1}{\mathrm{e}}$. Hence $\omega_{u,i}(z)$ is monotonically decreasing in z. To apply bisection, we derive bounds for z. The lower bound is $z_{lb}=0$. For the upper bound, observe that $\max_{\boldsymbol{M}_i}\{\omega_{u,i}(z)\}\geq\frac{1}{|\boldsymbol{M}_i|}$. Let $\phi_u=\frac{|\boldsymbol{M}_i|[\gamma_u^c(s_u)+J\beta\gamma_u^a(s_u)]\ln 2}{B(T_{\max}-T_{u,i}^*)}$, using the definition of the Lambert function, the upper bound is $z_{ub}=\max_{u\in\boldsymbol{M}_i}\{\frac{BN_0\lambda_{u,i}(T_{\max}-T_{u,i}^*)((\phi_u-1)e^{\phi_u}+1)}{h_{u,i}}\}$. Initialized on $[z_{lb},z_{ub}]$, bisection halves the interval at each iteration and stops when the tolerance ε is met. The time complexity is $\mathcal{O}(\log_2\frac{z_{ub}-z_{lb}}{\varepsilon})$.

Given a scheduled client set $\boldsymbol{M}_i$, aggregation weights $\boldsymbol{p}_i$, and bandwidth allocation policy $\boldsymbol{\omega}_i$, the cut layer decisions are decoupled across clients and affect the objective additively. Hence, each client's cut layer can be optimized independently. We decompose the cut layer selection subproblem of $\tilde{\mathcal{P}}$ as

$$\mathcal{P}_2: \min_{\boldsymbol{s}_i} \sum\nolimits_{u\in\boldsymbol{M}_i}\lambda_{u,i}E_{u,i} \quad (27)$$
$$\text{s. t. } (13b),(13e),$$

where $\mathcal{P}_2$ is nonconvex because (13e) imposes a discrete finite feasible set. Moreover, the sizes of the smashed activations, the smashed activations' gradients, and the client-side model,

Algorithm 2 Client Scheduling Algorithm

Initialize $\lambda_{u,i}$, and $\mathcal{V}$.
Sort $\boldsymbol{E}_{u,i}$ in ascending order, set $\Phi_0 = \{u : \lambda_{u,i} = 0\}$, $\Phi = \Phi_0$ and $\Omega = \{\Phi_0\}$.
for $v = |\Phi_0|+1, \cdots, V$ **do**
  Update $\Phi = \Phi \cup \{u\}$
  Algorithm 1 is applied for bandwidth allocation and cut layer selection, i.e., $\mathbf{X}(\Phi) = (\boldsymbol{\omega}_i, \boldsymbol{s}_i)$.
  if $-\mathcal{V}k_u + \lambda_{u,i}E_{u,i} \leq 0$ **then**
    $\Omega = \Omega \cup \Phi$
  else
    Break the loop.
end for
Compute the scheduling policy as $\boldsymbol{M}_i^* = \arg\min_{\Phi\in\Omega} \mathbf{Y}(\Phi)$.
return The optimal bandwidth allocation $\boldsymbol{\omega}_i^*$, cut layer decision $\boldsymbol{s}_i^*$, and client scheduling policy $\boldsymbol{M}_i^*$

as well as the computational workloads of the client-side FP and BP, are general functions of the cut layer. We therefore adopt a sample average approximation (SAA) based algorithm [20] to determine the optimal cut layer. As noted above, we separate the bandwidth allocation problem $\mathcal{P}_1$ and the cut layer selection problem $\mathcal{P}_2$ from the original formulation. We then solve for bandwidth allocation and cut layer decisions using block coordinate descent [5], alternating between $\mathcal{P}_1$ and $\mathcal{P}_2$ until convergence. The procedure is summarized in Algorithm 1, whose time complexity is $\mathcal{O}(\tau|\boldsymbol{M}_i|(\log_2 \frac{z_{ub}-z_{lb}}{\varepsilon} + |\mathcal{S}|))$. Algorithm 1 alternates between updating two variable blocks in the per-round objective in (22), namely the bandwidth allocation $\boldsymbol{\omega}_i$ and the cut-layer decision $\boldsymbol{s}_i$. Let L_τ be the objective value after the τ-th outer iteration. For fixed $\boldsymbol{s}_i$, the bandwidth subproblem is solved (numerically) optimally via $\tilde{P}_1$ and Lemma 2, so updating $\boldsymbol{\omega}_i$ cannot increase L_τ. For fixed $\boldsymbol{\omega}_i$, each cut layer is chosen by searching the finite set $\mathcal{S}$ in (13e), so updating $\boldsymbol{s}_i$ also cannot increase L_τ. Thus, L_τ is non-increasing and bounded below, so it converges. Meanwhile, since $\boldsymbol{s}_i$ takes values in a finite set, the algorithm reaches a stable point where neither update can further reduce the objective, which ensures convergence.

B. Energy-Aware Client Scheduling

For client scheduling, a straightforward approach is to evaluate the objective for every possible subset of clients and then select the subset that minimizes it. This brute-force enumeration requires considering $\sum_{u=0}^{U} C_U^u = 2^U$ scheduling sets and thus incurs exponential time complexity on the order of $\mathcal{O}(2^{U+1} \times \tau|\boldsymbol{M}_i|(\log_2 \frac{z_{ub}-z_{lb}}{\varepsilon} + |\mathcal{S}|))$. To overcome this intractability, we develop the following designs.

It worth noting that the scheduled subset is updated per-round to adapt to both the deviation degree and the time varying channel condition, which jointly determine the latency and energy costs. Under $\tilde{P}$, scheduling should prioritize clients with small $\lambda_{u,i}$ and $E_{u,i}$. A low $E_{u,i}$ typically reflects favorable channels and excellent computational efficiency. Accordingly, we first allocate bandwidth equally across all U clients and compute each client's induced energy $\bar{E}_{u,i}$. Specifically, each client u receives $\omega_{u,i} = \frac{1}{U}$ of the total bandwidth B and then solves $\mathcal{P}_2$ to determine the cut layer policy $\boldsymbol{s}_i$. Substituting $\omega_{u,i}$ and $\boldsymbol{s}_i$ into (10) and (12) yields $\bar{E}_{u,i} = E_{u,i}^{\mathrm{CP}} + E_{u,i}^{\mathrm{CM}}$. We apply the set-expansion algorithm [24] to construct the scheduling decision by ranking $\boldsymbol{E}_{u,i} = \lambda_{u,i}\bar{E}_{u,i}$ in ascending order. First, all clients with $\lambda_{u,i} = 0$ to Φ, denoted Φ_0. Next, we add clients with $\lambda_{u,i} > 0$ to Φ following the ascending order of $\boldsymbol{E}_{u,i}$. For each candidate set Φ, Algorithm 1 computes the bandwidth allocation and the cut layer selection, producing $\mathbf{X}^*(\Phi) = (\boldsymbol{\omega}^*(\Phi), \boldsymbol{s}^*(\Phi))$. Let $\mathbf{Y}(\Phi)$ denote the corresponding objective value. And let Ω be the collection of all candidate sets Φ. Note that $\mathbf{Y}(\Phi_0) = -\mathcal{V}\sum_{v\in\Phi_0} k_u$ since $\lambda_{u,i} = 0$ for all $v \in \Phi_0$. Because the energy terms of users in Φ_0 do not affect the objective, we we assign only the lowest needed bandwidth to these clients and reserve bandwidth for the clients in $(\Phi - \Phi_0)$. We then add users with $\lambda_{u,i} > 0$ to Φ in ascending order of $\boldsymbol{E}_{u,i}$, and for each candidate Φ compute $\mathbf{X}^*(\Phi)$ and $\mathbf{Y}(\Phi)$. If the resulting policy yields $-\mathcal{V}k_u + \lambda_{u,i}E_{u,i} > 0$ for the most recently added client u, we terminate the expansion and select the scheduling set with the smallest objective value, that is,

$$\boldsymbol{M}_i^* = \arg\min_{\Phi\in\Omega} \mathbf{Y}(\Phi). \tag{28}$$

The corresponding optimal bandwidth and cut layer decisions are $\boldsymbol{\omega}_i^*(\boldsymbol{M}_i^*)$ and $\boldsymbol{s}_i^*(\boldsymbol{M}_i^*)$. The procedure is summarized in Algorithm 2, which invokes Algorithm 1 at most U times. The resulting time complexity is $\mathcal{O}(\tau UV(\log_2 \frac{z_{ub}-z_{lb}}{\varepsilon} + |\mathcal{S}|))$, $V \leq U$, which is strictly lower than exhaustive enumeration.

Algorithm 3 Deviation-Aware Weighting Algorithm

Initialization: Globel model $\boldsymbol{w}$, Aggregation weight $\boldsymbol{p}_i$.
for $i = 0, \cdots, I-1$ **do**
  Server sends client-side model $\boldsymbol{w}_{u,i}^{\mathrm{c}}$ to each client.
  for $u \in \boldsymbol{M}_i$ **do**
    Calculate the deviation level k_u based on local dateset.
    Local training for J steps $\rightarrow \boldsymbol{w}_{u,i,J}^{\mathrm{c}}$.
    Send $\boldsymbol{w}_{u,i,J}^{\mathrm{c}}$ and the value of k_u to the server.
  end for
  Server computes the aggregation weight $\boldsymbol{p}_i$ based on (6).
  Server aggregates the full models $\boldsymbol{w}_{i+1} = \sum_{u\in\boldsymbol{M}_i} p_{u,i}\boldsymbol{w}_{u,i}$.
end for

C. Deviation-Aware Aggregation

Given the scheduled client set $\boldsymbol{M}_i$, cut layer decision $\boldsymbol{s}_i$, and bandwidth allocation policy $\boldsymbol{\omega}_i$, the clients' aggregation weight policies are separable: they do not couple and each enters the objective independently. Consequently, each client's aggregation weight can be optimized in isolation. We therefore decompose the cut layer selection subproblem of $\tilde{\mathcal{P}}$ as follows:

$$\begin{aligned} \mathcal{P}_3: \ & \min_{\boldsymbol{p}_i} \ \sum\nolimits_{u\in\boldsymbol{M}_i} p_{u,i}k_u \\ & \text{s. t.} \ \ (13\text{g}), (13\text{h}). \end{aligned} \tag{29}$$

Constraint (13g) is affine and therefore convex, whereas (13h) is a linear inequality that delineates a half-space in the decision-variable domain and is likewise convex. Note that $\mathcal{P}_3$ is linear in $\{p_{u,i}\}$ over the simplex, so its exact optimal solution is winner-take-all, i.e., assigning all weight to the client with the smallest k_u. The algorithmic procedure of deviation-aware weighting is detailed in Algorithm 3. To avoid this degenerate winner-take-all behavior, we do not directly use the exact solution of $\mathcal{P}_3$. Instead, Algorithm 3 adopts the heuristic deviation-aware weighting rule in (6) to produce practical aggregation weights.

V. Numerical Results

This section details the implementation of DA-SFL, including the simulation environment, datasets and models, and baseline methods. We then assess the proposed framework along three dimensions: (i) test accuracy gains relative to representative baselines; (ii) time and energy efficiency of the implemented algorithms; and (iii) robustness and scalability across heterogeneous settings.

TABLE I
System Parameters

Parameter	Value	Parameter	Value
U	100	M	10
B	1MHz	$\mathcal{V}$	$\{0.1, 1\}$
N_0	-174dBm/Hz	h_0	-30dB
v	2	κ	5×10^{-27}
a	0.5	b	0.1
f_s	20GHz	η_c	0.01
η_s	0.01	β	64
J	$\{2, 10\}$	I	100

A. Experimental Setting

Network setup: The simulation parameters follow representative wireless FL configurations (e.g., [23], [27], [39], [40]). Unless stated otherwise, the default experimental settings are summarised in Table I. Specifically, we consider a circular service area of radius 500m with the edge server located at the centre; U clients are randomly positioned within this region. The channel gain is designed as $h_{u,i} = h_0 \rho_u(i) d_u^{-v}$, where h_0 denotes the path loss constant, $\rho_u(i)$ is the small-scale fading coefficient for client u in round i, d_u is the distance from client u to the server, and v is the path loss exponent. Rayleigh fading is assumed, i.e., $\rho_u(i) \sim \exp(1)$, independent and identically distributed across clients and rounds. Following [9], [10], client heterogeneity is emulated by drawing each client's CPU frequency f_u from $\{0.5, 0.8, 1.0, 1.2, 1.6\}$GHz, with each CPU cycle processing 4 FLOPs. The client u's transmit power $\hat{p}_u$ is chosen from $\{0.01, 0.02, 0.03, 0.05\}$W randomly.

Dataset and model: We evaluate three standard image-classification datasets of increasing complexity: Fashion-MNIST (FMNIST), CIFAR-10, and CINIC-10. Implementation details are organised along two dimensions. (i) *Data distribution settings:* To emulate heterogeneity, we consider two local data distributions, denoted HD-1 and HD-2. In HD-1, client label proportions follow a Dirichlet distribution Dir(ι) [33], where a smaller ι implies greater heterogeneity. In HD-2, $U/2$ clients are *biased*, each containing data from $C/5$ classes, while the remaining $U/2$ clients are *unbiased*, containing all C classes. We further introduce μ as a global class-imbalance parameter, where $\mu = 1$ corresponds to a globally balanced dataset and $\mu > 1$ indicates imbalance. (ii) *Implemented models:* The neural network architectures used for each dataset are summarised in Table III, where 'C' denotes a convolutional module, 'M' a 2×2 max-pooling layer, 'F' a fully connected module, and the accompanying number is the neuron count (F) or filter count (C). The CNNs for CIFAR-10 and CINIC-10 use 5×5 kernels. All hidden layers employ ReLU activations.

TABLE III
Network Architecture

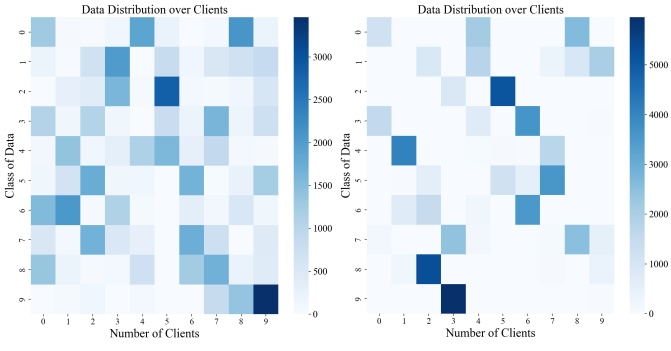
Dataset	Model Name	Architecture
FMNIST	MLP	F: [784, 256, 128, 64, 10]
CIFAR-10	CNN	C: [64, 64, M, 128, 128, M]
		F: [512, 10]
CINIC-10	CNN	C: [64, 64, M, 128, 256, M]
		F: [256, 10]

B. Performance Improvement of DA-SFL Framework

To substantiate the merits of the proposed DA-SFL framework, we benchmark its performance against four representative baselines: (i) FedAvg [4], wherein the server iteratively averages full local models from the selected clients; (ii) FedProx [13], which augments each client's objective with a proximal term tethering it to the current global model; (iii) FedDyn [14], which introduces a dynamic regulariser to counter client drift and align stationary points across clients; and (iv) FedNova [36], which normalises each client's update by its local step count prior to aggregation. In this subsection, we adopt the HD-1 local data distribution model and examine multiple heterogeneity levels (e.g., Dir(0.5) and Dir(0.1)), together with varying global class-imbalance regimes. This subsection primarily evaluates algorithmic efficacy, while the adaptive client scheduling, cut layer selection, and bandwidth allocation algorithms under explicit energy and wireless resource constraints are assessed in Section V-C.

In Fig. 2, we use the FMNIST dataset to illustrate per-client label distributions across ten classes under varying levels of local data heterogeneity. Specifically, Fig. 2(a) presents the HD-1 setting with Dirichlet concentration parameter $\iota = 0.5$, whereas Fig. 2(b) shows a more pronounced heterogeneity at $\iota = 0.1$. The global class distributions for all three datasets are summarized in Fig. 3: Fig. 3(a) corresponds to the globally balanced case $\mu = 1$, and Fig. 3(b) to a severely imbalanced case $\mu = 20$. These configurations are designed to demonstrate that the proposed DA-SFL is robust to both local data heterogeneity and global class imbalance.

Fig. 4 compares DA-SFL with four baselines on the FMNIST dataset under two levels of local data heterogeneity. With $\iota = 0.5$ in Fig. 4(a), DA-SFL attains the highest test accuracy across global class-imbalance levels, outperforming

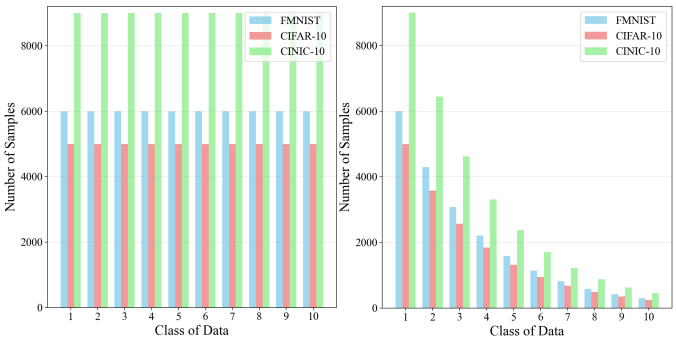


(a) $\iota = 0.5$ (b) $\iota = 0.1$

Fig. 2. Visualization of FMNIST distributions under varying heterogeneity.

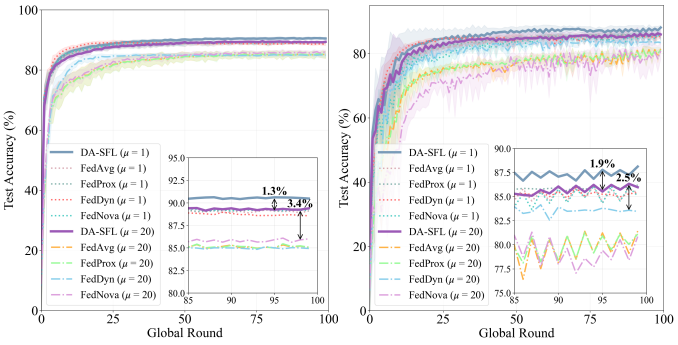
TABLE II
ACCURACY COMPARISON BETWEEN DA-SFL AND BASELINE APPROACHES

Dataset	ι	FedAvg		FedProx		FedDyn		FedNova		**DA–SFL**		Improvement	
		$\mu=1$	$\mu=20$	$\mu=1$	$\mu=20$	$\mu=1$	$\mu=20$	$\mu=1$	$\mu=20$	$\mu=1$	$\mu=20$	$\mu=1$	$\mu=20$
FMNIST	0.5	89.10%	85.08%	89.13%	85.08%	88.57%	84.94%	**89.21%**	**86.02%**	**90.48%**	**89.40%**	1.3%	3.4%
	0.1	85.42%	81.43%	**86.21%**	81.13%	85.42%	**83.47%**	85.07%	80.86%	**88.12%**	**85.97%**	1.9%	2.5%
CIFAR-10	0.5	66.57%	48.26%	66.47%	48.29%	**69.77%**	**52.21%**	66.93%	50.25%	**71.70%**	**68.82%**	1.9%	16.6%
	0.1	58.92%	40.06%	59.62%	39.49%	**62.70%**	**46.24%**	60.71%	36.08%	**65.22%**	**63.86%**	2.5%	17.6%
CINIC-10	0.5	53.21%	39.28%	53.34%	39.12%	**54.19%**	**40.97%**	53.14%	40.50%	**62.42%**	**61.66%**	8.2%	20.7%
	0.1	**45.24%**	**35.36%**	44.97%	34.86%	44.94%	31.35%	44.27%	30.07%	**57.58%**	**55.88%**	12.3%	20.5%



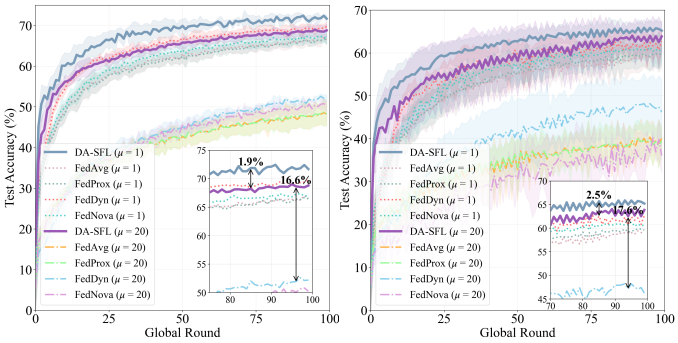
(a) Global Class-balanced $\mu = 1$ (b) Global Class-imbalanced $\mu = 20$

Fig. 3. Illustration of global data distributions on three distinct datasets.



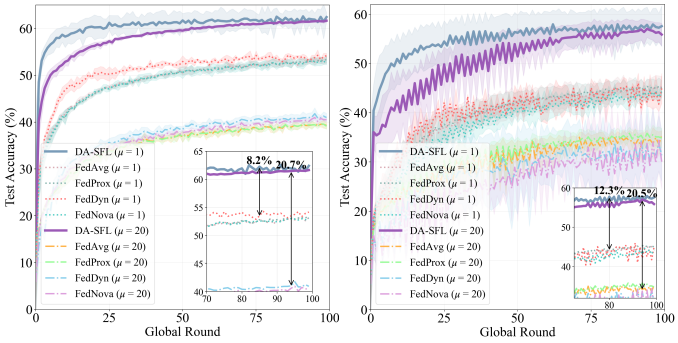
(a) $\iota = 0.5$ (b) $\iota = 0.1$

Fig. 4. Comparison of DA-SFL and baselines on FMNIST across varying local data deviation levels and global imbalance degrees.



(a) $\iota = 0.5$ (b) $\iota = 0.1$

Fig. 5. Comparison of DA-SFL and baselines on CIFAR-10 across varying local data deviation levels and global imbalance degrees.



(a) $\iota = 0.5$ (b) $\iota = 0.1$

Fig. 6. Comparison of DA-SFL and baselines on CINIC-10 across varying local data deviation levels and global imbalance degrees.

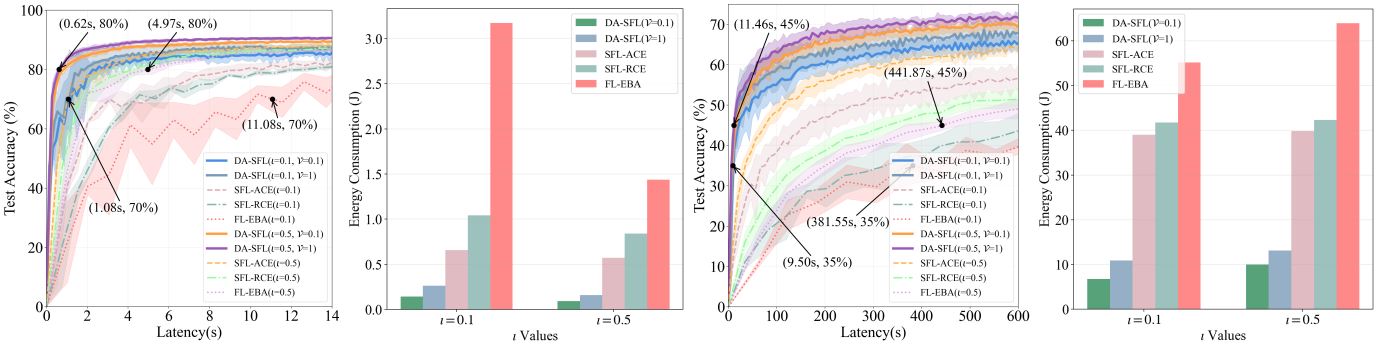
the best-performing baseline by 1.3% accuracy at $\mu = 1$ (globally balanced) and by 3.4% accuracy at $\mu = 20$ (severely imbalanced). Under the more heterogeneous setting $\iota = 0.1$ in Fig. 4(b), DA-SFL retains a clear advantage, improving accuracy by at least 1.9% and 2.5% at $\mu = 1$ and $\mu = 20$, respectively. As a result, these results show that DA-SFL is robust to both balanced and imbalanced global data distributions, with performance gains that become more pronounced as local data heterogeneity increases.

A parallel evaluation on CIFAR-10 is shown in Fig. 5. Relative to FMNIST, the gains are more pronounced on this more challenging dataset. With $\iota = 0.5$ (Fig. 5(a)), DA-SFL improves accuracy over the strongest baseline by 1.9% at $\mu = 1$ and by 16.6% at $\mu = 20$. With $\iota = 0.1$ (Fig. 5(b)), the improvements increase to 2.5% at $\mu = 1$ and 17.6% accuracy boosting at $\mu = 20$. These results indicate that as global class imbalance intensifies, DA-SFL maintains its advantage and often amplifies it, demonstrating robustness to data heterogeneity and imbalance.

To further substantiate effectiveness, we evaluate on CINIC-10, a composite benchmark that combines CIFAR-10 and ImageNet images and provides greater scale and diversity than either CIFAR-10 or grayscale FMNIST (Fig. 6). With $\iota = 0.5$ (Fig. 6(a)), DA-SFL exceeds the best baseline by at least 8.2% at $\mu = 1$ and by 20.7% at $\mu = 20$. Under $\iota = 0.1$ (Fig. 6(b)), the gains remain substantial, namely 12.3% and 20.5% at $\mu = 1$ and $\mu = 20$, respectively. Thus, on the most complex dataset considered, DA-SFL delivers the largest performance margins. Table II summarizes accuracy across datasets and settings and confirms that DA-SFL is robust to both local data heterogeneity and global class imbalance.

C. Time and Energy Efficiency

This subsection assesses the time and energy efficiency of the proposed framework against three baselines: 1) Traditional SFL with Adaptive Cut layer and Equal bandwidth allocation (SFL-ACE): In each round i, the cut layer is selected adaptively and the bandwidth is allocated equally on wireless



(a) Time Efficiency on FMNIST (b) Energy Consumption for MLP (c) Time Efficiency on CIFAR-10 (d) Energy Consumption for CNN

Fig. 7. Time and energy consumption comparison between DA-SFL and baselines for reaching the target accuracy on FMNIST and CIFAR-10 datasets.

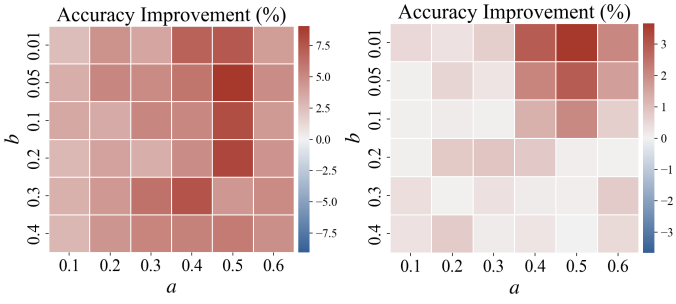
TABLE IV
COMPARATIVE ENERGY-TIME EFFICIENCY ANALYSIS OF DA-SFL

Dataset	ι	Target	FL-EBA		Best Baseline		DA-SFL ($\mathcal{V}=0.1$)		DA-SFL ($\mathcal{V}=1$)		Least Gains $\mathcal{V}=0.1$ / $\mathcal{V}=1$	
			Time	Energy	Time	Energy	Time	Energy	Time	Energy	Speed Up	Energy Reduction
FMNIST	0.1	70%	11.08s	3.17J	3.18s	0.66J	1.28s	0.14J	1.08s	0.26J	2.48x / 2.94x	78.8% / 60.6%
	0.5	80%	4.97s	1.44J	2.76s	0.57J	0.85s	0.09J	0.62s	0.16J	3.25x / 4.45x	84.2% / 71.9%
CIFAR-10	0.1	35%	381.55s	55.19J	74.67s	38.98J	10.70s	6.72J	9.50s	10.88J	6.98x / 7.86x	82.8% / 72.1%
	0.5	40%	441.87s	63.92J	76.23s	39.80J	16.00s	9.99J	11.46s	13.13J	4.76x / 6.65x	74.9% / 67.0%

SFL networks. 2) Traditional SFL with Random Cut layer and Equal bandwidth allocation (SFL-RCE): The cut layer is selected randomly and the bandwidth is allocated equally on wireless SFL networks. 3) Traditional FL with Equal Bandwidth Allocation (FL-EBA): The bandwidth is allocated equally on wireless FL networks. All experiments use FMNIST and CIFAR-10 with local heterogeneity $\iota \in \{0.5, 0.1\}$, weight parameter $\mathcal{V} = \{0.1, 1\}$, and the global class-imbalance degree fixed at $\mu = 1$.

Fig. 7 compares the time and energy requirement of DA-SFL and the baselines to reach target accuracies on FMNIST and CIFAR-10. On FMNIST (Fig. 7(a)), with $\mathcal{V} = 1$ and $\iota = 0.5$, DA-SFL reaches target accuracy 80% in 0.62 seconds, whereas FL-EBA requires 4.97 seconds. With $\iota = 0.1$, DA-SFL attains 70% accuracy in 1.08 seconds, compared with 11.08 seconds for FL-EBA. On CIFAR-10 (Fig. 7(c)), with $\iota = 0.5$, DA-SFL achieves 45% accuracy in 11.46 seconds, while FL-EBA needs 441.87 seconds. With $\iota = 0.1$, DA-SFL reaches 35% accuracy in 10.6 seconds, compared with 381.55 seconds for FL-EBA. With $\mathcal{V} = 0.1$ on FMNIST, DA-SFL reaches 70% in 1.28 seconds at $\iota = 0.1$ and 80% in 0.85 seconds at $\iota = 0.5$, which remains notably faster than the baselines.

Fig. 7(b) and Fig. 7(d) show that DA-SFL consistently consumes less energy than all baselines. Table IV summarizes the time and energy required to reach the target accuracies, reported relative to the representative baselines. With $\mathcal{V} = 1$ on FMNIST, DA-SFL achieves 2.94 times and 4.45 times speedups at $\iota = 0.1$ and $\iota = 0.5$, respectively, together with energy reductions of 60.6% and 71.9%. On CIFAR-10, the gains are larger: at $\iota = 0.1$, DA-SFL reaches the 35% target accuracy with 7.04 times speedup and an 82.5% energy reduction; at $\iota = 0.5$ it reaches the 40% target with a 6.65 times speedup and an 81.5% energy reduction. With $\mathcal{V} = 0.1$, speedups are slightly smaller than with $\mathcal{V} = 1$, while energy savings are larger, indicating that $\mathcal{V}$ effectively trades off training speed and client energy consumption. Overall, DA-SFL substantially reduces wall-clock time and energy across datasets and heterogeneity levels, demonstrating strong practical efficiency in wireless federated settings.



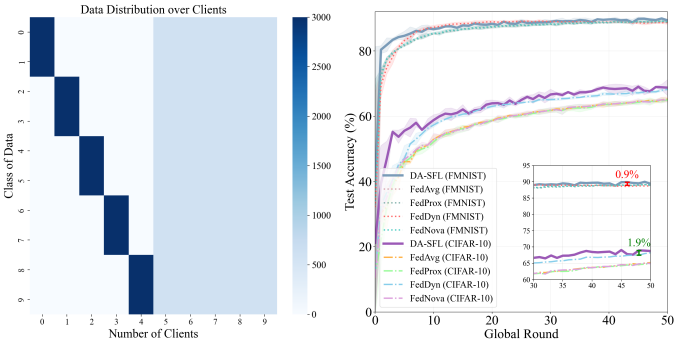
(a) Distribution HD-1 ($\iota = 0.1$) (b) Distribution HD-2

Fig. 8. Sensitivity study on CIFAR-10 under the HD-1 and HD-2 data distributions.

D. Sensitive study for a and b

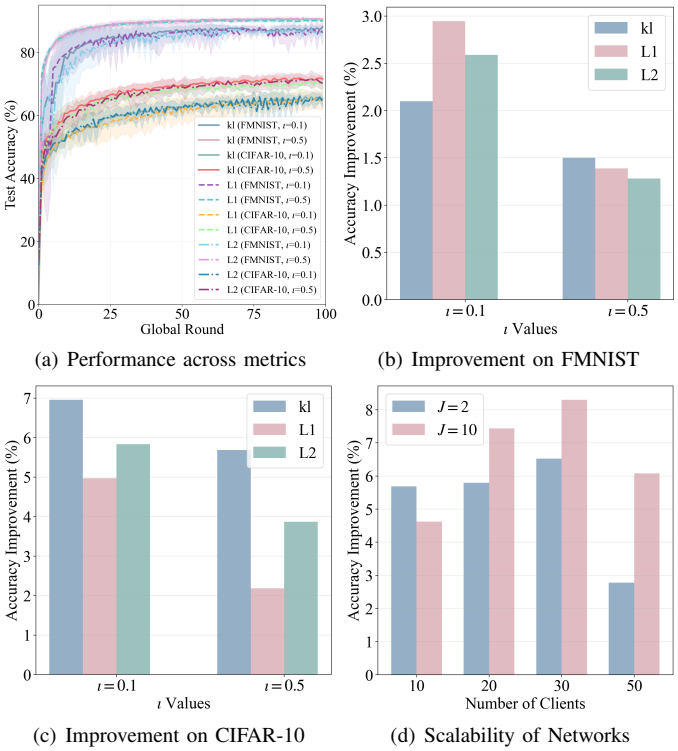
We evaluate $a \in [0.1, 0.6]$ and $b \in [0.01, 0.4]$, and report the resulting test accuracy on the CIFAR-10 dataset under two heterogeneity settings, HD-1 ($\iota = 0.1$) and HD-2. As shown in Fig. 8, DA-SFL achieves higher accuracy for most (a, b) pairs compared with the variant without deviation-aware weighting.

More specifically, Fig. 8(a) and Fig. 8(b) indicate that: (i) DA-SFL consistently improves performance across a wide range of (a, b) combinations, and the observed gains are robust under both heterogeneity settings on CIFAR-10; (ii) a in the range 0.4 to 0.6 and b in the range 0.01 to 0.1 represent safe hyperparameter choices that yield stable and effective performance.



(a) HD-2 Data Distribution (b) Performance Comparison

Fig. 9. Illustration of HD-2 data distribution and learning performance comparison of DA-SFL and baselines on FMNIST and CIFAR-10 datasets.



(a) Performance across metrics (b) Improvement on FMNIST

(c) Improvement on CIFAR-10 (d) Scalability of Networks

Fig. 10. Robustness of DA-SFL across datasets and metrics, and scalability with respect to the number of clients.

E. Robustness and Scalability of DA-SFL

This subsection assesses the robustness and scalability of DA-SFL along three dimensions: (1) the HD-2 local data distribution setting, which induces stronger client-level heterogeneity; (2) alternative discrepancy measures for quantifying label-distribution deviation; (3) larger-scale networks with increased client cardinality; (4) ablation study of DA-SFL; and (5) comparison with representative SFL baselines.

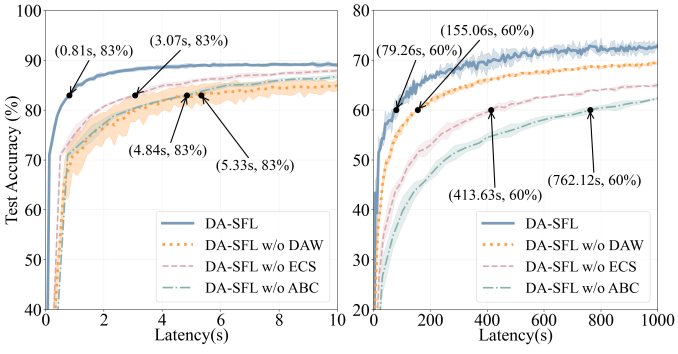
In Fig. 9, we adopt the HD-2 local data distribution and a globally balanced setting ($\mu = 1$), and evaluate performance on FMNIST and CIFAR-10. Fig. 9(a) depicts the per-client label distribution across ten classes for FMNIST under the HD-2 configuration. As shown in Fig. 9(b), DA-SFL continues to outperform all baselines under HD-2 on both datasets, exceeding the best baseline by 0.9% and 1.9% on FMNIST and CIFAR-10, respectively. These results provide additional evidence of the framework's robustness across distinct local heterogeneity regimes.

We further evaluate DA-SFL under different deviation metrics, including norm-based penalties (L1 and L2) and an information-theoretic divergence (KL), on FMNIST and CIFAR-10 with two heterogeneity settings, $\iota = 0.1$ and $\iota = 0.5$. All other components are kept unchanged to ensure a controlled comparison and to provide a systematic assessment of metric choice. As shown in Fig. 10(a), DA-SFL achieves comparable performance across datasets and data distributions under different deviation metrics, which indicates robustness to the metric selection. Moreover, Fig. 10(b) and Fig. 10(c) report the test accuracy gains of DA-SFL over the variant without deviation-aware weighting under different metrics, showing that the improvements brought by DA-SFL are consistent across metric choices. The resulting accuracies are closely matched across deviation levels, which indicates that DA-SFL is insensitive to the specific metric and thus robust to alternative discrepancy formulations.

For scalability, Fig. 10(d) reports results on CIFAR-10 with $\mu = 1$ and $\iota = 0.5$, comparing DA-SFL with the variant without deviation-aware weighting as the number of participating clients increases. DA-SFL consistently yields higher accuracy across all client cardinalities considered, demonstrating favorable scaling behavior across network sizes. As a consequence, these results show that DA-SFL is robust to metric choice and heterogeneous data conditions, and that it scales effectively to larger networks.

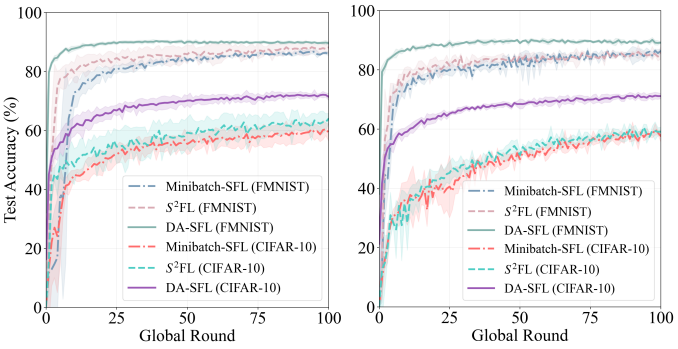
To validate the contribution of each algorithmic component, we consider three ablated variants: (i) DA-SFL without deviation-aware weighting (DA-SFL w/o DAW), (ii) DA-SFL without energy-aware client scheduling (DA-SFL w/o ECS), and (iii) DA-SFL without adaptive bandwidth allocation and cut-layer selection (DA-SFL w/o ABC). Under the HD-2 data partition and with $\mathcal{V} = 1$, we conduct experiments on FMNIST and CIFAR-10. The results are presented in Fig. 11.

In Fig. 11(a), the full DA-SFL consistently outperforms the three ablated variants on the FMNIST dataset. For a target test accuracy of 83%, DA-SFL reaches the target within 0.81s, whereas the best-performing ablation variant requires 3.07s. The remaining variants, DA-SFL w/o DAW and DA-SFL w/o



(a) FMNIST (b) CIFAR-10

Fig. 11. Ablation study on the FMNIST and CIFAR-10 datasets.



(a) Distribution HD-1 ($\iota = 0.5$) (b) Distribution HD-2

Fig. 12. Comparison of DA-SFL with representative SFL baselines on FMNIST and CIFAR-10 under the HD-1 and HD-2 data distributions.

ABC, require 5.33s and 4.84s, respectively. Fig. 11(b) reports the results on CIFAR-10. DA-SFL achieves the target accuracy of 60% in 79.26s, while the fastest ablation variant, DA-SFL w/o DAW, still requires 155.06s. The other two variants, DA-SFL w/o ECS and DA-SFL w/o ABC, take substantially longer, at 413.63s and 762.12s, respectively. Overall, these results show that the proposed DA-SFL provides consistently faster convergence than each ablated counterpart.

In Fig. 12, we consider two representative SFL baselines: (i) Minibatch-SFL [41], where clients train the client-side submodel and the server trains the server-side submodel using minibatch SGD to mitigate non-IID drift; and (ii) S^2FL [42], which employs an adaptive sliding split across devices and data-balanced server-side training to improve accuracy and alleviate straggler effects. To validate the effectiveness of the proposed DA-SFL, we compare DA-SFL with these baselines on FMNIST and CIFAR-10 under the HD-1 ($\iota = 0.5$) and HD-2 data distributions. As shown in Fig. 12(a) and Fig. 12(b), DA-SFL consistently outperforms Minibatch-SFL and S^2FL across both datasets and heterogeneity settings.

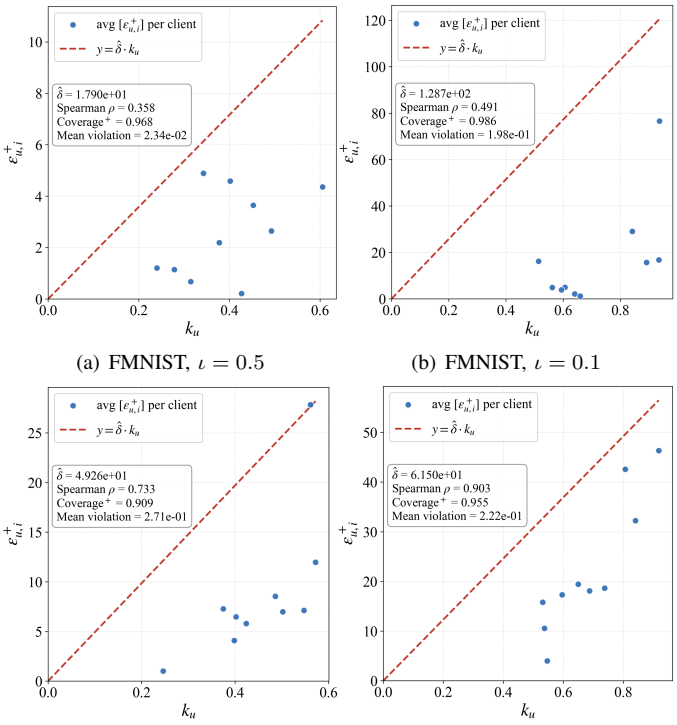
F. Empirical Support for Assumption 3

Assumption 3 in Section III-A is introduced as a deviation-aware modeling assumption that relates the local gradient norm to the client deviation level k_u. To examine its empirical plausibility, we evaluate the gradient behavior of DA-SFL on FMNIST and CIFAR-10 under the HD-1 setting with $\mu = 1$ and $\iota \in \{0.5, 0.1\}$. Specifically, at saved shared-model checkpoints during training, we compute the positive excess local gradient norm

$$\varepsilon_{u,i}^{+} = \max\left\{\|\nabla F_u(w_i)\|_2^2 - \|\nabla F(w_i)\|_2^2,\, 0\right\}, \qquad (30)$$

and compare its client-level average with the empirical upper-envelope $y = \hat{\delta} k_u$. Here, Coverage$^+$ denotes the fraction of positive cases satisfying $\varepsilon_{u,i}^{+} \le \hat{\delta} k_u$, while Mean violation denotes the average value of $[\varepsilon_{u,i}^{+} - \hat{\delta} k_u]_+$.

The FMNIST results are presented in Fig. 13(a) and Fig. 13(b). In both heterogeneity settings, a positive association is observed between the deviation score k_u and the averaged positive excess gradient norm. For $\iota = 0.5$, the Spearman rank correlation is 0.491 with Coverage$^+ = 0.986$, while for $\iota = 0.1$, the correlation is 0.358 with Coverage$^+ = 0.968$.



(a) FMNIST, $\iota = 0.5$ (b) FMNIST, $\iota = 0.1$

(c) CIFAR-10, $\iota = 0.5$ (d) CIFAR-10, $\iota = 0.1$

Fig. 13. Empirical support for Assumption 3 on FMNIST and CIFAR-10. Each point corresponds to one client, where the horizontal axis is the deviation score k_u and the vertical axis is the checkpoint-averaged positive excess gradient norm. The dashed line denotes the fitted upper-envelope $y = \hat{\delta} k_u$.

Although the monotone relation on FMNIST is moderate, the fitted upper-envelope covers most positive cases and the mean violations remain small, which is consistent with Assumption 3 in this setting.

Fig. 13(c) and Fig. 13(d) report the corresponding results on CIFAR-10. A clearer monotone relation emerges in this more challenging setting: the Spearman correlations are 0.903 and 0.733 for $\iota = 0.5$ and $\iota = 0.1$, respectively, while Coverage$^+$ remains high at 0.955 and 0.909. These results show that clients with larger data-distribution deviation tend to exhibit larger excess local gradient norms, particularly on CIFAR-10. Overall, across both datasets and heterogeneity levels, the observed positive association between k_u and $\varepsilon_{u,i}^{+}$, together with the high empirical coverage of the fitted upper-envelope, provides empirical support for the plausibility of Assumption 3 in the tested settings.

VI. Conclusion

In this work, we propose a DA-SFL framework for tackling data heterogeneity and imbalance, client heterogeneity, and constrained wireless resources. By providing a convergence analysis, we find a deviation level metric affects learning performance. Motivated by this, we formulate a joint optimization problem that integrates client scheduling, cut layer selection, bandwidth allocation, and aggregation weighting. Using a Lyapunov optimization approach, we transform the original problem into tractable subproblems and develop algorithms to solve each of them. Extensive experiments demonstrate that

DA-SFL outperforms baselines in accuracy, time and energy consumption, while also exhibiting robustness to heterogeneous conditions and scalability with increasing network size.

APPENDIX

A. Proof of Lemma 1

Using L_c-smooth of $F_u(*, \boldsymbol{w}_u^s)$, and L_s-smooth of $F_u(\boldsymbol{w}_u^c, *)$, we have $F_u(\boldsymbol{w}_u^{c'}, \boldsymbol{w}_u^{s'}) - F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}) \leq \frac{L_c}{2}\|\boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\|^2 + \langle \nabla_{\boldsymbol{w}^c} F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle$ and $F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}) - F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s) \leq \frac{L_s}{2}||\boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s||^2 + \langle \nabla_{\boldsymbol{w}^c} F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s\rangle$, summarizing them:

$$
\begin{aligned}
&F_u(\boldsymbol{w}_u^{c'}, \boldsymbol{w}_u^{s'}) - F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s) \leq \frac{L_c}{2}||\boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c||^2 \\
&+ \langle \nabla_{w^c} F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle + \frac{L_s}{2}||\boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s||^2 \\
&+ \langle \nabla_{\boldsymbol{w}^c} F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s\rangle. \qquad (31)
\end{aligned}
$$

Now focus on bounding $\langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle$:

$$
\begin{aligned}
&\langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle \overset{(a)}{=} \langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle \\
&+ \langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}) - \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle \\
&\overset{(b)}{\leq} \langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle \\
&+ \|\nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'}) - \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s)\|\|\boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\| \\
&\overset{(c)}{\leq} \langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle \\
&+ L_{cs}\|\boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s\|\|\boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\|, \\
&\overset{(d)}{\leq} \frac{1}{2}\chi L_c\|\boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\|^2 + \frac{1}{2}\chi L_s\|\boldsymbol{w}_u^{s'} - \boldsymbol{w}_u^s\|^2 \\
&+ \langle \nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s), \boldsymbol{w}_u^{c'} - \boldsymbol{w}_u^c\rangle, \qquad (32)
\end{aligned}
$$

where (a) is derived by adding and subtracting $\nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s)$ into $\nabla_c F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^{s'})$, (b) follows the Cauchy-Schwarz inequality, (c) comes from Assumption 1, (d) is due to the definition of χ. Substituting (32) into $F_u(\boldsymbol{w}_u^{c'}, \boldsymbol{w}_u^{s'}) - F_u(\boldsymbol{w}_u^c, \boldsymbol{w}_u^s)$, we obtain equation (16). The proof completes.

B. Proof of Theorem 1

From the Lipschitz-smooth assumption in Assumption 1, we have its equivalent form

$$
\begin{aligned}
&\mathbb{E}[F(\boldsymbol{w}_{i+1}^c, \boldsymbol{w}_{i+1}^s) - F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)] \\
&\leq \mathbb{E}[\langle \nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s), \boldsymbol{w}_{i+1}^c - \boldsymbol{w}_i^c\rangle] - \frac{1+\chi}{2}\mathbb{E}[\|\boldsymbol{w}_{i+1}^c - \boldsymbol{w}_i^c\|^2] \\
&+ \mathbb{E}[\langle \nabla_s F(\boldsymbol{w}_i^c, w_i^s), \boldsymbol{w}_{i+1}^s - \boldsymbol{w}_i^s\rangle] - \frac{1+\chi}{2}\mathbb{E}[\|\boldsymbol{w}_{i+1}^s - \boldsymbol{w}_i^s\|^2] \\
&= -J\eta_c \underbrace{\mathbb{E}[\langle \nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s), \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{h}_{u,i}^c\rangle]}_{A_1} \\
&+ \frac{(1+\chi)J^2\eta_c^2}{2} \underbrace{\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{h}_{u,i}^c\|^2]}_{A_2} \\
&- J\eta_s \underbrace{\mathbb{E}[\langle \nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s), \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{h}_{u,i}^s\rangle]}_{A_3} \\
&+ \frac{(1+\chi)J^2\eta_s^2}{2} \underbrace{\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{h}_{u,i}^s\|^2]}_{A_4}, \qquad (33)
\end{aligned}
$$

below we bound three terms in (33), for A_1,

$$
\begin{aligned}
A_1 &\overset{(a)}{=} \mathbb{E}[\langle \nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s), \sum_{u=1}^U \alpha_{u,i} p_{u,i} (\boldsymbol{h}_{u,i}^c - \boldsymbol{H}_{u,i}^c)\rangle] \\
&+ \mathbb{E}[\langle \nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s), \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\rangle], \\
&\overset{(b)}{=} \frac{1}{2}\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + \frac{1}{2}\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \\
&- \frac{1}{2}\mathbb{E}[\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \qquad (34)
\end{aligned}
$$

where (a) follows Assumption 2, and (b) follows $\mathbb{E}[\boldsymbol{h}_{u,i}^c - \boldsymbol{H}_{u,i}^c] = 0$ and the fact that $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$. Similarly, $A_3 = \frac{1}{2}\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + \frac{1}{2}\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^s\|^2 - \frac{1}{2}\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^s\|^2$. For bounding A_2,

$$
\begin{aligned}
A_2 &\overset{(a)}{=} 2\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \\
&+ \mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} (\boldsymbol{h}_{u,i}^c - \boldsymbol{H}_{u,i}^c) + \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \\
&\overset{(b)}{=} 2\sum_{u=1}^U (\alpha_{u,i} p_{u,i})^2 \mathbb{E}[\|\boldsymbol{h}_{u,i}^c - \boldsymbol{H}_{u,i}^c\|^2] \\
&+ 2\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \\
&\overset{(c)}{=} \frac{2}{J^2}\sum_{u=1}^U (\alpha_{u,i} p_{u,i})^2 \sum_{j=0}^{J-1} \mathbb{E}[\|\tilde{\nabla} F_u(\boldsymbol{w}_{u,i,j}^c) - \nabla F_u(\boldsymbol{w}_{u,i,j}^c)\|^2] \\
&+ 2\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \\
&\overset{(d)}{\leq} \frac{2\sigma^2}{J}\sum_{u=1}^U (\alpha_{u,i} p_{u,i})^2 + 2\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2] \qquad (35)
\end{aligned}
$$

where (a) subtract and add $\boldsymbol{H}_{u,i}^c$, (b) follows $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ and adopts the truth that clients are independent to each other so that $\mathbb{E}\langle \boldsymbol{h}_{u,i} - \boldsymbol{H}_{u,i}, \boldsymbol{h}_{v,i} - \boldsymbol{H}_{v,i}\rangle = 0$, (c) utilizes $\mathbb{E}\|\sum_{t=1}^T A_t\|_F^2 = \sum_{t=1}^T \mathbb{E}\|A_t\|_F^2$ as $\{A_t\}_{t=1}^T$ is a random matrices' sequence while each A_t is orthogonal, and (d) follows Assumption 2. Similarly, A_4 is $\frac{2\sigma^2}{J}\sum_{u=1}^U (\alpha_{u,i} p_{u,i})^2 + 2\mathbb{E}[\|\sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^s\|^2]$. Plug A_1, A_2, A_3, A_4 back into (33), when $\eta_c \leq \frac{1}{2J(1+\chi)}$ and $\eta_s \leq \frac{1}{2J(1+\chi)}$, we have

$$
\begin{aligned}
&\mathbb{E}[F(\boldsymbol{w}_{i+1}^c, \boldsymbol{w}_{i+1}^s) - F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)] \\
&\leq -\frac{J\eta_c}{2}\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + (1+\chi)J\eta_c^2\sigma^2\sum_{u=1}^U (\alpha_{u,i} p_{u,i})^2 \\
&+ \frac{J\eta_c}{2} \underbrace{\mathrm{E}[\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2]}_{B_1} \\
&- \frac{J\eta_s}{2}\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + (1+\chi)J\eta_s^2\sigma^2\sum_{u=1}^U (\alpha_{u,i} p_{u,i})^2 \\
&+ \frac{J\eta_s}{2} \underbrace{\mathrm{E}[\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - \sum_{u=1}^U \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^s\|^2]}_{B_2}. \qquad (36)
\end{aligned}
$$

Now focus on bounding B_1 and B_2:

$$B_1 = \mathbb{E}[\|\sum_{u=1}^{U} \alpha_{u,i} p_{u,i} \nabla_c F_u(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} \boldsymbol{H}_{u,i}^c\|^2]$$
$$\overset{(a)}{\leq} \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} \mathbb{E}[\|\nabla_c F_u(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - \boldsymbol{H}_{u,i}^c\|^2]$$
$$\overset{(b)}{\leq} \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} \{2\|\nabla_c F_u(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + 2\|\boldsymbol{H}_{u,i}^c\|^2\}$$
$$\overset{(c)}{\leq} 2\sum_{u=1}^{U} \alpha_{u,i} p_{u,i} [\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + \delta k_u]$$
$$+ 2\sum_{u=1}^{U} \alpha_{u,i} p_{u,i} \|\boldsymbol{H}_{u,i}^c\|^2, \tag{37}$$

where (a) uses Jensen's Inequality, (b) follows inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, and (c) uses bounded similarity assumption in Assumption 3. Similarly, $B_2 \leq 2\sum_{u=1}^{U} \alpha_{u,i} p_{u,i} [\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + \delta k_u] + 2\sum_{u=1}^{U} \alpha_{u,i} p_{u,i} \|\boldsymbol{H}_{u,i}^s\|^2$. We bound $\|\boldsymbol{H}_{u,i}^c\|^2$ and $\|\boldsymbol{H}_{u,i}^s\|^2$:

$$\|\boldsymbol{H}_{u,i}^c\|^2 = \|\frac{1}{J}\sum_{j=0}^{J-1} \nabla_c F_u(\boldsymbol{w}_{u,i,j}^c, \boldsymbol{w}_{u,i,j}^s)\|^2$$
$$\overset{(a)}{\leq} \frac{1}{J}\sum_{j=0}^{J-1} \|\nabla_c F_u(\boldsymbol{w}_{u,i,j}^c, \boldsymbol{w}_{u,i,j}^s)\|^2$$
$$\overset{(b)}{\leq} \frac{1}{J}\sum_{j=0}^{J-1} \{\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + \delta k_u\}, \tag{38}$$

where (a) uses Jensen's Inequality and b follows Assumption 3. Similarly, $\|\boldsymbol{H}_{u,i}^s\|^2 \leq \frac{1}{J}\sum_{j=0}^{J-1}\{\|\nabla_s F(\boldsymbol{w}_{i,0}^c, \boldsymbol{w}_{i,0}^s)\|^2 + \delta k_u\}$. By substituting B_1, B_2, $\|\boldsymbol{H}_{u,i}^c\|^2$ and $\|\boldsymbol{H}_{u,i}^s\|^2$ back into (36), we have (17), where it follows Cauchy-Schwarz inequality and the fact that $\alpha_{u,i}^2 = \alpha_{u,i}$ when $\alpha_{u,i} \in \{0,1\}$. Proof completed.

C. Proof of Corollary 1

According to the L-smooth in Assumption 1,

$$F(\boldsymbol{w}^*) \leq F(\boldsymbol{w} - \frac{1}{L}\nabla F(\boldsymbol{w}))$$
$$\leq F(\boldsymbol{w}) - \langle \nabla F(\boldsymbol{w}), \frac{1}{L}\nabla F(\boldsymbol{w})\rangle + \frac{1}{2L}\|\nabla F(\boldsymbol{w})\|^2$$
$$= F(\boldsymbol{w}) - \frac{1}{2L}\|\nabla F(\boldsymbol{w})\|^2, \tag{39}$$

where $F(\boldsymbol{w}^*)$ denote the optimal loss, i.e., $F(\boldsymbol{w}^*) \leq F(\boldsymbol{w}), \forall \boldsymbol{w}$. By rearranging the inequality, we have

$$\|\nabla F(\boldsymbol{w})\|^2 \leq 2L(F(\boldsymbol{w}) - F(\boldsymbol{w}^*)). \tag{40}$$

Subtracting $F(\boldsymbol{w}^*)$ into both $F(\boldsymbol{w}_{i+1})$ and $F(\boldsymbol{w}_i)$, we have $F(\boldsymbol{w}_{i+1}) - F(\boldsymbol{w}^*) \leq F(\boldsymbol{w}_i) - F(\boldsymbol{w}^*) + \{-\frac{J\eta_c}{2} + 2J\eta_c \sum_{u=1}^{U} \alpha_{u,i} p_{u,i}\}\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + 2J\eta_c\delta \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + (1+\chi)J\eta_c^2\sigma^2 \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2 + \{-\frac{J\eta_s}{2} + 2J\eta_s \sum_{u=1}^{U} \alpha_{u,i} p_{u,i}\}\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 + 2J\eta_s\delta \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + (1+\chi)J\eta_s^2\sigma^2 \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2$. Deriving from inequality (40), we have $\|\nabla_c F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 \leq 2L_c[F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - F(\boldsymbol{w}_i^{c,*}, \boldsymbol{w}_i^{s,*})]$, and $\|\nabla_s F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s)\|^2 \leq 2L_s[F(\boldsymbol{w}_i^c, \boldsymbol{w}_i^s) - F(\boldsymbol{w}_i^{c,*}, \boldsymbol{w}_i^{s,*})]$, substituting them back into $F(\boldsymbol{w}_{i+1}) - F(\boldsymbol{w}^*)$, we have

$$F(\boldsymbol{w}_{i+1}) - F(\boldsymbol{w}^*) \leq F(\boldsymbol{w}_i) - F(\boldsymbol{w}^*)$$
$$+ \{-\frac{J\eta_c}{2} + J\eta_c \sum_{u=1}^{U} \alpha_{u,i} p_{u,i}\}\{2L_c[F(\boldsymbol{w}_i) - F(\boldsymbol{w}^*)]\}$$
$$+ J\eta_c\delta \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + (1+\chi)J\eta_c^2\sigma^2 \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2$$
$$+ \{-\frac{J\eta_s}{2} + J\eta_s \sum_{u=1}^{U} \alpha_{u,i} p_{u,i}\}\{2L_s[F(\boldsymbol{w}_i) - F(\boldsymbol{w}^*)]\}$$
$$+ J\eta_s\delta \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + (1+\chi)J\eta_s^2\sigma^2 \sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2, \tag{41}$$

let $a_1 = 1 - J\eta_c L_c + 2J\eta_c L_c \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} - J\eta_s L_s + 2J\eta_s L_s \sum_{u=1}^{U} \alpha_{u,i} p_{u,i}$, which is constant, $a_2 = J\delta(\eta_c + \eta_s)$, and $a_3 = (1+\chi)J\sigma^2(\eta_c^2 + \eta_s^2)\sum_{u=1}^{U} \alpha_{u,i} \sum_{u=1}^{U} p_{u,i}^2$, we have

$$F(\boldsymbol{w}_{i+1}) - F(\boldsymbol{w}^*)$$
$$\leq a_1[F(\boldsymbol{w}_i) - F(\boldsymbol{w}^*)] + a_2 \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + a_3, \tag{42}$$

By doing the operation of telescoping for the above inequality, we have

$$F(\boldsymbol{w}_{i+1}) - F(\boldsymbol{w}^*) \leq a_1^I[F(\boldsymbol{w}_0) - F(\boldsymbol{w}^*)]$$
$$+ a_2 \sum_{i=1}^{I-1} a_1^{I-1-i} \sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u + \frac{1 - a_1^I}{1 - a_1} a_3. \tag{43}$$

According to the convergence analysis results, we transform problem into minimize $\sum_{i=1}^{I-1}\sum_{u=1}^{U} \alpha_{u,i} p_{u,i} k_u$ in each round for client scheduling, cut layer selection, bandwidth allocation, and aggregation weight determination policies design.

D. Proof of Lemma 2

The first order and the second order of (25) with respect to $\omega_{u,i}$ are $\frac{\partial \mathcal{R}(\boldsymbol{\omega}_i, z)}{\partial \omega_{u,i}} = \frac{BN_0\lambda_{u,i}(T_{\max} - T_{u,i}^*)}{h_{u,i}}(q_u(\omega_{u,i}) + \omega_{u,i}q_u'(\omega_{u,i})) + z = \frac{BN_0\lambda_{u,i}(T_{\max} - T_{u,i}^*)}{h_{u,i}} q_u(\omega_{u,i})(1 - \frac{[\gamma_u^c(s_u) + J\beta\gamma_u^a(s_u)]\ln 2}{\omega_{u,i}B(T_{\max} - T_{u,i}^*)}) + z$, and $\frac{\partial^2 \mathcal{R}(\boldsymbol{\omega}_i, z)}{\partial^2 \omega_{u,i}} = \frac{BN_0\lambda_{u,i}(T_{\max} - T_{u,i}^*)([\gamma_u^c(s_u) + J\beta\gamma_u^a(s_u)]\ln 2)^2}{\omega_{u,i}^3[B(T_{\max} - T_{u,i}^*)]^2 h_{u,i}} q_u(\omega_{u,i}) \geq 0$. Hence, problem $\tilde{\mathcal{P}}_1$ is a typical convex optimization problem. Utilizing the KKT conditions, the Lagrange function of problem $\tilde{\mathcal{P}}_1$ is $\mathcal{R}(\boldsymbol{\omega}_i, z) = \lambda_{u,i}\vartheta(\boldsymbol{\omega}_i) + z(\sum_{u=1}^{U} \omega_{u,i} - 1)$, where z is the Lagrange multiplier associated with (13c). The first order derivative of $\mathcal{R}(\boldsymbol{\omega}_i, z)$ is $\frac{\partial \mathcal{R}(\boldsymbol{\omega}_i, z)}{\partial \omega_{u,i}} = \frac{BN_0\lambda_{u,i}(T_{\max} - T_{u,i}^*)}{h_{u,i}}(q_u(\omega_{u,i}) + \omega_{u,i}q_u'(\omega_{u,i})) + z$. Let $\frac{\partial \mathcal{R}(\boldsymbol{\omega}_i, z)}{\partial \omega_{u,i}} = 0$, we have

$$q_u(\omega_{u,i}) + \omega_{u,i}q_u'(\omega_{u,i}) = -\frac{zh_{u,i}}{BN_0\lambda_{u,i}(T_{\max} - T_{u,i}^*)}, \tag{44}$$

where (26) is the inverse function. With (22a), we have bandwidth allocation policy $\omega_{u,i}^* = \max\{\omega_{u,i}(z), \omega_{u,i}^{\min}\}$. Proof completed.

REFERENCES

[1] C. Xie, Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "DevSFL: Deviation-Aware split federated learning in resource-constrained wireless networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2025, p. 5.97.

[2] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, 2022.

[3] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6g: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, 2022.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, 20–22, Apr. 2017.

[5] Z. Lin, W. Wei, Z. Chen, C.-T. Lam, X. Chen, Y. Gao, and J. Luo, "Hierarchical split federated learning: Convergence analysis and system optimization," *IEEE Trans. Mobile Comput.*, 2025.

[6] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, and *et al.*, "Gemini: A family of highly capable multimodal models," 2025. [Online]. Available: https://arxiv.org/abs/2312.11805

[7] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proc. AAAI*, vol. 36, no. 8, 2022, pp. 8485–8493.

[8] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," 2018. [Online]. Available: https://arxiv.org/abs/1812.00564

[9] Z. Chen, W. Yi, H. Shin, A. Nallanathan, and G. Y. Li, "Efficient wireless federated learning with partial model aggregation," *IEEE Trans. Commun.*, vol. 72, no. 10, pp. 6271–6286, 2024.

[10] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Knowledge-aided federated learning for energy-limited wireless networks," *IEEE Trans. Commun.*, vol. 71, no. 6, pp. 3368–3386, 2023.

[11] P. Li, G. Cheng, X. Huang, J. Kang, R. Yu, Y. Wu, M. Pan, and D. Niyato, "Snowball: Energy efficient and accurate federated learning with coarse-to-fine compression over heterogeneous wireless edge devices," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6778–6792, 2023.

[12] N. Huang, M. Dai, Y. Wu, T. Q. Quek, and X. Shen, "Wireless federated learning with hybrid local and centralized training: A latency minimization design," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 248–263, 2022.

[13] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[14] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh, "Federated learning based on dynamic regularization," in *International conference on learning representations*, 2021.

[15] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "Feddisco: Federated learning with discrepancy-aware collaboration," in *International Conference on Machine Learning*. PMLR, 2023, pp. 39 879–39 902.

[16] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 174–10 183.

[17] J. Zhang, A. Li, M. Tang, J. Sun, X. Chen, F. Zhang, C. Chen, Y. Chen, and H. Li, "Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction," in *International Conference on Machine Learning*. PMLR, 2023, pp. 41 354–41 381.

[18] C. Xie, Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "Tackling class imbalance and client heterogeneity for split federated learning in wireless networks," *IEEE Trans. Wireless Commun.*, 2025.

[19] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, 2019.

[20] W. Wu, M. Li, K. Qu, C. Zhou, X. Shen, W. Zhuang, X. Li, and W. Shi, "Split learning over wireless networks: Parallel design and resource management," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1051–1066, 2023.

[21] C. Xu, J. Li, Y. Liu, Y. Ling, and M. Wen, "Accelerating split federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 5587–5599, 2024.

[22] J. Yan, S. Bi, and Y.-J. A. Zhang, "Optimal model placement and online model splitting for device-edge co-inference," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8354–8367, 2022.

[23] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2021.

[24] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, 2021.

[25] Y. Sun, S. Zhou, Z. Niu, and D. Gndz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 227–242, 2022.

[26] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with cpu-gpu heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, 2021.

[27] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2021.

[28] M. Zhang, G. Zhu, S. Wang, J. Jiang, Q. Liao, C. Zhong, and S. Cui, "Communication-efficient federated edge learning via optimal probabilistic device scheduling," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8536–8551, 2022.

[29] Y. Liang, Q. Chen, G. Zhu, H. Jiang, Y. C. Eldar, and S. Cui, "Communication-and-energy efficient over-the-air federated learning," *IEEE Trans. Wireless Commun.*, 2024.

[30] L. Qiao, Z. Gao, M. B. Mashhadi, and D. Gündüz, "Massive digital over-the-air computation for communication-efficient federated edge learning," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 11, pp. 3078–3094, 2024.

[31] Y. Liang, Q. Chen, R. Li, G. Zhu, M. K. Awan, and H. Jiang, "Communication-and-computation efficient split federated learning in wireless networks: Gradient aggregation and resource management," *IEEE Trans. Wireless Commun.*, 2025.

[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," 2020. [Online]. Available: https://arxiv.org/abs/1907.02189

[33] Z. Chen, W. Yi, H. Shin, and A. Nallanathan, "Adaptive semi-asynchronous federated learning over wireless networks," *IEEE Trans. Commun.*, vol. 73, no. 1, pp. 394–409, 2025.

[34] E. Abbasnejad, J. Q. Shi, and A. van den Hengel, "Deep lipschitz networks and dudley gans," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:125599652

[35] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," *in Proc. Neural Inf. Process. Syst. (NIPS)*, vol. 31, 2018.

[36] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *in Proc. Neural Inf. Process. Syst. (NIPS)*, vol. 33, pp. 7611–7623, 2020.

[37] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," *in Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[38] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers, 2010.

[39] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, 2020.

[40] 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.

[41] C. Huang, G. Tian, and M. Tang, "When minibatch sgd meets splitfed learning: Convergence analysis and performance evaluation," *arXiv preprint arXiv:2308.11953*, 2023.

[42] D. Yan, M. Hu, Z. Xia, Y. Yang, J. Xia, X. Xie, and M. Chen, "Have your cake and eat it too: Toward efficient and accurate split federated learning," *arXiv preprint arXiv:2311.13163*, 2023.