This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3600269

Privacy-Aware Split Federated Learning for LLM Fine-Tuning over Internet of Things

Xiaopei Chen, *Student Member, IEEE*, Wen Wu, *Senior Member, IEEE*, Fei Ji, *Member, IEEE*, Yongguang Lu, and Liang Li, *Member, IEEE*

***Abstract*—The proliferation of Internet of Things (IoT)-generated distributed personal data enables user-specific large language model (LLM) adaptation at the edge. The split federated learning (SFL) facilitates collaborative learning and reduces memory footprint by model splitting, which necessitates the transmission of intermediate activations, rendering it susceptible to reconstruction attacks and privacy breaches. In this paper, we present a privacy-aware SFL scheme addressing the accuracy-efficiency-privacy trilemma in LLM fine-tuning over heterogeneous IoT devices. Particularly, we develop a privacy quantification metric based on Fisher information to assess layer-wise privacy risks in smashed data transmission. Guided by this metric, we establish an analytical model that captures the intricate relationships between privacy leakage, fine-tuning convergence time, and device energy consumption. To optimize these three aspects, we formulate a multi-objective mixed-integer programming problem. Then, an ϵ-constraint-based block coordinate descent (BCD) algorithm is proposed to jointly determine the optimal LLM split layer, transmit power, and bandwidth allocation for IoT devices under their memory and network constraints. Extensive simulation results demonstrate the proposed scheme's effectiveness in achieving 24% faster convergence, 40% lower energy consumption, and 7% reduced privacy leakage compared to baseline approaches, while maintaining competitive model accuracy.**

***Index Terms*—LLM fine-tuning, split federated learning, privacy leakage, multi-objective optimization.**

I. Introduction

The rapid advancement of artificial intelligence (AI) has driven large language models (LLMs), such as GPT and LLaMa, to achieve remarkable progress in natural language processing and intelligent decision-making. These models rely on large-scale, diverse data for continual improvement [1]. Meanwhile, with the rapid growth of the Internet of Things (IoT), the number of global IoT devices is expected to be 3.56 billion [2], generating rich, real-time, and domain-specific data streams [3]. These IoT devices offer rich and diverse data to support LLM development, while LLMs are expected to enhance the intelligence of IoT [4–6]. This convergence of IoT and LLM technologies thus creates unprecedented opportunities to advance smart services and connected applications. As IoT devices persistently generate decentralized data with privacy implications [7, 8], there is a critical requirement to adapt LLMs through localized fine-tuning to meet specialized service needs while ensuring information privacy.

Split federated learning (SFL) offers an efficient framework for fine-tuning LLMs in IoT networks by harnessing distributed device data while overcoming memory limitations [9, 10]. Unlike conventional federated learning (FL) that requires storing the whole model and processing on each device [11], SFL divides the LLM into client-side and server-side submodels [12]. Each IoT device exclusively trains its assigned client submodel, dramatically reducing local memory and processing requiremets, making it particularly suitable for resource-constrained IoT devices. In the SFL scheme, the devices locally update their submodels and the edge server updates the corresponding submodels, where the intermediate outputs (smashed data) and the gradients at the split layer are exchanged between the device and the server. A critical challenge lies in optimizing model split strategies to accommodate devices with varying computational capacities and memory resources. This directly impacts resource efficiency, fine-tuned model accuracy, and the system's ability to mitigate performance bottlenecks caused by slower devices.

In the literature, several pioneering studies [13–18] have explored integrating SFL with parameter-efficient fine-tuning (PEFT) methods like low-rank adaptation (LoRA) [19], where only minimal activated parameters (rather than full model parameters) undergo updates during training. Existing works primarily focus on optimizing training latency [13–15] and energy consumption [15–17] through strategic split layer selection, computation resource scheduling, and communication resource allocation. The core challenge resides in balancing training performance against device resource constraints—shallower split layer reduces client-side computation at the cost of increased resource conflict at the edge server, while deeper split layer alleviates server-side load but impose heavier computational and memory burdens on resource-constrained IoT devices.

However, critical limitations persist in current SFL-based LLM fine-tuning frameworks. *Firstly*, most existing approaches neglect the inherent privacy vulnerabilities in client-

This work was supported in part by the Pengcheng Laboratory Major Key Project under Grant 2025QYA002 and 2025QYB041, in part by the Natural Science Foundation of China under Grant 62201071, 62201311, and 62192712. *(Corresponding author: Wen Wu.)*

Xiaopei Chen is with the School of Future Technology, South China University of Technology, Guangzhou 511442, China, and also with the Department of Strategic and Advanced Interdisciplinary Research, Pengcheng Laboratory, Shenzhen 518000, China (e-mail: ftchenxp@mail.scut.edu.cn).

Wen Wu and Liang Li are with the Department of Strategic and Advanced Interdisciplinary Research, Pengcheng Laboratory, Shenzhen 518000, China (e-mail: {wuw02, lil03 }@pcl.ac.cn).

Fei Ji is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: eefeiji@scut.edu.cn).

Yongguang Lu is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: luyg5@mail2.sysu.edu.cn).

server smashed data exchanges. LLM fine-tuning exhibits the "not-too-far" property [20], meaning that the pre-trained model features remain largely preserved after fine-tuning. This property makes it easier for adversaries to leverage pre-training knowledge to perform data reconstruction attacks, especially when intermediate data transmissions are not protected. *Secondly*, split layer configuration creates a complex tripartite trade-off: Shallow split layer (closer to input) exposes more raw data patterns in smashed outputs, increasing privacy leakage susceptibility, while deeper split layer shifts computational burdens to resource-constrained devices. Furthermore, privacy-enhancing techniques like gradient noise injection, though effective for shallow split layer, disproportionately degrade model convergence compared to their application in deeper split layer. *Thirdly*, the heterogeneous nature of device capabilities and communication conditions compounds these challenges. Coordinating privacy preservation, training efficiency, and energy conservation across devices with varying capabilities forms a non-trivial multi-objective optimization problem. These interdependent constraints resist conventional optimization approaches, particularly when managing diverse IoT nodes with disparate resource profiles and privacy thresholds.

In this paper, we propose a privacy-aware SFL scheme for LLM fine-tuning over heterogeneous IoT devices. Specifically, we develop a privacy quantification metric based on Fisher information to assess layer-wise privacy risks in smash data transmissions. Guided by it, we establish analytical models that capture the intricate relationships between privacy leakage, fine-tuning convergence time, and device energy consumption. To optimize the efficiency-privacy trade-offs, we formulate a multi-objective mixed-integer programming (MIP) problem that aims to minimize these three performance metrics. An ϵ-constraint-based block coordinate descent (BCD) method is proposed to jointly determine optimal split layer, transmit power, and bandwidth allocation for IoT devices under their memory and network constraints. Specifically, the ϵ-constraint method is employed to transform the multi-objective optimization problem into a single-objective problem while ensuring weak Pareto optimality. Leveraging the BCD method, the problem is decomposed into three subproblems. The optimal bandwidth allocation is first derived through monotonicity analysis. Next, the transmit power is determined via a bisection method. Finally, the optimal split layer is obtained using a successive convex approximation (SCA)-based approach. The main contributions of this paper are as follows:

- We establish a unified model for SFL-based LLM fine-tuning that quantifies privacy risk, memory footprint, and energy consumption.
- We formulate a multi-objective mixed-integer programming problem by optimizing split layer, transmit power, and bandwidth allocation.
- We design an efficient ϵ-constraint-based BCD algorithm to solve the multi-objective MIP problem.

The remainder of this paper is organized as follows. Section II presents the related works. In Section III, the system model is presented. The problem formulation and transformation are given in Section IV. Section V proposes the optimization algorithms. The simulation results are shown in Section VI. Section VII concludes this work.

II. Related Work

A. LLM Fine-Tuning over IoT Devices

Recent advances in PEFT have enabled preliminary attempts to fine-tune LLMs on resource-constrained IoT devices. In [21], a split federated LoRA framework is proposed for LLM fine-tuning by integrating LoRA with FL, and an online algorithm for effective device scheduling and bandwidth allocation is developed to enhance learning performance. In [22], a device-edge cooperative fine-tuning paradigm is proposed to fine-tune different layers of an LLM, and the bandwidth and layer are jointly allocated to minimize the training time. In [23], a sparsely activated LoRA algorithm that freezes the pre-trained foundation model parameters and inserts low-rank adaptation matrices into transformer blocks is designed to adapt to the limited computational resources. These approaches maintain a full pre-trained LLM locally for fine-tuning. This demands significant device memory, creating a critical bottleneck for resource-constrained IoT devices that lack the capacity to store and execute full-scale LLMs.

To reduce local computation and memory usage, SL and SFL have been explored to distribute the fine-tuning process. [9] and [10] propose the full fine-tuning-based SFL and LoRA fine-tuning-based SFL, respectively. Since the LoRA approach can reduce computational resource consumption while maintaining model performance. Subsequent studies have incorporated LoRA-based fine-tuning into SL or SFL architectures to facilitate further research [14–17]. In [16], a time and memory-efficient collaborative split fine-tuning framework is proposed to break down the resource wall of fine-tuning LLMs at individual devices, where the training time is minimized by optimizing the split layer selection and data and pipeline parallelization design under the memory constraint. In [14], under an SL architecture, split layer selection and computation resources are optimized to minimize training delay and energy consumption. In [17], a pipeline parallel training mechanism is developed to ensure fast and efficient distributed training, where split layer selection and backend scheduler are designed to minimize memory usage and training delay. In [15], a LoRA fine-tuning-based SFL is considered, and the total energy consumption and training delay are minimized by optimizing the split layer selection, communication resource, and computation resource. Most existing studies primarily focus on improving energy efficiency and training latency, with limited attention to potential privacy risks during client-server interactions.

B. Privacy Leakage in Split Learning

Privacy leakage has been noticed in SL for traditional DNN models. In [24], a privacy-aware adaptive model splitting approach is proposed to balance privacy preservation and communication-computation performance, where the structural similarity index measure (SSIM) is used as an evaluation

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3600269

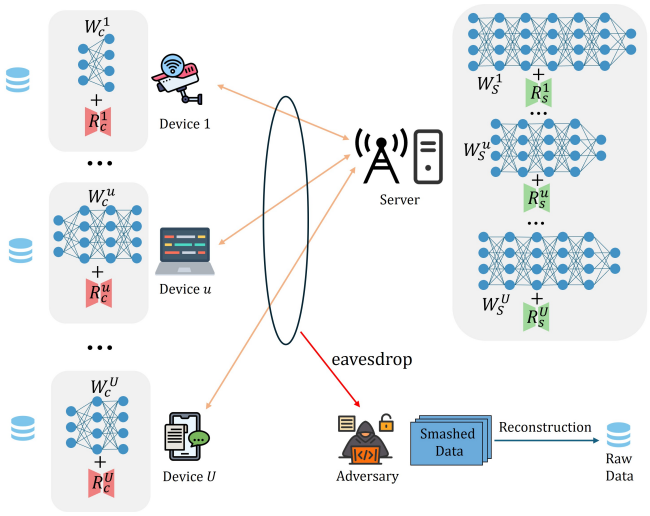


Fig. 1. System model.

metric for data privacy. In [25], a hybrid federated split learning architecture that combines efficiency and privacy is suggested, where the privacy leakage is evaluated by an attacker encoder model. In [2], DNN model partitioning and resource allocation are optimized to reduce the energy consumption under the amount of privacy leakage, where the privacy leakage is defined as the size of the intercepted data. [26] and [27] address privacy-latency and privacy-energy tradeoffs, respectively, through optimized split layer selection. However, existing privacy metrics are largely designed for conventional models and may not generalize to LLMs. For instance, SSIM [24, 26, 27] is tailored to image tasks and is inapplicable to textual data. Intercepted data size metrics [2] lost discriminative power since transformer layers produce identically-sized smashed data. Methods requiring attacker model training [25] impose impractical computational burdens on resource-limited systems. This necessitates simpler, LLM-applicable privacy assessment approaches.

III. System Model

A. Considered Scenario

As shown in Fig. 1, this work considers a typical two-tier IoT network, which consists of an edge server and a set of IoT devices denoted by $\mathcal{U} = \{1, 2, ..., u, ..., U\}$. The edge server with powerful computing capacity C_s is primarily responsible for the model fine-tuning training task and manages resource allocation and training schedule. These IoT devices are heterogeneous, with different computing capabilities and memory. Device u with computing capability C_u and memory M_u has a local dataset D_u for fine-tuning and the local datasets of the devices are non-independent and identically distributed (non-IID). The edge server and the IoT devices collaboratively fine-tune a transformer-based LLM for a specific downstream task. This framework targets intelligent IoT applications such as environmental monitoring, smart healthcare, and interactive smart home systems. These applications require customized and context-aware LLM services, which calls for efficient on-site fine-tuning of pre-trained LLMs at the network edge.

TABLE I: Summary of Notations

Symbol	Description
D_u	Local dataset of client u
C_u	Computing capability of client u
M_u	Available memory of client u
C_s	Computing capability of the edge server
W_c^u	Client sub-model of client u
R_c^u	LoRA module on client u
W_s^u	Server sub-model for client u
R_s^u	LoRA module on the server for client u
l_u	Selected split layer of client u
ψ_u	Privacy leakage of client u
B_u	Bandwidth allocated to client u
P_u	Transmit power of client u
P_s	Transmit power of the server
h_u	Channel gain (client u to server)
n_0	Noise power spectral density
B, S, H	Batch size, sequence length, hidden dimension
$t_u^{\text{up}}, t_u^{\text{down}}$	Upload/download delay of client u
t_u^{comp}	Local computation delay of client u
t_u^{somp}	Server computation delay for client u
E_u^{comp}	Computation energy consumption of client u
E_u^{com}	Communication energy consumption of client u

The system operates under an SFL framework that implements LoRA fine-tuning [19] across edge servers and client devices via model partitioning. Specially, a fully pre-trained LLM with a total of L layers is split into two parts: The client sub-model $\mathbf{W}_c^u$ encompassing layers from the first layer up to layer l_u and the corresponding client-side LoRA $\mathbf{R}_c^u$ are deployed on device (client) u; The server sub-model $\mathbf{W}_s^u$ encompassing layers from layer $l_u + 1$ up to layer L and the corresponding server-side LoRA $\mathbf{R}_s^u$ are deployed on the server. Each client u randomly samples a mini-batch $\{x_u, y_u\} \in D_u$ with batch size B to perform forward propagation, where x_u and y_u represent the input sequences and the corresponding label, respectively. During the forward propagation, the clients compute the initial layers and transmit the smashed data to the server via orthogonal multiple access (OMA) links. The server then completes the remaining forward computation, performs the entire backward propagation, and sends the gradients of smashed data back to the corresponding client. The client subsequently completes its own backward propagation. After several training rounds, each client transmits its LoRA module to the server, where model aggregation and updates are performed. The goal is to learn an optimal LoRA model that minimizes the global loss function across all the clients:

$$\begin{aligned}\mathbf{R}_f^\star &= \arg\min_{\mathbf{R}_f} F(\mathbf{W}_u, \mathbf{R}_u; \{D_u\}) \\ &= \frac{1}{U}\sum_{u=1}^{U} f_u(\mathbf{W}_c^u, \mathbf{R}_c^u; \mathbf{W}_s^u, \mathbf{R}_s^u; \{D_u\}),\end{aligned} \tag{1}$$

where $f_u(\cdot)$ is the local loss function of device u.

In the process of SFL, smashed data becomes exposed in open wireless networks and edge servers, where adversaries may exist, leading to privacy leakage. In this work, we focus

on the data reconstruction attack (DRA), where an adversary attempts to reconstruct the raw data through the captured smashed data. Specifically, given the smashed data z, the adversary aims to reconstruct the raw data $\mathbf{x}$ by an attack function $h(\cdot;\cdot)$ and get $\hat{\mathbf{x}} = h(z;\alpha)$, where α is some auxiliary knowledge (e.g., the model parameters, the input data distribution of victims). Our threat model aligns with the assumptions and methodologies used in most prior DRA studies [25, 28]. To enhance the clarity and consistency of the mathematical expressions, we provide a summary of key notations used throughout the paper in Table I.

B. Privacy Leakage Model

To quantify the information leakage of smashed data in the split layer, we leverage Fisher information that provides an implicit privacy definition by bounding the error of DRA [29–31]. The Fisher information of the split layer l_u is defined as[1]

$$\mathcal{I}_u(\mathbf{x}) = \mathbf{J}^T_{\mathbf{W}^u_c}(\mathbf{x})\mathbf{J}_{\mathbf{W}^u_c}(\mathbf{x}), \tag{2}$$

where $\mathbf{x}$ is the input raw data, $\mathbf{J}_{\mathbf{W}^u_c}(\mathbf{x}) = \frac{\partial f(\mathbf{W}^u_c;\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian of client sub-model $\mathbf{W}^u_c$ with respect to the input $\mathbf{x}$.

Since computing the full matrix $\mathcal{I}_u(\mathbf{x})$ is computationally expensive, we use the trace of the Fisher information matrix $\mathrm{Tr}(\mathcal{I}_u(\mathbf{x}))$ as a surrogate. Prior work has demonstrated that the trace correlates well with the full matrix and captures similar trends in practice [35]. Therefore, we use diagonal Fisher information leakage (DFIL) as the privacy leakage metric, with a lower value indicating stronger privacy protection. The privacy leakage of device u can be given as [29–31]

$$\psi_u = \frac{\mathrm{Tr}(\mathcal{I}_u(\mathbf{x}))}{d}, \tag{3}$$

where d is the dimension of $\mathbf{x}$ and $\mathrm{Tr}(\cdot)$ is the trace of a matrix. When an attacker tries to reconstruct the training data from the trained model, the average lower bound of the reconstruction error can be deduced by DFIL.

C. Communication Model

In the SFL framework mentioned above, it is assumed that orthogonal channels are allocated to different devices. Therefore the data rates from device u to the server and from the server to device u can be expressed, respectively, as

$$R^u_{\mathrm{up}} = B_u \log_2\left(1 + \frac{P_u h_u}{n_0 B_u}\right), \tag{4}$$

$$R^u_{\mathrm{down}} = B_u \log_2\left(1 + \frac{P_s h_u}{n_0 B_u}\right), \tag{5}$$

where B_u is the bandwidth allocated to device u, P_u is the transmit power of device u, P_s is the transmit power of the server, h_u is the channel gain between device u and the server, and n_0 is noise power spectral density.

[1]Prior studies [32–34] have shown that the ranking of the Fisher information across layers remains largely unchanged during fine-tuning. Importantly, we focus only on their relative order to analyze the degree of privacy leakage, rather than relying on their absolute values. Therefore, similar to [32–34], we compute the Fisher information using the pre-trained model before fine-tuning.

In transformer-based models, the data size of the smashed data remains consistent across all layers. Consequently, the corresponding gradient size also stays uniform throughout the layers. Therefore, the transmission delays of smashed data and the corresponding gradients in floating point 32-bit (FP32) precision can be expressed, respectively, as [36–38]

$$t^u_{\mathrm{up}} = \frac{32BSH}{R^u_{\mathrm{up}}}, \tag{6}$$

$$t^u_{\mathrm{down}} = \frac{32BSH}{R^u_{\mathrm{down}}}, \tag{7}$$

where B is the batch size, S is the sequence length, H is the hidden dimension.

Therefore, the energy consumption of wireless transmission for device u is:

$$E^u_{\mathrm{com}} = P_u t^u_{\mathrm{up}}. \tag{8}$$

D. Memory and Computation Model

In transformer-based LLMs, memory usage and computation workload are key considerations when computing models on resource-constrained IoT devices. We consider that the LoRA adapters are configured in the attention modules to reduce the computation and memory pressure. For the client sub-model $\mathbf{W}^u_c$ and the corresponding LoRA $\mathbf{R}^u_c$, the memory usage (in bytes) under FP32 precision can be expressed as [36–38][2]

$$\begin{aligned} M(\mathbf{W}^u_c;\mathbf{R}^u_c) = &\underbrace{l_u(32h^2 + 72h + 16hH + 8H + 32Hr)}_{\text{model parameters}} \\ &+ \underbrace{l_u(68BSH + 10BNS^2 + 8BS(r+H))}_{\text{activations}} \\ &+ \underbrace{l_u(64Hr)}_{\text{optimizer states and gradients}}, \end{aligned} \tag{9}$$

where h is the embedding dimension, H is the hidden dimension, N is the head number, r is the LoRA rank, B is the batch size, S is the sequence length.

Let $A(\mathbf{W}^u_c;\mathbf{R}^u_c)$ denote the computation workload of the client sub-model $\mathbf{W}^u_c$ and the corresponding LoRA $\mathbf{R}^u_c$ during the forward and backward propagation for a batch. Given the split layer l_u, the computation workload $A(\mathbf{W}^u_c;\mathbf{R}^u_c)$ can be given as [15]

$$A(\mathbf{W}^u_c;\mathbf{R}^u_c) = l_u(72BSH^2 + 12BS^2H), \tag{10}$$

where B is the batch size, S is the sequence length, and H is the hidden dimension. The local computation delay can be given by:

$$t^u_{\mathrm{lcomp}} = \frac{A(\mathbf{W}^u_c;\mathbf{R}^u_c)}{C_u}, \tag{11}$$

where $C_u = f_u N_u \theta_u$ is the computing capability of device u. f_u is the GPU frequency of device u, N_u is the number of

[2]Compared to the transformer block, the number of parameters in the embedding and header layers is very small, so we ignore the analysis of these two layers.

cores and θ_u is the number of FLOPs per cycle per core. The corresponding energy consumption is given by [15, 39]

$$E^u_{\text{comp}} = \kappa f_u^3 t^u_{\text{lcomp}}, \tag{12}$$

where κ is the computation constant factor reflecting the power usage per cubic cycle per second.

Similar to the local computation delay, the computation delay of server sub-model $\mathbf{W}_s^u$ and the corresponding LoRA $\mathbf{R}_s^u$ during the forward and backward propagation for a batch can be given as

$$t^u_{\text{scomp}} = \frac{A(\mathbf{W}_s^u; \mathbf{R}_s^u)}{C_s}, \tag{13}$$

where C_s is the computing capability of the server. We assume that the server has sufficient energy and memory resources; therefore, we do not analyze its energy consumption and memory usage in this work.

IV. Problem Formulation and Transformation

A. Problem Formulation

During the model aggregation phase, each client uploads its corresponding LoRA adapters to the server for model aggregation. Due to their small size, the aggregation time of the LoRA adapters can be negligible. With the computation and communication model above, the training time for all models to complete training for one round can be expressed as

$$T = \max_{u \in \mathcal{U}} (t^u_{\text{lcomp}} + t^u_{\text{scomp}} + t^u_{\text{up}} + t^u_{\text{down}}). \tag{14}$$

The energy consumption of the devices for one round can be given as

$$E = \sum_{u \in \mathcal{U}} (E^u_{\text{com}} + E^u_{\text{comp}}). \tag{15}$$

The privacy leakage of the devices can be given as

$$\Psi = \sum_{u \in \mathcal{U}} \psi_u. \tag{16}$$

We aim to minimize the training time, the energy consumption of the devices, and the privacy leakage, by optimizing the split layer $\boldsymbol{L} = \{l_u\}_{\forall u \in \mathcal{U}}$, bandwidth allocation $\boldsymbol{B} = \{B_u\}_{\forall u \in \mathcal{U}}$ and transmit power of the devices $\boldsymbol{P} = \{P_u\}_{\forall u \in \mathcal{U}}$. The joint optimization problem can be formulated as follows:

$$\mathbf{P1}: \min_{\boldsymbol{L},\boldsymbol{B},\boldsymbol{P}} \quad \{T(\boldsymbol{L},\boldsymbol{B},\boldsymbol{P}), E(\boldsymbol{L},\boldsymbol{B},\boldsymbol{P}), \Psi(\boldsymbol{L})\} \tag{17a}$$

$$\text{s.t.} \quad l_u \in \{1, \ldots, L\}, \quad \forall u \in \mathcal{U}, \tag{17b}$$

$$P_u \in [0, P_{max}], \quad \forall u \in \mathcal{U}, \tag{17c}$$

$$B_u \in [0, B_{all}], \quad \forall u \in \mathcal{U}, \tag{17d}$$

$$\sum_{u \in \mathcal{U}} B_u \le B_{all}, \tag{17e}$$

$$M(\mathbf{W}_c^u; \mathbf{R}_c^u) \le M_u, \quad \forall u \in \mathcal{U}. \tag{17f}$$

It can be observed that the problem **P1** is a multi-objective mixed-integer programming (MIP) problem. Constraint (17b) restricts model splitting (or, equivalently, split layer selection). Constraint (17c) restricts the transmit power. Constraints (17d) (17e) ensure the communication bandwidth. Constraint (17f) ensures that the model execution does not exceed the available memory capacity. Solving this problem is challenging due to the combinatorial explosion caused by integer decision variables, which significantly increases computational complexity. Additionally, handling multiple conflicting objectives requires careful consideration of trade-offs and the implementation of efficient optimization techniques to derive Pareto-optimal solutions.

B. Problem Transformation

We first transform the MIP multi-objective optimization problem **P1** into a single-objective optimization problem. There are currently various methods to achieve this goal, such as the linear weighted sum [40] and the Tchebycheff method [41]. Here, we adopt the ϵ-constraint method [42], which at least guarantees weak Pareto optimality and ensures that the transformed problem retains practical physical significance. By introducing an auxiliary variable $\widetilde{T}$, the single-objective optimization problem is as follows:

$$\mathbf{P2}: \min_{\boldsymbol{L},\boldsymbol{B},\boldsymbol{P},\widetilde{\boldsymbol{T}}} \quad \widetilde{T} \tag{18a}$$

$$\text{s.t.} \quad (17b) - (17f), \tag{18b}$$

$$E \le \epsilon_1, \tag{18c}$$

$$\Psi \le \epsilon_2, \tag{18d}$$

$$t^u_{\text{lcomp}} + t^u_{\text{scomp}} + t^u_{\text{up}} + t^u_{\text{down}} \le \widetilde{T}, \forall u \in \mathcal{U}. \tag{18e}$$

Lemma 1. *The solution to problem **P2** is at least weakly Pareto optimal for problem **P1**.*

Proof. Please refer to Appendix A. □

The above lemma demonstrates the effectiveness of the problem transformation. However, problem **P2** still involves non-convex constraints (18c) and (18d) as well as integer variables $\mathbf{L}$, and is generally classified as an NP-hard problem. To address this, we design an iterative algorithm capable of efficiently obtaining near-optimal solutions. The algorithm is based on the block coordinate descent (BCD) method, where problem **P2** is decomposed into three subproblems. In each subproblem, one optimization variable is updated while the other two are fixed. These subproblems are solved iteratively until convergence.

V. Algorithm Design

A. Subproblem 1: Optimize Bandwidth Allocation with Fixed Transmit Power and Split Layer

This subproblem optimizes the bandwidth allocation while keeping the transmission power and split layer fixed. It can be formulated as follows:

$$\mathbf{P3}: \min_{\boldsymbol{B},\widetilde{\boldsymbol{T}}} \quad \widetilde{T} \tag{19a}$$

$$\text{s.t.} \quad (17d) - (17f), \tag{19b}$$

$$E \le \epsilon_1, \tag{19c}$$

$$t^u_{\text{lcomp}} + t^u_{\text{scomp}} + t^u_{\text{up}} + t^u_{\text{down}} \le \widetilde{T}, \forall u \in \mathcal{U}. \tag{19d}$$

The function $f(x) = 1/xlog_2(1+\frac{\gamma}{\lambda x})$ is convex for all $\gamma > 0$, $\lambda > 0$, and $x > 0$. Given this, the objective function and all constraints in problem **P3** are either convex or linear. Therefore, problem **P3** is a standard convex optimization problem that can be efficiently solved using existing optimization tools, such as CVX with an interior-point method.

B. Subproblem 2: Optimize Transmit Power with Fixed Bandwidth and Split Layer

This subproblem focuses on optimizing the transmit power under fixed bandwidth and split layer settings, and is formulated as:

$$\mathbf{P4}: \min_{\boldsymbol{P}, \widetilde{\boldsymbol{T}}} \quad \widetilde{T} \tag{20a}$$

$$\text{s.t.} \quad (17c), \tag{20b}$$

$$E \le \epsilon_1, \tag{20c}$$

$$t^u_{\text{lcomp}} + t^u_{\text{scomp}} + t^u_{\text{up}} + t^u_{\text{down}} \le \widetilde{T}, \forall u \in \mathcal{U}. \tag{20d}$$

The function $f(x) = \frac{\alpha x}{\beta log_2(1+\frac{\gamma x}{\lambda})}$ is a concave function when $\alpha, \beta, \gamma, \lambda, x > 0$. Given this, constraint (20c) is non-convex, and then problem **P4** is a non-convex problem, which makes it difficult to obtain the optimal solution. Fortunately, we observe that the constraints (20c) and (20d) possess certain special monotonic properties. Based on these properties, we can apply a bisection method to find the optimal solution of problem **P4** in polynomial time. The pseudocode of the algorithm is provided in Algorithm 1.

Given the fixed variables for bandwidth allocation and split layer selection, computation time bounds become solely dependent on transmit power through a monotonic relationship. This monotonicity enables efficient determination of feasible computation time ranges satisfying all constraints. The outer `while` loop identifies the optimal computation time, and the inner `while` loop determines the power P_i that makes (18e) hold with equality for each client.

Lemma 2. *Algorithm 1 can obtain the optimal solution for problem **P4** with sub-polynomial time.*

Proof. Function $f(x) = \frac{\alpha}{\beta log_2(1+\frac{\gamma x}{\lambda})}$ is a decreasing function when $\alpha, \beta, \gamma, \lambda > 0$, and function $f(x) = \frac{\alpha x}{\beta log_2(1+\frac{\gamma x}{\lambda})}$ is an increasing function at the same situation. Therefore, at the optimal solution of problem **P4**, constraint (20c) must be satisfied with equality. Otherwise, reducing the transmit power to lower energy consumption would lead to an increase in task completion time, thereby violating optimality. Accordingly, the inner loop of Algorithm 1 employs a bisection method to efficiently determine the transmit power that enforces equality in (20c). Similarly, the optimization variable $\widetilde{T}$ in the outer loop exhibits a monotonic relationship, allowing its optimal value to be found via bisection as well. The overall time complexity of the bisection method is sub-polynomial, ensuring computational efficiency. □

Algorithm 1: Bisection Method for Subproblem 2

Input: An upper bound $\widetilde{T}_{\max}$ and lower bound $\widetilde{T}_{\min}$ of computation time, convergence accuracy δ.

Output: Transmit power $\mathbf{P}$.

while $\widetilde{T}_{\max} - \widetilde{T}_{\min} \ge \delta$ **do**
  $\widetilde{T} = (\widetilde{T}_{\max} + \widetilde{T}_{\min})/2$;
  for $i = 1, 2, \ldots, U$ **do**
    $P^{\min} = 0$, $P^{\max} = P_{\max}$;
    while $P^{\max} - P^{\min} \ge \delta$ **do**
      $P_i = (P^{\max} + P^{\min})/2$;
      Calculate the total time T_i based on Eq. (14);
      if $T_i \ge \widetilde{T}$ **then**
        $P^{\min} = P_i$;
      end
      else
        $P^{\max} = P_i$;
      end
    end while
  end for
  Calculate the total energy E based on Eq. (15);
  if $E \le \epsilon_1$ **then**
    $\widetilde{T}_{\max} = \widetilde{T}$;
  end
  else
    $\widetilde{T}_{\min} = \widetilde{T}$;
  end
end while

C. Subproblem 3: Optimize Split Layer with Fixed Transmit Power and Bandwidth

This subproblem can be written as:

$$\mathbf{P5}: \min_{\boldsymbol{L}, \widetilde{\boldsymbol{T}}} \quad \widetilde{T} \tag{21a}$$

$$\text{s.t.} \quad (17b), (17f), \tag{21b}$$

$$E \le \epsilon_1, \tag{21c}$$

$$\Psi \le \epsilon_2, \tag{21d}$$

$$t^u_{\text{lcomp}} + t^u_{\text{scomp}} + t^u_{\text{up}} + t^u_{\text{down}} \le \widetilde{T}, \forall u \in \mathcal{U}. \tag{21e}$$

This problem involves integer optimization variables $\boldsymbol{L}$ and is further complicated by constraint (21d), which introduces a nonlinear lookup-table condition. Compared to the other constraints, this nonlinearity significantly increases the difficulty of solving the problem **P5**.

To solve the problem **P5**, we introduce an auxiliary variable $Z = \{z_{i,j} \in \{0, 1\}\}$ of dimension $L \times U$ to represent the split layer selections for each client. We reformulate the problem

P5 into a 0-1 integer linear programming problem as follows

$$\mathbf{P6}: \min_{\boldsymbol{Z},\widetilde{\boldsymbol{T}}} \quad \widetilde{T} \tag{22a}$$

$$\text{s.t.} \quad \sum_{u}^{U} E_{\text{com}}^{u} + \sum_{u}^{U} k f_u^3 \tau_{\text{lcomp}}^{u} \mathbf{\Gamma Z} \leq \epsilon_1, \tag{22b}$$

$$\sum_{u}^{U} \mathbf{\Lambda Z} \leq \epsilon_2, \tag{22c}$$

$$\tau_{\text{com}}^{u} + \tau_{\text{lcomp}}^{u} \mathbf{\Gamma Z} + \tau_s(\mathbf{\Gamma} - \mathbf{\Gamma Z}) \leq \widetilde{T}, \forall u \in \mathcal{U}, \tag{22d}$$

$$\sum_{u}^{U} m \mathbf{\Gamma Z} \leq M_u, \forall u \in \mathcal{U}, \tag{22e}$$

$$z_{l,u} \in \{0, 1\}, \tag{22f}$$

$$\sum_{l}^{L} z_{l,u} = 1, \forall u \in \mathcal{U}, \tag{22g}$$

where $\Gamma = [1, 2, 3, ..., L]$ is a row vector representing the selectable number of layers; thus, $\mathbf{\Gamma Z}$ is also a row vector. Λ is a row vector representing the privacy leakage risk value of each layer. $\tau_{\text{com}}^{u} = t_{\text{up}}^{u} + t_{\text{down}}^{u}$ is the transmission delay. $\tau_{\text{lcomp}}^{u} = (72BSH^2 + 12BS^2H)/C_u$is the local computation time of each layer, $\tau_s = (72BSH^2 + 12BS^2H)/C_s$ is the server's computation time of each layer. m is the memory usage of each layer. The rest of the parameters are all fixed values when the bandwidth and power are fixed.

Then, we perform the linear relaxation on $\mathbf{Z}$ and add a penalty term to the objective function. The problem is formulated as follows:

$$\mathbf{P7}: \min_{\boldsymbol{Z},\widetilde{\boldsymbol{T}}} \quad \widetilde{T} + \theta \sum_{u}^{U} \sum_{l}^{L} (z_{l,u} - z_{l,u}^2) \tag{23a}$$

$$\text{s.t.} \quad 0 \leq z_{l,u} \leq 1, \tag{23b}$$

$$(22\text{b}) - (22\text{e}), (22\text{g}). \tag{23c}$$

Lemma 3. *Problem **P7** is equivalent to problem **P6** when θ is large sufficiently.*

Proof. Please refer to Appendix B. □

Theoretically, it needs $\sum_{u}^{U} \sum_{l}^{L} (z_{l,u} - z_{l,u}^2) = 0$ which means $\theta \to +\infty$, and this is unrealistic. However, in numerical, it is enough to accept a tolerance for some small η to make $\sum_{u}^{U} \sum_{l}^{L} (z_{l,u} - z_{l,u}^2) \leq \eta$ with a large enough value of θ.

Eq. (23a) is a concave function, and we can obtain an approximate solution by minimizing its upper bound. By applying successive convex approximation (SCA), (23a) can be rewritten as:

$$\widetilde{T} + \theta \sum_{u}^{U} \sum_{l}^{L} (z_{l,u}^{fea})^2 + z_{l,u}(1 - 2z_{l,u}^{fea}), \tag{24}$$

where $z_{l_u}^{fea}$ is an initial first-order Taylor feasible point. After replacing (23a) with (24), the problem **P7** can be solved iteratively by Algorithm 2. Algorithm 2 can ensure convergence to Fritz John's point, which has been proved in [43].

Algorithm 2: SCA Method for Subproblem 3

Input: Initial solution Z_0, number of iterations N, convergence accuracy δ.

Output: Split layer $\boldsymbol{L}$.

```
Set iteration number i = 0;
while T_{i-1} - T_i ≥ δ and i ≤ N do
    i = i + 1;
    Z_fea = Z_{i-1};
    Replacing (23a) with (24);
    Get Z_i by solving (23) with inter-point method;
    L = ΓZ;
    Calculate complete time T_i;
end while
```

Algorithm 3: ϵ-Constraint-based BCD Algorithm

Input: Initial solution $\mathbf{L}, \mathbf{P}, \mathbf{B}$, number of iterations N, convergence accuracy δ, zero martix Z.

Output: Split layer $\boldsymbol{L}$, transmit power $\mathbf{P}$, and bandwidth allocation $\mathbf{B}$.

```
Iteration number i = 0;
while The change of task complete time after each
  iteration ≥ δ and i ≤ N do
    i = i + 1;
    Z(L) = 1;
    Updating L by Alg. 2 with parameters Z;
    Updating P by Alg. 1;
    Updating B by solving problem P3 with
      inter-point method;
    Calculate task complete time;
end while
```

Combining BCD method and SCA method, we can obtain an approximate solution to problem **P2**. The overall algorithm to solve problem **P2** is shown by Algorithm 3.

D. Computational Complexity

The complexity of Algorithm 1 is $\mathcal{O}((log\frac{1}{\delta})^2)$, where δ is the convergence accuracy. Based on the SCA method, the complexity of Algorithm 2 is $\mathcal{O}((LU)^{3.5})$, where L is the number of layers and U is the number of devices. In Algorithm 3, the complexity of solving problem (19) is $\mathcal{O}(U^{3.5})$ due to the inter-point method. Since Algorithm 3 iteratively invokes Algorithm 1 and Algorithm 2, and its most computationally intensive step is solving problem **P3** with an order of $U^{3.5}$, the overall complexity of Algorithm 3 remains polynomial with a dominant order of 3.5. This indicates that although multiple subproblems are involved, the algorithm scales polynomially with respect to the number of devices and layers, making it feasible for practical deployment.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3600269

TABLE II: Simulation parameters

Parameter	Value
B_{all}, bandwidth of the system	40 MHz
P_{max}, maximum transmit power of each device	0.5 w
P_s, transmit power of the server	10 w
M_u, available memory of each device	1 GB
κ, computation constant factor	10^{-27}
n_0, noise power spectral density	-174 dbm/Hz
B, batch size	16
S, sequence length	128
H, hidden dimension	768
h, embedding dimension	768
r, LoRA rank	16
N, head number	12
Learning rate	1e-5

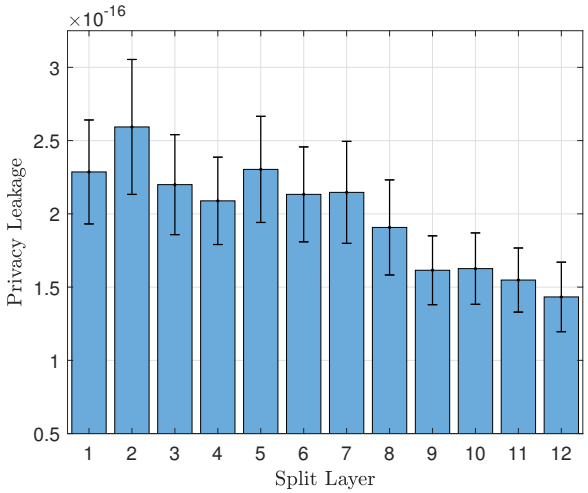


Fig. 2. Privacy leakage of different split layer.

VI. Performance Evaluation

A. Simulation Settings

We leverage Bert-base [44] as the pre-trained model for text analysis tasks using CARER dataset [45]. To simulate realistic non-IID data distributions across clients, we adopt a label distribution skewing approach based on the Dirichlet distribution, where the concentration parameter is set to 0.5. A square area of 10m × 10m is considered, where the server is deployed in the center and the clients are randomly deployed. The channel gain is assumed to follow the path-loss model, and the path-loss factor is set to 2. We use an RTX 4080s server with a computational capability of 52.2 TFLOPS and consider six heterogeneous clients: a Jetson Nano (0.472 TFLOPS, 0.921 GHz), a Jetson TX2 (1.33 TFLOPS, 1.3 GHz), a Snapdragon 8s Gen 3 (1.689 TFLOPS, 1.1 GHz), a Snapdragon 8 Gen 3 (2.774 TFLOPS, 0.903 GHz), an A17 Pro (2.147 TFLOPS, 1.4 GHz), and an M3 (3.533 TFLOPS, 1.6 GHz). Other related parameters are summarized in Table II.

To evaluate the effectiveness of the proposed scheme, the following three baseline schemes are considered.

- Time-myopic SL [14]: Time-myopic SL enables collaborative training by splitting the model between the client and the server, where the client only processes a portion of the model. The split layer design is optimized for minimizing the training time.
- Privacy-myopic SL: Under the SL framework, the split layer is designed by the proposed method for minimizing the privacy leakage.
- Federated learning (FL) [21]: FL enables distributed training by allowing each client to train the full model independently and periodically aggregate updates through the server.
- NSGA-II algorithm [46]: NSGA-II is a widely used evolutionary algorithm designed for solving multi-objective optimization problems. It identifies a set of Pareto-optimal solutions through fast non-dominated sorting and crowding distance mechanisms to maintain diversity and convergence.

B. Simulation Results

Figure 2 illustrates the privacy leakage associated with different split layers of the Bert-base. The privacy leakage denotes the mean DFIL across 64,000 data samples, and the error bars indicate the standard deviation, reflecting the variance in privacy leakage across trials. As observed, privacy leakage exhibits a decreasing trend with increasing split depth. Shallow split layers (e.g., layers 1-3) result in significantly higher values, indicating that early-layer activations retain more sensitive information that may be exploited for data reconstruction attacks. In contrast, deeper split layers (e.g., layers 9-12) yield lower values, suggesting reduced exposure of private information when the model is partitioned closer to the output.

Table III presents the memory usage, privacy leakage, fine-tuning time, energy consumption, and accuracy of different fine-tuning schemes. All schemes achieve comparable accuracy, indicating that neither resource-efficient nor privacy-preserving optimizations degrade fine-tuning performance. As shown, FL requires the highest per-device memory usage (2118.7 MB) and even encounters out-of-memory (OOM) issues on typical IoT devices. In contrast, both SL- and SFL-based schemes significantly reduce memory consumption, making them more suitable for resource-constrained environments. In terms of fine-tuning time, the proposed privacy-aware SFL achieves the fastest convergence (35,336 seconds), outperforming time-myopic SL (47,478 seconds) and privacy-myopic SL (67,199 seconds). This demonstrates the efficiency of collaborative training and parallel updates in SFL. Regarding privacy leakage, privacy-myopic SL and privacy-aware SFL achieve lower leakage values compared to time-myopic SL, validating the effectiveness of privacy-driven optimization strategies. Furthermore, under the same privacy objective, the proposed SFL framework reduces convergence time by approximately 47.4% compared to privacy-myopic SL. The superior performance of privacy-aware SFL can be attributed to its parallel client update mechanism, which alleviates the sequential bottleneck inherent in SL and enables more efficient

TABLE III: Performance Comparison of Different Schemes

Schemes	Device's Memory Usage (MB)	Privacy Leakage (1e-15)	Fine-Tuning Time (s)	Energy Consumption (J)	Accuracy
FL	2118.7 (OOM)	0.85974	68996	224730	0.883
Time-myopic SL	176.56	1.3716	47478	21159	0.881
Privacy-myopic SL	706.23	1.2532	67199	50525	0.881
Privacy-aware SFL (Ours)	706.23	1.2532	35336	92045	0.883

utilization of distributed computational resources.

Figure 3 illustrates the training accuracy versus training time under different schemes. It can be clearly observed that the proposed privacy-aware SFL achieves the fastest convergence among all schemes while maintaining comparable final accuracy. Specifically, it reduces the training time by approximately 48.76% compared to FL and by 47.42% compared to privacy-myopic SL, benefiting from parallel client updates and optimized resource allocation. Compared to time-myopic SL, the privacy-aware SFL also shows a significant acceleration of about 25.57%, further validating its efficiency advantage. Although privacy-aware schemes generally converge more slowly than time-myopic schemes due to deeper split layers for privacy protection, the proposed privacy-aware SFL effectively overcomes this drawback, striking a balance between privacy and efficiency.

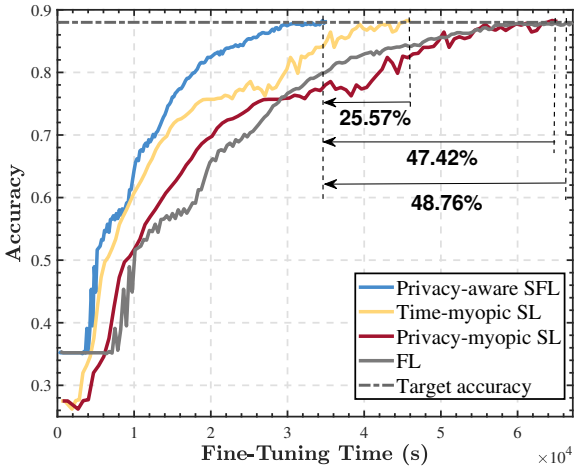


Fig. 3. Training accuracy versus fine-tuning time under different schemes.

To further evaluate the overall resource efficiency of the compared schemes, Fig. 4 presents a joint comparison of total energy consumption and convergence time for all participating devices. Specially, compared to conventional FL, both SFL and SL can significantly reduces convergence time and energy consumption of the devices. When compared with SL, SFL methods exhibit a clear trade-off between energy and latency. The higher energy consumption of SFL mainly results from the increased number of training rounds required for federated aggregation, which leads to more frequent local computations and communication. Although SFL introduces additional energy overhead due to federated updates, it provides substantial convergence speedup. Although SFL consumes 1.82× more energy than SL at the same privacy level, it reduces the training time to merely 52.62%, highlighting its efficiency advantage. These results clearly demonstrate that the proposed SFL framework delivers substantial acceleration in training at the cost of moderate additional energy, offering a favorable trade-off for latency-sensitive applications in IoT environments.

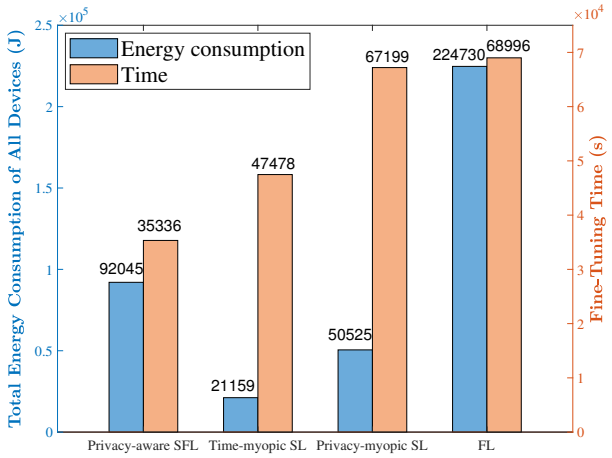
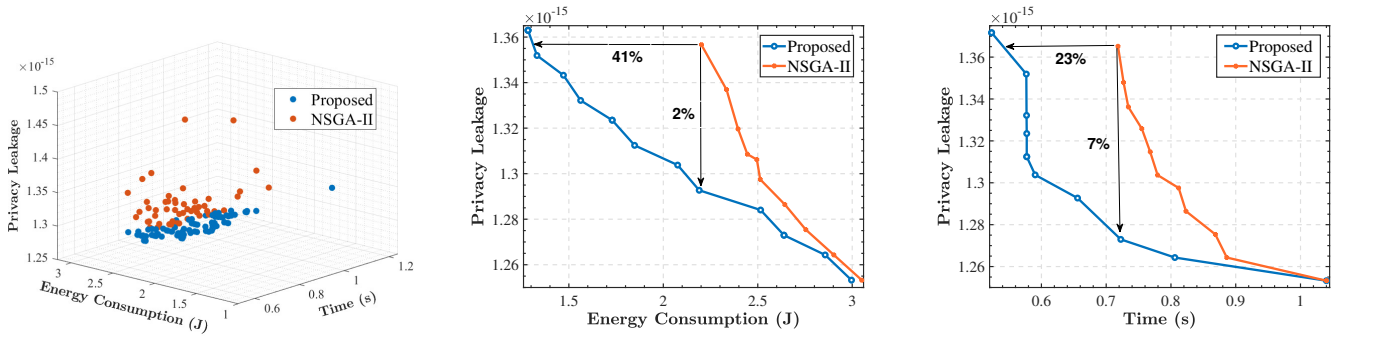


Fig. 4. Comparison of energy consumption and fine-tuning time across different schemes.

Figure 5a illustrates the Pareto fronts representing the trade-offs among privacy leakage, energy consumption, and training time for the proposed method and the NSGA-II baseline. Each point represents a solution obtained by the respective algorithm. Compared with NSGA-II, the proposed method consistently achieves lower privacy leakage and energy consumption while maintaining shorter training time. It can be observed that the solutions of the proposed method are predominantly distributed toward the front and lower regions of the plot, indicating superior performance across all objectives. These results demonstrate that the proposed optimization framework effectively balances the multi-objective trade-offs essential for efficient and privacy-preserving LLM fine-tuning over IoT devices.

To better illustrate the performance gains of the proposed method, the 3D Pareto front is projected onto two planes: the privacy leakage–energy consumption plane and the privacy leakage–training time plane. As shown in Fig. 5b, on the privacy leakage–energy consumption plane, the proposed method consistently outperforms NSGA-II, achieving up to 41% reduction in energy consumption and a 2% improvement in privacy leakage. Similarly, Fig. 5c presents the projection onto the privacy leakage–training time plane, where the proposed method achieves up to 23% reduction in training time and 7% improvement in privacy leakage compared to NSGA-II. These results demonstrate that the proposed optimization framework

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2025.3600269



(a) 3D Pareto fronts

(b) Pareto front projection onto the privacy leakage–energy consumption plane.

(c) Pareto front projection onto the privacy leakage–time plane.

Fig. 5. Privacy-energy-time trade-off comparison of the proposed scheme and NSGA-II.

better balances the multi-objective trade-offs, leading to more energy-efficient, faster, and privacy-preserving fine-tuning of LLMs over IoT devices.

VII. Conclusion

In this paper, we have presented a privacy-aware and resource-efficient SFL scheme for fine-tuning LLMs on heterogeneous IoT devices. We have introduced a Fisher information-based metric to quantify layer-wise privacy risks and developed analytical models capturing the relationships among privacy leakage, convergence time, and energy consumption. We have proposed an ϵ-constraint-based BCD method that jointly optimizes the split layer, transmit power, and bandwidth allocation under device resource constraints, transforming the multi-objective problem into tractable sub-problems while ensuring weak Pareto optimality. Extensive experiments confirmed that the proposed scheme can effectively accelerate convergence, reduce energy consumption, and mitigate privacy leakage while maintaining competitive accuracy. This work enables privacy-preserving LLM fine-tuning on resource-constrained IoT devices, promoting intelligent and personalized edge services. In future work, we will explore dynamic model re-partitioning and adaptive privacy protection to further enhance flexibility.

Appendix A
Proof of Lemma 1

First, note that the problem **P2** is equivalent to minimizing the worst-case task completion time. Indeed, if at any feasible point, $\widetilde{T}$ were strictly larger than $\max_{u\in\mathcal{U}}(t^u_{\text{lcomp}} + t^u_{\text{scomp}} + t^u_{\text{up}} + t^u_{\text{down}})$, one could always decrease $\widetilde{T}$ to reduce the objective until constraint (18e) becomes active (i.e. holds with equality). Then, denote by $T(X)$ the maximum task completion time, $E(X)$ the total energy consumption, and $\Psi(X)$ the privacy leakage associated with any feasible solution X. Let X^* be an optimal solution to problem **P2**. For the sake of contradiction, suppose that X^* is not weakly Pareto optimal. Then there exists another feasible solution X such that $T(X) < T(X^*), E(X) < E(X^*), \Psi(X) < \Psi(X^*)$. Since $\Psi(X^*) \le \epsilon_1$ and $E(X^*) \le \epsilon_2$, it follows that $T(X) < T(X^*) \le \epsilon_1, E(X) < E(X^*), \Psi(X) < \Psi(X^*) \le \epsilon_2$, so X also satisfies all feasibility constraints of the problem **P2**. But then $T(X) < T(X^*)$ contradicts the assumption that X^* minimizes the objective of the problem **P2**. Therefore, X^* must be weakly Pareto optimal.

Appendix B
Proof of Lemma 3

This lemma can be proven by using Lagrangian duality. Firstly, the problem **P7** can be viewed as the Lagrangian dual problem of the following problem.

$$\mathbf{P8}: \min_{\boldsymbol{Z},\widetilde{\boldsymbol{T}}} \quad \widetilde{T} \tag{25a}$$

$$\text{s.t.} \quad 0 \le z_{l,u} \le 1, \tag{25b}$$

$$z_{l,u} - z_{l,u}^2 \le 0, \tag{25c}$$

$$(22\text{b}) - (22\text{e}), (22\text{g}). \tag{25d}$$

It can be seen the problem **P8** has the same optimal solution as the problem **P6**, because $z_{l,u} \in \{0,1\}^{LU}$ is the union set of $[0,1]^{LU}$ and $\{z_{l,u} : z_{l,u} - z_{l,u}^2 \le 0\}$. Let $A(\theta)$ and $(\widetilde{T}_u, z_u)$ be the optimal value and solution of the problem **P7** for a given θ. Then, using the Lagrangian $\mathcal{L}(\widetilde{T}, z, \theta) = \widetilde{T} + \theta \sum_u^U \sum_l^L (z_{l,u} - z_{l,u}^2)$ with only one Lagrangian multiplier to handle the difference convex constraint (24c). Therefore, the optimal value of **P8** can be written as $A^* = \min_{\widetilde{T},z} \max_{\theta>0} \mathcal{L}(\widetilde{T}, z, \theta)$, and its Lagranginal duality problem is $\sup_{\theta>0} \min_{\widetilde{T},z} \mathcal{L}(\widetilde{T}, z, \theta)$.

Since the sequence $\{(\widetilde{T}_u, z_u)\}_{u\ge 0}$ is bounded, it has convergent subsequences. We assume $(\widetilde{T}_u, z_u) \to (\widetilde{T}_\infty, z_\infty)$ with $\sum_u^U \sum_l^L (z_{l,u}^\infty - (z_{l,u}^\infty)^2) = 0$, otherwise, $A(\theta) \to +\infty$, which is contradicts to the weak duality, i.e., $sup_{\theta>0} A(\theta) = \sup_{\theta>0} \min_{\widetilde{T},z} \mathcal{L}(\widetilde{T}, z, \theta) \le A^* = \min_{\widetilde{T},z>0} \max_{\theta\ge 0} \mathcal{L}(\widetilde{T}, z, \theta)$, where A^* is the optimal value of **P8**. This shows that $(\widetilde{T}_\infty, z_\infty)$ is feasible to **P8**, and we can obtain $sup_\theta A(\theta) \ge A(\theta) = \widetilde{T}_\theta + \theta Z_\theta \ge A^*$. Letting $\theta \to +\infty$, we can obtain $sup_\theta A(\theta) \ge \widetilde{T}_{+\infty} \ge A^*$. Finally, combining the weak duality, we have $sup_\theta A(\theta) = A^*$, which proves $(\widetilde{T}_\infty, z_\infty)$ is an optimal solution of **P6**. The proof also demonstrates that strong duality holds.

REFERENCES

[1] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv:2001.08361*, 2020.

[2] S. Han, W. Zhang, X. Xu, B. Wang, M. Sun, X. Tao, and P. Zhang, "S2E-DECI: Secrecy and energy-efficient dual-aware device-edge co-inference for AIoT," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 39 142–39 157, 2024.

[3] ITU, "Framework and overall objectives of the future development of imt for 2030 and beyond," ITU, Tech. Rep. ITU-R M.2160-0, 2023.

[4] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 2022.

[5] W. Wu, C. Zhou, M. Li, H. Wu, H. Zhou, N. Zhang, S. Xuemin, and W. Zhuang, "AI-native network slicing for 6G networks," *IEEE Wireless Commun.*, vol. 29, no. 1, p. 96–103, 2022.

[6] X. Chen, W. Wu, L. Li, and F. Ji, "LLM-empowered IoT for 6G networks: Architecture, challenges, and solutions," *IEEE Internet of Things Mag.*, early access, 2025.

[7] Z. Liu, L. Hu, T. Zhou, Y. Tang, and Z. Cai, "Prevalence overshadows concerns? understanding chinese users' privacy awareness and expectations towards LLM-based healthcare consultation," in *Proc. IEEE Symp. Secur. Privacy*, 2025, pp. 2716–2734.

[8] L. Hu, T. Zhou, Z. Liu, F. Liu, and Z. Cai, "Split learning on segmented healthcare data," *IEEE Trans. Big Data*, early access, 2025.

[9] Y. Tian, Y. Wan, L. Lyu, D. Yao, H. Jin, and L. Sun, "FedBERT: When federated learning meets pre-training," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, 2022.

[10] Z. Lin, X. Hu, Y. Zhang, Z. Chen, Z. Fang, X. Chen, A. Li, P. Vepakomma, and Y. Gao, "SplitloRA: A split parameter-efficient fine-tuning framework for large language models," *arXiv:2407.00952*, 2024.

[11] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *Proc. IEEE INFOCOM*, 2021, pp. 1–10.

[12] 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.

[13] S. Zhang, G. Cheng, W. Wu, X. Huang, L. Song, and X. Shen, "Split fine-tuning for large language models in wireless networks," *IEEE J. Sel. Topics Signal Process.*, early access, 2025.

[14] Z. Li, S. Wu, L. Li, and S. Zhang, "Energy-efficient split learning for fine-tuning large language models in edge networks," *IEEE Netw. Lett.*, early access, 2025.

[15] C. Liu and J. Zhao, "Resource allocation for stable LLM training in mobile edge computing," in *Proc. ACM MobiHoc*, 2024, pp. 81–90.

[16] B. Ouyang, S. Ye, L. Zeng, T. Qian, J. Li, and X. Chen, "Pluto and charon: A time and memory efficient collaborative edge AI framework for personal LLMs fine-tuning," in *Proc. ACM ICPP*, 2024, pp. 762–771.

[17] Y. Chen, Y. Yan, Q. Yang, Y. Shu, S. He, and J. Chen, "Confidant: Customizing transformer-based LLMs via collaborative edge training," *arXiv:2311.13381*, 2023.

[18] X. Chen, L. Li, F. Ji, and W. Wu, "Memory-efficient split federated learning for LLM fine-tuning on heterogeneous mobile devices," *arXiv:2506.02940*, 2025.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *arXiv:2106.09685*, 2021.

[20] G. Chen, Z. Qin, M. Yang, Y. Zhou, T. Fan, T. Du, and Z. Xu, "Unveiling the vulnerability of private fine-tuning in split-based frameworks for large language models: A bidirectionally enhanced attack," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2024, pp. 2904–2918.

[21] Z. Wang, Y. Zhou, Y. Shi, and K. B. Letaief, "Federated fine-tuning for pre-trained foundation models over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 24, no. 4, pp. 3450–3464, 2025.

[22] H. Wu, X. Chen, and K. Huang, "Resource management for low-latency cooperative fine-tuning of foundation models at the network edge," *IEEE Trans. Wireless Commun.*, early access, 2025.

[23] P. Wu, K. Li, T. Wang, Y. Dong, V. C. M. Leung, and F. Wang, "FedFMSL: Federated learning of foundation models with sparsely activated LoRA," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 15 167–15 181, 2024.

[24] G. Jiang, S. Han, X. Xu, and X. Tao, "Privacy-aware adaptive model splitting for device-edge co inference," in *Proc. IEEE Globecom Workshops*, 2023, pp. 184–189.

[25] Z. Zhang, A. Pinto, V. Turina, F. Esposito, and I. Matta, "Privacy and efficiency of communications in federated split learning," *IEEE Trans. Big Data*, vol. 9, no. 5, pp. 1380–1391, 2023.

[26] J. Lee, M. Seif, J. Cho, and H. Vincent Poor, "Optimizing privacy and latency tradeoffs in split federated learning over wireless networks," *IEEE Wireless Commun. Lett.*, vol. 13, no. 12, pp. 3439–3443, 2024.

[27] J. Lee, M. Seif, J. Cho, and H. V. Poor, "Exploring the privacy-energy consumption tradeoff for split federated learning," *IEEE Netw.*, vol. 38, no. 6, pp. 388–395, 2024.

[28] R. Deng, S. Hu, J. Lin, J. Yang, Z. Lu, J. Wu, S.-C. Huang, and Q. Duan, "Invmetrics: Measuring privacy risks for split model–based customer behavior analysis," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 4168–4177, 2024.

[29] A. Hannun, C. Guo, and L. van der Maaten, "Measuring data leakage in machine-learning models with fisher information," in *Proc. UAI*, 2021, pp. 760–770.

[30] C. Guo, B. Karrer, K. Chaudhuri, and L. van der Maaten, "Bounding training data reconstruction in private (deep) learning," in *Proc. ICML*, 2022, pp. 8056–8071.

[31] K. Maeng, C. Guo, S. Kariyappa, and G. E. Suh, "Bounding the invertibility of privacy-preserving instance encoding using fisher information," in *Proc. NeurIPS*, 2023, pp. 51 904–51 925.

[32] A. Lodha, G. Belapurkar, S. Chalkapurkar, Y. Tao, R. Ghosh, S. Basu, D. Petrov, and S. Srinivasan, "On surgical fine-tuning for language encoders," in *Proc. EMNLP*, 2023, pp. 3105–3113.

[33] Y.-L. Sung, V. Nair, and C. A. Raffel, "Training neural networks with fixed sparse masks," in *Proc. NeurIPS*, 2021, pp. 24 193–24 205.

[34] C. Liu, J. Pfeiffer, I. Vulić, and I. Gurevych, "FUN with fisher: Improving generalization of adapter-based cross-lingual transfer with scheduled unfreezing," in *Proc. NAACL*, 2024, pp. 1998–2015.

[35] A. Achille, M. Rovere, and S. Soatto, "Critical learning periods in deep networks," in *Proc. ICLR*, 2019.

[36] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "ZeRO: Memory optimizations toward training trillion parameter models," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2020, pp. 1–16.

[37] S. Rajbhandari, O. Ruwase, J. Rasley, S. Smith, and Y. He, "Zero-infinity: Breaking the GPU memory wall for extreme scale deep learning," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2021, pp. 1–14.

[38] G. Xu, Z. Hao, Y. Luo, H. Hu, J. An, and S. Mao, "DeViT: Decomposing vision transformers for collaborative inference in edge devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5917–5932, 2024.

[39] Z. Lin, X. Chen, X. He, D. Tian, Q. Zhang, and P. Chen, "Energy-efficient cooperative task offloading in NOMA-enabled vehicular fog computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 7223–7236, 2024.

[40] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, "Hedonic coalition formation for distributed task allocation among wireless agents," *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1327–1344, 2011.

[41] J.-H. Cho, Y. Wang, I.-R. Chen, K. S. Chan, and A. Swami, "A survey on modeling and optimizing multi-objective systems," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1867–1901, 2017.

[42] Y. Lu, K. Wang, P. Liu, L. Chen, H. Shin, and T. Q. S. Quek, "Tradeoff analysis of unintentional interference and communication rate of UAV," *IEEE Trans. Veh. Technol.*, vol. 74, no. 4, pp. 5941–5953, 2025.

[43] T. T. Vu, D. T. Ngo, M. N. Dao, S. Durrani, and R. H. Middleton, "Spectral and energy efficiency maximization for content-centric C-RANs with edge caching," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6628–6642, 2018.

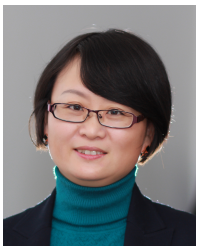
[44] J. Devlin, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018.

[45] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "CARER: Contextualized affect representations for emotion recognition," in *Proc. EMNLP*, 2018, pp. 3687–3697.

[46] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

Xiaopei Chen (S'25) received the B.S. degree in electronics and information engineering and the M.S. degree in electronics and communication engineering from Fuzhou University, Fuzhou, China, in 2020 and 2023, respectively. He is currently a joint Ph.D. student in Information and Communication Engineering with the School of Future Technology, South China University of Technology, Guangzhou, China, and the Pengcheng Laboratory, Shenzhen, China. His current research interests include edge intelligence and vehicular networks.

Wen Wu (S'13-M'20-SM'22) earned the Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Waterloo, ON, Canada, in 2019. He received the B.E. degree in Information Engineering from South China University of Technology, Guangzhou, China, and the M.E. degree in Electrical Engineering from University of Science and Technology of China, Hefei, China, in 2012 and 2015, respectively. He worked as a Post-doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. He is currently an Associate Researcher at the Frontier Research Center, Pengcheng Laboratory, Shenzhen, China. His research interests include 6G networks, network intelligence, and network virtualization. Dr. Wu serves as Track Co-Chairs for IEEE VTC, and Workshop Co-Chairs for IEEE INFOCOM, IEEE Globecom, and IEEE ICCC. He serves on the editorial board for IEEE Networking Letter, Hindawi WCMC, China Communications, and Springer PPNA. He has published over 100 refereed IEEE journal and conference papers and 6 books/book chapters. Dr. Wu received the World Top 2% Scientist Award 2023-2024, USENIX Security Distinguished Paper Award, IEEE HITC Award for Excellence (Early Career Researcher), and IEEE CIC/ICCC Best Paper Award.



Fei Ji (M'06) received the B.S. degree in applied electronic technologies from Northwestern Polytechnical University, Xi'an, China, in 1992, and the M.S. degree in bioelectronics and the Ph.D. degree in circuits and systems from the South China University of Technology, Guangzhou, China, in 1995 and 1998, respectively. She was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada, from June 2009 to June 2010. She worked with the City University of Hong Kong, Hong Kong, as a Research Assistant from March 2001 to July 2002 and a Senior Research Associate from January 2005 to March 2005. She is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology. Her research focuses on wireless communication systems and networking.

Yongguang Lu received the B.S. and M.S. degree in communication engineering from the Wuhan University of Technology, Wuhan, China, in 2020 and 2023, respectively. He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include open and virtual radio access network, network slicing and real-time scheduling.

Liang Li (S'19-M'21) is currently an assistant researcher with the Department of Strategic and Advanced Interdisciplinary Research, Pengcheng Laboratory, Shenzhen, China. She received the Ph.D. degree in the School of Telecommunications Engineering at Xidian University, China, in 2021. From 2018 to 2020, she was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA. From 2021 to 2023, she was a postdoctoral faculty member with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications (BUPT), Beijing, China. Her research interests include edge intelligence, distributed learning, data-driven robust optimization, and differential privacy.