Energy-Efficient Federated Edge Learning for Small-Scale Datasets in Large IoT Networks

Haihui Xie, *Member, IEEE*, Wenkun Wen, *Member, IEEE*, Shuwu Chen, Zhaogang Shu, *Senior Member, IEEE*, and Minghua Xia, *Senior Member, IEEE*

***Abstract*—Large-scale Internet of Things (IoT) networks enable intelligent services such as smart cities and autonomous driving, but often face resource constraints. Collecting heterogeneous sensory data, especially in small-scale datasets, is challenging, and independent edge nodes can lead to inefficient resource utilization and reduced learning performance. To address these issues, this paper proposes a collaborative optimization framework for energy-efficient federated edge learning with small-scale datasets. We first derive an expected learning loss to quantify the relationship between the number of training samples and learning objectives. A stochastic online learning algorithm is then designed to adapt to data variations, and a resource optimization problem with a convergence bound is formulated. Finally, an online distributed algorithm efficiently solves large-scale optimization problems with high scalability. Extensive simulations and autonomous navigation case studies with collision avoidance demonstrate that the proposed approach significantly improves learning performance and resource efficiency compared to state-of-the-art benchmarks.**

***Index Terms*—Edge intelligence, federated learning (FL), Internet of Things (IoT), learning-oriented communication, small-scale datasets.**

I. Introduction

With the rapid growth of the Internet of Things (IoT), applications such as smart cities, smart factories, intelligent transportation, and autonomous driving are generating massive volumes of data. Edge intelligence enables local data processing on IoT devices, improving communication efficiency and responsiveness. Wireless and cellular communications are particularly well-suited for such applications due to their ability to support high data rates and wide coverage. Achieving energy-efficient edge intelligence requires not only selecting the right communication technology but also optimizing resource allocation strategies [1].

Federated learning (FL) has emerged as a key paradigm for edge intelligence, enabling distributed edge devices to collaboratively train models without requiring centralized data aggregation. Despite extensive research on FL [2], [3], [4], [5], [6], most existing designs prioritize learning performance while overlooking communication constraints. Since reliable communication is indispensable in distributed networks, it is crucial to jointly optimize learning objectives and resource allocation.

A. Related Works and Motivation

To ensure reliable communication in energy-efficient and low-cost IoT networks, recent studies have explored resource allocation under constrained environments [7], [8], [9]. For example, the study [9] proposed a sum-rate maximization (SRM) scheme to exploit wireless gains. However, such throughput-oriented schemes often fail to deliver sufficient learning performance [10]. To overcome these limitations, *task-oriented* resource allocation has been investigated [11], [12], [13], [14], [15], [16], [17], [18], [19]. Specifically, the works [11], [12], [13] introduced learning-centric power allocation models to allocate limited energy resources more effectively, while [14], [15] developed quality-of-training maximization (QoT-Max) schemes to preserve task information under bandwidth-limited inference. Other works [16], [17], [18] jointly optimized sensing, communication, and computation to maximize accuracy. Additionally, [19] proposed semantic transmission methods that extract key task-related features from multimodal IoT data to accelerate learning.

In addition, *task-aware* communication schemes have been studied to improve resource efficiency in federated edge learning frameworks [20], [21], [22], [23], [24]. For instance, [20] proposed a federated edge learning model that minimizes energy costs by selecting optimal local data samples, which is solved via the distributed alternating direction method of multipliers (ADMM). The work in [21] introduced gradient norms as indicators of data importance, enabling joint data selection and communication allocation to improve efficiency. Similarly, [22] designed a sample-level importance metric based on maximum categorical probability, pruning less significant samples

Received 28 August 2025; revised 12 February 2026; accepted 8 April 2026. Date of publication 21 April 2026; date of current version 22 April 2026. This work was supported in part by the Guangzhou Techphant Technologies Company Ltd.-Sun Yat-sen University Joint Project under Grant SYSU-76120-20260126-0002, in part by Fujian Industry-University-Institute Cooperation Project of China under Grant 2024H6007 and Grant 2024H6030, and in part by the Overseas Research and Further Education Program for Young and Middle-Aged Backbone Teachers from Universities in Fujian Province of China. The associate editor coordinating the review of this article and approving it for publication was X. Cao. *(Corresponding authors: Wenkun Wen; Minghua Xia.)*

Haihui Xie, Shuwu Chen, and Zhaogang Shu are with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China, and also with the Engineering Research Center of Smart Sensing and Agricultural Chip Technology, Fujian Province University, Fuzhou 350002, China (e-mail: xiehh@fafu.edu.cn; chenshuwu@fafu.edu.cn; zgshu@fafu.edu.cn).

Wenkun Wen is with the Research and Development Department, Techphant Technologies Company Ltd., Guangzhou 510310, China (e-mail: wenwenkun@techphant.net).

Minghua Xia is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: xiamingh@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TWC.2026.3683911

to reduce energy consumption. Furthermore, hierarchical FL frameworks [23], [24] optimized communication scheduling under budget constraints to improve scalability.

Another challenge arises from the data acquisition process in distributed IoT networks. Heterogeneous sensory data enhances coverage and generality [25], but most existing studies are designed under *offline modes*, where models rely on static pre-collected datasets [11], [12], [14], [15], [16], [17], [19], [20], [21], [22], [23], [24]. Offline frameworks, including reinforcement learning methods [26], can improve efficiency but often suffer from limited adaptability when datasets lack diversity. To address this, online reinforcement learning approaches [27], [28] have been proposed, which enable continuous data collection and model adaptation. Techniques such as prioritized experience replay and stochastic policies improve both sample efficiency and exploration, making online learning more robust to environmental variations.

Motivated by these insights, analyzing the theoretical relationship between learning performance and resource utilization is essential for federated edge learning. Existing *task-oriented* schemes [11], [12], [23], [24] require large-scale deterministic datasets, making real-time optimization computationally expensive. Meanwhile, *task-aware* methods [20], [21], [22] often lead to low resource utilization in distributed networks, where nodes operate independently. Moreover, the interplay between task-oriented and task-aware schemes is rarely studied, leading to redundant computation or degraded performance.

These challenges motivate the design of an energy-efficient federated edge learning framework that bridges task-oriented and task-aware paradigms, leverages expected learning loss to capture dataset–performance relationships, and adapts resource allocation to dynamic IoT environments.

B. Our Approach

To address the limitations above, we propose an *energy-efficient federated edge learning framework* designed for large-scale IoT networks with small-scale training datasets. The core idea is to jointly optimize learning objectives and wireless resource allocation while adapting to dynamic environments.

First, we develop a cloud-edge-end collaborative framework to balance resource allocation and learning performance. Unlike conventional task-oriented methods that rely heavily on deterministic datasets, we derive an expected learning loss that characterizes the fundamental relationship between the number of training samples and model performance. This provides a theoretical foundation for capturing learning dynamics under limited datasets.

Second, to support dynamic data environments, we design a stochastic online learning algorithm that progressively adjusts the dataset size during training. The algorithm leverages a convergence bound, derived via the majorization–minimization (MM) framework, to ensure that model updates remain stable as new mini-batches are incorporated. This enables efficient learning from small-scale data samples while preserving adaptability to real-time end-device conditions.

Third, for scalability in large-scale IoT networks, we propose a distributed optimization algorithm that eliminates redundant variables and fixes an optimization direction. This design significantly improves convergence probability compared with conventional ADMM-based approaches, achieving faster training and higher resource utilization.

Finally, to validate the effectiveness of our framework, we deploy the proposed algorithms on high-fidelity simulation platforms, including the Intelligent Robot Simulator (IR-SIM) [29] and Car Learning to Act (CARLA) [23]. Experimental results demonstrate that our approach achieves superior convergence speed and energy efficiency, highlighting the practical benefits of resource-aware federated edge learning in real-world autonomous systems.

C. Main Contributions

The main contributions of this paper are summarized below:

1) Expected learning loss formulation: We derive an expected learning loss that rigorously characterizes the relationship between training dataset size and learning performance. This provides a principled approach to linking communication resources with model convergence, particularly in small-scale dataset scenarios.
2) Stochastic online learning algorithm: We design an adaptive online learning algorithm that dynamically adjusts the dataset size during training. By deriving a convergence bound through the MM framework, the algorithm ensures stable updates and efficient learning from streaming data.
3) Distributed optimization for large-scale IoT: We propose a distributed optimization algorithm that eliminates redundant variables and enforces a fixed optimization direction. Compared with conventional ADMM, this approach improves convergence probability, scalability, and resource utilization in large-scale IoT networks.
4) Practical validation: We implement the proposed framework on high-fidelity platforms, including IR-SIM [29] and CARLA [23]. Experimental results and case studies demonstrate faster convergence, reduced energy consumption, and improved performance in autonomous navigation and collision-avoidance tasks.

The rest of this paper is organized as follows. Section II introduces the system model and formulates the optimization problem. Section III analyzes the relationship between edge learning performance and the training dataset size, and develops an adaptive learning model for online data collection. Section IV proposes a centralized MM-based algorithm for small- and medium-sized IoT networks. For large-scale networks, Section V presents a distributed algorithm and an accelerated momentum-based variant. Section VI provides simulation results and a case study to evaluate the proposed scheme. Finally, Section VII concludes the paper.

Notation: Scalars, column vectors, and matrices are denoted by italic letters, bold lower-case letters, and bold upper-case letters, respectively. The vectors $\mathbf{1}$ and $\mathbf{0}$ denote all-ones and all-zeros column vectors, respectively. The superscripts $(\cdot)^T$ and $(\cdot)^H$ represent the transpose and Hermitian transpose, respectively. The ℓ_2-norm of a vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|_2$. The notation $\mathcal{CN}(\mathbf{0}, \varrho \boldsymbol{I})$ denotes a circularly symmetric complex Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\varrho \boldsymbol{I}$. For a set $\mathcal{X}$, $|\mathcal{X}|$ denotes its cardinality, and $\mathcal{Y} \setminus \mathcal{X}$ denotes

TABLE I
SUMMARY OF MAIN NOTATIONS

Symbol	Definition
$\mathcal{K}$, $\mathcal{K}_i$	User set; user set at edge node i
$\mathcal{X}(t)$, $\mathcal{X}_i(t)$	Collected dataset (all devices); dataset at node i
$\boldsymbol{w}(t)$, $\boldsymbol{w}_i(t+1)$	Global model; local model at node i
$A(t)$, $A_i(t)$	Initial number of collected samples at epoch t (all nodes; node i)
$F(\boldsymbol{w})$, $F_i(\boldsymbol{w})$	Global loss; local loss at node i
$\mathcal{D}$, $\mathcal{D}_i$	The whole data distribution, data distribution at edge node i
D, D_i	Total dataset size; dataset size at node i
α, α_i	Data-deficit factor (all nodes; node i)
$\bar{F}$, $\bar{F}_i$	Expected learning loss (cloud); expected loss at node i
$\boldsymbol{p}(t)$, $\boldsymbol{p}_i(t)$	Power allocation vector; power vector at node i
$G_{k,l}$	Channel gain from user l when decoding user k
P, B, T, σ^2	Power budget; bandwidth; transmission time; noise power
$\boldsymbol{s}_j$, ξ_1, ξ_2, L	Random sample; SGD noise constants; smoothness constant
r_l, r_c, r_g	Learning rate; collision rate; goal rate
T_l, N_c, N_g	Training time; number of collisions; number of goals
T_o, N_a, N_u	Optimization time; number of avoidances; failed goals

the set difference. The operator $\mathbb{E}(\cdot)$ denotes expectation, and $\mathcal{O}(\cdot)$ denotes the order of arithmetic operations. The main symbols used throughout the paper are summarized in Table I.

II. System Model and Problem Formulation

In this section, we first describe the system model of cloud-edge-end collaboration. Then, we formulate the learning-oriented resource allocation problem.

A. System Model

Fig. 1 illustrates a federated edge learning system consisting of a cloud server with an aggregated parameter vector $\boldsymbol{w}(t)$, I edge servers each equipped with N antennas serving user sets $\mathcal{K} \triangleq \{\mathcal{K}_1, \mathcal{K}_2, \cdots, \mathcal{K}_I\}$, local learning parameters $\{\boldsymbol{w}_1(t), \cdots, \boldsymbol{w}_I(t)\}$ and variable training datasets $\mathcal{X}(t) \triangleq \{\mathcal{X}_1(t), \mathcal{X}_2(t), \cdots, \mathcal{X}_I(t)\}$, respectively. Here, edge node i also has an edge server, user set $\mathcal{K}_i$, local parameter $\boldsymbol{w}_i(t)$, and training dataset $\mathcal{X}_i(t)$. In particular, each user set contains various IoT devices or sensors.

Fig. 1 also demonstrates the complete data collection pipeline, model training, and information exchange. The cloud server coordinates this federated edge learning framework through global parameter aggregation and dynamic power allocation. Specifically, the cloud server synthesizes model updates from edge nodes and optimizes wireless resource utilization. At the edge node, each server primarily collects heterogeneous training data from associated IoT devices, trains a local model using the collected datasets, and then transmits the model parameters to the cloud server. Each edge node is associated with various IoT devices that generate diverse training data through their interactions with the environment. These IoT devices should transmit sensory data to their designated edge servers, balancing communication overhead and data utility. During the iterative processes, edge servers continuously train local models using newly collected data while the cloud server aggregates these updates into an improved global model. This cyclic process continues until convergence, yielding a robust global model that is well-suited for distributed networks.

In resource-constrained wireless networks, we perform a dynamic power allocation for efficient communications. To

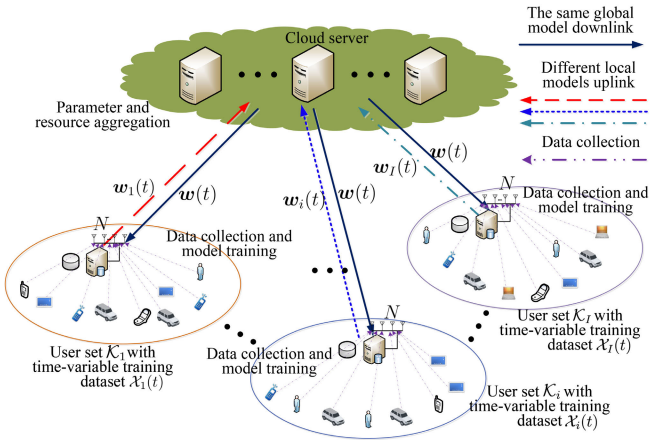


Fig. 1. The cloud-edge-end collaborative system in resource-constrained large-scale IoT networks.

maximize the network utility function of long-term average data rates, by recalling the seminal Shannon formula, the achievable data rate of device k can be expressed as [11]

$$R_k = \log_2\left(1 + \frac{G_{k,k}p_k}{\sum_{\ell\in\mathcal{K}\setminus k} G_{k,\ell}p_\ell + \sigma^2}\right), k \in \mathcal{K}_i, \quad (1)$$

where σ^2 denotes the variance of additive white Gaussian noise; $G_{k,\ell}$ represents the composite channel gain from the ℓ^{th} device to the edge server when decoding data of the k^{th} device, computed as $G_{k,k} = \rho_k\|\boldsymbol{h}_k\|_2^2$ if $\ell = k$, and $G_{k,\ell} = \rho_\ell|\boldsymbol{h}_k^H\boldsymbol{h}_\ell|^2/\|\boldsymbol{h}_k\|_2^2$ if $\ell \neq k$, with $\boldsymbol{h}_k \in \mathbb{C}^{N\times 1}$ being the complex-valued channel fast-fading vector from the k^{th} device to the edge server and ρ_k being the path loss of the k^{th} device. By (1), the number of collected data samples at edge node i is computed by

$$|\mathcal{X}_i| = \sum_{k\in\mathcal{K}_i}\left\lfloor\frac{BTR_k}{V}\right\rfloor + A_i \approx \sum_{k\in\mathcal{K}_i}\frac{BTR_k}{V} + A_i, \quad (2)$$

where B denotes the total bandwidth in Hz; T represents the transmission time in seconds; V is the bit number of each data sample, and A_i is the initial number of historical data for the i^{th} edge node.

B. Problem Formulation

This paper aims to design an energy-efficient federated edge learning model by jointly optimizing the expected loss function and wireless resource allocation strategies. Reinforcement learning typically optimizes a non-parametric learning loss under a parameterized distribution. Motivated by this principle, we introduce a learning loss formulation where the distribution of wireless resource parameters serves as the objective of optimization. To this end, we first derive an expected learning loss that captures the fundamental trade-off between the number of training data samples and the resulting learning performance.

Specifically, consider the federated learning system illustrated in Fig. 1, the local loss function at edge server i is defined as

$$F_i(\boldsymbol{w}) \triangleq \frac{1}{|\mathcal{D}_i|}\mathbb{E}_{\mathcal{D}_i}\left(\sum_{j\in\mathcal{D}_i} f(\boldsymbol{w} \mid \boldsymbol{s}_j)\right), \tag{3}$$

where $\boldsymbol{w}$ is a global aggregated parameter; $\boldsymbol{s}_j$ denotes an arbitrary training data; $f(\cdot \mid \boldsymbol{s}_j)$ is a loss function given the training data $\boldsymbol{s}_j$, and $\mathcal{D}_i$ sufficiently describes the data distribution at edge node i. Therefore, $\mathcal{D} \triangleq \cup_{i=1}^{I}\mathcal{D}_i$ describes the distribution of the whole dataset [30]. Moreover, suppose that $\mathcal{D}_i \cap \mathcal{D}_{i'} = \emptyset$ for all $i \neq i'$, then, the global loss can be expressed as

$$F(\boldsymbol{w}) \triangleq \frac{1}{D}\sum_{i=1}^{I} D_i F_i(\boldsymbol{w}) = \sum_{i=1}^{I} \alpha_i F_i(\boldsymbol{w}), \tag{4}$$

where $D_i \triangleq |\mathcal{D}_i|$ and $D \triangleq |\mathcal{D}|$ denote the number of datasets, $\alpha_i \triangleq D_i/D$ reflects the fraction of local models that are involved in aggregation.

However, obtaining comprehensive datasets $\mathcal{D}_i$ for learning $F_i(\boldsymbol{w})$, for all $i = 1, \cdots, I$, is a significantly challenge. Therefore, we should employ the collected dataset $\mathcal{X}_i(t)$ to asymptotically optimize the true learning model $F_i(\boldsymbol{w})$. Accordingly, we formulate a resource optimization problem to minimize the learning loss concerning wireless resource constraints, given by

$$\mathcal{P}_1: \min_{\boldsymbol{p},\{P_i\}_{i=1}^{I}} \frac{1}{D}\sum_{i=1}^{I} D_i F_i(\boldsymbol{w}(\mathcal{X}_i(t))) \tag{5a}$$

$$\text{s.t. } |\mathcal{X}_i(t)| \leq D_i,\ i = 1,\ \cdots,\ I, \tag{5b}$$

$$\sum_{i=1}^{I} |\mathcal{X}_i(t)| \leq D, \tag{5c}$$

$$\sum_{k\in\mathcal{K}_i} p_k \leq P_i,\ p_k \geq 0,\ \forall k \in \mathcal{K}, \tag{5d}$$

$$\sum_{i=1}^{I} P_i = P, \tag{5e}$$

where (5a) implies that the global parameter $\boldsymbol{w}$ is determined by $\mathcal{X}_i(t)$ due to (2); (5b) and (5c) denote the maximum local stored capacity of dataset at the edge node; (5d) denotes that the power allocation of devices adapts to the distributed framework; (5e) denotes that the power allocation of edge nodes should not exceed the total power budget P [11].

Compared with the optimization model with independent resource allocation [31], the proposed model implements the resource optimization more efficiently, since it is deployed at the cloud server, e.g., the gateway of a large-scale IoT network, which informs the devices of their transmit powers and other parameters through the downlink control channel, e.g., the narrow-band physical downlink control channel in NB-IoT networks [32]. However, the explicit form of $F_i(\boldsymbol{w})$ related to $\mathcal{X}_i(t)$ is undetermined, due to the randomness of $\mathcal{X}_i(t)$ and the non-convexity of $F_i(\boldsymbol{w})$. This implies that finding an exact analytical expression for the power allocation $\boldsymbol{p}$ in $\mathcal{P}_1$ is mathematically intractable. Thus, we derive an analytical upper bound in Section III-A to establish the connection between wireless resource allocation and the learning performance.

Note that this work focuses on optimizing the communication energy via transmit power control for online data collection. The energy consumption of local computation (e.g., SGD updates) is not included in $\mathcal{P}_1$, since the learning architecture and training hyperparameters are fixed and the computational cost is largely independent of the transmit power decisions. Moreover, our formulation is quantity-oriented, assuming that uploaded samples from each node are representative of its local distribution. We acknowledge that, in heterogeneous IoT environments, computation energy and data importance may both be significant; incorporating these factors is feasible by augmenting the energy model with standard CPU-cycle-based formulations and introducing importance weights estimated from local training feedback (e.g., validation loss or gradient statistics), which is an interesting direction for future work.

III. Convergence Analysis and Problem Transformation

This section first derives the relationship between learning performance and training datasets. Then, we design a learning algorithm that simultaneously collects data and trains a model. Next, we prove the convergence and derive the upper bound of learning algorithms. At last, we transform the original problem $\mathcal{P}_1$ into a more mathematical tractable one.

A. Relationship Between Learning Performance and Training Datasets

We begin by examining the relationship between the performance of the learning model and the training datasets $\mathcal{X}_i(t)$ used to optimize the model parameters $\boldsymbol{w}$. The datasets $\mathcal{X}_i(t)$ are collected from the i^{th} edge node in the network and are utilized to approximate the local learning models $F_i(\boldsymbol{w})$. However, the optimal performance of these models cannot be rigorously evaluated without knowledge of the underlying data distribution at this optimal state.

To address this, we first analyze the optimal state of the learning model under the conditions provided by $\mathcal{X}_i(t)$, and then establish a connection between the learning performance and the dataset characteristics, as presented in Lemma 1.

Lemma 1: Let $X \in \mathcal{D}_i$ be a training dataset and $Y \in \mathbb{R}$ be the corresponding output (learning loss), where $\mathcal{D}_i$ denotes the dataset associated with the i^{th} edge node and $F_i(\boldsymbol{w})$ is the learning model. Let the joint probability density function of X and Y be denoted by $f_{X,Y}(x, y)$. Then, the expected learning loss $\bar{F}_i(\boldsymbol{w})$ at the i^{th} edge node is given by the conditional expectation:

$$\bar{F}_i(\boldsymbol{w}) = \mathbb{E}_{f_{X,Y}(x,y)}\left(Y \mid X \in \mathcal{D}_i\right). \tag{6}$$

Proof: Given the dataset X, we define the squared error loss function as:

$$L(Y, F_i(\boldsymbol{w} \mid X)) = (Y - F_i(\boldsymbol{w} \mid X))^2, \tag{7}$$

where $F_i(\boldsymbol{w} \mid X)$ is the model's prediction based on the dataset X.

Using the law of total expectation, we decompose the expected squared error as follows:

$$\mathbb{E}\left((Y - F_i(\boldsymbol{w} \mid X))^2\right) = \mathbb{E}_X \mathbb{E}_{Y|X}\left((Y - F_i(\boldsymbol{w} \mid X))^2 \mid X\right). \tag{8}$$

This expresses the total expected loss as the expectation over the input dataset X and the conditional expectation over the outputs Y, given X.

The goal is to minimize the expected squared error with respect to the model prediction $F_i(\boldsymbol{w} \mid X)$. This minimization problem is equivalent to finding the value ℓ that minimizes:

$$\bar{F}_i(\boldsymbol{w}) = \underset{\ell}{\operatorname{argmin}}\, \mathbb{E}_{Y|X=x}\left((Y-\ell)^2 \mid X = x \in \mathcal{D}_i\right), \tag{9}$$

which leads to the optimal prediction being the conditional expectation:

$$\ell = \mathbb{E}[Y \mid X = x]. \tag{10}$$

Thus, the expected learning loss is given by the conditional expectation $\mathbb{E}[Y \mid X]$, which completes the proof. ■

Lemma 1 shows that the expected learning loss $\bar{F}_i(\boldsymbol{w})$ is the optimal prediction given by (6), while using the collected dataset $\mathcal{X}_i(t)$ to optimize $\mathcal{P}_1$. Thus, we can use (6) to define the objective function for the training dataset, especially for small-scale datasets. Although (6) is an implicit mathematical expression, a fitting nonlinear model can be used to estimate its shape [11], [12]. However, this fitting model typically relies on large-scale datasets, making it challenging to fit it accurately with the limited data available. To address this issue, Section III-B introduces a framework where data collection and model training occur concurrently. Additionally, we derive an upper bound for (6) to optimize resource allocation under a learning-oriented principle.

Remark 1: (Parameter Fitting of Expected Learning Losses): Consider a sequence of nested sample datasets $\mathcal{X}(t_1) \subseteq \mathcal{X}(t_2) \subseteq \cdots \subseteq \mathcal{X}(t_Q)$ collected over ordered time periods $t_1 \leq t_2 \leq \cdots \leq t_Q$. For each $\mathcal{X}(t_m)$, we train the model to convergence and record the learning loss ℓ_m. Using the pairs $\{\mathcal{X}(t_m), \ell_m\}_{m=1}^{Q}$, the parameters (a, b) of the expected loss model [11], [12]

$$\bar{F}(\mathcal{X}(t) \mid a, b) \approx a|\mathcal{X}(t)|^{-b} \tag{11}$$

are fitted via nonlinear least squares:

$$\min_{a>0, b>0} \frac{1}{Q}\sum_{m=1}^{Q}\left|\ell_m - \bar{F}(\mathcal{X}(t_m) \mid a, b)\right|^2. \tag{12}$$

This fitting can be efficiently solved by grid search or gradient-based methods. The adopted power-law form serves as a surrogate diminishing-return model, which is consistent with classical generalization and optimization scaling laws, where the expected excess risk often decays with sample size as $\mathcal{O}(|\mathcal{D}|^{-1/2})$ or $\mathcal{O}(|\mathcal{D}|^{-1})$ under standard assumptions [30], [33], [34], and with SGD training where the stochastic gradient variance reduces as more samples become available [30]. Motivated by these scaling behaviors, we adopt $\bar{F}(\mathcal{X}(t) \mid a, b) \approx a|\mathcal{X}(t)|^{-b}$, where (a, b) are task-dependent and fitted from data rather than assumed universal. The robustness of the proposed framework to learning-loss model mismatch is empirically validated in Figs. 3 and 8, where stable training behavior and consistent performance gains are observed under heterogeneous learning dynamics.

When the dataset turns large, the learning curve saturates, and the online optimization can be simplified (e.g., longer control intervals, fixed/quantized power, or early stopping). In extremely data-scarce regimes, the fitting of (a, b) may be noisy; the proposed algorithm mainly relies on the monotonic diminishing-return structure and continuously refines the fitting online as more samples become available. A short warm-up (data-acquisition) phase or conservative default parameters can also be used to improve robustness in very small-sample regimes.

B. Learning Algorithm

In contrast to traditional static learning problems, $\mathcal{P}_1$, as given by (5), presents a stochastic learning problem with dynamic optimization objectives and datasets. Thus, we design a stochastic online learning algorithm to address this issue. In principle, this algorithm sequentially processes incoming data, making decisions or updates based on stochastic samples or noisy observations. Specifically, we regard $\mathcal{X}_i(t) \subseteq \mathcal{D}_i$ as a random mini-batch dataset,[1] then employ the stochastic gradient descent to minimize $F(\boldsymbol{w})$:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) - \frac{\eta}{|\mathcal{X}_i(t)|}\sum_{j \in \mathcal{X}_i(t)} f\left(\boldsymbol{w}(t) \mid \boldsymbol{s}_j\right), \tag{13a}$$

$$\boldsymbol{w}(t+1) = \sum_{i=1}^{I} \alpha_i \boldsymbol{w}_i(t+1), \tag{13b}$$

where (13a) denotes that edge node i updates the learning parameter $\boldsymbol{w}_i(t+1)$ using the collected dataset $\mathcal{X}_i(t)$ at the t^{th} iteration, and (13b) represents that the cloud server aggregates learning parameters from I edge nodes.

As the training samples $\mathcal{X}_i(t)$ in (13a) are not dedicated and change over time, this learning process can supplement the dataset of small training samples periodically. To ensure the reliability and stability of the stochastic online gradient descent algorithm, we provide theoretical guarantees for convergence analysis in the following Section III-C.

C. Convergence Analysis

Now, we analyze the convergence behavior of the learning algorithm as the mini-batch dataset size increases. For this, we make the following standard assumptions.

Assumption 1 [30, Assumption 4.1]: The global loss function $F : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and has an L-Lipschitz continuous gradient, i.e.,

$$\|\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d,$$

where $L > 0$ is the smoothness constant of F (also referred to as the gradient Lipschitz constant), which characterizes the smoothness (curvature upper bound) of the loss landscape induced by the adopted model architecture and regularization.

[1] As the number of iterations increases, the collected sample sets grow progressively over time. For analytical tractability, we treat each sampled dataset as a mini-batch, thereby mitigating stochastic variations during training. In practice, even with a large-scale dataset, mini-batch learning remains an effective approach for efficiently training models while maintaining stability.

Assumption 2 [30, Assumption 4.3]: There exist scalars $\xi_1 > 0$ and $\xi_2 > 0$ such that

$$\mathbb{E}_{\boldsymbol{s}_j}\left[\|\nabla f(\boldsymbol{x} \mid \boldsymbol{s}_j)\|_2^2\right] \leq \xi_1 + \xi_2 \|\nabla F(\boldsymbol{x})\|_2^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^d,$$

where $\nabla f(\boldsymbol{x} \mid \boldsymbol{s}_j)$ is the stochastic gradient evaluated on a random sample $\boldsymbol{s}_j$. Here, ξ_1 quantifies the intrinsic stochastic gradient noise (i.e., a variance-related term that remains even near stationarity), while ξ_2 controls how the second moment of the stochastic gradient scales with the squared norm of the full gradient.

Assumption 1 ensures the smoothness of $F(\boldsymbol{x})$, which is satisfied by many neural network losses, including convolutional neural networks [35], [36]. Assumption 2 captures the bias–variance tradeoff inherent in stochastic gradient descent [2]. Both assumptions are widely adopted in convergence analysis, even for non-convex objectives [5], [6].

Under these assumptions, we derive an upper bound on the expected optimality gap $\mathbb{E}[F(\boldsymbol{w}(t)) - F(\boldsymbol{w}^*)]$, as stated below. In particular, building upon Lemma 1, we incorporate the effect of *online data collection* by explicitly linking the dataset size to time, i.e., $|\mathcal{X}| \to |\mathcal{X}(t)|$. By introducing the scaling factor α to characterize the data-deficit effect in small-scale dataset regimes, we obtain the α-dependent convergence bound in Theorem 1, which directly couples learning progress with the number of collected samples and enables the subsequent learning-driven power allocation.

Theorem 1: Let $\boldsymbol{w}^*$ denote the optimal model parameters and set the learning rate to $\eta = 1/L$. Then, the expected optimality gap satisfies

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{w}(t)) - F(\boldsymbol{w}^*)\right)\\ &\leq \mathbb{E}\left((4\xi_2\alpha)^t\left(F(\boldsymbol{w}(0)) - F(\boldsymbol{w}^*) + \frac{2\alpha\xi_1}{(1-4\xi_2\alpha)L}\right)\right.\\ &\qquad \left. + \frac{2\alpha\xi_1}{(1-4\xi_2\alpha)L}\right),\end{aligned} \tag{14}$$

where $\alpha \triangleq (D - |\mathcal{X}(t)|)^2/D^2$ represents the squared fraction of uncollected data, serving as a data-deficit/statistical-efficiency factor, and $\mathcal{X}(t) \triangleq \cup_{i=1}^{I} \mathcal{X}_i(t)$ denotes the total dataset collected from all devices.

Proof: See Appendix A. ■

Theorem 1 shows that the expected optimality gap at iteration t is upper-bounded by three components: *i)* the initial gap $F(\boldsymbol{w}(0)) - F(\boldsymbol{w}^*)$, independent of the transmitted data size; *ii)* a descent term $(4\xi_2\alpha)^t$, which decreases with more iterations but is only weakly affected by data size; and *iii)* a variance term $2\alpha\xi_1/((1-4\xi_2\alpha)L)$, which is strongly dependent on the number of transmitted samples. To tighten the bounds and improve convergence, it is essential to reduce both the descent and variance terms—primarily by increasing the size of the collected dataset $\mathcal{X}(t)$. Intuitively, the data-deficit factor α quantifies how far the currently available training dataset is from the "statistically sufficient" regime: a larger α corresponds to a more severe sample deficiency and thus slower effective convergence, whereas $\alpha \to 0$ indicates that the system approaches a data-sufficient regime where classical learning behavior is recovered.

It is noteworthy that Theorem 1 builds on standard SGD convergence tools; the key novelty is that the factor α explicitly captures the data-deficit effect under small-scale datasets, thereby coupling learning progress with the online data-collection process. We also emphasize that the bound is worst-case and mainly used to reveal the monotonic diminishing-return dependence on the number of collected samples, rather than to provide a numerically tight prediction. The constants ξ_1 and ξ_2 are problem-dependent and vary with the model architecture, optimizer, and data distribution. In practice, they can be estimated during a short warm-up (data-acquisition) phase and further refined online. The proposed framework does not require their exact closed-form values; conservative estimates yield only conservative resource allocation.

To ensure convergence of the training process, we derive the following corollary based on Theorem 1.

Corollary 1: Training convergence is guaranteed if the number of transmitted samples $\mathcal{X}(t)$ satisfies the lower bound:

$$|\mathcal{X}(t)| > D - \frac{D}{2\sqrt{\xi_2}}. \tag{15}$$

Proof: From Theorem 1, convergence is ensured if the descent factor decays to zero as $t \to \infty$, i.e.,

$$\begin{aligned}&\lim_{t\to\infty}\left(4\xi_2\left(\frac{D-|\mathcal{X}(t)|}{D}\right)^2\right)^t = 0\\ &\Longrightarrow 4\xi_2\left(\frac{D-|\mathcal{X}(t)|}{D}\right)^2 \ll 1.\end{aligned} \tag{16}$$

Solving this inequality yields the condition in (15). ■

Corollary 1 highlights that a minimum number of initial training samples is necessary to guarantee stable convergence. This emphasizes the importance of initializing each model with sufficiently informative samples A_i as in (2). When a new node joins with very few samples ($A_i \approx 0$), it is placed into a short warm-up (data-acquisition) phase to collect sufficient local data. The edge orchestrator applies a simple admission control rule and only includes nodes with $A_i(t) \geq A_{\min}$ in the online power allocation and training loop. This ensures that the learning updates remain well-conditioned and that the assumptions underlying Theorem 1 are satisfied for the active node set. Additionally, Theorem 1 implies that a tighter upper bound on the convergence gap leads to greater algorithmic stability. Therefore, minimizing this bound becomes a key objective in subsequent Section III-D.

Furthermore, Corollary 1 reveals a phase transition in the convergence condition based on the learning loss parameter ξ_2. When $0 < \xi_2 < \frac{1}{4}$, the right-hand side of (15) is negative, indicating a looser convergence requirement and tolerating small initial sample sizes. In contrast, if $\xi_2 > \frac{1}{4}$, the lower bound becomes positive, enforcing a stricter requirement on initial data size for convergence.

To bridge these regimes, we propose an online learning framework that incrementally satisfies the convergence condition by progressively increasing the dataset size. While this approach may slow convergence, it ensures robustness across varying system parameters and model characteristics.

Remark 2 (Dataset Distribution Shifts): The analysis in this work assumes stationary local data distributions $\{D_i\}$. In dynamic IoT environments, local distributions may evolve over time, i.e., $D_i \to D_i(t)$, making the learning objective time-varying. The proposed algorithm remains applicable, as it updates the model using newly collected samples and online stochastic gradients; however, the theoretical guarantee should be interpreted in a tracking/piecewise-stationary sense rather than as convergence to a single fixed optimum. In practice, adaptivity to drift can be improved by periodically re-fitting the surrogate learning-loss model using a sliding window or an exponential forgetting factor, and refreshing the allocation state upon drift detection.

D. Problem Transformation

The convergence analysis in Theorem 1 highlights the impact of the number of transmitted samples on learning performance. Combining (14) with (2), it is evident that power control also affects learning outcomes. Accordingly, we relax problem $\mathcal{P}_1$ to the following:

$$\begin{aligned}\mathcal{P}_2: \min_{\boldsymbol{p}} \ & (4\xi_2\alpha)^t \left(C + \frac{2\alpha\xi_1}{(1-4\xi_2\alpha\left(|\mathcal{X}(t)|,\boldsymbol{p}\right))L}\right) \\ & + \frac{2\alpha\xi_1}{(1-4\xi_2\alpha)\,L} \\ \text{s.t. } & \text{(5b)-(5d), (2),}\end{aligned} \tag{17}$$

where $C \triangleq F(\boldsymbol{w}(0)) - F(\boldsymbol{w}^*)$ denotes the initial loss gap.

Under the conditions of Theorem 1 and Corollary 1 (i.e., (15)), $\mathcal{P}_2$ can be further simplified to (18), as shown at the bottom of the page, where $A \triangleq \sum_{i=1}^{I} A_i$ is the total number of initial samples. We assume $A_i \geq D_i - D_i/(2\sqrt{\xi_2})$, which suffices to ensure stable convergence according to Corollary 1.

IV. Centralized MM-Based Algorithm

The problem $\mathcal{P}_3$ remains a nonlinear integer programming problem that is challenging to solve due to the presence of an exponential term. To address this issue, we propose an iterative MM-based algorithm that sequentially solves optimization sub-problems until convergence. Specifically, we construct a sequence of upper bounds $\{\widetilde{\Phi}\}$ on $\{\Phi\}$ and substitutes $\{\Phi\}$ in $\mathcal{P}_3$ with $\{\widetilde{\Phi}\}$ to obtain the surrogate problems. Given any feasible solution $\boldsymbol{p}^*$ to $\mathcal{P}_3$, we define surrogate functions $\widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}^*)$ as Eq. (19) and (20a) (20d), shown at the bottom of the page, and the following proposition can be established.

Proposition 1: (MM Surrogate: Majorization and Tangency): For any feasible expansion point $\boldsymbol{p}^*$, the surrogate function $\widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}^*)$ defined in (19) satisfies the standard MM conditions:

(1) Majorization: $\Phi(\boldsymbol{p}) \leq \widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}^*)$ for all feasible $\boldsymbol{p}$;
(2) Tangency: $\widetilde{\Phi}(\boldsymbol{p}^* \mid \boldsymbol{p}^*) = \Phi(\boldsymbol{p}^*)$;

$$\mathcal{P}_3: \min_{\boldsymbol{p}} \ \underbrace{\left(\frac{BT}{V}\sum_{k\in\mathcal{K}} \log_2\left(1+\frac{G_{k,k}p_k}{\sum_{\ell\in\mathcal{K}\setminus k} G_{k,\ell}p_\ell + \sigma^2}\right) + A - D\right)^2}_{\triangleq \Phi(\boldsymbol{p})}$$
$$\text{s.t. (5b)-(5d), (2)} \tag{18}$$

$$\begin{aligned}\widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}^*) \triangleq \Bigg(\frac{BT}{V\ln 2}\sum_{k=1}^{K}\Bigg(\ln\left(\sum_{\ell=1}^{K}\frac{G_{k,\ell}p_\ell}{\sigma^2}+1\right) - \ln\left(\sum_{\ell=1,\,\ell\neq k}^{K}\frac{G_{k,\,\ell}p_\ell^*}{\sigma^2}+1\right)+1 \\ - \left(\sum_{\ell=1,\,\ell\neq k}^{K}\frac{G_{k,\,\ell}p_\ell^*}{\sigma^2}+1\right)^{-1} \times \left(\sum_{\ell=1,\,\ell\neq k}^{K}\frac{G_{k,\,\ell}p_\ell}{\sigma^2}+1\right)\Bigg) + A - D\Bigg)^2,\end{aligned} \tag{19}$$

$$\mathcal{P}_4: \min_{\boldsymbol{p}} \ \widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}(t)) \tag{20a}$$
$$\text{s.t. } \widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}(t)) \leq \frac{1}{2}D^2, \tag{20b}$$
$$\begin{aligned}&\frac{BT}{V\ln 2}\sum_{k\in\mathcal{K}_i}\Bigg(\ln\left(\sum_{\ell=1}^{K}\frac{G_{k,\ell}p_\ell}{\sigma^2}+1\right) - \ln\left(\sum_{\ell=1,\,\ell\neq k}^{K}\frac{G_{k,\,\ell}p_\ell(t)}{\sigma^2}+1\right)+1 \\ &- \left(\sum_{\ell=1,\,\ell\neq k}^{K}\frac{G_{k,\,\ell}p_\ell(t)}{\sigma^2}+1\right)^{-1} \times \left(\sum_{\ell=1,\,\ell\neq k}^{K}\frac{G_{k,\,\ell}p_\ell}{\sigma^2}+1\right)\Bigg) + A_i - D_i \leq 0,\end{aligned} \tag{20c}$$
$$\text{(5d).} \tag{20d}$$

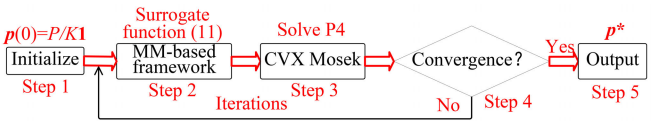


Fig. 2. Block diagram of the centralized MM-based algorithm.

(3) First-order consistency: $\nabla_{\boldsymbol{p}}\widetilde{\Phi}(\boldsymbol{p} \mid \boldsymbol{p}^*)\big|_{\boldsymbol{p}=\boldsymbol{p}^*} = \nabla_{\boldsymbol{p}}\Phi(\boldsymbol{p}^*)$.

Proof: See Appendix B. ■

With part (1) of Proposition 1, an upper bound can be directly obtained by replacing the functions $\{\Phi\}$ by $\{\widetilde{\Phi}\}$ around a feasible point. However, a tighter upper bound can be achieved if we treat the obtained solution as another feasible point, which constructs the next-round surrogate function. Specifically, assuming that the solution at the t^{th} iteration is denoted as $\boldsymbol{p}(t)$, the problem $\mathcal{P}_4$ is formulated (shown at the top of the next page) and considered at the $(t+1)^{\text{th}}$ iteration. According to part (2) of Proposition 1, $\mathcal{P}_4$ is convex and can be efficiently solved using an iterative procedure. Part (3) of Proposition 1 indicates that every limit point of $\{\boldsymbol{p}(t)\}$ is a Karush-Kuhn-Tucker point of $\mathcal{P}_3$.

Fig. 2 describes the detailed workflow of the centralized MM-based algorithm. Specifically, Step 1 initializes the power allocation as $\boldsymbol{p}(0) = P/K\mathbf{1}$, where the symbol $\mathbf{1}$ represents a column vector with all elements being one. In Step 2, the algorithm iteratively constructs a surrogate function using the MM-based framework, as per (19). Then, Step 3 utilizes standard software packages such as CVX and Mosek to solve $\mathcal{P}_4$, and Step 4 evaluates whether the solution has converged. Last, Step 5 outputs the final solution $\boldsymbol{p}^*$ while the convergence condition is satisfied.

Regarding computational complexity, this algorithm exhibits medium or high time complexity, making it suitable for small- or medium-scale IoT networks. Specifically, $\mathcal{P}_4$ involves K primal variables and $I+K+2$ dual variables. The dual variables correspond to $I+K+2$ constraints in $\mathcal{P}_4$, where $I+1$ constraints come from the sample bounds, K constraints come from non-negative power constraints, and one constraint is derived from the power budget. Consequently, the worst-case complexity for solving $\mathcal{P}_4$ is $\mathcal{O}\left((I+2K+2)^{7/2}\right)$, where the number of iterations required for convergence is ignored.

In real-world applications, the proposed algorithm can be deployed at the cloud server, such as the gateway of a small-scale edge network. The server can then inform devices of their transmit powers and other parameters through the downlink control channel, such as the narrowband physical downlink control channel in NB-IoT networks [32]. This algorithm reduces communication delays during parameter transmission. However, as large-scale IoT networks require optimization for distributed parallelism, communication efficiency across multiple edge nodes tends to degrade. Moreover, the per-iteration complexity also remains high. Thus, in the next Section V, we design a distributed first-order method (FoM) to address these issues.

V. FoM-Based Distributed Algorithm for Large-Scale IoT Networks

So far, we have addressed $\mathcal{P}_3$ using the centralized MM-based algorithm. However, the computational complexity of the solution increases as the number of IoT devices and edge nodes grows. Additionally, the proposed algorithm cannot resolve the coupling among different edge nodes, making distributed deployment challenging. To address these issues, we design a FoM-based distributed algorithm. Due to the dependence on power and co-channel interference terms in large-scale IoT networks, parallelization across edge nodes is difficult. Leveraging Theorem 1 and Proposition 1, we further relax $\mathcal{P}_4$ to handle these independently. This method yields various sub-problems through variable relaxation, facilitating an easier parallel solution [37].

Now we begin extracting the relevant sub-problems. Given Theorem 1 and Proposition 1, $\mathcal{P}_4$ can be readily relaxed as follows:

$$\mathcal{P}_5: \min_{\{\boldsymbol{p}_i,\, P_i\}_{i=1}^I} \ \frac{1}{2}\sum_{i=1}^{I}\phi_i(\boldsymbol{p}_i)^2 \tag{21a}$$

$$\text{s.t. } \phi_i(\boldsymbol{p}_i) \le 0,\ i=1,\ \cdots,\ I, \tag{21b}$$

$$\mathbf{1}^T\boldsymbol{p}_i \le P_i, p_k \ge 0,\ k \in \mathcal{K}_i, \tag{21c}$$

$$P_i \in \left\{ x_i \ge 0 \,\middle|\, \sum_{i=1}^{I} x_i = P \right\},\ i=1,\ \cdots,\ I, \tag{21d}$$

where $\phi_i(\boldsymbol{p}_i)$ is defined in (21e), as shown at the bottom of the next page.

Although $\mathcal{P}_5$ is a distributed model that can be deployed across different edge nodes, it is non-convex and challenging to solve due to its dependency on the total power across different edge nodes. Thus, we propose a first-order algorithm to solve it. Accordingly, we begin to write the augmented Lagrangian function (ALF) of $\mathcal{P}_5$ as follows:

$$\begin{aligned} &L\left(\{\boldsymbol{p}_i\}_{i=1}^I, \{P_i\}_{i=1}^I; \{\lambda_i\}_{i=1}^I, \{\beta_i\}_{i=1}^I\right) \\ &= \frac{1}{2}\sum_{i=1}^{I}\phi_i(\boldsymbol{p}_i)^2 + \sum_{i=1}^{I}\lambda_i\phi_i(\boldsymbol{p}_i) \\ &\quad + \sum_{i=1}^{I}\beta_i\left(\mathbf{1}^T\boldsymbol{p}_i - P_i\right) + \frac{\mu}{2}\sum_{i=1}^{I}\left(\mathbf{1}^T\boldsymbol{p}_i - P_i\right)^2, \end{aligned} \tag{22}$$

where μ is a penalty factor. As shown shortly, (22) enables the decomposition of sub-problems across different edge nodes. Based on (22), we formalize the following proposition.

Proposition 2: For the ALF given by (22), we can make the following iteration concerning variables P_i:

$$P_i(t) \triangleq \mathbf{1}^T\boldsymbol{p}_i(t) - \frac{1}{I}\left(\mathbf{1}^T\boldsymbol{p}(t) - P\right). \tag{23}$$

The relative dual variables $\{\beta_i\}_{i=1}^I$ are updated by

$$\beta(t+1) = \max\left(\beta(t) + \frac{\mu}{I}\left(\mathbf{1}^T\boldsymbol{p}(t+1) - P\right), 0\right), \tag{24}$$

and $\beta_i(t+1) = \beta(t+1)$, for all $i = 1,\ \cdots,\ I$.

Proof: See Appendix C. ■

By Proposition 2, it is evident that we have efficiently dealt with (21d) and reduced the dimension of dual variables. Next, we decompose the ALF in (22) into a set of sub-functions, each representing the ALF for the i^{th} edge node, given by

$$L_i(\boldsymbol{p}_i;\lambda_i,\beta)=\frac{1}{2}\phi_i(\boldsymbol{p}_i)^2+\lambda_i\phi_i(\boldsymbol{p}_i)+\beta\left(\mathbf{1}^T\boldsymbol{p}_i-P_i\right)+\frac{\mu}{2}\left(\mathbf{1}^T\boldsymbol{p}_i-P_i\right)^2. \quad (25)$$

By (25), we compute partial derivatives of the relevant variables and apply the gradient descent algorithms in distributed edge nodes, as detailed below.

1) Update $\boldsymbol{p}_i$ and P_i With Other Variables Fixed:

It is observed that $L_i(\boldsymbol{p}_i;\lambda_i,\beta)$ given by (25) is differentiable with respect to $\boldsymbol{p}_i$. Then, its gradient can be easily computed as

$$\nabla_{\boldsymbol{p}_i}L_i(\boldsymbol{p}_i;\lambda_i,\beta)=\phi_i(\boldsymbol{p}_i)\nabla_{\boldsymbol{p}_i}\phi_i(\boldsymbol{p}_i)+\lambda_i\nabla_{\boldsymbol{p}_i}\phi_i(\boldsymbol{p}_i)+\beta\mathbf{1}+\mu\left(\mathbf{1}^T\boldsymbol{p}_i-P_i\right)\mathbf{1}. \quad (26)$$

We then apply the gradient descent algorithm to update $\boldsymbol{p}_i(t+1)$, yielding

$$\boldsymbol{p}_i(t+1)=\max(\boldsymbol{p}_i(t)-\eta\nabla_{\boldsymbol{p}_i}L_i(\boldsymbol{p}_i(t);\lambda_i(t),\beta(t)),\,\mathbf{0}), \quad (27)$$

where η denotes the step-size. Also, by Proposition 2, P_i can be efficiently computed using (23) and thus, $\mathbf{1}^T\boldsymbol{p}_i(t)-P_i(t)$ can be readily updated.

2) Update Relative Dual Variables With Others Fixed:

It is evident that the ALF given by (25) is linear with respect to all dual variables. Thus, we update the dual variables as follows:

$$\lambda_i(t+1)=\max(\lambda_i(t)+\eta\phi_i(\boldsymbol{p}_i(t+1)),\,0). \quad (28)$$

Apart from these dual variables, $\beta(t+1)$ can be directly updated using (24). In real-world applications, edge node i updates (27) and (28) in parallel, and then transmits $\boldsymbol{p}_i(t+1)$ and $\lambda_i(t+1)$ to the cloud server. This cloud server updates (24) and broadcasts $\beta(t+1)$ to all edge nodes. Compared with the dimensionality of learning parameters, the optimization parameters have a lower dimensionality, making their communication delay negligible.

So far, we have developed a distributed algorithm to solve $\mathcal{P}_5$. However, the convergence rate slows down as multiple variables must be relaxed for parallelism. Therefore, we add momentum to accelerate this first-order algorithm. Specifically, $\boldsymbol{p}_i(t+1)$ is updated by

$$\boldsymbol{z}_{p_i}(t+1)=\max(\boldsymbol{p}_i(t)-\eta\nabla_{\boldsymbol{p}_i}L_i(\boldsymbol{p}_i(t);\lambda_i(t),\beta(t)),\,\mathbf{0}), \quad (29a)$$

$$\boldsymbol{p}_i(t+1)=(1-\theta(t))\boldsymbol{p}_i(t)+\theta(t)\boldsymbol{z}_{p_i}(t+1), \quad (29b)$$

$$\theta(t+1)=\frac{1}{2}\left(-\theta^2(t)+\sqrt{\theta^4(t)+4\theta^2(t)}\right). \quad (29c)$$

The procedure above is formalized in Algorithm 1, which is implemented in a distributed manner across edge nodes. In Algorithm 1, the primal variables $\{\boldsymbol{p}_i\}$ are updated locally in parallel, while the auxiliary/dual variables (e.g., λ_i) and the acceleration-related parameters are updated iteratively based on the current primal iterates.

Regarding computational complexity, the number of decision variables scales linearly with the number of devices K. Since most updates are separable across edge nodes, the per-iteration computational cost is $\mathcal{O}(K/I)$ at each edge node, and $\mathcal{O}(K)$ in total across the network, up to constant factors. Moreover, when the surrogate subproblem is convex and smooth, the accelerated first-order updates achieve the standard convergence-rate improvement from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$ with respect to the number of FoM inner iterations k.

Moreover, the convergence behavior of Algorithm 1 can be characterized by the following proposition, which summarizes standard descent and rate results for MM with inexact first-order inner solvers.

Proposition 3: (MM Descent and FoM Convergence [38], [39], [40]): Let $\{\boldsymbol{x}^{(t)}\}$ be generated by the MM framework with surrogate $g(\boldsymbol{x}\mid\boldsymbol{x}^{(t)})$ satisfying Proposition 1, and let $\boldsymbol{x}^{(t+1)}$ be obtained by approximately minimizing $g(\cdot\mid\boldsymbol{x}^{(t)})$ using a FoM until a prescribed accuracy ϵ is reached (e.g., as in Algorithm 1). If the inner solution satisfies the sufficient-decrease condition

$$g(\boldsymbol{x}^{(t+1)}\mid\boldsymbol{x}^{(t)})\le g(\boldsymbol{x}^{(t)}\mid\boldsymbol{x}^{(t)}), \quad (30)$$

then the MM outer objective is non-increasing:

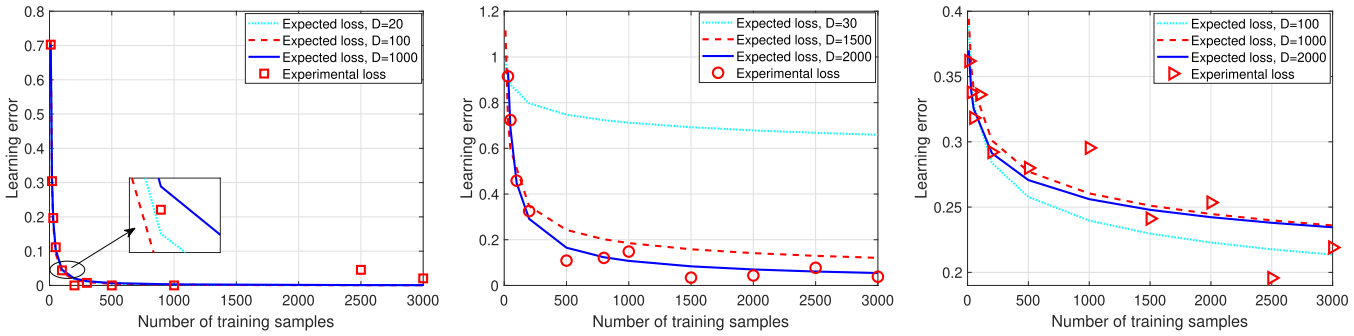
$$f(\boldsymbol{x}^{(t+1)})\le f(\boldsymbol{x}^{(t)}),\quad \forall t. \quad (31)$$

Moreover, under standard regularity conditions for MM, every limit point of $\{\boldsymbol{x}^{(t)}\}$ is a stationary point of the original problem.

For each fixed outer iterate t, if the surrogate subproblem is convex and L_g-smooth, then the FoM inner iterates $\boldsymbol{x}^{(t,k)}$ satisfy

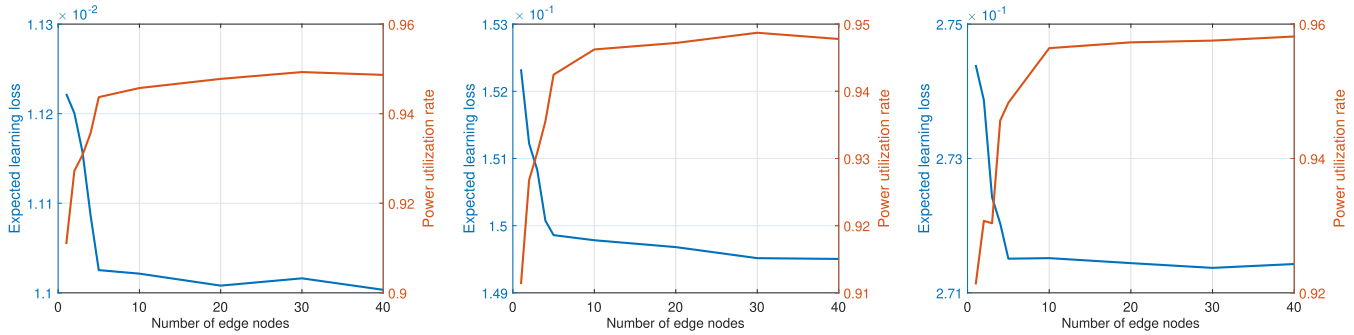
$$g(\boldsymbol{x}^{(t,k)}\mid\boldsymbol{x}^{(t)})-g^\star(\boldsymbol{x}^{(t)})=\mathcal{O}(1/k), \quad (32)$$

$$\phi_i(\boldsymbol{p}_i)\triangleq\frac{BT}{V\ln 2}\sum_{k\in\mathcal{K}_i}\left(\ln\left(\sum_{\ell\in\mathcal{K}_i}\frac{G_{k,\ell}p_\ell}{\sigma^2}+\sum_{\ell\in\mathcal{K}\setminus\mathcal{K}_i}\frac{G_{k,\ell}p_\ell(t)}{\sigma^2}+1\right)+1-\ln\left(\sum_{\ell\in\mathcal{K},\,\ell\neq k}\frac{G_{k,\ell}p_\ell(t)}{\sigma^2}+1\right)-\left(\sum_{\ell\in\mathcal{K},\,\ell\neq k}\frac{G_{k,\ell}p_\ell(t)}{\sigma^2}+1\right)^{-1}\times\left(\sum_{\ell\in\mathcal{K}_i,\,\ell\neq k}\frac{G_{k,\ell}p_\ell}{\sigma^2}+\sum_{\ell\in\mathcal{K}\setminus\mathcal{K}_i,\,\ell\neq k}\frac{G_{k,\ell}p_\ell(t)}{\sigma^2}+1\right)\right)+A_i-D_i. \quad (21e)$$



(a) Weather classification. (b) Traffic sign recognition. (c) Object detection.

Fig. 3. Learning loss vs. the number of training samples.



(a) Weather classification. (b) Traffic sign recognition. (c) Object detection.

Fig. 4. Expected learning loss/Power utilization ratio vs. the number of edge nodes.

Algorithm 1 The FoM-Based Distributed Algorithm With an Acceleration

Input: Setting $(I, N, K, P, T, B, V, \sigma^2, D_i, A_i)$, channels $\{\boldsymbol{h}_k\}$, user sets $\{\mathcal{K}_i\}$, learning rate η, optimization parameter μ, and error tolerance ϵ.

Output: The optimal solution $\boldsymbol{p}^*$.

1 Initialize $t = 0$, $\boldsymbol{p}_i(t) = P/I\mathbf{1}$, $\lambda_i(t) = 1$, $\beta(t) = 1$;
2 **repeat**
3 Update $\boldsymbol{z}_{p_i}(t+1)$ as per (29a);
4 Update $\boldsymbol{p}_i(t+1)$ as per (29b);
5 Update $\lambda_i(t+1)$ as per (28);
6 Update $\beta(t+1)$ as per (24);
7 Update $\theta(t+1)$ as per (29c);
8 $t = t + 1$;
9 **until** $\|\boldsymbol{p}_i(t) - \boldsymbol{p}_i(t-1)\|_2 \leq \epsilon$.

and the accelerated variant achieves $\mathcal{O}(1/k^2)$ under the usual convexity and smoothness assumptions.

VI. Simulation Results and Discussion

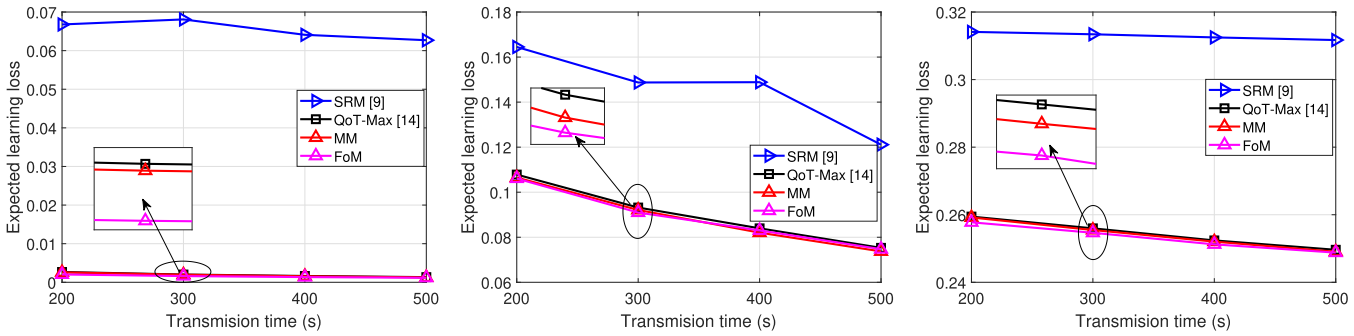
This section presents simulation results evaluating the performance of the designed algorithms, comparing them with state-of-the-art benchmarks. We primarily consider three perception tasks in autonomous driving [23], [24]: weather classification from RGB images, traffic sign recognition from RGB images, and object detection from point cloud data. The CarlaFLCAV framework, an open-source autonomous driving simulation platform available online at https://github.com/SIAT-INVS/CarlaFLCAV, generates all the datasets. The simulation parameter settings are as follows, unless otherwise specified. The size of each RGB image sample is $V_1 = V_2 = 0.7$ MB and that of each point cloud sample is $V_3 = 1.6$ MB. The number of historical data samples is $A_1 = A_2 = A_3 = 50$; the number of total IoT devices is $|\mathcal{K}| = 20$, and the number of edge nodes is $I = 10$. As for the wireless parameters, the Tx time $T = 200$ s, the number of antennas $N = 4$, the total Tx power $P = 50$ mW, the noise power $\sigma^2 = -77$ dBm, the communication bandwidth $B = 4$ MHz, the path loss of the k^{th} device $\varrho_k = -90$ dB, and the channel $\boldsymbol{h}_k$ generated by $\mathcal{CN}(\mathbf{0}, \varrho_k \boldsymbol{I})$ are set in this simulation experiments.

We consider two typical benchmark schemes: SRM [9] and QoT-Max [14]. In particular, the SRM scheme only considers the wireless channel information, ignoring the learning factors. We also simulate two proposed schemes: the MM- and FoM-based distributed algorithms. To evaluate convergence performance, we further define a mean squared error (MSE) as

$$\text{MSE} \triangleq \|\boldsymbol{p}(t) - \boldsymbol{p}(t-1)\|_2 + \|P_i(t) - P_i(t-1)\|_2 + \|\lambda_i(t) - \lambda_i(t-1)\|_2 + \|\beta(t) - \beta(t-1)\|_2 . \quad (33)$$

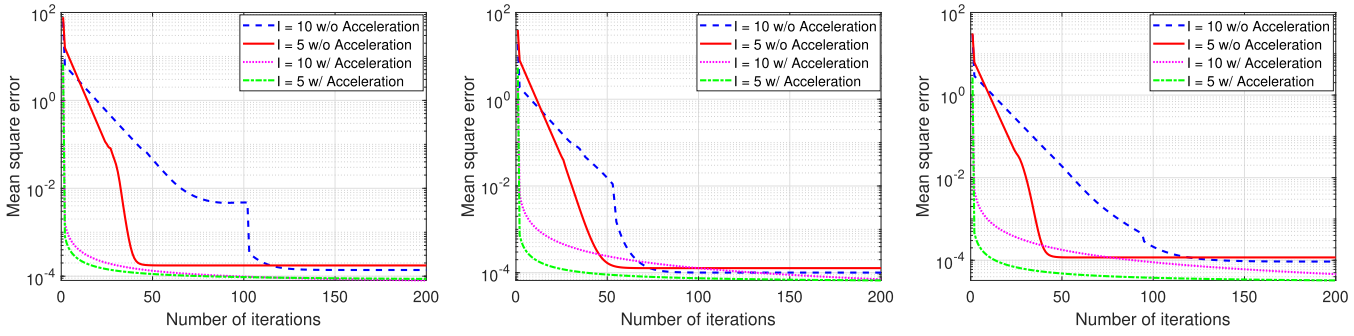
Next, we present and discuss simulation results on learning error, convergence, and complexity, followed by a case study of autonomous navigation with collision avoidance.

Due to the high computational cost of the end-to-end training–optimization loop, the curves in this section report



(a) Weather classification. (b) Traffic sign recognition. (c) Object detection.

Fig. 5. Learning loss vs. the transmission time.



(a) Weather classification. (b) Traffic sign recognition. (c) Object detection.

Fig. 6. Mean squared errors (MSE) vs. the number of iterations, with the learning step $\eta = 1 \times 10^{-3}$.

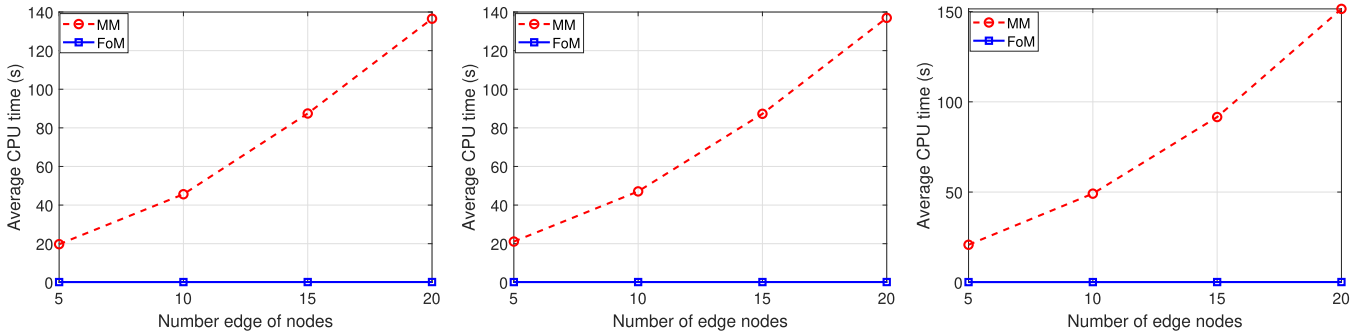
representative averaged results to illustrate the qualitative energy–learning tradeoff. In additional tests with different random seeds, we observed that the relative performance ordering remains consistent. The proposed framework also supports standard statistical reporting (e.g., mean $\pm$ standard deviation or confidence intervals) when additional computational budget is available.

A. Learning Error Performance

Fig. 3 depicts the experimental learning loss versus the number of training samples, where the expected learning losses with various training sample sizes are also plotted. Notably, we approximate the expected learning curves across different datasets using the fitting metrics described in Remark 1. Fig. 3a (Weather classification) shows that the experimental and expected learning losses decrease with the training samples, and they agree very well. This is because the expected loss converges after a sufficient number of iterations, even when a dedicated training dataset is provided. Moreover, three expected learning curves are approximately similar when the number of samples is set to $D = 20, 100, 1000$. Since the learning model attains competitive performance with small-scale datasets, scaling up the datasets yields diminishing marginal improvements. On the other hand, Figs. 3b (Traffic sign recognition) and 3c (Object detection) show that the expected learning loss significantly deviates from the actual data distribution when the number of training samples is relatively small. The reason is that a small-scale training sample results in high overfitting, and the learning model fails to capture the distribution of the actual data.

Fig. 4 demonstrates that the expected learning loss decreases as the number of edge nodes increases, while the power utilization ratio rises. This trend holds across various tasks, including weather classification, traffic sign recognition, and object detection. The primary reason for this improvement is the collaborative resource allocation framework, which enhances the efficiency of wireless resource utilization as more edge nodes are added. Additionally, increasing the number of edge nodes yields a larger sample set, thereby improving learning performance. Notably, the curves converge even with only a few edge nodes, highlighting the effectiveness of our collaborative resource allocation framework.

Fig. 5 compares the proposed MM- and FoM-based algorithms against benchmark schemes. The results demonstrate that the proposed algorithms outperform traditional SRM algorithms in terms of expected learning loss performance. This advantage arises from the fact that the proposed algorithms focus on minimizing expected learning loss and convergence bounds while optimizing wireless resource allocation based on a learning-oriented principle. This finding aligns with other task-oriented resource allocation schemes [11], [15], [16], [23], [24]. However, the SRM algorithm exhibits a weak connection between wireless resource allocation and edge learning models. Additionally, the two proposed algorithms show slightly better performance compared to the QoT-Max algorithm. It is noteworthy that the MM-based algorithm performs somewhat less effectively than the FoM-based algo-



(a) Weather classification. (b) Traffic sign recognition. (c) Object detection.

Fig. 7. Average CPU time vs. the number of edge nodes, with $|\mathcal{K}_1| = |\mathcal{K}_2| = \cdots = |\mathcal{K}_I| = 10$.

rithm. The reason is that the proposed FoM-based algorithm is adaptive to distributed learning and optimization, whereas the MM-based algorithm is designed for centralized learning and optimization.

B. Convergence Performance and Complexity Analysis

To evaluate the convergence performance of the algorithms, Fig. 6 shows the MSE, defined by (33), as a function of the number of iterations. On the one hand, we observe from Fig. 6 that the proposed algorithm without a momentum acceleration term converges faster when $I = 5$ than when $I = 10$. This trend is consistent across various tasks, including weather classification, traffic sign recognition, and object detection. The difference in convergence rates can be attributed to the greater constraint relaxation required when deploying distributed networks with $I = 10$ rather than $I = 5$. As the number of edge nodes increases, the convergence rate of the proposed algorithm gradually decreases, making it more challenging to deploy large-scale distributed networks.

On the other hand, the proposed algorithm, incorporating a momentum acceleration term, exhibits a steeper decrease, as evident in Fig. 6. This improvement occurs because the momentum acceleration term reduces the convergence rate from $\mathcal{O}(1/t)$ to $\mathcal{O}(1/t^2)$. Furthermore, this term helps prevent the algorithm from becoming trapped in local oscillations. Therefore, adding momentum makes the algorithm more suitable for deploying large-scale distributed networks. Overall, we conclude that an increase in the dynamic growth of edge nodes leads to a decrease in the convergence of the collaborative model, as shown in Figs. 6(a)-(c). This decline is attributed to reduced distributed information and heterogeneous communication, leading to significant fluctuations in the accuracy of the collaborative model.

Fig. 7 illustrates the relationship between average CPU time and the number of edge nodes. The computation time of the distributed FoM-based algorithm remains relatively stable as the number of edge nodes increases. Specifically, Figs. 7a-7c indicate that the average CPU times for three different tasks are approximately 0.15, 0.1, and 0.18 seconds, respectively. This stability is attributed to the FoM algorithm's parallel nature, which enables easy deployment across distributed networks and thereby reduces computational complexity. In contrast, the average CPU time for the centralized MM-based algorithm increases significantly as the number of edge nodes grows.

TABLE II
PERFORMANCE COMPARISON

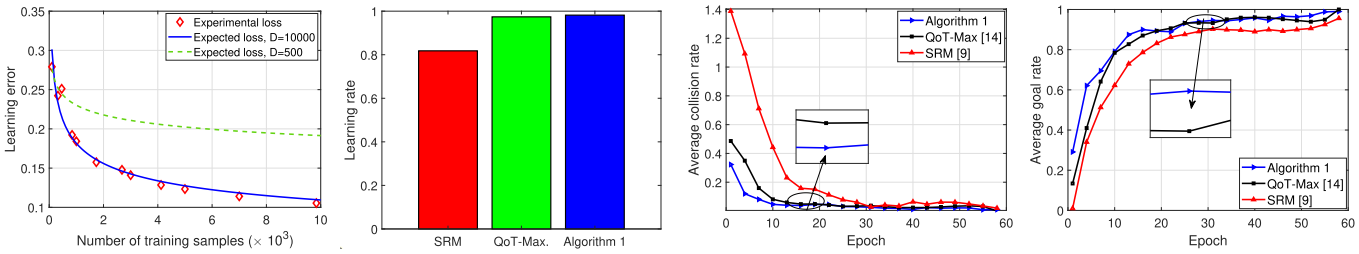
Algorithm	Complexity	Distribution[a]	Online	Learning Performance
SRM [9]	$\mathcal{O}\left((K-1)^7\right)$	✗	✗	low
QoT-Max [14]	$\mathcal{O}\left(K/I\right)$	✓	✗	high
ADMM-based LCPA [12]	$\mathcal{O}\left((K^2+K)/I\right)$	✓	✗	high
MM-based Proposal	$\mathcal{O}\left((I+K^2+K)^{3.5}\right)$	✗	✗	high
FoM-based Proposal	$\mathcal{O}\left(K/I\right)$	✓	✓	high

[a] The "✓" indicates a functionality supported, whereas the "✗" not supported.

Regarding computational complexity, the centralized MM-based algorithm involves $I + 2K + 2$ optimization variables and requires the CVX or Mosek package to solve the convex optimization problem. As a result, the worst-case complexity of this algorithm can be computed as $\mathcal{O}\left((I + 2K + 2)^{3.5}\right)$. On the other hand, the first-order FoM algorithm involves $2I + 2K + 1$ optimization variables and has a linear complexity of $\mathcal{O}(2I + 2K + 1)$. Additionally, since most of the variables can be updated in a distributed manner, the complexity of the first-order FoM algorithm can be approximated as $\mathcal{O}(K/I)$. Overall, we present the complexity results of three benchmark algorithms and two proposed algorithms in the first column of Table II.

In addition to examining computational complexity, Table II compares three other factors: distributed capability, online capability, and learning performance. The proposed FoM-based algorithm is effective for distributed networks and online learning due to its low computational complexity, strong distribution, and online capabilities. It also demonstrates high learning performance, as illustrated in Fig. 5. In contrast, the MM-based algorithm is suited for small- to medium-scale IoT networks, thanks to its high learning performance.

In practice, the proposed online optimization is executed at the edge orchestrator once per control interval (spanning multiple SGD steps) and can be warm-started across consecutive intervals. The control signaling for distributing power plans consists of only a few scalars per node and can be piggybacked on existing control messages. In our simulations, we focus on the energy–learning tradeoff and thus do not explicitly model the solver runtime and signaling



(a) The expected loss. (b) The learning rate. (c) The average collision rate. (d) The average goal rate.

Fig. 8. The training behavior in autonomous navigation.

cost; nevertheless, these overheads are typically negligible compared with training and payload communication in large-scale IoT networks, and can be further reduced by using longer control intervals or quantized power levels in small-scale deployments.

C. Case Study: Autonomous Navigation With Collision Avoidance

We evaluate the proposed algorithms through a case study on vehicle navigation with shape-based collision avoidance [41], [42]. Simulations are conducted on the IR-SIM platform [29], an open-source navigation environment (https://github.com/reiniscimurs/DRL-robot-navigation). Unless otherwise specified, we deploy $|\mathcal{K}| = 20$ sensors on a single edge server with wireless settings consistent with earlier experiments. Each LiDAR packet has a size of $V = 1.6$ MB, and the initial allocation is $A = 500$. LiDAR point-cloud datasets are generated via CarlaFLCAR and used for training within IR-SIM, while the trained models are validated in CARLA for visualizing obstacle avoidance.

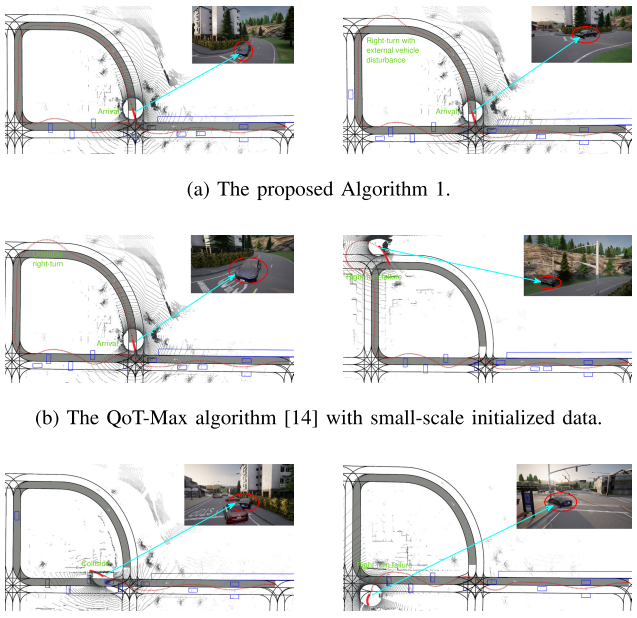
To quantify training performance, in addition to learning loss, we define three metrics. The learning rate r_l reflects the ratio of learning time to total time, the average collision rate r_c measures collisions per obstacle, and the average goal rate r_g counts successful arrivals per trial:

$$r_l = \frac{T_l}{T_l + T_o}, \quad r_c = \frac{N_c}{N_c + N_a}, \quad r_g = \frac{N_g}{N_g + N_u}, \tag{34}$$

where r_l, r_c, r_g denote the learning rate, collision rate, and goal rate, respectively; T_l, N_c, N_g are training time, number of collisions, and number of goals, while T_o, N_a, N_u are optimization time, number of avoidances, and failed goals.

Fig. 8 illustrates the training behavior. In Fig. 8a, the expected learning loss decreases with more samples, but small-scale fitting (500 samples) diverges from actual LiDAR data, while large-scale fitting (10000 samples) aligns well. Fig. 8b shows that Algorithm 1 and QoT-Max maintain learning rates above 0.95, outperforming SRM (<0.85) due to their lower computational complexity. Fig. 8c compares collision rates: Algorithm 1 consistently achieves the lowest, benefiting from its task-oriented optimization that collects more samples. Fig. 8d shows goal rates, where Algorithm 1 again surpasses QoT-Max and SRM, which suffer from insufficient data.

In the CARLA navigation task, the learning loss measures the policy's training objective (e.g., prediction/behavioral-cloning error) and serves as a proxy for control quality.



(a) The proposed Algorithm 1.

(b) The QoT-Max algorithm [14] with small-scale initialized data.

(c) The SRM algorithm [9] without task-oriented communication.

Fig. 9. Visualization of autonomous navigation in the CARLA simulator. Each subfigure pair displays the navigation trajectories (main panels) and final states (top-right panels) of the autonomous vehicles. Black and red vehicles indicate agents and obstacles, gray regions represent roads, red dashed lines show trajectories, blue polygons denote obstacles, and white circular markers indicate vehicle positions.

As training proceeds, lower loss generally leads to more stable driving behavior, consistent with the improved task-level outcomes in Fig. 8, such as a higher goal rate r_g and a lower collision rate r_c.

Fig. 9 visualizes autonomous navigation outcomes across different algorithms. The top-right subfigures show the final navigation states, while the main panels display the corresponding trajectories. Black and red vehicles in the top-right subfigures denote autonomous agents and obstacles, respectively. In the main panels, gray regions represent roads, red dashed lines indicate trajectories, blue polygons denote obstacles, and white circular markers mark the positions of autonomous vehicles.

Performance comparison across algorithms reveals distinct behaviors. For Algorithm 1 (Fig. 9a), the left subfigure shows stable navigation, and the right demonstrates successful goal completion under disturbances, enabled by efficient resource utilization and extensive training sample collection. For QoT-

Max (Fig. 9b), the left subfigure depicts an unstable but partially effective trajectory, while the right shows a right-turn failure due to divergence between small-scale initial data and the accurate LiDAR distribution. SRM (Fig. 9c) exhibits collisions and right-turn failures, as its non-task-oriented design limits data collection and learning performance.

VII. Conclusion

This paper has proposed an energy-efficient federated edge learning model for processing small-scale datasets in large-scale IoT networks. To address low resource utilization, a collaborative power allocation framework was developed to align with distributed network structures. An expected learning loss and a stochastic online learning algorithm were introduced to establish the relationship between dataset size and learning performance, enabling effective small-scale data collection. Additionally, a highly distributed algorithm was designed to solve large-scale optimization problems efficiently. Extensive experiments and case studies demonstrated that the proposed model significantly improves resource utilization and learning performance, supporting efficient distributed processing with low computational complexity. The model also showed scalability and effectiveness in autonomous driving and navigation scenarios. Future work includes extending the framework to multi-task learning in multi-modal environments.

Appendix A
Proof of Theorem 1

We begin by relaxing $F(\boldsymbol{w}(t+1))$ as follows:

$$F(\boldsymbol{w}(t+1)) \leq F(\boldsymbol{w}(t)) + \nabla F(\boldsymbol{w}(t))^T(\boldsymbol{w}(t+1)-\boldsymbol{w}(t)) + \frac{L}{2}\|\boldsymbol{w}(t+1)-\boldsymbol{w}(t)\|^2, \tag{35}$$

where (35) is derived from Assumption 1. Taking the expectation on both sides of (35) yields

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{w}(t+1))\right)\\ &\leq \mathbb{E}\Big(F(\boldsymbol{w}(t)) + (\boldsymbol{w}(t+1)-\boldsymbol{w}(t))^T\nabla F(\boldsymbol{w}(t)) + \frac{L}{2}\|\boldsymbol{w}(t+1)-\boldsymbol{w}(t+1)\|^2\Big)\\ &\leq \mathbb{E}\left(F(\boldsymbol{w}(t))\right) - \frac{1}{2L}\|\nabla F(\boldsymbol{w}(t))\|^2 + \frac{1}{2L}\mathbb{E}(\|\boldsymbol{o}(t)\|^2),\end{aligned} \tag{36}$$

where (36) is obtained by (13a)-(13b). Specifically, $\boldsymbol{o}(t)$ is defined as

$$\begin{aligned}\boldsymbol{o}(t) &\triangleq \frac{D-|\mathcal{X}(t)|}{D|\mathcal{X}(t)|}\sum_{i\in\mathcal{X}(t)}\nabla f_i(\boldsymbol{w}(t)) - \frac{1}{D}\sum_{i\in\mathcal{X}^C(t)}\nabla f_i(\boldsymbol{w}(t)),\end{aligned} \tag{37}$$

where $\mathcal{X}^C(t)$ denote the complement of $\mathcal{X}(t)$. By (37), we can derive $\mathbb{E}(\|\boldsymbol{o}(t)\|^2)$ as follows

$$\begin{aligned}&\mathbb{E}(\|\boldsymbol{o}(t)\|^2)\\ &\leq \mathbb{E}\left(\left[\left(\frac{D-|\mathcal{X}(t)|}{D|\mathcal{X}(t)|}\right)\sum_{i\in\mathcal{X}(t)}\|\nabla f_i(\boldsymbol{w}(t))\| + \frac{1}{|\mathcal{X}^C(t)|}\sum_{i\in\mathcal{X}^C(t)}\|\nabla f_i(\boldsymbol{w}(t))\|\right]^2\right)\\ &\leq 4\,\mathbb{E}\left(\left(\frac{D-|\mathcal{X}(t)|}{D}\right)^2(\xi_1+\xi_2\|\nabla F(\boldsymbol{w}(t))\|^2)\right),\end{aligned}$$

where the triangle inequality is applied twice, and the resulting terms are simplified. Substituting this result into (36) gives:

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{w}(t+1))\right)\\ &\leq \mathbb{E}\left(F(\boldsymbol{w}(t))\right) - \frac{1}{2L}\|\nabla F(\boldsymbol{w}(t))\|^2 + \frac{1}{2L}\mathbb{E}(\|\boldsymbol{o}\|^2)\\ &\leq \mathbb{E}\left(F(\boldsymbol{w}(t)) - \frac{1}{2L}\|\nabla F(\boldsymbol{w}(t))\|^2 + \frac{2\alpha}{L}(\xi_1+\xi_2\|\nabla F(\boldsymbol{w}(t))\|^2)\right)\\ &\leq \mathbb{E}\left(F(\boldsymbol{w}(t)) + \left(\frac{2\xi_2 A}{L}-\frac{1}{2L}\right)\|\nabla F(\boldsymbol{w}(t))\|^2 + \frac{2\xi_1 A}{L}\right)\\ &\leq \mathbb{E}\left(F(\boldsymbol{w}(t)) + \left(4\xi_2\alpha^2-1\right)\left(F(\boldsymbol{w}(t))-F(\boldsymbol{w}^*)\right) + \frac{2\xi_1 A}{L}\right),\end{aligned}$$

where $\alpha \triangleq (D-|\mathcal{X}(t)|/D)^2$. Therefore, we obtain $\mathbb{E}\left(F(\boldsymbol{w}(t+1))-F(\boldsymbol{w}^*)\right)$, as shown at the bottom of the page. Additionally, we can further derive:

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{w}(t+1))-F(\boldsymbol{w}^*) - \frac{2\alpha\xi_1}{(1-4\xi_2 A)L}\right)\\ &\leq \mathbb{E}\left(4\xi_2 A\left(F(\boldsymbol{w}(t))-F(\boldsymbol{w}^*) - \frac{2\alpha\xi_1}{(1-4\xi_2 A)L}\right)\right)\\ &\leq \mathbb{E}\left((4\xi_2 A)^{t+1}\left(F(\boldsymbol{w}(0))-F(\boldsymbol{w}^*) - \frac{2\alpha\xi_1}{(1-4\xi_2 A)L}\right)\right),\end{aligned}$$

which concludes (14) in Theorem 1. This completes the proof.

$$\begin{aligned}&\mathbb{E}\left(F(\boldsymbol{w}(t+1))-F(\boldsymbol{w}^*)\right)\\ &\leq \mathbb{E}\left(F(\boldsymbol{w}(t))-F(\boldsymbol{w}^*) + \left(4\xi_2\alpha^2-1\right)\left(F(\boldsymbol{w}(t))-F(\boldsymbol{w}^*)\right) + \frac{2\alpha\xi_1}{L}\right)\\ &\leq \mathbb{E}\left(4\xi_2\alpha^2(F(\boldsymbol{w}(t))-F(\boldsymbol{w}^*)) + \frac{2\alpha\xi_1}{L}\right).\end{aligned}$$

APPENDIX B
PROOF OF PROPOSITION 1

Recalling the objective function of $\mathcal{P}_3$, define the (unsquared) learning-progress term

$$\psi(\boldsymbol{p}) \triangleq \frac{BT}{V\ln 2}\sum_{k=1}^{K}\ln\left(1+\frac{G_{k,k}p_k}{\sum_{\ell\neq k}G_{k,\ell}p_\ell+\sigma^2}\right)+A-D, \tag{38}$$

so that $\Phi(\boldsymbol{p})=\big(\psi(\boldsymbol{p})\big)^2$. Likewise, let $\widetilde{\psi}(\boldsymbol{p}\mid\boldsymbol{p}^*)$ denote the expression inside the square in (19), so that $\widetilde{\Phi}(\boldsymbol{p}\mid\boldsymbol{p}^*)=\big(\widetilde{\psi}(\boldsymbol{p}\mid\boldsymbol{p}^*)\big)^2$.

Step 1: A lower bound on each log-SINR term. For each k, rewrite the log-SINR term as a difference of logarithms:

$$\ln\left(1+\frac{G_{k,k}p_k}{\sum_{\ell\neq k}G_{k,\ell}p_\ell+\sigma^2}\right) = \ln\left(1+\sum_{\ell=1}^{K}\frac{G_{k,\ell}p_\ell}{\sigma^2}\right)-\ln\left(1+\sum_{\ell\neq k}\frac{G_{k,\ell}p_\ell}{\sigma^2}\right). \tag{39}$$

Let

$$u_k(\boldsymbol{p})\triangleq 1+\sum_{\ell\neq k}\frac{G_{k,\ell}p_\ell}{\sigma^2},\qquad u_k(\boldsymbol{p}^*)\triangleq 1+\sum_{\ell\neq k}\frac{G_{k,\ell}p_\ell^*}{\sigma^2}.$$

Since $\ln(x)$ is concave on $x>0$, its first-order Taylor expansion at $x'=u_k(\boldsymbol{p}^*)$ yields the global *upper* bound

$$\ln\big(u_k(\boldsymbol{p})\big)\le\ln\big(u_k(\boldsymbol{p}^*)\big)+\frac{1}{u_k(\boldsymbol{p}^*)}\big(u_k(\boldsymbol{p})-u_k(\boldsymbol{p}^*)\big), \tag{40}$$

with equality at $\boldsymbol{p}=\boldsymbol{p}^*$. Multiplying (40) by -1 gives the corresponding global *lower* bound on $-\ln(\cdot)$:

$$-\ln\big(u_k(\boldsymbol{p})\big)\ge-\ln\big(u_k(\boldsymbol{p}^*)\big)-\frac{1}{u_k(\boldsymbol{p}^*)}\big(u_k(\boldsymbol{p})-u_k(\boldsymbol{p}^*)\big), \tag{41}$$

which is tight at $\boldsymbol{p}=\boldsymbol{p}^*$. Substituting (41) into (39) yields, for each k,

$$\ln\left(1+\frac{G_{k,k}p_k}{\sum_{\ell\neq k}G_{k,\ell}p_\ell+\sigma^2}\right) \ge \ln\left(1+\sum_{\ell=1}^{K}\frac{G_{k,\ell}p_\ell}{\sigma^2}\right)-\ln\big(u_k(\boldsymbol{p}^*)\big)+1-\frac{u_k(\boldsymbol{p})}{u_k(\boldsymbol{p}^*)}, \tag{42}$$

which is exactly the inner surrogate term used in (19).

Step 2: Lower bound on $\psi(\boldsymbol{p})$ and majorization of $\Phi(\boldsymbol{p})$. Summing (42) over k and scaling by $\frac{BT}{V\ln 2}$ yields

$$\psi(\boldsymbol{p})\ \ge\ \widetilde{\psi}(\boldsymbol{p}\mid\boldsymbol{p}^*),\quad\forall\boldsymbol{p}. \tag{43}$$

In the considered data-deficit operating regime, $\psi(\boldsymbol{p})\ge 0$ holds over the feasible set, and the surrogate construction also ensures $\widetilde{\psi}(\boldsymbol{p}\mid\boldsymbol{p}^*)\ge 0$. Therefore, the pointwise inequality (43) implies

$$\Phi(\boldsymbol{p})=\psi(\boldsymbol{p})^2\ \le\ \widetilde{\psi}(\boldsymbol{p}\mid\boldsymbol{p}^*)^2=\widetilde{\Phi}(\boldsymbol{p}\mid\boldsymbol{p}^*),$$

which establishes the majorization property in Proposition 1.

Step 3: Tangency and first-order consistency. The bound (40)–(41) is tight at $\boldsymbol{p}=\boldsymbol{p}^*$ because it is the first-order Taylor expansion at $u_k(\boldsymbol{p}^*)$. Hence (42) holds with equality at $\boldsymbol{p}=\boldsymbol{p}^*$ for every k, which implies $\widetilde{\psi}(\boldsymbol{p}^*\mid\boldsymbol{p}^*)=\psi(\boldsymbol{p}^*)$ and thus $\widetilde{\Phi}(\boldsymbol{p}^*\mid\boldsymbol{p}^*)=\Phi(\boldsymbol{p}^*)$. Moreover, since the surrogate is constructed via first-order Taylor expansion at $\boldsymbol{p}^*$, the gradients coincide at the expansion point, i.e., $\nabla_{\boldsymbol{p}}\widetilde{\Phi}(\boldsymbol{p}\mid\boldsymbol{p}^*)\big|_{\boldsymbol{p}=\boldsymbol{p}^*}=\nabla_{\boldsymbol{p}}\Phi(\boldsymbol{p}^*)$. This completes the proof.

APPENDIX C
PROOF OF PROPOSITION 2

Fix $\{\boldsymbol{p}_i\}_{i=1}^I$ and consider minimizing the ALF (22) w.r.t. $\{P_i\}_{i=1}^I$ under the coupling constraint $\sum_{i=1}^I P_i=P$ (and $P_i\ge 0$). To obtain a distributed-friendly form, we enforce the consensus structure $\beta_i=\beta$ for all i (this is standard in dual decomposition / augmented-Lagrangian treatments).

Introduce a Lagrange multiplier ν for the equality $\sum_i P_i=P$. The partial derivatives of the augmented Lagrangian w.r.t. P_i give the optimality conditions

$$\frac{\partial}{\partial P_i}\left[\sum_{j=1}^{I}\beta(\mathbf{1}^T\mathbf{p}_j-P_j)+\frac{\mu}{2}\sum_{j=1}^{I}(\mathbf{1}^T\mathbf{p}_j-P_j)^2\right]-\nu=0, \tag{44}$$

which reduces to

$$-\beta-\mu(\mathbf{1}^T\mathbf{p}_i-P_i)-\nu=0\implies P_i=\mathbf{1}^T\mathbf{p}_i-\frac{\beta+\nu}{\mu}. \tag{45}$$

Summing this identity over $i=1,\cdots,I$ and using $\sum_i P_i=P$ yields

$$P=\sum_{i=1}^{I}\mathbf{1}^T\mathbf{p}_i-I\frac{\beta+\nu}{\mu}\implies\frac{\beta+\nu}{\mu}=\frac{\sum_i\mathbf{1}^T\mathbf{p}_i-P}{I}. \tag{46}$$

Substituting back gives the intended (23). Finally, treating β as the dual variable associated with the (global) equality constraint, a projected gradient (dual ascent) step for β on the augmented Lagrangian with step-size μ/I yields the desired (24). This completes the proof.

REFERENCES

[1] Q. Duan, J. Huang, S. Hu, R. Deng, Z. Lu, and S. Yu, "Combining federated learning and edge computing toward ubiquitous intelligence in 6G network: Challenges, recent advances, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2892–2950, 2023.

[2] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Robust federated learning for unreliable and resource-limited wireless networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9793–9809, Aug. 2024.

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

[4] S. Yue, J. Ren, J. Xin, D. Zhang, Y. Zhang, and W. Zhuang, "Efficient federated meta-learning over multi-access wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1556–1570, May 2022.

[5] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 219–232, Jan. 2021.

[6] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[7] X. Pei, H. Yu, Y. Chen, M. Wen, and G. Chen, "Hybrid multicast/unicast design in NOMA-based vehicular caching system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16304–16308, Dec. 2020.

[8] S. Yu, X. Gong, Q. Shi, X. Wang, and X. Chen, "EC-SAGINs: Edge-computing-enhanced space–air–ground-integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5742–5754, Apr. 2022.

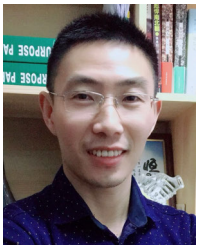
[9] N. S. Perovic, L.-N. Tran, M. D. Renzo, and M. F. Flanagan, "On the maximum achievable sum-rate of the RIS-aided MIMO broadcast channel," *IEEE Trans. Signal Process.*, vol. 70, pp. 6316–6331, 2022.
[10] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.
[11] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, "Machine intelligence at the edge with learning centric power allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7293–7308, Nov. 2020.
[12] H. Xie, M. Xia, P. Wu, S. Wang, and H. V. Poor, "Edge learning for large-scale Internet of Things with task-oriented efficient communication," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9517–9532, Dec. 2023.
[13] H. Xie, M. Xia, P. Wu, S. Wang, and K. Huang, "Decentralized federated learning with asynchronous parameter sharing for large-scale IoT networks," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34123–34139, Nov. 2024.
[14] X. Li, T. Zhang, S. Wang, G. Zhu, R. Wang, and T.-H. Chang, "Large-scale bandwidth and power optimization for multi-modal edge intelligence autonomous driving," *IEEE Wireless Commun. Lett.*, vol. 12, no. 6, pp. 1096–1100, Jun. 2023.
[15] J. Shao, Y. Mao, and J. Zhang, "Learning task-oriented communication for edge inference: An information bottleneck approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 197–211, Jan. 2022.
[16] D. Wen et al., "Task-oriented sensing, computation, and communication integration for multi-device edge AI," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 2486–2502, Mar. 2024.
[17] P. Liu et al., "Toward ambient intelligence: Federated edge learning with task-oriented sensing, computation, and communication integration," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 158–172, Jan. 2023.
[18] C. Y. Liu, M. Xia, J. Zhao, H. Li, and Y. Gong, "Optimal resource allocation for integrated sensing and communications in Internet of Vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 74, no. 2, pp. 3028–3038, Feb. 2024.
[19] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented multi-user semantic communications for VQA," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 553–557, Mar. 2022.
[20] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.
[21] Y. He, J. Ren, G. Yu, and J. Yuan, "Importance-aware data selection and resource allocation in federated edge learning system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13593–13605, Nov. 2020.
[22] S. Prakash et al., "Coded computing for low-latency federated learning over wireless edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 233–250, Jan. 2021.
[23] S. Wang et al., "Federated deep learning meets autonomous vehicle perception: Design and verification," *IEEE Netw.*, vol. 37, no. 3, pp. 16–25, May 2023.
[24] W.-B. Kou et al., "Communication resources constrained hierarchical federated learning for end-to-end autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 9383–9390.
[25] C. Qiao, M. Li, Y. Liu, and Z. Tian, "Transitioning from federated learning to quantum federated learning in Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 1, pp. 509–545, Feb. 2025.
[26] S. Liu, Y. Zhang, W. Chen, and P. Wu, "DASP: Hierarchical offline reinforcement learning via diffusion autodecoder and skill primitive," *IEEE Robot. Autom. Lett.*, vol. 10, no. 2, pp. 1649–1655, Feb. 2025.
[27] T. Li et al., "Applications of multi-agent reinforcement learning in future internet: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1240–1279, 2nd Quart., 2022.
[28] J. Zhao, H. Quan, M. Xia, and D. Wang, "Adaptive resource allocation for mobile edge computing in Internet of Vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5834–5848, Apr. 2024.
[29] R. Cimurs, I. H. Suh, and J. H. Lee, "Goal-driven autonomous exploration through deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 730–737, Apr. 2022.
[30] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM*, vol. 60, no. 2, pp. 223–311, May 2018.
[31] E. Chen, P. Wu, Y.-C. Wu, and M. Xia, "UGV-assisted wireless powered backscatter communications for large-scale IoT networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3147–3161, May 2022.
[32] S. Zhang et al., "System-level simulation framework for NB-IoT: Key features and performance evaluation," *IEEE Syst. J.*, vol. 19, no. 2, pp. 577–588, Jun. 2025.
[33] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.:, Cambridge Univ. Press, 2014.
[34] S. Bubeck, *Convex Optimization: Algorithms and Complexity*. Hanover, MA, USA: Now Publishers, 2015.
[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, Dec. 2018, pp. 3835–3844.
[36] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2019, pp. 1–26.
[37] C. Lu, J. Feng, S. Yan, and Z. Lin, "A unified alternating direction method of multipliers by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 527–541, Mar. 2018.
[38] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
[39] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Boston, MA, USA: Kluwer Academic Publishers, 2004.
[40] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.
[41] R. Han et al., "RDA: An accelerated collision free motion planner for autonomous navigation in cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1715–1722, Mar. 2023.
[42] Z. Ji et al., "Robotic sensor network: Achieving mutual communication control assistance with fast cross-layer optimization," *IEEE Wireless Commun. Lett.*, vol. 14, no. 2, pp. 385–389, Feb. 2025.

Haihui Xie (Member, IEEE) received the B.S. and M.S. degrees in photonic and electronic engineering from Fujian Normal University, Fuzhou, China, in 2014 and 2016, respectively, and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2023. He is currently a Lecturer with Fujian Agriculture and Forestry University (FAFU), Fuzhou. His research interests include edge learning, optimization, and wireless communications.

Wenkun Wen (Member, IEEE) received the Ph.D. degree in telecommunications and information systems from Sun Yat-sen University, Guangzhou, China, in 2007.

Since 2020, he has been with Techphant Technologies Company Ltd., Guangzhou, as a Chief Engineer. From 2008 to 2009, he was with Guangdong-Nortel Research and Development Center, Guangzhou, where he was a System Engineer for 4G systems. From 2009 to 2012, he was with the LTE Research and Development Center, New Postcom Equipment Company Ltd., Guangzhou, where he was the 4G Standard Team Manager. From 2012 to 2018, he was with the 7th Institute of China Electronic Technology Corporation (CETC), as an expert in wireless communications. From 2018 to 2020, he was the Deputy Director of the 5G Innovation Center, CETC. His research interests include 5G/B5G mobile communications, machine-type communications, narrow-band wireless communications, and signal processing.

Shuwu Chen received the M.S. degree in radio physics from Xiamen University in 2003. He is currently a Professor with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University. He is also the Director of the Engineering Research Center of Smart Sensing and Agricultural Chip Technology, Fujian Province University. His research interests include the Internet of Things, edge computing, and artificial intelligence algorithms.



Zhaogang Shu (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Shantou University, China, in 2002 and 2005, respectively, and the Ph.D. degree from South China University of Technology, Guangzhou, China, in 2008.

From September 2008 to July 2012, he was a Senior Engineer and the Project Manager of Ruijie Network Corporation, Fuzhou, China. From October 2018 to October 2019, he was a Visiting Professor with the MOSIAC Laboratory, Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland. He is currently an Associate Professor with the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou, China, and the Director of the Network System and Cloud Computing Laboratory, Fujian Agriculture and Forestry University. He led more than ten research projects and authored more than 40 articles and ten patents. His research interests include software-defined networks, network function virtualization, 5G networks, next-generation network architecture, network security, machine learning-based network optimization, cloud computing, and edge computing. He is a Senior Member of China Computer Federation and Fujian Computer Society.

Minghua Xia (Senior Member, IEEE) received the Ph.D. degree in telecommunications and information systems from Sun Yat-sen University, Guangzhou, China, in 2007.

From 2007 to 2009, he was with the Electronics and Telecommunications Research Institute (ETRI), South Korea, Beijing Research and Development Center, Beijing, China, where he was a member and then as a Senior Member of the Engineering Staff. From 2010 to 2014, he was in sequence with The University of Hong Kong, Hong Kong, China; King Abdullah University of Science and Technology, Jeddah, Saudi Arabia; and the Institut National de la Recherche Scientifique (INRS), University of Quebec, Montreal, Canada, as a Post-Doctoral Fellow. Since 2015, he has been a Professor with Sun Yat-sen University. Since 2019, he has also been an Adjunct Professor with the Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai). His research interests include wireless communications and signal processing.