

Closing the Generalization Gap in Parameter-efficient Federated Edge Learning

Xinnong Du, Zhonghao Lyu, *Member, IEEE*, Xiaowen Cao, *Member, IEEE*, Chunyang Wen,
Shuguang Cui, *Fellow, IEEE*, and Jie Xu, *Fellow, IEEE*

Abstract—Federated edge learning (FEEL) provides a promising foundation for edge artificial intelligence (AI) by enabling collaborative model training while preserving data privacy. However, limited and heterogeneous local datasets, as well as resource-constrained deployment, severely degrade both model generalization and resource utilization, leading to a compromised learning performance. Therefore, we propose a parameter-efficient FEEL framework that jointly leverages model pruning and client selection to tackle such challenges. First, we derive an information-theoretic generalization statement that characterizes the discrepancy between training and testing function losses and embed it into the convergence analysis. It reveals that a larger local generalization statement can undermine the global convergence. Then, we formulate a generalization-aware average squared gradient norm bound minimization problem, by jointly optimizing the pruning ratios, client selection, and communication-computation resources under energy and delay constraints. Despite its non-convexity, the resulting mixed-integer problem is efficiently solved via an alternating optimization algorithm. Extensive experiments demonstrate that the proposed design achieves superior learning performance than state-of-the-art baselines, validating the effectiveness of coupling generalization-aware analysis with system-level optimization for efficient FEEL.

Index Terms—Federated edge learning (FEEL), generalization analysis, model pruning, client selection, joint resource management.

I. INTRODUCTION

A. Background

Driven by the deep integration of communication networks and artificial intelligence (AI), edge AI is expected to become a key application scenario for next-generation wireless systems [1]. However, in edge AI, the distributed data resources across edge environments are difficult to be fully exploited for effective model training [2]. To address this issue, federated edge learning (FEEL) has emerged as a promising paradigm

Xinnong Du, Shuguang Cui, and Jie Xu are with the School of Science and Engineering (SSE), the Shenzhen Future Network of Intelligence Institute (FNii-Shenzhen), and the Guangdong Provincial Key Laboratory of Future Networks of Intelligence, The Chinese University of Hong Kong (Shenzhen), Longgang, Shenzhen 518172, China (e-mail: xinnongdu@link.cuhk.edu.cn, shuguangcui@cuhk.edu.cn, xujie@cuhk.edu.cn).

Zhonghao Lyu is with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: lzhon@kth.se).

Xiaowen Cao is with the College of Electronic and Information Engineering, Shenzhen University, and Guangdong Provincial Key Laboratory of Future Networks of Intelligence, The Chinese University of Hong Kong (Shenzhen), Longgang, Shenzhen 518172, China (email: caoxwen@szu.edu.cn). Xiaowen Cao is the corresponding author.

Chunyang Wen is with University of Science and Technology of China (USTC), Hefei 230026, China (email: chywen@mail.ustc.edu.cn).

that enables collaborative model training while preserving data privacy [1].

Despite the benefits, the FEEL framework often faces the data isolation problem and suffers from overfitting, as locally trained models at distributed edge nodes tend to bias toward their own data, leading to degraded generalization performance [3]. This issue becomes more severe in non-independent and identically distributed (non-IID) scenarios [4], where heterogeneous client data distributions result in slower convergence. This fundamentally limits the robustness and scalability of edge AI, underscoring the need to enhance model generalization in distributed learning.

B. Related Works

In FEEL, existing studies generally adopt three representative architectures, namely centralized, decentralized, and hierarchical architectures, corresponding to model aggregation at a central server [5], direct inter-client coordination [6], and joint edge-cloud cooperation [7], respectively. Within these architectures, achieving reliable learning performance in practical deployment over wireless edge networks remains challenging, due to the limited communication and computation capabilities at distributed edge nodes [8], [9]. To address this challenge, several studies have characterized learning performance in terms of the optimality gap through convergence analysis, and leveraged these insights to guide system-level optimization [10], [11]. Specifically, some prior works have investigated the joint allocation of communication (e.g., bandwidth and transmit power) and computation resources (e.g., CPU frequency) [12], as well as training configurations (e.g., batch size and training rounds) [13], to minimize the optimality gap while reducing the learning delay and energy consumption. Moreover, to handle data heterogeneity, existing works have incorporated client selection and algorithm designs such as gradient correction to mitigate the impact of non-IID data distributions [14], [15], [16]. Meanwhile, beyond optimizing learning performance, other studies have further studied the minimization of overall energy consumption or learning latency to further enhance system efficiency, while ensuring the learning performance [17], [18].

In addition to optimizing the convergence speed or energy efficiency, another line of work focuses on enhancing the generalization performance of FEEL models via techniques, such as data sharing [19], knowledge transferring [20], and mathematical analysis [21]. In particular, the authors in [19] have used the data sharing and global datasets synthesizing

to achieve more balanced data distributions across clients, thereby enabling unbiased local learning and mitigating the detrimental effects of statistical heterogeneity. [20] has leveraged knowledge transfer, such as the aggregated global model outputs or the average local predictions, to refine local update directions and enhance cross-client output consistency, leading to improved model alignment. In addition, other works have adjusted aggregation strategies or adopted two-stage learning frameworks that sequentially perform global training and local fine-tuning, effectively strengthening the generalization performance [22], [23]. However, such data sharing and knowledge transfer strategies compromise the privacy-preserving capability of FEEL. Meanwhile, the additional transmission of the sharing data exacerbates the challenge of constrained edge resources. Moreover, recent works have employed information-theoretic frameworks to derive generalization analysis by quantifying the mutual information between data labels and extracted features [21], [24] or by establishing tighter probably approximately correct (PAC) Bayesian generalization bounds under non-IID settings [25]. Although these studies provide interpretable theoretical analyses of generalization, they do not consider how such insights can be incorporated into system design aspects.

On the other hand, with the increasing scale of model size, deploying AI models on resource-limited edge devices becomes challenging [26]. Model compression techniques such as sparsification [27], quantization [28], and pruning [29]–[33] provide an effective solution for alleviating the resource bottlenecks. Among them, model pruning has been widely used for designing efficient lightweight FEEL recently. Existing works mainly focus on the characterization of parameter importance and the adjustment of pruning ratio. Specifically, some works have estimated parameter redundancy based on local accuracy gains to determine the pruning ratio [30], while others have jointly optimized pruning ratios with respect to (w.r.t.) the computational and communication resources of edge devices [31]. Once the pruning ratio is determined, the clients prune less important parameters to achieve lightweight training, where the importance is represented by gradient variations [32] or model weight magnitudes [33]. However, existing lightweight FEEL frameworks mainly focus on reducing the communication and computation costs through isolated compression mechanisms, while overlooking their coupling with learning dynamics and generalization behavior.

C. Motivations and Contributions

Considering the above issues, it is essential to accurately analyze the generalization behavior of FEEL and incorporate such analysis into its parameter-efficient design. However, characterizing such behavior is fundamentally challenging. First, there is no clear analytical way to describe the mismatch between local training data and the target task, as each client observes only a limited and biased dataset. Consequently, it is non-trivial to establish a practical measure of how well a client's local update can generalize to unseen data. Second, incorporating generalization into convergence analysis to reflect the model's reliability on unseen tasks is also challenging, as it

requires capturing the effects of key system parameters such as client selection and data imbalance. Finally, the deployment of generalization-aware and parameter-efficient FEEL at the network edge introduces another layer of complexity. Effectively exploiting heterogeneous and constrained resources at edge clients is highly challenging. Moreover, parameter-efficient strategies are closely coupled to the resource management schemes. This coupling makes it challenging to balance learning performance, energy efficiency, and delay. These challenges have not been well investigated in existing works, thus motivate our work.

In this paper, we propose a parameter-efficient FEEL system to enhance generalization performance in heterogeneous and resource-constrained edge environments. We derive a novel learning performance bound by analyzing the generalization gap, which quantitatively captures the discrepancy between training and testing behaviors. Based on this analysis, we jointly optimize system resources, client selection, and model pruning ratios to accelerate the learning convergence. The main contributions are summarized as follows.

- **Generalization analysis:** We derive a novel theoretical generalization statement, grounded in information-theoretic principles, that quantifies the deviation between local training and testing distributions. We incorporate the derived generalization statement into the convergence analysis and couple it with the client selection indicator to mitigate the impact of data heterogeneity on generalization. It is noteworthy that when the local training distribution of the client is more closely aligned with the sampled testing distribution, the gap between its training and test losses is reduced. This observation sheds light on how to schedule participating devices to improve generalization performance.
- **Joint optimization framework:** Building on the convergence and generalization analysis, we formulate an optimization problem that minimizes the generalization-aware average squared gradient norm bound, by jointly optimizing client participation, model pruning ratios, and communication and computation parameters under the constraints on overall system energy consumption and delay. Subsequently, we design an efficient algorithm to tackle this complex non-convex problem by leveraging alternating optimization based on successive convex approximation (SCA).
- **Simulation evaluation:** We conduct extensive simulations to evaluate the performance of the proposed design, where data heterogeneity across clients is simulated using a Dirichlet-based non-IID setting. The results demonstrate that, under given energy and latency constraints, our design significantly achieves higher test accuracy compared with other benchmark schemes. This improvement stems from the system-level joint resource optimization guided by generalization analysis.

The remainder of this paper is organized as follows. Section II describes the proposed FEEL system model. Section III provides convergence and generalization analysis. Section IV presents the joint optimization framework that integrates gen-

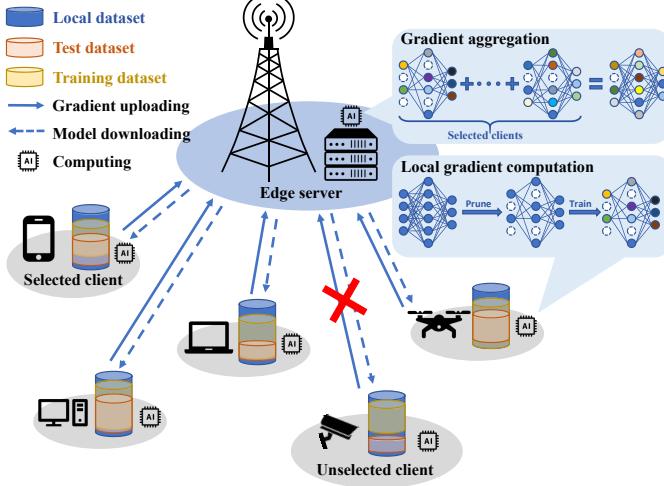


Figure 1. Illustration of the considered FEEL system over wireless communication networks with model pruning.

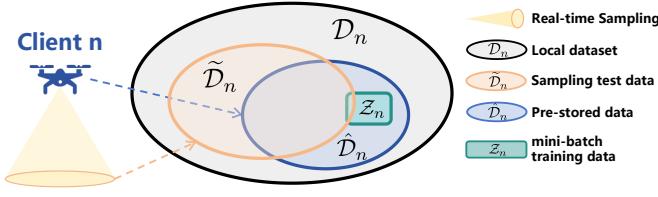


Figure 2. Hierarchical structure of the local dataset at client n .

eralization analysis with resource constraints. Section V provides numerical results on benchmark datasets under various non-IID settings. Finally, Section VI concludes the paper.

II. SYSTEM MODEL

We consider a FEEL system consisting of an edge server and multiple clients with a set of $\mathcal{N} = \{1, \dots, N\}$ as shown in Fig. 1. Suppose that client $n \in \mathcal{N}$ has a local dataset \mathcal{D}_n comprising the task-oriented data, as illustrated in Fig. 2. Denote $\mathcal{D}_n = \{(x_{n,i}, y_{n,i})\}_{i=1}^{D_n}$ with size of D_n , where $x_{n,i}$ and $y_{n,i}$ represent the data sample and its corresponding ground-truth label, respectively. It is assumed that client n has a pre-stored training dataset $\hat{\mathcal{D}}_n$ with size of \hat{D}_n and a testing dataset $\tilde{\mathcal{D}}_n$ with size of \tilde{D}_n . To address the resource constraints in distributed edge AI scenarios, we adopt a parameter-efficient training strategy in the following.

A. Parameter-efficient FEEL

The FEEL enables the server and clients to collaboratively learn a shared model, represented by the parameter vector $\omega \in \mathbb{R}^M$, where M is the model size. In general, the FEEL algorithm aims at minimizing the global loss to obtain optimized model parameters, i.e.,

$$\omega^* = \arg \min_{\omega} \mathcal{L}(\omega, \hat{\mathcal{D}}), \quad (1)$$

where $\mathcal{L}(\omega, \hat{\mathcal{D}})$ denotes the loss function w.r.t. model parameters ω over the entire training dataset $\hat{\mathcal{D}} \triangleq \cup_{n=1}^N \hat{\mathcal{D}}_n$. To

solve this problem, we adopt the federated stochastic gradient descent (FedSGD) framework [34] over training rounds with a set of $\mathcal{S} = \{0, 1, \dots, S\}$, during which client selection and model pruning are incorporated to improve the parameter efficiency.

First, we introduce the client selection strategy. Denote $a_n^{(s)}$ as the binary selection indicator, where $a_n^{(s)} = 1$ if client n is selected in round $s \in \mathcal{S}$. We also assume that the set of selected clients is denoted as $\tilde{\mathcal{N}}^{(s)}$ with size of $\tilde{N}^{(s)}$.

Then, we introduce the model pruning. We define the pruning ratio of client n at round s as $\lambda_n^{(s)}$, which is the ratio of the pruned model size $\hat{M}_n^{(s)}$ to the full model size M , i.e.,

$$\lambda_n^{(s)} = \frac{\hat{M}_n^{(s)}}{M}, \forall n, s. \quad (2)$$

We denote $\mathbf{Q}_n^{(s)} = \{Q_{n,m}^{(s)}\}_{m=1}^M$ as the importance score of model weights, which is measured by the squared error in the loss caused by the removal of the corresponding weight [32]. For each parameter m , the corresponding importance score at client n in round s is given by

$$Q_{n,m}^{(s)} = (\bar{\mathcal{L}}_n^{(s)}(\omega_n^{(s)}, \hat{\mathcal{D}}_n) - \bar{\mathcal{L}}_n^{(s)}(\omega_n^{(s)}|_{\rho_{n,m}^{(s)}=0}, \hat{\mathcal{D}}_n))^2, \forall m, n, s, \quad (3)$$

where $\omega_n^{(s)}|_{\rho_{n,m}^{(s)}=0}$ is the unpruned model parameter vector $\omega_n^{(s)}$ with the weight $\rho_{n,m}^{(s)}$ setting to be zero and $\bar{\mathcal{L}}_n^{(s)}(\omega_n^{(s)}, \hat{\mathcal{D}}_n)$ is the local loss function for client n in round s . However, the evaluation in (3) is with high computation complexity. Denote $v_m^{(s-1)}$ as the global gradient of specific model weight $\rho_{n,m}^{(s-1)}$. To reduce the computational overhead, we adopt the first-order Taylor approximation as a surrogate [31]. Specifically, by expanding $\bar{\mathcal{L}}_n^{(s)}(\omega_n^{(s)}|_{\rho_{n,m}^{(s)}=0}, \hat{\mathcal{D}}_n)$, we have $\bar{\mathcal{L}}_n^{(s)}(\omega_n^{(s)}|_{\rho_{n,m}^{(s)}=0}, \hat{\mathcal{D}}_n) \approx \bar{\mathcal{L}}_n^{(s)}(\omega_n^{(s)}, \hat{\mathcal{D}}_n) - v_m^{(s-1)} \rho_{n,m}^{(s-1)}$. Substituting this approximation into (3) yields the first-order importance estimation

$$Q_{n,m}^{(s)} = (v_m^{(s-1)} \rho_{n,m}^{(s-1)})^2, \forall m = 1, \dots, M. \quad (4)$$

With client selection and model pruning, the FEEL is implemented as follows. At the beginning of each round, selected clients compute $\mathbf{Q}_n^{(s)}$ from the received global gradient and its local parameters through (4). Each selected client prunes its local model by removing weights of low importance according to $\mathbf{Q}_n^{(s)}$ and $\lambda_n^{(s)}$, and then obtain the pruned model $\tilde{\omega}_n^{(s)}$. Denote $l(\tilde{\omega}_n^{(s)}; x_{n,i}, y_{n,i})$ as the loss function of $\tilde{\omega}_n^{(s)}$ on data point $(x_{n,i}, y_{n,i})$. Then, the selected client n randomly generates a mini-batch $Z_n^{(s)}$ from $\hat{\mathcal{D}}_n$ with size of $Z_n^{(s)} \geq 1$ in round s to compute the local gradient, i.e.,

$$\nabla \bar{\mathcal{L}}_n^{(s)}(\tilde{\omega}_n^{(s)}, Z_n^{(s)}) = \frac{1}{Z_n^{(s)}} \sum_{i=1}^{Z_n^{(s)}} \nabla l(\tilde{\omega}_n^{(s)}; x_{n,i}, y_{n,i}), \forall n, s. \quad (5)$$

Next, all selected clients upload their local gradients and the corresponding pruning indices to the server. To clearly define the global gradient, we use $\tilde{\omega}^{(s)}$ to denote the global pruned model obtained by aggregating the selected local pruned models. Then, the server aggregates local gradients

to obtain the global gradient over the entire training batch $\mathcal{Z}^{(s)} \triangleq \bigcup_{n \in \tilde{\mathcal{N}}^{(s)}} \mathcal{Z}_n^{(s)}$:

$$\nabla \mathcal{L}^{(s)}(\tilde{\omega}^{(s)}, \mathcal{Z}^{(s)}) = \frac{1}{\tilde{N}^{(s)}} \sum_{n \in \tilde{\mathcal{N}}^{(s)}} \nabla \bar{\mathcal{L}}_n^{(s)}(\tilde{\omega}_n^{(s)}, \mathcal{Z}_n^{(s)}), \forall n, s, \quad (6)$$

based on which the server can update the global model as

$$\omega^{(s+1)} = \omega^{(s)} - \eta \nabla \mathcal{L}^{(s)}(\tilde{\omega}^{(s)}, \mathcal{Z}^{(s)}), \quad (7)$$

where η denotes the learning rate. Finally, the server sends the global gradient back to all clients for local model update and importance score computation, while the pruning ratios are sent to the selected clients to start the next round. Notably, client selection and pruning ratio are determined by the server through the optimization process discussed in Section IV.

B. Wireless Communication Model

We then introduce the uplink communication model. All clients transmit their local gradients by using the technique of frequency-division multiple access (FDMA). Denote $p_n^{(s)}$ as the transmit power of client n in round s . Then, the uplink rate (in bits-per-second (bps)) for client n is

$$r_n^{(s)}(p_n^{(s)}) = c_n \log_2 \left(1 + \frac{p_n^{(s)} h_n^{(s)}}{c_n U_0^{(s)}} \right), \forall n, s, \quad (8)$$

where c_n denotes the bandwidth allocated to client n , $h_n^{(s)}$ is the channel power gain from client n to the server, and $U_0^{(s)}$ denotes the power spectral density (PSD) of the additive white Gaussian noise (AWGN) at the server in round s .

Next, we introduce the downlink communication model. The server adopts a multicast transmission strategy, enabling the dissemination of identical data streams to multiple clients. Denote \hat{p} as the transmit power of the server and $\hat{h}_n^{(s)}$ as the channel power gain from the server to client n in round s . Then, the achievable downlink communication rate is

$$\hat{r}_n^{(s)} = \hat{c} \log_2 \left(1 + \frac{\hat{p} \hat{h}_n^{(s)}}{\hat{c} U_{0,n}^{(s)}} \right), \forall n, s, \quad (9)$$

where \hat{c} is the bandwidth for broadcasting the global gradient and $U_{0,n}^{(s)}$ is the PSD of AWGN at client n in round s .

C. System Delay and Energy Consumption

1) *System Delay*: In general, the system delay in FEEL arises from both computation and communication. First, we analyze the computation delay, which primarily stems from local training, while we assume that the time for global aggregation is negligible. Reducing neurons or connections decreases the model size by eliminating operations associated with pruned nodes, such as multiplication, addition, and activation, thereby significantly lowering the needed number of floating-point operations (FLOPs). Moreover, the reduction in computational load is assumed to scale proportionally with the number of FLOPs [31]. Denote $f_n^{(s)}$ as the processor clock

frequency of client n in round s . Then, the computation delay at client n is

$$\tau_n^{(s)}(\lambda_n^{(s)}, f_n^{(s)}) = \frac{(1 - \lambda_n^{(s)}) Z_n^{(s)} e_n}{f_n^{(s)} q_n}, \forall n, s, \quad (10)$$

where e_n denotes the number of FLOPs for computing the complete gradient of one data sample for client n , $(1 - \lambda_n^{(s)}) e_n$ corresponds to that for computing the pruned gradient, and q_n denotes the number of FLOPs per clock cycle of the processor.

Next, we analyze the communication delay. Denote $H_n^{(s)}$ as the data size (in bits) of the unpruned model gradients. Accordingly, we have $(1 - \lambda_n^{(s)}) H_n^{(s)}$ as the data size to be transmitted after pruning. Then, the communication delay of client n in round s is

$$\hat{\tau}_n^{(s)}(\lambda_n^{(s)}, p_n^{(s)}) = \frac{(1 - \lambda_n^{(s)}) H_n^{(s)}}{r_n^{(s)}(p_n^{(s)})} + \frac{H_n^{(s)}}{\hat{r}_n^{(s)}}. \quad (11)$$

Finally, the overall delay refers to the cumulative maximum latency among the selected clients over the S rounds, i.e.,

$$\begin{aligned} & T(\{a_n^{(s)}, \lambda_n^{(s)}, p_n^{(s)}, f_n^{(s)}\}) \\ &= \sum_{s=0}^S \max_{n \in \mathcal{N}} (a_n^{(s)} (\tau_n^{(s)}(\lambda_n^{(s)}, f_n^{(s)}) + \hat{\tau}_n^{(s)}(\lambda_n^{(s)}, p_n^{(s)}))) \\ &= \sum_{s=0}^S \max_{n \in \mathcal{N}} \left(a_n^{(s)} \left(\frac{(1 - \lambda_n^{(s)}) Z_n^{(s)} e_n}{f_n^{(s)} q_n} + \frac{(1 - \lambda_n^{(s)}) H_n^{(s)}}{r_n^{(s)}(p_n^{(s)})} \right. \right. \\ & \quad \left. \left. + \frac{H_n^{(s)}}{\hat{r}_n^{(s)}} \right) \right). \end{aligned} \quad (12)$$

2) *Energy Consumption*: The energy consumption primarily arises from computation and communication. First, we analyze the energy consumption of local training, which is proportional to the number of FLOPs through model pruning. Accordingly, an approximately proportional relationship to the retained model size is applied [35]. Then, the computation energy consumption of client n in round s is

$$\tilde{E}_n^{(s)}(\lambda_n^{(s)}, f_n^{(s)}) = (1 - \lambda_n^{(s)}) \kappa_n \varpi_n (f_n^{(s)})^2 \frac{Z_n^{(s)} e_n}{q_n}, \quad (13)$$

where κ_n denotes the power usage effectiveness (PUE) of client n , and ϖ_n represents the effective switched capacitance coefficient determined by the processor characteristics.

Then, the energy consumption for gradients uploading at client n in round s is

$$\hat{E}_n^{(s)}(\lambda_n^{(s)}, p_n^{(s)}) = \frac{(1 - \lambda_n^{(s)}) p_n^{(s)} H_n^{(s)}}{r_n^{(s)}(p_n^{(s)})}. \quad (14)$$

Thus, the overall energy consumption in round s consists of local computation, local gradients uploading, and global information broadcasting for all clients, i.e.,

$$\begin{aligned} & E(\{a_n^{(s)}, \lambda_n^{(s)}, p_n^{(s)}, f_n^{(s)}\}) \\ &= \sum_{s=0}^S \left(\sum_{n=1}^N a_n^{(s)} (\tilde{E}_n^{(s)}(\lambda_n^{(s)}, f_n^{(s)}) + \hat{E}_n^{(s)}(\lambda_n^{(s)}, p_n^{(s)})) \right. \\ & \quad \left. + \hat{p} \max_{n \in \mathcal{N}} \left\{ \frac{H_n^{(s)}}{\hat{r}_n^{(s)}} \right\} \right) \end{aligned}$$

$$= \sum_{s=0}^S \left(\sum_{n=1}^N a_n^{(s)} ((1 - \lambda_n^{(s)}) \kappa_n \varpi_n(f_n^{(s)})^2 \frac{Z_n^{(s)} e_n}{q_n} + p_n^{(s)} \frac{(1 - \lambda_n^{(s)}) H_n^{(s)}}{r_n^{(s)}(p_n^{(s)})}) + \hat{p} \max_{n \in \mathcal{N}} \left\{ \frac{H_n^{(s)}}{\hat{r}_n^{(s)}} \right\} \right). \quad (15)$$

III. CONVERGENCE ANALYSIS

A. Assumptions for Convergence Analysis

For the convergence analysis, we first introduce standard assumptions on the loss function and gradient estimation, as widely adopted in prior studies (see, e.g., [31]).

Assumption 1 (Smoothness): The gradient $\nabla \mathcal{L}(\omega, \mathcal{D})$ is Lipschitz continuous w.r.t. the model parameters. Accordingly, for any ω and ω' , it holds that

$$\|\nabla \mathcal{L}(\omega, \mathcal{D}) - \nabla \mathcal{L}(\omega', \mathcal{D})\| \leq L \|\omega - \omega'\|, \quad (16)$$

where L denotes the Lipschitz constant. The Hessian matrix of (16) is presented as $\nabla^2 \mathcal{L}(\omega, \mathcal{D}) \preceq L \mathbf{I}$, where \mathbf{I} is an identity matrix. Also we have

$$\mathcal{L}(\omega, \mathcal{D}) - \mathcal{L}(\omega', \mathcal{D}) \leq \nabla \mathcal{L}(\omega', \mathcal{D})^\top (\omega - \omega') + \frac{L}{2} \|\omega - \omega'\|^2, \quad (17)$$

where \mathbf{T} represents the transpose operation.

From Assumption 1, the gradients of loss function variations w.r.t. model parameters are bounded, ensuring smooth changes rather than abrupt fluctuations.

Assumption 2 (Unbiased gradient): The global mini-batch stochastic gradient $G(\omega)$ is assumed to be an unbiased estimate of the full-batch gradient $\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})$, expressed as

$$\mathbb{E}\{G(\omega)\} = \nabla \mathcal{L}(\omega, \hat{\mathcal{D}}), \quad (18)$$

where $\mathbb{E}\{\cdot\}$ denotes the statistical expectation.

Assumption 3 (Bounded gradient and model): The second moments of the local mini-batch stochastic gradients $g(\omega)$ and model parameters are upper bounded by non-negative constants A^2 and B^2 , respectively, i.e.,

$$\mathbb{E}\{\|g(\omega)\|^2\} \leq A^2, \quad (19)$$

$$\mathbb{E}\{\|\omega\|^2\} \leq B^2. \quad (20)$$

Assumption 4 (Bounded pruning level): The expected squared difference between the model parameters before and after pruning is upper bounded by the pruning ratio $\lambda_n^{(s)}$ with the expected squared parameter norm [32], i.e.,

$$\mathbb{E}\left\{\|\omega_n^{(s)} - \tilde{\omega}_n^{(s)}\|^2\right\} \leq \lambda_n^{(s)} \mathbb{E}\left\{\|\omega_n^{(s)}\|^2\right\}. \quad (21)$$

B. Convergence Analysis Based on Generalization Gap

In the following, we analyze the convergence property of the proposed FEEL framework w.r.t. the generalization gap. Denote $\tilde{\mathcal{L}}(\omega^{(s)}, \tilde{\mathcal{D}})$ as the global loss function under global test dataset $\tilde{\mathcal{D}}$ in round s . First, we introduce the generalization performance. Specifically, the generalization gap is defined as a theoretical measure of the discrepancy between the training loss (empirical risk) and the test loss (population risk) [24], given by

$$\varphi^{(s)} \triangleq \mathcal{L}^{(s)}(\omega^{(s)}, \hat{\mathcal{D}}) - \tilde{\mathcal{L}}^{(s)}(\omega^{(s)}, \tilde{\mathcal{D}}). \quad (22)$$

Next, we use the generalization gap and gradient decomposition during the FEEL training to characterize the upper bounds of model gradients on both the training and test sets. Specifically, given a data point z , we denote $p(z|\hat{\mathcal{D}})$ and $p(z|\tilde{\mathcal{D}})$ as the probability distributions over the training and test dataset, respectively, while $p'(z|\hat{\mathcal{D}})$ denotes the probability of the least frequent data point in the training set. Denote $H(p(z|\tilde{\mathcal{D}}))$ as the entropy of test distribution, and $I(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}}))$ as the mutual information between the training and test distributions. With these preliminaries, we establish the following lemma to characterize the gradient discrepancy between the training and test sets.

Lemma 1: For any model ω , the norm of the difference between the gradients evaluated over the training set $\hat{\mathcal{D}}$ and test set $\tilde{\mathcal{D}}$ is upper bounded by

$$\|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \leq \phi \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\|, \quad (23)$$

where ϕ is defined as the generalization statement, derived as

$$\phi = \left[\frac{(\hat{D} + \tilde{D})}{p'(z|\hat{\mathcal{D}})} \cdot \left| \frac{\sqrt{2(H(p(z|\tilde{\mathcal{D}})) - I(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}})))}}{1 - \tilde{D} \sqrt{2(H(p(z|\tilde{\mathcal{D}})) - I(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}})))}} \right| \right].$$

Proof: See Appendix A.

Next, building on the previous analysis, we introduce the following proposition that captures the generalization gap across each FEEL round.

Proposition 1: The generalization gap between training rounds s and $(s+1)$ is theoretically derived as

$$\varphi^{(s+1)} - \varphi^{(s)} \leq \frac{1}{2} \left(\eta^2 + \left| \sum_{n=1}^N a_n^{(s)} \phi_n \right|^2 \right) \mathbb{E}\left\{\|G(\tilde{\omega}^{(s)})\|^2\right\}, \quad (24)$$

where ϕ_n is the generalization statement at client n , derived as

$$\phi_n = \left[\frac{(\hat{D}_n + \tilde{D}_n)}{p'(z|\hat{\mathcal{D}}_n)} \cdot \left| \frac{\sqrt{2(H(p(z|\tilde{\mathcal{D}}_n)) - I(p(z|\hat{\mathcal{D}}_n), p(z|\tilde{\mathcal{D}}_n)))}}{1 - \tilde{D}_n \sqrt{2(H(p(z|\tilde{\mathcal{D}}_n)) - I(p(z|\hat{\mathcal{D}}_n), p(z|\tilde{\mathcal{D}}_n)))}} \right| \right].$$

Proof: See Appendix B.

Remark 1: Proposition 1 characterizes an upper bound on the variation of the generalization gap across consecutive training rounds, where the bound explicitly depends on the generalization statement ϕ_n . We prefer reducing the generalization gap $\varphi^{(s+1)}$ in round $(s+1)$ compared with round s by considering ϕ_n , thereby achieving improved generalization. In particular, a smaller ϕ_n indicates that the local training distribution of a client is more consistent with the sampled testing distribution, thereby reducing the discrepancy between training and test losses. Consequently, prioritizing clients with smaller values of ϕ_n in the selection process can enhance the alignment between local updates and the global objective, thus improving model generalization performance. This observation highlights the importance of client selection for improving the generalization in FEEL.

Finally, building on Proposition 1, we analyze the gradient convergence with generalization gap dynamics and client heterogeneity, as presented in Theorem 1.

Theorem 1: With the FEEL model initialized at $\omega^{(0)}$, satisfying Assumptions 1~4, with fixed learning rate η and

batch size $Z = Z_n^{(s)}$, $\forall n, s$, the average squared gradient norm after S training rounds has the following upper bound:

$$\begin{aligned} \frac{1}{S+1} \sum_{s=0}^S \mathbb{E} \left\{ \|\nabla \tilde{\mathcal{L}}(\tilde{\omega}^{(s)}, \tilde{\mathcal{D}})\|^2 \right\} &\leq \theta(\{a_n^{(s)}, \lambda_n^{(s)}\}) \\ &\triangleq \alpha + \beta \sum_{s=0}^S \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &+ \sum_{s=0}^S \frac{\gamma_1 |\sum_{n=1}^N a_n^{(s)} \phi_n|^2 + \gamma_2 \sum_{n=1}^N a_n^{(s)} \lambda_n^{(s)}}{\sum_{n=1}^N a_n^{(s)}}, \end{aligned} \quad (25)$$

where $\alpha = \frac{2(\mathcal{L}(\omega^{(0)}) - \mathcal{L}(\omega^*))}{\eta(S+1)}$, $\beta = \frac{\eta^3 A^2 (L+1)}{Z(S+1)}$, $\gamma_1 = \frac{\eta A^2}{Z(S+1)}$, and $\gamma_2 = \frac{L^2 B^2}{(S+1)}$.

Proof: See Appendix C.

Remark 2: The upper bound $\theta(\{a_n^{(s)}, \lambda_n^{(s)}\})$ in Theorem 1 indicates that both the client selection and the pruning ratio directly affect the convergence behavior. Increasing the number of participated clients enhances update reliability, while smaller pruning ratios contribute to faster convergence. These jointly tighten the gradient bound and accelerate convergence, but heavier communication and computation from more clients and less-pruned models increase latency and energy consumption. Moreover, the system prioritizes the client with smaller ϕ_n that further accelerates convergence. Consequently, jointly optimizing $\{a_n^{(s)}\}$ and $\{\lambda_n^{(s)}\}$ alongside $\{p_n^{(s)}\}$ and $\{f_n^{(s)}\}$ is essential to balance learning (generalization) performance, energy consumption, and delay, thereby providing theoretical guidance for resource optimization.

IV. JOINT RESOURCE OPTIMIZATION

This section first formulates a joint optimization problem to enhance the convergence performance subject to system energy consumption and latency requirements. Then, we develop an efficient algorithm to solve the formulated problem.

A. Problem Formulation

We design the optimization problem to minimize the expected squared gradient norm bound $\theta(\{a_n^{(s)}, \lambda_n^{(s)}\})$ in (25) by jointly optimizing the pruning ratios $\{\lambda_n^{(s)}\}$, client selection indicators $\{a_n^{(s)}\}$, computation frequencies $\{f_n^{(s)}\}$, and transmit powers $\{p_n^{(s)}\}$. Accordingly, the learning performance optimization problem is formulated as

$$(P1) : \min_{\{\lambda_n^{(s)}, a_n^{(s)}, p_n^{(s)}, f_n^{(s)}\}} \theta(\{a_n^{(s)}, \lambda_n^{(s)}\}) \quad (26a)$$

$$\text{s.t. } E(\{a_n^{(s)}, \lambda_n^{(s)}, p_n^{(s)}, f_n^{(s)}\}) \leq E_0, \quad (26b)$$

$$T(\{a_n^{(s)}, \lambda_n^{(s)}, p_n^{(s)}, f_n^{(s)}\}) \leq T_0, \quad (26c)$$

$$0 \leq \lambda_n^{(s)} \leq \lambda^{\max}, \forall n \in \mathcal{N}, \quad (26d)$$

$$0 \leq f_n^{(s)} \leq f_n^{\max}, \forall n \in \mathcal{N}, \quad (26e)$$

$$0 \leq p_n^{(s)} \leq p_n^{\max}, \forall n \in \mathcal{N}, \quad (26f)$$

$$a_n^{(s)} \in \{0, 1\}, \forall n \in \mathcal{N}, \quad (26g)$$

where λ^{\max} , f_n^{\max} , and p_n^{\max} in (26c)~(26e) denote the maximum pruning ratio, clock frequency, and transmit power

during parameter-efficient FEEL, respectively, E_0 and T_0 denote the overall energy consumption and system delay requirements. Appropriate adjustment of E_0 and T_0 ensures a trade-off between convergence performance, energy consumption, and system latency over the learning process. However, in problem (P1), the objective and constraint functions in (26a) and (26b) are non-convex, due to the tight coupling between $\{a_n^{(s)}\}$ and $\{\lambda_n^{(s)}\}$, as well as $\{p_n^{(s)}\}$ appearing in both the numerator and the denominator of (26a). Moreover, due to the binary variable $\{a_n^{(s)}\}$, problem (P1) is a mixed-integer nonlinear program (MINLP), which is highly non-convex and hard to be optimally solved.

B. Proposed Solution to Problem (P1)

We propose an efficient alternating optimization (AO) framework to tackle problem (P1). The framework decouples pruning ratios, selection indicators, and resource variables across the objective and constraints in an iterative manner.

1) Optimization of System Resources: First, with fixed $\{a_n^{(s)}\}$ and $\{\lambda_n^{(s)}\}$, we jointly optimize the communication resource allocation and computation frequency in the following problem (P2).

$$\begin{aligned} (P2) : \min_{\{p_n^{(s)}, f_n^{(s)}\}} \quad &\theta(\{a_n^{(s)}, \lambda_n^{(s)}\}) \\ \text{s.t.} \quad &\sum_{s=0}^S \left(\sum_{n=1}^N a_n^{(s)} ((1 - \lambda_n^{(s)}) \kappa_n \varpi_n(f_n^{(s)})^2 \frac{Z_n^{(s)} e_n}{q_n} \right. \\ &\left. + \frac{(1 - \lambda_n^{(s)}) p_n^{(s)} H_n^{(s)}}{c_n \log_2(1 + \frac{p_n^{(s)} h_n^{(s)}}{c_n U_0^{(s)}})} \right) + \hat{p} \max_{n \in \mathcal{N}} \left(\frac{H_n^{(s)}}{\hat{r}_n^{(s)}} \right) \leq E_0, \end{aligned} \quad (27a)$$

$$\begin{aligned} \sum_{s=0}^S \max_{n \in \mathcal{N}} \left(a_n^{(s)} \left(\frac{(1 - \lambda_n^{(s)}) Z_n^{(s)} e_n}{f_n^{(s)} q_n} \right. \right. \\ \left. \left. + \frac{(1 - \lambda_n^{(s)}) H_n^{(s)}}{c_n \log_2(1 + \frac{p_n^{(s)} h_n^{(s)}}{c_n U_0^{(s)}})} + \frac{H_n^{(s)}}{\hat{r}_n^{(s)}} \right) \right) \leq T_0, \end{aligned} \quad (27b)$$

$$(26d), (26e).$$

Problem (P2) is still non-convex and hard to be optimally solved directly. We apply SCA to approximate the non-convex constraint into a convex counterpart. Specifically, we take the first-order Taylor expansion on the middle term of non-convex constraint (27a) at point $p_n^{(s)(k)}$ in iteration $k \geq 1$, i.e.,

$$\begin{aligned} \frac{p_n^{(s)} H_n^{(s)}}{c_n \log_2(1 + \frac{p_n^{(s)} h_n^{(s)}}{c_n U_0^{(s)}})} &\leq \frac{p_n^{(s)(k)} H_n^{(s)}}{c_n \log_2(1 + \frac{p_n^{(s)(k)} h_n^{(s)}}{c_n U_0^{(s)}})} \\ &+ b_n^{(s)(k)} (p_n^{(s)} - p_n^{(s)(k)}) \triangleq \xi^{(k)}(p_n^{(s)}), \end{aligned} \quad (28)$$

where $b_n^{(s)(k)}$ is the partial derivative over $p_n^{(s)(k)}$, i.e.,

$$\left(\frac{\partial}{\partial p_n^{(s)(k)}} \frac{p_n^{(s)(k)} H_n^{(s)}}{c_n \log_2(1 + \frac{p_n^{(s)(k)} h_n^{(s)}}{c_n U_0^{(s)}})} \right) = \frac{H_n^{(s)}}{c_n \log_2(1 + \frac{p_n^{(s)(k)} h_n^{(s)}}{c_n U_0^{(s)}})}$$

TABLE I
EXPERIMENT SETUPS

Parameter	Model on MNIST	Model on CIFAR-10
Model/gradient size H	1.42 Mbit	21.07 Mbit
Computation workload e	1.8 MFLOPs	0.59 GFLOPs
Bandwidth of devices c_n^U	100 kHz	2 MHz
PSD of AWGN $U_0^{(s)}$	3.98×10^{-21} W/Hz	3.98×10^{-21} W/Hz
Maximum pruning ratio λ_{\max}	0.5	0.7
Maximum clock frequency of clients f_n^{\max}	500 MHz	2000 MHz
Maximum transmit power of clients p_n^{\max}	500 mW	500 mW
FLOPs performed per clock cycle q_n	4	8
The PUE of clients κ_n	1	1
Power coefficient of clients $\{\varpi_n\}$	$\{0.88, 0.84, 1.41, 1.33, 0.94, 1.37, 1.8, 1.01, 0.26, 0.96\} \times 10^{-27}$	$\{0.88, 0.84, 1.41, 1.33, 0.94, 1.37, 1.8, 1.01, 0.26, 0.96\} \times 10^{-28}$

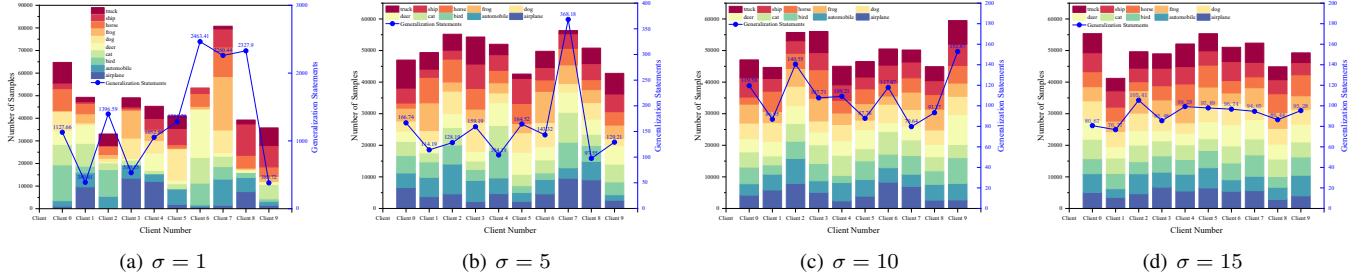


Figure 3. Impact of data heterogeneity on sample distributions and generalization statements with different values of σ .

$$-\frac{p_n^{(s)(k)} H_n^{(s)} h_n^{(s)}}{c_n \left(\log_2 \left(1 + \frac{p_n^{(s)(k)} h_n^{(s)}}{c_n U_0^{(s)}} \right) \right)^2 (c_n U_0^{(s)} + p_n^{(s)(k)} h_n^{(s)}) \ln 2}.$$

By substituting (28) into (27a), we obtain the following approximate convex version of problem (P2) in the k -th iteration as problem (P2.1):

$$(P2.1) : \begin{aligned} & \min_{\{p_n^{(s)}, f_n^{(s)}\}} \theta(\{a_n^{(s)}, \lambda_n^{(s)}\}) \\ \text{s.t. } & \sum_{s=0}^S \left(\sum_{n=1}^N a_n^{(s)} (1 - \lambda_n^{(s)}) (\kappa_n \varpi_n(f_n^{(s)})^2 \frac{Z_n^{(s)} e_n}{q_n} \right. \\ & \left. + \xi^{(k)}(p_n^{(s)})) + \hat{p} \max_{n \in \mathcal{N}} \left(\frac{H_n^{(s)}}{\hat{r}_n^{(s)}} \right) \right) \leq E_0, \quad (29) \end{aligned}$$

(27b), (26d), (26e).

2) *Optimization of Pruning Ratio:* Then, under fixed $\{a_n^{(s)}\}$, $\{p_n^{(s)}\}$, and $\{f_n^{(s)}\}$, the pruning ratio optimization problem is formulated as

$$(P3) : \begin{aligned} & \min_{\{\lambda_n^{(s)}\}} \theta(\{a_n^{(s)}, \lambda_n^{(s)}\}) \\ \text{s.t. } & (26a) \sim (26c). \end{aligned}$$

Problem (P3) is a linear programming (LP) problem which is solved by standard convex optimization tools, such as CVX [36].

3) *Optimization of Client Selection:* Finally, given fixed computation frequency $\{f_n^{(s)}\}$, transmit power $\{p_n^{(s)}\}$, and

pruning ratio $\{\lambda_n^{(s)}\}$, we optimize the client selection indicator $\{a_n^{(s)}\}$. And thus the problem is reformulated as

$$(P4) : \begin{aligned} & \min_{\{a_n^{(s)}\}} \sum_{s=0}^S \frac{\gamma_1 |\sum_{n=1}^N a_n^{(s)} \phi_n|^2 + \gamma_2 \sum_{n=1}^N a_n^{(s)} \lambda_n^{(s)}}{\sum_{n=1}^N a_n^{(s)}} \\ & + \alpha + \beta \sum_{s=0}^S \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ \text{s.t. } & (26a), (26b), (26f). \end{aligned}$$

Although constraints (26a), (26b), and (26f) are convex w.r.t. $\{a_n^{(s)}\}$, the objective function involves coupled fractional and quadratic terms, making Problem (P3) still non-convex. To deal with the non-convexity, we define an auxiliary variable $\mu^{(s)}$, which satisfies $[\gamma_1 |\sum_{n=1}^N a_n^{(s)} \phi_n|^2 + \gamma_2 \sum_{n=1}^N a_n^{(s)} \lambda_n^{(s)}] \leq \mu^{(s)}$. Then, problem (P3) is reformulated as

$$(P5) : \begin{aligned} & \min_{\{a_n^{(s)}, \mu^{(s)}\}} \alpha + \beta \sum_{s=0}^S \frac{1}{\sum_{n=1}^N a_n^{(s)}} + \sum_{s=0}^S \frac{\mu^{(s)}}{\sum_{n=1}^N a_n^{(s)}} \\ \text{s.t. } & \gamma_1 \left| \sum_{n=1}^N a_n^{(s)} \phi_n \right|^2 + \gamma_2 \sum_{n=1}^N a_n^{(s)} \lambda_n^{(s)} \leq \mu^{(s)}, \\ & (26a), (26b), (26f). \end{aligned} \quad (32a)$$

Problem (P5) is still non-convex due to the coupling between $a_n^{(s)}$ and $\mu^{(s)}$ in the objective function. This subproblem is solved by iteratively optimizing $\{a_n^{(s)}\}$ and $\{\mu^{(s)}\}$ until convergence, where one variable is optimized while the other is fixed in each iteration.

Algorithm 1 Proposed Algorithm for Solving Problem (P1)

```

1: Initialize the auxiliary variable  $\{\mu^{(s),0}\}$  and optimization
variables  $\{a_n^{(s),0}, \lambda_n^{(s),0}, p_n^{(s),0}, f_n^{(s),0}\}$ .
2: for  $o = 1 : O$  do
3:   Initialize  $\{p_n^{(s),(o-1)}, f_n^{(s),(o-1)}\} =$   

 $\{p_n^{(s),(o-1)}, f_n^{(s),(o-1)}\}$ ; Set  $k = 1$ .
4:   repeat
5:     Solve the problem (P2.1) under the local points  

 $\{p_n^{(s),(o-1)}, f_n^{(s),(o-1)}, a_n^{(s),(o-1)}, \lambda_n^{(s),(o-1)}\}$  to  

obtain the solution  $\{p_n^{(s),(o)}, f_n^{(s),(o)}\}$ .
6:   Set  $k = k + 1$ .
7:   until the decrease of the objective value is below a  

predefined threshold
8:   Update  $\{p_n^{(s),(o)}, f_n^{(s),(o)}\} = \{p_n^{(s),(k)}, f_n^{(s),(k)}\}$ .
9:   Solve problem (P3) to obtain  $\{\lambda_n^{(s),(o)}\}$  under given  

 $\{a_n^{(s),(o-1)}, p_n^{(s),(o)}, f_n^{(s),(o)}\}$ .
10:  Solve problem (P5) to obtain  $\{a_n^{(s),(o)}\}$  under given  

 $\{\lambda_n^{(s),(o)}, p_n^{(s),(o)}, f_n^{(s),(o)}\}$ .
11:  Obtain the final solution leading to non-increasing  

objective value.
12: end for
13: Output final solution  $\{a_n^{(s)*}, p_n^{(s)*}, f_n^{(s)*}, \lambda_n^{(s)*}\}$  with the  

minimum objective value of problem (P1).

```

4) *Overall algorithm:* By alternately solving the subproblems (P2) ~ (P5), we can obtain a suboptimal solution to problem (P1); via iteratively optimizing the pruning ratios, client selection, as well as the communication and computation resources, we thus obtain a series of solutions $\{a_n^{(s)*}, \lambda_n^{(s)*}, p_n^{(s)*}, f_n^{(s)*}\}$, which lead to non-increasing objective values. The proposed algorithm for solving problem (P1) is summarized in Algorithm 1. Since the objective function of problem (P1) monotonically decreases with each iteration and is lower bounded, the convergence of Algorithm 1 is guaranteed.

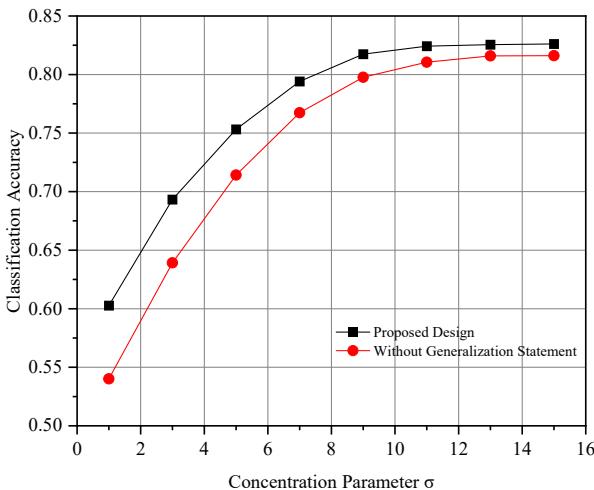


Figure 4. Classification accuracy of ResNet-110 versus Dirichlet parameter σ with/without generalization statement under $T_0=3600$, and $E_0=7100$.

V. NUMERICAL RESULTS

This section presents numerical results to evaluate the performance of our proposed system. We set the number of clients $N = 10$, learning rate $\eta = 0.01$, server transmit power $\hat{p} = 0.5$ W, and maximum client transmit power $p_n^{\max} = 0.5$ W, $\forall n \in \mathcal{N}$. Channel coefficients are modeled as IID Rayleigh fading with an average path loss of 10^{-5} , and remain constant during all rounds. Parameter settings are summarized in Table I. To simulate statistical heterogeneity in distributed learning, we employ the Dirichlet-based partitioning strategy, which splits non-IID data by sampling label proportions for clients from a Dirichlet distribution $p_{n,z} \sim \text{Dirichlet}(\sigma)$, where the concentration parameter σ controls data heterogeneity.

We evaluate LeNet [37] on the MNIST [38] dataset, followed by experiments with ResNet-110 [39] on CIFAR-10 [40]. MNIST contains 60,000 training and 10,000 testing grayscale images (28×28), while CIFAR-10 consists of 50,000 training and 10,000 testing RGB images ($32 \times 32 \times 3$) from 10 object categories. All experiments are implemented using PyTorch and conducted on NVIDIA RTX 3090 GPUs.

For comparison, we consider the following benchmark schemes.

- **Fixed pruning:** The model is trained without pruning, i.e., $\lambda_n^{(s)} = 0$, $\forall n, s$, to evaluate the effect of pruning ratio optimization.
- **Fixed selection:** All clients participate in each training round by setting $a_n^{(s)} = 1$, $\forall n, s$, serving as the baseline without client selection.
- **Without generalization statement:** This scheme follows the conventional convergence analysis in [31], without generalization statement based optimization.
- **Fixed power design:** Each client transmits with a constant power $p_n^{(s)} = 0.5$ W, $\forall n, s$, to examine the benefit of adaptive power control.
- **Fixed clock frequency design:** We set the computation frequency of each client as $f_n^{(s)} = f^{\max}$, $\forall n, s$, for comparison with adaptive computation design.

Figs. 3(a) and 3(d) illustrate the impact of client data heterogeneity on label distribution and generalization statement under different Dirichlet parameters σ . As σ increases, the label distributions among clients become more balanced, and the variation in generalization performance decreases. Fig. 4 shows that incorporating the generalization statement improves classification accuracy across different levels of data heterogeneity, consistently outperforming the baseline without it. To evaluate the generalization ability, we conduct experiments on $\sigma = 5$ to simulate data heterogeneity [15].

Figs. 5(a) and 5(b) demonstrate the convergence performance in terms of training loss w.r.t. the training time. It is observed that, within the permissible training delay threshold, the proposed system consistently attains the lowest loss values across all schemes. While certain baseline methods for model ResNet-110 show a faster initial decrease in loss, they finally converge to a higher loss compared to the proposed design. This demonstrates the effectiveness of the proposed system in dynamically adjusting client participation and compression

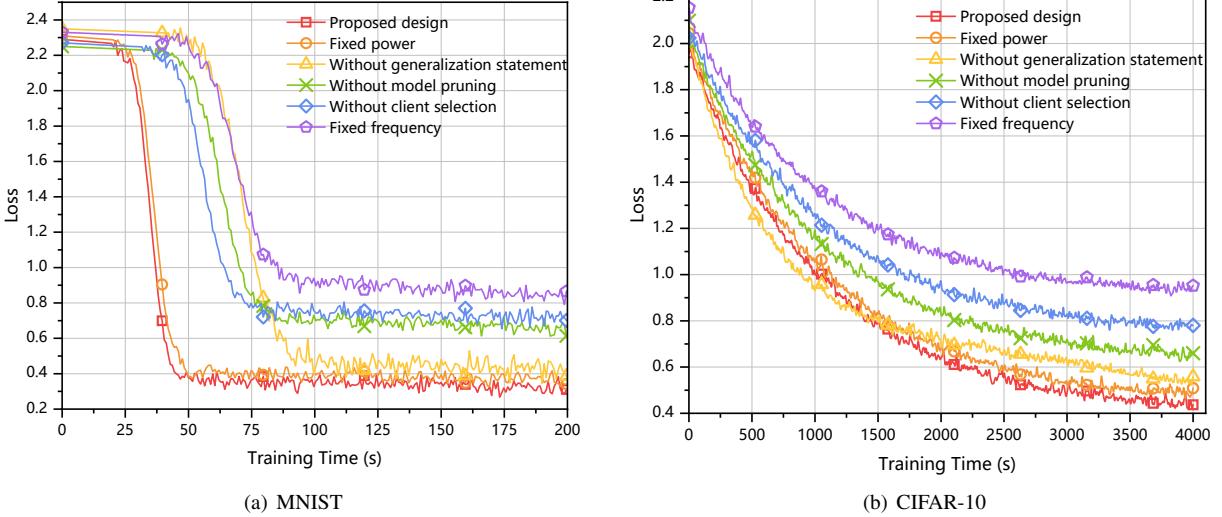


Figure 5. Convergence behavior in terms of the training loss over the overall training time. (a) LeNet on MNIST under $E_0 = 250$ J; (b) ResNet-110 on CIFAR-10 under $E_0 = 7100$ J.

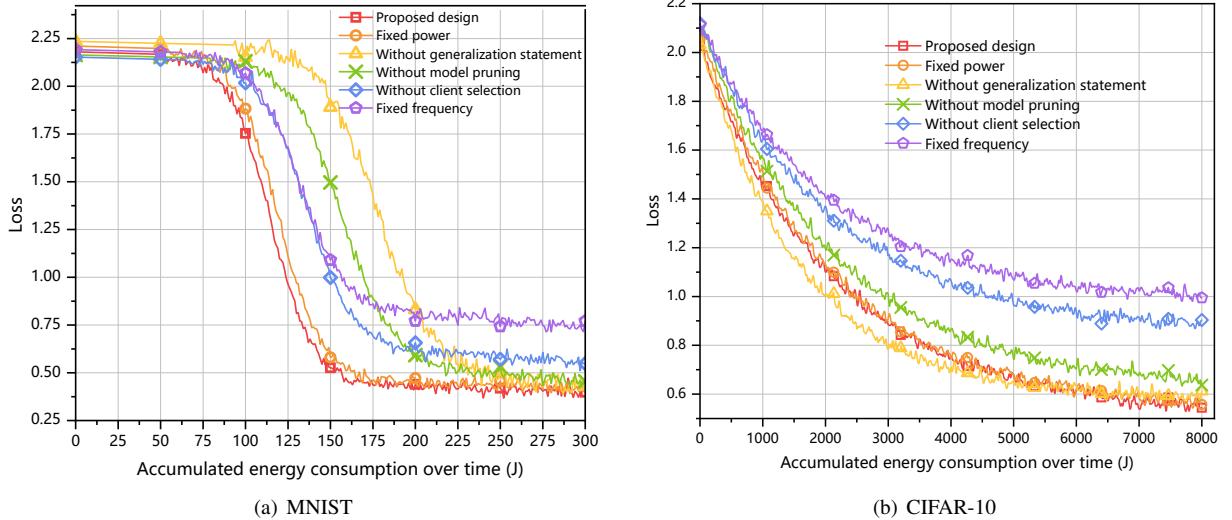


Figure 6. Convergence behavior in terms of the training loss w.r.t. the accumulated overall system energy consumption over time. (a) LeNet on MNIST under $T_0 = 150$ s; (b) ResNet-110 on CIFAR-10 under $T_0 = 3600$ s.

levels based on local data quality, particularly under non-IID conditions.

Figs. 6(a) and 6(b) demonstrate the convergence performance of training loss w.r.t the accumulated system energy consumption. It is observed that under a permissible energy constraint, the proposed framework achieves superior training performance, as evidenced by lower final loss values. Although some baselines exhibit faster loss reduction in the early training stages, their final performance remains suboptimal. This demonstrates that, by efficiently coordinating model selection, pruning, and resource allocation under constrained energy resources, the proposed design leads to more favorable final convergence behavior.

Figs. 7(a) and 7(b) show the classification accuracy under different system delay thresholds. It is observed that the proposed design achieves higher accuracy than baseline

schemes. As the energy consumption threshold increases, the performance improvement of all baselines gradually saturates, and latency emerges as the dominant system constraint. Notably, the proposed design significantly outperforms the design without generalization statement. This demonstrates that, in joint resource optimization, generalization statements enable effective selection of both clients and pruning levels, thereby yielding higher test accuracy and superior model generalization.

Figs. 8(a) and 8(b) show the classification accuracy under energy consumption thresholds. It is observed that under tight resource limitations, the proposed design exhibits a considerable accuracy advantage, demonstrating the effectiveness of joint resource optimization. Furthermore, compared with the system without generalization statements, the proposed design achieves higher accuracy, indicating the advantage of

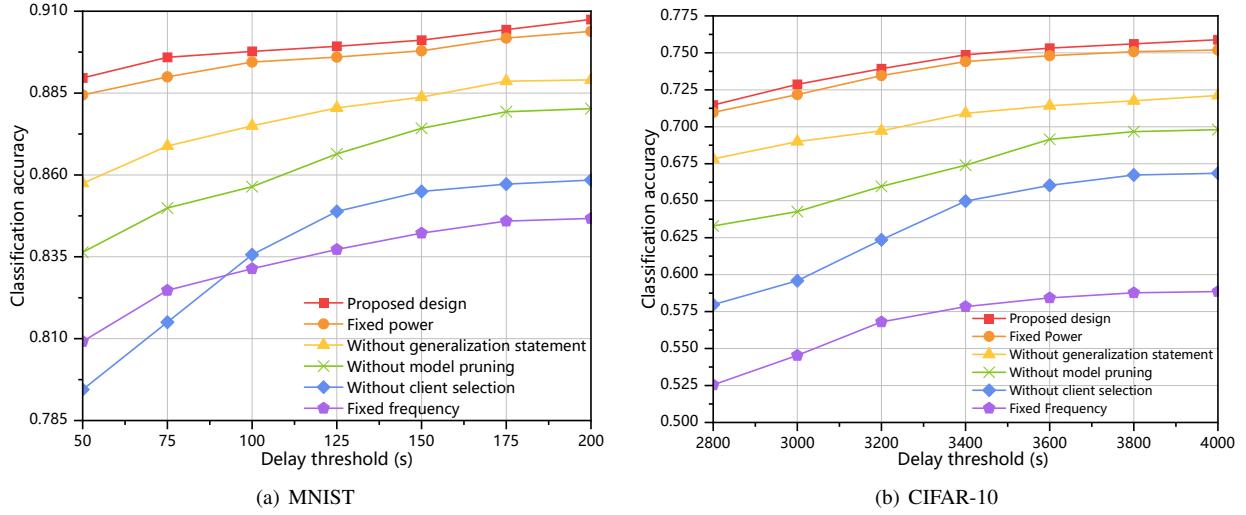


Figure 7. Classification accuracy w.r.t. different system delay thresholds (a) LeNet on MINST under $E_0 = 250$ J; (b) ResNet-110 on CIFAR-10 under $E_0 = 7100$ J.

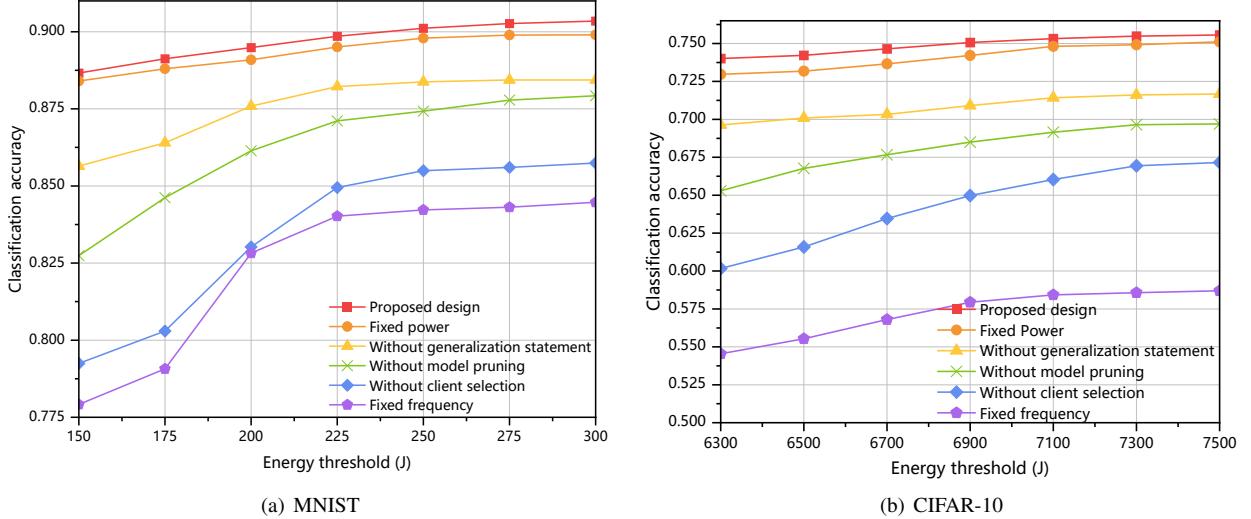


Figure 8. Classification accuracy w.r.t. different system energy thresholds (a) LeNet on MINST under $T_0 = 150$ s; (b) ResNet-110 on CIFAR-10 under $T_0 = 3600$ s.

incorporating generalization analysis into system optimization.

VI. CONCLUSIONS

This paper presented a parameter-efficient FEEL framework optimized through generalization analysis to improve both model generalization and resource utilization in resource-constrained edge deployment. We first established an information-theoretic generalization statement to quantify the divergence between local training and testing function losses, and analyzed the bound of average squared gradient norm. Then, we designed an efficient algorithm to solve the joint optimization problem of minimizing the above bound over client selection, model pruning, and resource allocation. Finally, numerical results demonstrated that under energy and delay constraints, the proposed design achieves better convergence and generalization performance. Future work will further explore adaptive local training data sampling strategies under dynamic environments to enhance robustness against temporal

data shifts. In addition, extending the proposed framework to large-scale model fine-tuning is another important direction toward realizing more general and scalable edge AI.

APPENDIX A PROOF OF LEMMA 1

To analyze generalization, we first derive the expression for the gradient discrepancy norm between training and testing sets by decomposing the loss gradient as

$$\begin{aligned}
 & \|\nabla \mathcal{L}(\boldsymbol{\omega}, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\boldsymbol{\omega}, \tilde{\mathcal{D}})\| \\
 &= \left\| \sum_{z \in \mathcal{D}} \nabla l(\boldsymbol{\omega}, z) [p(z|\hat{\mathcal{D}}) - p(z|\tilde{\mathcal{D}})] \right\| \\
 &= \left\| \sum_{z \in \hat{\mathcal{D}} \cup \tilde{\mathcal{D}}} \nabla l(\boldsymbol{\omega}, z) |p(z|\hat{\mathcal{D}}) - p(z|\tilde{\mathcal{D}})| \right\| \\
 &\leq \left\| \sum_{z \in \hat{\mathcal{D}} \cup \tilde{\mathcal{D}}} \nabla l(\boldsymbol{\omega}, z) \sum_{z \in \hat{\mathcal{D}} \cup \tilde{\mathcal{D}}} |p(z|\hat{\mathcal{D}}) - p(z|\tilde{\mathcal{D}})| \right\|, \quad (33)
 \end{aligned}$$

where $l(\omega, z)$ denotes the loss function of model ω on data sample z from the dataset. According to Pinsker's inequality, we have $|p(\hat{\mathcal{D}}) - p(\tilde{\mathcal{D}})| \leq \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}$, where $p(\hat{\mathcal{D}})$ and $p(\tilde{\mathcal{D}})$ denote the training and test data distributions, respectively, and $D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]$ denotes the Kullback–Leibler (KL) divergence of the above distributions. Accordingly, inequality (33) is established as

$$\begin{aligned} & \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \\ & \leq \left\| \sum_{z \in \hat{\mathcal{D}} \cup \tilde{\mathcal{D}}} \nabla l(\omega, z) \right\| \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} \\ & \leq \left\| \sum_{z \in \hat{\mathcal{D}}} \nabla l(\omega, z) \right\| \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} \\ & \quad + \left\| \sum_{z \in \tilde{\mathcal{D}}} \nabla l(\omega, z) \right\| \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} \\ & \leq \hat{D} \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\| \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} \\ & \quad + \tilde{D} \|\nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}. \end{aligned} \quad (34)$$

Then, we apply the reverse triangle inequality $\|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\| - \|\nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \leq \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\|$ to (34), which yields

$$\frac{1 - \tilde{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}}{1 + \hat{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}} \|\nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \leq \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\|. \quad (35)$$

Next, to analyze the above inequality, we consider two cases based on the sign of $(1 - \tilde{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]})$. Firstly, when $1 - \tilde{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} > 0$, we substitute (35) into (34) and obtain

$$\begin{aligned} & \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \\ & \leq (\hat{D} \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\| + \tilde{D} \|\nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\|) \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} \\ & \leq (\hat{D} \cdot \frac{1 + \hat{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}}{1 - \tilde{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}} \\ & \quad + \hat{D}) \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\| \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]} \\ & = \left| \frac{(\hat{D} + \tilde{D}_n) \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}}{1 - \tilde{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}} \right| \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\| \\ & = \phi' \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\|, \end{aligned} \quad (36)$$

where we define the preliminary generalization gap term as ϕ' . Next, when $(1 - \tilde{D} \sqrt{2D_{KL}[p(\hat{\mathcal{D}})\|p(\tilde{\mathcal{D}})]}) \leq 0$, it follows that $\phi' \geq 1$. Consequently, we have $|p(z|\hat{\mathcal{D}}) - p(z|\tilde{\mathcal{D}})| \leq 1 \leq \phi'$, which leads to

$$\begin{aligned} & \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \\ & = \left\| \sum_{z \in \hat{\mathcal{D}} \cup \tilde{\mathcal{D}}} \nabla l(\omega, z) [p(z|\hat{\mathcal{D}}) - p(z|\tilde{\mathcal{D}})] \right\| \end{aligned}$$

$$\begin{aligned} & \leq \left\| \sum_{z \in \hat{\mathcal{D}} \cup \tilde{\mathcal{D}}} \nabla l(\omega, z) \phi' p(z|\hat{\mathcal{D}}) \frac{1}{p(z|\hat{\mathcal{D}})} \right\| \\ & \leq \phi' \frac{1}{p'(z|\hat{\mathcal{D}})} \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\|, \end{aligned} \quad (37)$$

where $p'(z|\hat{\mathcal{D}})$ denotes the probability of the least frequent element in the distribution. KL divergence is decomposed as

$$\begin{aligned} D_{KL}[p(z|\hat{\mathcal{D}})\|p(z|\tilde{\mathcal{D}})] &= H(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}})) - H(p(z|\hat{\mathcal{D}})) \\ &= H(p(z|\tilde{\mathcal{D}})) - [H(p(z|\hat{\mathcal{D}})) \\ &\quad + H(p(z|\tilde{\mathcal{D}})) - H(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}}))] \\ &= H(p(z|\tilde{\mathcal{D}})) - I(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}})). \end{aligned} \quad (38)$$

Finally, the norm of the difference between the gradients computed on the training and test datasets is bounded as

$$\|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}}) - \nabla \tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})\| \leq \phi \|\nabla \mathcal{L}(\omega, \hat{\mathcal{D}})\|, \quad (39)$$

where the generalization statement is defined as

$$\phi = \frac{(\hat{D} + \tilde{D})}{p'(z|\hat{\mathcal{D}})} \cdot \left| \frac{\sqrt{2(H(p(z|\tilde{\mathcal{D}})) - I(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}})))}}{1 - \tilde{D} \sqrt{2(H(p(z|\tilde{\mathcal{D}})) - I(p(z|\hat{\mathcal{D}}), p(z|\tilde{\mathcal{D}})))}} \right|.$$

Here, (39) bounds the gradient gap between training and test sets via gradients. ϕ reflects distributional shift and scale variation. Smaller values indicate better generalization, while larger ones imply performance degradation on unseen data.

APPENDIX B PROOF OF PROPOSITION 1

For notational simplicity, we use $\mathcal{L}(\omega)$ to denote $\mathcal{L}(\omega, \hat{\mathcal{D}})$ and $\tilde{\mathcal{L}}(\omega)$ to denote $\tilde{\mathcal{L}}(\omega, \tilde{\mathcal{D}})$ in the following derivations. To start with, by leveraging gradient descent and Taylor expansion, we decompose the loss function as

$$\begin{aligned} \mathcal{L}(\omega^{(s+1)}) &= \mathcal{L}(\omega^{(s)} - \eta G(\tilde{\omega}^{(s)})) \\ &= \mathcal{L}(\omega^{(s)}) - \eta G(\tilde{\omega}^{(s)})^T \nabla \mathcal{L}(\omega^{(s)}) + \mathcal{O}(\eta G(\tilde{\omega}^{(s)})). \end{aligned} \quad (40)$$

The generalization gap between two iterations is

$$\begin{aligned} & \varphi^{(s+1)} - \varphi^{(s)} \\ &= \mathbb{E}[[\mathcal{L}(\omega^{(s+1)}) - \tilde{\mathcal{L}}(\omega^{(s+1)})] - [\mathcal{L}(\omega^{(s)}) - \tilde{\mathcal{L}}(\omega^{(s)})]] \\ &= \mathbb{E}[[\mathcal{L}(\omega^{(s+1)}) - \mathcal{L}(\omega^{(s)})] - [\tilde{\mathcal{L}}(\omega^{(s+1)}) - \tilde{\mathcal{L}}(\omega^{(s)})]] \\ &= \mathbb{E}[\eta G(\tilde{\omega}^{(s)})^T [\nabla \tilde{\mathcal{L}}(\omega^{(s)}) - \nabla \mathcal{L}(\omega^{(s)})]]. \end{aligned} \quad (41)$$

Then, using the Young's inequality $a^T b \leq \frac{1}{2}(\|a\|^2 + \|b\|^2)$, we bound (41) as

$$\begin{aligned} & \varphi^{(s+1)} - \varphi^{(s)} \\ &= \mathbb{E}[\eta G(\tilde{\omega}^{(s)})^T [\nabla \tilde{\mathcal{L}}(\omega^{(s)}) - \nabla \mathcal{L}(\omega^{(s)})]] \\ &\leq \mathbb{E}\left[\frac{1}{2}(\eta^2 \|G(\tilde{\omega}^{(s)})\|^2 + \|\nabla \tilde{\mathcal{L}}(\omega^{(s)}) - \nabla \mathcal{L}(\omega^{(s)})\|^2)\right]. \end{aligned} \quad (42)$$

Next, to further bound (42), we make the following analysis. In FEEL, each client is restricted to a limited and biased local subset that poorly represents the global distribution, leading to larger generalization statements. Consequently, the generalization statement derived from the global dataset is

considerably lower than that of the selected clients. Based on this observation and Lemma 1, we establish the bound

$$\begin{aligned}\|\nabla \mathcal{L}(\boldsymbol{\omega}^{(s)}) - \nabla \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)})\| &\leq \hat{\phi} \|\nabla \mathcal{L}(\boldsymbol{\omega}^{(s)})\| \\ &\leq \sum_{n=1}^N a_n \phi_n \|\nabla \mathcal{L}(\boldsymbol{\omega}^{(s)})\|,\end{aligned}\quad (43)$$

where $\hat{\phi}$ denotes the generalization statement of global dataset. Accordingly, we bound the generalization statement between rounds s and $(s+1)$ as

$$\varphi^{(s+1)} - \varphi^{(s)} \leq \frac{1}{2} (\eta^2 + |\sum_{n=1}^N a_n \phi_n|^2) \mathbb{E} \left\{ \|G(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \right\}. \quad (44)$$

APPENDIX C

PROOF OF THEOREM 1

To analyze the convergence property, we first decompose the loss function as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}^{(s+1)}) &= \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s+1)}) + \varphi^{(s+1)} \\ &\leq \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}) - \eta \langle \nabla \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}), \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}) \rangle \\ &\quad + \frac{\eta^2 L}{2} \mathbb{E} \left\{ \|G(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \right\} + \varphi^{(s+1)}.\end{aligned}\quad (45)$$

We suppose that the local model trained with a fixed batch size $Z_n^{(s)} = Z, \forall n, s$. Then, the aggregated global gradient $G(\tilde{\boldsymbol{\omega}}^{(s)})$ is decomposed as

$$\begin{aligned}\|G(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 &= \left\| \frac{1}{\tilde{N}^{(s)}} \sum_{n=1}^N a_n^{(s)} g(\tilde{\boldsymbol{\omega}}_n^{(s)}) \right\|^2 \\ &= \left\| \frac{1}{\tilde{N}^{(s)}} \sum_{n=1}^N \frac{a_n^{(s)}}{Z} \sum_{i=1}^Z g^{(i)}(\tilde{\boldsymbol{\omega}}_n^{(s)}) \right\|^2 \\ &= \left\| \frac{1}{Z \tilde{N}^{(s)}} \sum_{n=1}^N \sum_{i=1}^Z a_n^{(s)} g^{(i)}(\tilde{\boldsymbol{\omega}}_n^{(s)}) \right\|^2.\end{aligned}\quad (46)$$

Then, from Assumption 3, we have

$$\mathbb{E}[\|G(\tilde{\boldsymbol{\omega}}^{(s)})\|^2] \leq \frac{A^2}{Z \tilde{N}^{(s)}} = \frac{A^2}{Z^{(s)}}, \quad (47)$$

where $Z^{(s)} = \sum_{n=1}^N a_n^{(s)} Z$ is the total batch size in round s . Accordingly, we bound the loss function as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}^{(s+1)}) &\leq \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}) - \eta \langle \nabla \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}), \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}) \rangle + \varphi^{(s)} \\ &\quad + \frac{1}{2} \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n \phi_n|^2 \right) \mathbb{E} \left\{ \|G(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \right\} \\ &\leq \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}) - \eta \langle \nabla \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}), \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}) \rangle + \varphi^{(s)} \\ &\quad + \frac{A^2}{2Z^{(s)}} \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n \phi_n|^2 \right) \\ &\leq \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}) + \varphi^{(s)} - \eta \langle \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}), \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}) \rangle \\ &\quad + \frac{A^2}{2Z} \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n \phi_n|^2 \right) \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &\quad + \eta \langle \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}) - \nabla \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}), \nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)}) \rangle.\end{aligned}\quad (48)$$

By applying Assumption 1 and Young's inequality to (48), we further obtain

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}^{(s+1)}) &\leq \frac{A^2}{2Z} \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n \phi_n|^2 \right) \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &\quad + \frac{\eta L^2}{2\tilde{N}^{(s)}} \sum_{n \in \tilde{N}^{(s)}} \|\boldsymbol{\omega}_n^{(s)} - \tilde{\boldsymbol{\omega}}_n^{(s)}\|^2 - \frac{\eta}{2} \|\nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \\ &\quad + \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}) + \varphi^{(s)}.\end{aligned}\quad (49)$$

Based on $\mathcal{L}(\boldsymbol{\omega}^{(s)}) = \tilde{\mathcal{L}}(\boldsymbol{\omega}^{(s)}) + \varphi^{(s)}$, we update (49) as

$$\begin{aligned}\mathbb{E} \left\{ \|\nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \right\} &\leq -\frac{2}{\eta} (\mathbb{E} \left\{ \mathcal{L}(\boldsymbol{\omega}^{(s+1)}) \right\} - \mathbb{E} \left\{ \mathcal{L}(\boldsymbol{\omega}^{(s)}) \right\}) \\ &\quad + \frac{\eta A^2}{Z} \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n \phi_n|^2 \right) \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &\quad + \frac{L^2}{\tilde{N}^{(s)}} \sum_{n \in \tilde{N}^{(s)}} \|\boldsymbol{\omega}_n^{(s)} - \tilde{\boldsymbol{\omega}}_n^{(s)}\|^2.\end{aligned}\quad (50)$$

Then, we accumulate the inequality (50) over $s = 0$ to $s = S$ as

$$\begin{aligned}\frac{1}{S+1} \sum_{s=0}^S \mathbb{E} \left\{ \|\nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \right\} &\leq -\frac{2}{\eta(S+1)} \sum_{s=0}^S (\mathbb{E} \left\{ \mathcal{L}(\boldsymbol{\omega}^{(s+1)}) \right\} - \mathbb{E} \left\{ \mathcal{L}(\boldsymbol{\omega}^{(s)}) \right\}) \\ &\quad + \frac{\eta A^2}{Z(S+1)} \sum_{s=0}^S \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n^{(s)} \phi_n|^2 \right) \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &\quad + \frac{L^2}{(S+1)} \sum_{s=0}^S \frac{\sum_{n=1}^N a_n^{(s)} \mathbb{E} \left\{ \|\boldsymbol{\omega}_n^{(s)} - \tilde{\boldsymbol{\omega}}_n^{(s)}\|^2 \right\}}{\sum_{n=1}^N a_n^{(s)}},\end{aligned}\quad (51)$$

where the third term captures the effect of model pruning and the second term shows the effect of generalization gap analysis. From Assumption 4, the pruning effect term and the inequality is transferred as

$$\begin{aligned}\frac{1}{S+1} \sum_{s=0}^S \mathbb{E} \left\{ \|\nabla \tilde{\mathcal{L}}(\tilde{\boldsymbol{\omega}}^{(s)})\|^2 \right\} &\leq \frac{2(\mathcal{L}(\boldsymbol{\omega}^{(0)}) - \mathcal{L}(\boldsymbol{\omega}^*))}{\eta(S+1)} + \frac{L^2 B^2}{(S+1)} \sum_{s=0}^S \frac{\sum_{n=1}^N a_n^{(s)} \lambda_n^{(s)}}{\sum_{n=1}^N a_n^{(s)}} \\ &\quad + \frac{\eta A^2}{Z(S+1)} \sum_{s=0}^S \left(\eta^2 L + \eta^2 + |\sum_{n=1}^N a_n^{(s)} \phi_n|^2 \right) \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &= \frac{2(\mathcal{L}(\boldsymbol{\omega}^{(0)}) - \mathcal{L}(\boldsymbol{\omega}^*))}{\eta(S+1)} + \frac{\eta^3 A^2 (L+1)}{Z(S+1)} \sum_{s=0}^S \frac{1}{\sum_{n=1}^N a_n^{(s)}} \\ &\quad + \sum_{s=0}^S \frac{\gamma_1 |\sum_{n=1}^N a_n^{(s)} \phi_n|^2 + \gamma_2 \sum_{n=1}^N a_n^{(s)} \lambda_n^{(s)}}{\sum_{n=1}^N a_n^{(s)}},\end{aligned}\quad (52)$$

where $\boldsymbol{\omega}^*$ is the optimal model, $\gamma_1 = \frac{\eta A^2}{Z(S+1)}$, and $\gamma_2 = \frac{L^2 B^2}{(S+1)}$. Finally, we obtain the average squared gradient norm, which is presented as the convergence rate of the FEEL model with model pruning and generalization analysis.

REFERENCES

- [1] G. Zhu, Z. Lyu, X. Jiao, P. Liu, M. Chen, J. Xu, S. Cui, and P. Zhang, “Pushing AI to wireless network edge: An overview on integrated sensing, communication, and computation towards 6G,” *Sci. China Inf. Sci.*, vol. 66, no. 3, p. 130301, 2023.
- [2] L. Yuan, Z. Wang, L. Sun, P. S. Yu, and C. G. Brinton, “Decentralized federated learning: A survey and perspective,” *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34617–34638, Nov. 2024.
- [3] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang, “Federated learning for generalization, robustness, fairness: A survey and benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9387–9406, Dec. 2024.
- [4] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, “Federated learning with Non-IID data: A survey,” *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19188–19209, 2024.
- [5] H. Zhang, M. Tao, Y. Shi, X. Bi, and K. B. Letaief, “Federated multi-task learning with non-stationary and heterogeneous data in wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 2653–2667, Apr. 2024.
- [6] L. Kong, T. Lin, A. Koloskova, M. Jaggi, and S. Stich, “Consensus control for decentralized deep learning,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5686–5696.
- [7] Z. Wang, H. Xu, J. Liu, Y. Xu, H. Huang, and Y. Zhao, “Accelerating federated learning with cluster construction and hierarchical aggregation,” *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3805–3822, Jul. 2023.
- [8] S. A. Khowaja, K. Dev, P. Khowaja, and P. Bellavista, “Toward energy-efficient distributed federated learning for 6G networks,” *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 34–40, Dec. 2021.
- [9] Z. Cai, X. Cao, X. Chen, Y. Cui, G. Zhu, K. Huang, and S. Cui, “Ai-in-the-loop sensing and communication joint design for edge intelligence,” *arXiv:2502.10203*, 2025.
- [10] X. Cao, Z. Lyu, G. Zhu, J. Xu, L. Xu, and S. Cui, “An overview on over-the-air federated edge learning,” *IEEE Wireless Communications*, vol. 31, no. 3, pp. 202–210, 2024.
- [11] M. F. Pervej and A. F. Molisch, “Resource-aware hierarchical federated learning in wireless video caching networks,” *IEEE Trans. Wireless Commun.*, vol. 24, no. 1, pp. 165–180, Jan. 2025.
- [12] G. Zhu, Y. Du, D. Gündüz, and K. Huang, “One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.
- [13] Y. Park, D. J. Han, D. Y. Kim, J. Seo, and J. Moon, “Few-round learning for federated learning,” *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, pp. 28612–28622, 2021.
- [14] Y. Ruan, X. Zhang, S. C. Liang, and C. Joe-Wong, “Towards flexible device participation in federated learning,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2021, pp. 3403–3411.
- [15] S. Guo, Y. Guo, H. Zhang, and J. Wang, “Mitigating update conflict in Non-IID federated learning via orthogonal class gradients,” *IEEE Trans. Mobile Comput.*, vol. 24, no. 4, pp. 2967–2978, Apr. 2025.
- [16] S. Zawad, X. Ma, J. Yi, C. Li, M. Zhang, L. Yang, F. Yan, and Y. He, “Fedcust: Offloading hyperparameter customization for federated learning,” *Perform. Eval.*, vol. 167, p. 102450, Mar. 2025.
- [17] X. Liu, H. Zhang, C. Ren, H. Li, C. Sun, and V. C. M. Leung, “Multi-task learning resource allocation in federated integrated sensing and communication networks,” *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 11612–11623, Sep. 2024.
- [18] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy efficient federated learning over wireless communication networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
- [19] X. Hu, R. Li, L. Wang, Y. Ning, and K. Ota, “A data sharing scheme based on federated learning in IoV,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11644–11656, Sep. 2023.
- [20] L. Meng, Z. Qi, L. Wu, X. Du, Z. Li, L. Cui, and X. Meng, “Improving global generalization and local personalization for federated learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 76–87, Jan. 2025.
- [21] Z. Li, Z. Lin, J. Shao, Y. Mao, and J. Zhang, “Fedcir: Client-invariant representation learning for federated Non-IID features,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 11, pp. 10509–10522, Nov. 2024.
- [22] Z. Lyu, Y. Li, G. Zhu, J. Xu, H. V. Poor, and S. Cui, “Rethinking resource management in edge learning: A joint pre-training and fine-tuning design paradigm,” *IEEE Trans. Wireless Commun.*, vol. 24, no. 2, pp. 1584–1601, Feb. 2025.
- [23] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, “No fear of heterogeneity: Classifier calibration for federated learning with Non-IID data,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 5972–5984.
- [24] L. P. Barnes, A. Dytso, and H. V. Poor, “Improved information theoretic generalization bounds for distributed and federated learning,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022, pp. 1465–1470.
- [25] M. G. Boroujeni, A. Krause, and G. F. Trecate, “Personalized federated learning of probabilistic models: A PAC-Bayesian approach,” *arXiv:2401.08351*, 2025.
- [26] W. Ni, H. Sun, H. Ao, and H. Tian, “Federated intelligence: When large AI models meet federated fine-tuning and collaborative reasoning at the network edge,” *IEEE Internet Things Mag.*, pp. 1–8, Sep. 2025.
- [27] X. Deng, J. Li, K. Wei, L. Shi, Z. Xiong, M. Ding, W. Chen, S. Jin, and H. V. Poor, “Towards communication-efficient federated learning via sparse and aligned adaptive optimization,” *IEEE Trans. Signal Process.*, pp. 1–16, Sep. 2025.
- [28] N. Lang, E. Sofer, T. Shaked, and N. Shlezinger, “Joint privacy enhancement and quantization in federated learning,” *IEEE Trans. Signal Process.*, vol. 71, pp. 295–310, Feb. 2023.
- [29] Z. Lyu, M. Xiao, J. Xu, M. Skoglund, and M. D. Renzo, “The larger the merrier? efficient large ai model inference in wireless edge networks,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.09214>
- [30] L. Yi, X. Shi, N. Wang, J. Zhang, G. Wang, and X. Liu, “FedPE: Adaptive model pruning-expanding for federated learning on mobile devices,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 11, pp. 10475–10493, Nov. 2024.
- [31] S. Liu, G. Yu, R. Yin, J. Yuan, L. Shen, and C. Liu, “Joint model pruning and device selection for communication-efficient federated edge learning,” *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 231–244, Jan. 2022.
- [32] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, “Importance estimation for neural network pruning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.
- [33] J. Pei, W. Li, and S. Mumtaz, “From routine to reflection: Pruning neural networks in communication-efficient federated learning,” *IEEE Trans. Artif. Intell.*, vol. 6, no. 11, pp. 2896–2905, Nov. 2025.
- [34] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.
- [35] F. Guan, X. Hou, X. Wang, J. Wang, J. Du, and Y. Ren, “Energy-efficient federated learning: Integrating model pruning, compressive sensing, and outage compensation,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2025.
- [36] M. Grant and S. Boyd. (2016) CVX: MATLAB Software for Disciplined Convex Programming. [Online]. Available: <http://cvxr.com/cvx>.
- [37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proc. IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324.
- [38] Y. LeCun, C. Cortes, and C. Burges. *The MNIST Database of Handwritten Digits*. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
- [40] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Univ. Toronto, Tech. Rep., 2009.