# FEELPGen: Data-Free Knowledge Distillation for Personalized Federated Learning Across Heterogeneous Edge Silos

Xiao Jiang, Hong Zhong, Depeng Chen, Jing Zhang, *Member, IEEE*, Qingyang Zhang, *Member, IEEE*, and Jie Cui, *Senior Member, IEEE*

***Abstract*—Deploying machine learning models on large-scale Internet of Things (IoT) devices in edge networks is challenging. Federated edge learning (FEEL) has emerged as a potential solution based on a hierarchical architecture. However, existing research relies primarily on an idealized cross-device assumption, overlooking more realistic cross-silo scenarios where devices typically belong to different organizational silos. To facilitate multigroup collaboration, we first propose a semidecentralized FEEL structure called FEELPGen, in which different silos collaborate in training to maximize local model benefits without relying on trusted third-party coordination. Based on that, a two-layer aggregation algorithm is proposed to enhance the generalization ability under highly heterogeneous data distribution. For inner-silo learning, we devise a heterogeneity-aware, synchronous inner-silo aggregation algorithm utilizing data-free knowledge distillation (DF-KD) based on generative learning (Gen). Feature vectors are generated to approximate silo knowledge. For inter-silo learning, a personalized (P), asynchronous inter-silo aggregation algorithm is proposed with adaptive selection and dynamic weight queues. To further improve efficiency, we introduce an optional optimized scheme, FEELPGen$^+$, which integrates a privacy-preserving dimension-reduction algorithm. Finally, we provide a detailed analysis for convergence and complexity to verify the feasibility of FEELPGen. Extensive experiments demonstrate that FEELPGen achieves significant improvement in accuracy compared to the state-of-the-art schemes.**

***Index Terms*—Cross-silo federated learning (FL), federated edge learning (FEEL), heterogeneity-aware, personalization, privacy-preserving.**

## NOMENCLATURE

| | |
|---|---|
| $N/K$ | Number of training end/edge. |
| $D$ | Training datasets. |
| $\mathcal{K}/\mathcal{K}'$ | Number of edge for aggregate/update. |
| $T$ | Number of inner-silo aggregation interval. |
| $\mathcal{T}$ | Number of inter-silo aggregation round. |
| $E$ | Local training epochs. |
| $\eta/\eta'$ | Edge/end learning rate. |
| $\tau_k/\tau_*^i$ | $k$th edge/$i$th end silo. |
| $\omega$ | Model parameter. |
| $\mathbf{X}/\mathbf{Y}$ | Feature/label distribution. |
| $\mathcal{Q}$ | True feature distribution. |
| $\mathcal{Z}^*$ | Generated feature distribution. |
| $\boldsymbol{G}$ | Generator. |
| $\hat{\omega}$ | Generator model parameter. |
| $\hat{\eta}$ | Generator learning rate. |
| $t$ | Timestamp. |
| $\boldsymbol{M}$ | Dynamic queue. |
| $\alpha$ | Level of heterogeneity. |
| $\sigma$ | Weight of global model. |

Received 30 July 2025; accepted 6 October 2025. Date of publication 9 October 2025; date of current version 8 December 2025. This work was supported in part by the National Natural Science Foundation of China under Grant U24A20243, Grant U23A20308, Grant 62572003, and Grant 62472003; and in part by the Natural Science Foundation of Anhui Province, China, under Grant 2408085JX010. *(Corresponding author: Hong Zhong.)*

The authors are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China, and also with Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2025.3619561

## I. INTRODUCTION

THE rapid expansion of the Internet of Things (IoT) devices has led to a significant influx of perception data, offering both opportunities and challenges for data-driven artificial intelligence (AI) applications. Federated edge learning (FEEL) [1] has attracted considerable attention in IoT applications [2]. Traditional IoT approaches [3] involve transmitting data to a central computing server for training, which incurs significant communication overhead, particularly when dealing with numerous edge devices. FEEL offers inherent cost advantages by shifting computing tasks to the edge server and replacing data updates with model updates.

To achieve practical implementation in real-world scenarios, different FEEL paradigms are designed to meet individual needs. The simple two-layer architecture [4], similar to horizontal federated learning (FL), is easily extensible. To reduce the computational burden, the hierarchical FEEL framework, which includes the end, edge, and cloud layers, has gained significant attention. Liu et al. [1] first proposed a three-layer framework that combines a FedAvg aggregation algorithm. Despite the multilayer structure reducing the computational load, security remains a concern from the centralized cloud. Furthermore, to eliminate the risk of centralization, semidecentralized [5] and cluster-based [6] designs are outstanding for complex optimization problems. These studies focus on *cross-device networks* where the edge commonly relays data to a central authority, as seen in applications like Gboard. However,

none of these approaches can be directly applied to more realistic edge networks emphasizing multigroup collaboration.

Existing FEEL solutions assume that end devices are controlled by multiple cloud servers, resembling a cross-device setting with similar features. However, in real-world scenarios, these devices are often divided into silos with unique features. For example, banks may build credit models using transaction records, and insurance companies may have similar needs. They may seek to train a global model while their devices are working collaboratively in private. In the two-layer FL, it contributes to cross-silo FL [7], [8]. Unlike virtual domains, silos emphasize fixed entity boundaries between different organizations. However, more practical cross-silo FEEL research has yet to be conducted, as interactions between organizations have been overlooked. *Collaborative entities exhibit autonomy marked by heterogeneous data, diverse requirements, and disparate performance metrics.* For example, the business requirements and data features of banks and insurance companies are not aligned. Cross-silo FEEL places greater demands on program design, including personalization [9], heterogeneity, and privacy protection [10].

*Personalization:* Personalization is a hallmark of cross-silo FEEL, where diverse organizations engage in collaborative training while aiming to maximize self-benefit.

*Heterogeneity:* In cross-silo FEEL, heterogeneity stems from data heterogeneity. It is characterized by statistical heterogeneity from separate data domains (such as noninindependent and identically distributed NIID), which leads to global model drift and degrades model performance.

*Privacy:* Privacy concerns arise due to organizational interests and legal limitations, as data is collected from individuals who may not trust the server or other silos. Section II will offer a comprehensive overview of these recent developments and outline our goals. Therefore, our primary objective is to develop a collaborative cross-silo FEEL framework for more realistic edge networks.

To address the concerns mentioned above, we propose a practical scheme for cross-silo FEEL through the combinatorial optimization of the FEEL framework's components. This scheme aims to personalize optimal returns under heterogeneous constraints while ensuring both efficiency and privacy protection in FEEL. To this end, we introduce a semidecentralized structure, *FEELPGen*. Without compromising privacy, an optimized data-free knowledge distillation (DF-KD) algorithm is designed to generate pseudo-global knowledge which enhances the inner-silo model and facilitates personalized aggregation across silos. Then, an optional optimization scheme, *FEELPGen*$^+$, leverages dimension-reduced pseudo-global knowledge to improve communication efficiency and privacy. The injection of random noise enhances the scheme's privacy while causing minimal loss of accuracy. To the best of our knowledge, this is the first comprehensive study that investigates these optimization problems within the context of cross-silo FEEL. The main contributions of this study are as follows.

1) We first propose FEELPGen, a novel semidecentralized two-layer FEEL framework tailored for realistic cross-silo settings. It integrates combinatorial optimization mechanisms to address data heterogeneity and personalization jointly.
2) To achieve heterogeneity-aware, a DF-KD algorithm is designed via a lightweight generator. Instead of accessing updated models, we approximate silo knowledge using low-dimensional vectors. With the extracted knowledge, a personalized global aggregation algorithm is proposed to maximize the profit of each edge. These methods comply with the privacy constraints inherent to FEEL.
3) Furthermore, an enhanced scheme, FEELPGen$^+$, is designed, which incorporates a privacy-preserving dimension reduction strategy to reduce communication overhead.
4) We provide a detailed theoretical analysis to establish the convergence guarantees of FEELPGen. Extensive empirical evaluations across diverse benchmark datasets demonstrate that FEELPGen outperforms state-of-the-art baselines, achieving utility gains of 2.43%–7.93%.

The remainder of this article is organized as follows. In Section II, we present a thorough review and comparison of the latest studies in this field. We elaborate the core design of FEELPGen and optimized FEELPGen$^+$ in Section III. In Section IV, we present a theoretical analysis of the scheme's availability and limitations. We conduct sufficient experiments to verify the performance of our design in Section V. The last Section VI concludes the study.

## II. Related Work

Unlike traditional FL [18], FEEL encounters more intricate requirements and challenges, stemming from the complex edge networks. Simple methods [4], [19] transfer the traditional two-layer FL to the edge computing scenario, but these fail to meet the complex demands of real-world applications. Research on hierarchical FEEL is more popular. Moreover, numerous studies on specific challenges, like data heterogeneity, personalized learning, and privacy and efficiency optimization, have been partially explored.

### A. Federated Edge Learning

In terms of hierarchical architecture, Liu et al. [1] were the first to propose HierFAV, a representative cloud–edge–end architecture for FEEL. Three-layer architecture is typically easier to deploy and optimize, making it the foundation for many studies. However, a centralized server can lead to privacy concerns and lacks flexibility. To eliminate dependency on the top-level central server, Sun et al. [5] proposed a semidecentralized FEEL framework. Building on this structure, Xu et al. [20] further designed dynamic scheduling at the edge to enhance efficiency. Furthermore, Chen et al. [14] proposed PED$^2$FL for decentralized FL in highly dynamic edge environments, which suffers from substantial communication overhead. However, these architectures are built upon an idealized cross-device setting.

### B. Heterogeneous Optimization

Data heterogeneity remains a key challenge that limits the generalization ability of the target model. Regard-

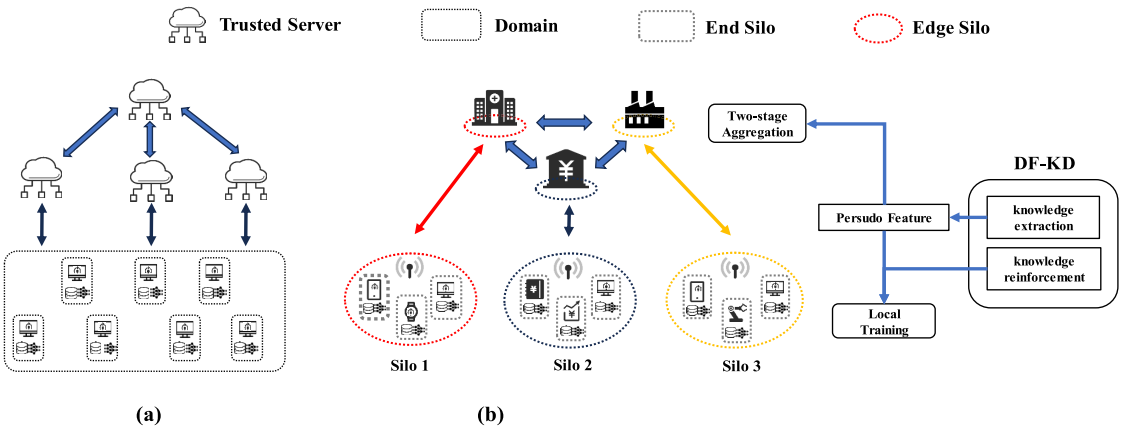



Fig. 1. Overview of FEEL and FEELPGen. Unlike the hierarchical structure in (a) FEEL, (b) FEELPGen features a semidecentralized structure for various edge organizations. Each edge initially trains within its private silo and seeks features from other organizations to expand the generalization capability of its private model. Unlike the domain boundary in FEEL, silo sets the boundaries of an individual's abilities and privacy with different colors.

TABLE I
COMPARISON OF THE RECENT FEEL SCHEMES

| | Year | Hierarchical | Heterogeneity | Privacy | Personalization | Efficiency |
|---|---|---|---|---|---|---|
| HierFAVG[1] | 2020 | ✓ | ✗ | ✗ | ✗ | ✗ |
| HierIoV[11] | 2021 | ✓ | ✓ | ✗ | ✗ | ✓ |
| HiFlash[12] | 2023 | ✓ | ✓ | ✗ | ✗ | ✓ |
| FedCH[13] | 2023 | ✓ | ✓ | ✗ | ✗ | ✓ |
| PED$^2$FL[14] | 2023 | ✗ | ✗ | ✓ | ✗ | ✗ |
| SD-FEEL[5] | 2023 | ✓ | ✓ | ✗ | ✗ | ✗ |
| HPFL[15] | 2023 | ✓ | ✗ | ✗ | ✓ | ✓ |
| AFL-HCS[16] | 2023 | ✗ | ✓ | ✗ | ✗ | ✓ |
| ESPerHFL[17] | 2024 | ✓ | ✓ | ✗ | ✓ | ✗ |

ing model drift caused by data heterogeneity in FEEL, a common idea is to cluster ends based on the similarity of their feature distributions. Wu et al. [12] introduced the HiFlash approach, which clusters edges based on the Jensen–Shannon divergence between the label distribution of ends and edges. Wang et al. [13] proposed FedCH, which is designed for resource-constrained edge computing with adaptive reclustering. *However, as shown in Fig. 1(b), such cluster-based methods require accurate estimation of ends' features and flexible scheduling of targeted end devices, potentially posing security risks and conflicting with cross-silo constraints.* Besides, adaptive aggregation schemes based on client selection have also proven beneficial for achieving efficient optimization for heterogeneity. Zhou et al. [11] proposed HierIoV, a multilayer heterogeneous model selection and context-aware learning mechanism. Based on the arrival sequence, Tang et al. [16] proposed a heterogeneous edge client selection method with an efficiency guarantee. Wang et al. [21] initially devised a staleness-aware mechanism grounded in round-based global aggregation, forming the fundamental framework for the hierarchical scheme. A limitation of these schemes is the necessity for the server to evaluate and schedule the edge devices.

### C. Personalized Optimization

Personalization aims at enhancing generalization ability of local models, and is a classic requirement in cross-silo settings [9]. In traditional FL, several personalized schemes have already been proposed. Jiang et al. [22] proposed model-agnostic metalearning to pretrain a global model with good generalization and fast adaptation. Tang et al. [23] proposed FedCor, which adaptively selects target clients based on the target loss. Zhang et al. [24] proposed adaptive local aggregation by capturing the desired information from the global model. Shamsian et al. [25] proposed pFedHN, which generates a set of personalized local models for different clients. Regularization-based methods [26] attempt to mitigate the impact of local knowledge by incorporating an approximate term. In cross-silo FL, Huang et al. [9] proposed FedAMP where each client maintains its own personalized local model, enabling it to better adapt to local data distribution. Based on a similar idea, Luo and Wu [27] learn multiple local models with the help of a dynamic directed relationship vector. Qin et al. [28] proposed FedAPEN, a method that requires each client to adaptively learn a set of additional weight coefficients and model ensembles. Xie et al. [29] achieve personalization for medical image segmentation by leveraging local feature enhancement. However, these strategies are infeasible for large-scale cross-silo FEEL, which involve a vast number of end devices and edge servers. Learning personalized local models directly on these resource-constrained end devices is challenging and incurs prohibitive computational overhead.

### D. Privacy and Efficiency

As fundamental constraints in FL, privacy and efficiency are overlooked in some optimization approaches. For instance, most existing heterogeneity optimization approaches, such as aggregation and client selection, often rely on precise prior knowledge, which raises potential privacy concerns. Inference attacks and inversion attacks can extract private information about the target from the model [30]. Chen et al. [14] ensure the privacy and security by introducing differential privacy. By incorporating cryptographic techniques, such as homomorphic encryption [31] and secure multiparty computation [32], data privacy can also be ensured. However, the complex computations and scheduling involved in these methods

lead to substantial communication or computational overhead. Therefore, achieving a balance between efficiency and privacy is important. Tang et al. [33] proposed a dynamic and flexible scheduling strategy for edge networks, which helps alleviate this conflict.

## III. FEELPGen Framework

In this section, we first present the problem definition of the work and provide an overview of FEELPGen. Then, the core algorithms are detailed to achieve heterogeneity-aware and personalized aggregation. Finally, we present the workflow of FEELPGen and discuss its optimizations FEELPGen$^{+}$.

### A. Problem Definition

*1) System Model:* This work focuses on a novel cross-silo FEEL, where $K$ edge organizations (short as edge) collaboratively train their target models for their respective end devices (short as end). As shown in Fig. 1, at the inner-silo level, multiple end devices collaboratively train a shared global model under the coordination of the edge server. At the inter-silo level, multiple edge servers select available models from each other in real time, perform a secondary aggregation, and then update the models on their local end devices. The overall target is designing a cross-silo FEEL framework that enhances the performance of personalized models under heterogeneous data distribution while adhering to privacy constraints.

*2) Optimization Objective:* For edges, the primary goal is to learn an optimal set of personalized models, $\{\omega_k\}_{k=1}^{K}$, for the collection of end devices it governs. Accordingly, the local objective function for edge $k$ is defined as the minimization of the average empirical loss across all its constituent ends

$$\mathcal{L}_k = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(\omega_k, D_k^i\right) \tag{1}$$

where $D_k^i$ is the local dataset of the end device $i$ in the edge $k$. $\mathcal{L}_k$ is the target loss, which quantifies the model consensus of silo $\tau_k$. In the cross-silo FEEL setting, the system's global objective is to jointly minimize the aggregate of the local objective functions from all $K$ participating organizations

$$\min_{k\in K} F\left(\{\omega_1, \ldots, \omega_K\}\right) = \sum_{k=1}^{K}\mathcal{L}_k. \tag{2}$$

*3) Challenges:* Our design aims to address key challenges: silo's reward maximization, data heterogeneity, and privacy constraints. The statistical heterogeneity of the training data adversely affects the utility of the target model. Moreover, the requirements of cross-silo settings inherently constrain the sovereignty and privacy of ends and edges.

*4) Assumptions:* As our work centers on the optimization algorithm, we simplify the problem setting with the following assumptions. First, we make the typical assumption of a lossless and reliable communication environment. Second, we assume each edge can schedule its inner-silo devices synchronously, and that inter-silo coordination is asynchronous. Finally, the training data is stored locally on the end devices and is never shared or transmitted.

### B. Overview

We propose a semidecentralized structure that leverages DF-KD and personalization, eliminating dependence on the central server, as shown in Fig. 1(b). For the silo, the edge collaborates with local ends for distributed training with heterogeneity-aware optimization. Then, without relying on a trusted third party, the edge conducts personalized aggregation operations based on pseudofeatures. We divide the system model into the following two components.

1) *Heterogeneity-Aware Inner-Silo Training:* Following the standard FL procedure, each end holds a private sample space $D_k^i \sim (\mathbf{X}, \mathbf{Y})$ and performs local training over $E$ rounds. The edge conducts $T$ rounds of local training to minimize the expected loss for each end

$$\omega_k = \underset{\omega_1,\ldots,\omega_N}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}\frac{D_k^i}{D_k}\mathcal{L}_i\left(\omega_k^i; D_k^i\right) \tag{3}$$

where $\omega_k$ denotes the edge model after aggregation.

2) *Personalized Inter-Silo Aggregation:* Due to the high feature heterogeneity across silos, edge nodes need to communicate with each other and personalize the selection of optimal models for aggregation. Here, we formulate the inter-silo aggregation for edge $k$ as follows:

$$\omega_k = \omega_k - \eta\sum_{j=1}^{\mathcal{K}}\frac{\sigma_j}{\sum\left[\sigma_j\right]}\omega_j \tag{4}$$

where $\mathcal{K}$ denotes the number of silo edges selected with a specific selection algorithm, and $\sigma$ means aggregated weights. In our approach, the selection strategy and weights are carefully designed to meet personalized requirements while incorporating asynchronous optimization.

### C. Core Algorithm

We use DF-KD [34] to approximate the heterogeneous data distribution. The core idea is training a generator through a *knowledge extraction* algorithm which can get the pseudo-feature distribution. Based on this approximate feature, the edges perform a two-step optimization to enhance model performance. First, a *knowledge reinforcement* operation is performed that returns the approximate features to the ends for boosting the generalization ability of the local models. Second, the approximate features are broadcast to other edges to perform more *accurate personalized aggregation*.

*1) Heterogeneity-Aware via Regularized DF-KD:* To achieve heterogeneity-aware for edge silo $\tau_k$, we have to observe the data distribution $\mathcal{Q}_k$ from a global perspective. A naive approach is to sample $(x, y) \in D_k^i$ randomly, which will converge to the true distribution in the condition of enough samples. Unfortunately, this violates privacy constraints. Therefore, the core target of our scheme is to recover a conditional distribution $\mathcal{Z}$ that approximates the ground-truth data distribution for each silo $\tau_k$ without any privacy violation. Next, we will gradually develop the design approach for the generator approximation, starting with real sampling.

First, we assume that the conditional distribution $\mathcal{Z}_k^*$ can output a space $\mathbf{Y}_k$ that is approximated from the ground-truth space $\mathbf{X}_k$, the objective function can be reformulated as an expectation-solving problem from the true label distribution $p(y)$ and features distribution $\mathcal{Q}_k(x|y)$

$$\mathcal{Z}_k^* = \underset{\mathbf{Y}_k \to \mathbf{X}_k}{\operatorname{argmax}} \mathbb{E}_{y\sim\hat{p}(y)} \mathbb{E}_{x\sim\mathcal{Q}_k(x|y)} \left[\log \hat{p}\left(y|x\right)\right] \tag{5}$$

where $\hat{p}(y)$ denotes the approximated prior ground-truth distribution $p(y)$. $x$ is sampled from raw data. Following the label sampling [34], we can approximate the prior label distributions $p(y|x)$ and $p(y)$ with local models and local training label counts as follows:

$$\hat{p}\left(y|x\right) \approx \frac{c}{n} \sum_i p\left(y|x;\omega_k^i\right) \tag{6}$$

$$\hat{p}\left(y\right) \approx \sum_n c * \mathbb{E}_{y^*\sim\mathcal{D}_k^n} \left[I\left(y^* = y\right)\right] \tag{7}$$

where $y^*$ denotes the ground-truth label in training data. $I(\cdot)$ is an indicator function which outputs 0 or 1. $c$ is an arbitrary constant and $n$ is the number of sampled set. In (6), we extract the ensemble knowledge from update models that act as Teachers to get an optimal approximation.

However, we notice that (7) relies on counting true labels $y^*$ and $x$ from clients' original datasets, which is prohibited due to privacy concerns. Therefore, further optimized KD without sharing or counting raw data is necessary. We introduce a data-free method to recover an induced distribution $\mathcal{Z}_k : \mathbf{Y}_k \to \mathcal{Z}_k$ over a latent space $\mathbb{R}$ via generative learning [34], which is more approximate to raw data. Specifically, we train a simplified model, $\boldsymbol{G}$, as the generator, which is parameterized by $\hat{\omega}$. $\boldsymbol{G}$ is a lightweight generation model that will approximate a target feature. Thus, we further reformulate (5) as follows:

$$\mathcal{Z}_k = \underset{\mathbf{Y}_k \to \mathcal{Z}_k}{\operatorname{argmax}} \mathbb{E}_{y\sim\hat{p}(y)} \mathbb{E}_{z\sim G_{\hat{\omega}_k}(z|y)} \left[\log \hat{p}\left(y|z\right)\right] \tag{8}$$

where $y$ is sampled arbitrarily from a latent label space. $z \in \mathcal{Z}$ means the latent feature space is generated by $\boldsymbol{G}_{\hat{\omega}_k}(z|y)$. $\hat{p}(y|z)$ can be approximated with (6). With the convergence of the generator, we can approximate $\mathcal{Z}_k \sim \mathcal{Q}_k$ which contains the ensemble feature information. More precisely, we present the inner-silo training as follows.

*a) Generator learning by knowledge extraction:* We can transfer the heterogeneity-aware to the generating optimization problem, which optimizes the generator with knowledge from local updates to get a more generalized induced distribution. In experiments, we find that the generator tends to overfit specific domain features. Inspired by regularization techniques [26], [35], we introduce *a random noise term*, and a *regularization term* constrains the update momentum direction to reduce deviation caused by noise

$$\begin{aligned}\mathcal{L}\left(\hat{\omega}_k\right) = \frac{1}{b^*} \sum_{b^*} f\left(\left(\frac{1}{N}\sum_{i=1}^{N} h\left(z;\omega_k^i\right)\right), y\right) \\ + \left\|\epsilon - f\left(h\left(z,\hat{\omega}_{k-1}\right), y\right)\right\| + \lambda\left\|\hat{\omega}_k - \hat{\omega}_{k-1}\right\|\end{aligned} \tag{9}$$

where $h(\cdot)$ represents the logistic output and nonlinear activation and $f(\cdot)$ is the loss function. $b^*$ denotes the batch size of generated $y$. $\epsilon$ is a random Gaussian noise introduced to improve the stochastic generalization capability during the optimization process. $\lambda$ is the balance factor preset as 0.1. Through compound loss, the generator can converge rapidly while ensuring approximation ability. Finally, we finish the knowledge extraction (KE) by

$$F\left(\hat{\omega}_k\right) := \mathbb{E}_{y\sim\mathbb{R}} \mathbb{E}_{z\sim\boldsymbol{G}_{\hat{\omega}_k}(z|y)} \left[\mathcal{L}\left(\hat{\omega}_k\right)\right] \tag{10}$$

$$\hat{\omega}_k = \hat{\omega}_k - \hat{\eta}\nabla F\left(\hat{\omega}_k\right). \tag{11}$$

*b) Local learning by knowledge reinforcement:* To improve the generalization, each end learns the ensemble representation from generator $\boldsymbol{G}$ by introducing inductive bias to regularize the local training. Specifically, the edge broadcast generator is directed to its ends. End sample induces distribution $z \sim \boldsymbol{G}(\cdot|y)$, which is similar to global distribution over the latent space, and gets the empirical loss

$$\hat{\mathcal{L}}_k\left(\omega_k^i\right) := \frac{1}{b^*} \sum_{b^*} f\left(\left(h\left(z;\boldsymbol{\omega}_k\right)\right), y\right). \tag{12}$$

By combining the aforementioned two loss functions, each end finishes knowledge reinforcement (KR) as follows:

$$F\left(\omega_k^i\right) := \mathbb{E}_{x\sim D_k^i}\left[\mathcal{L}\left(\boldsymbol{\omega}_k^i\right)\right] + \frac{b^*}{\left|D_k^i\right|} \mathbb{E}_{y\sim\mathbb{R}, z\sim G(z|y)} \hat{\mathcal{L}}_k\left(\omega_k^i\right) \tag{13}$$

$$\mathcal{L}\left(\boldsymbol{\omega}_k^i\right) := \frac{1}{\left|D_k^i\right|} f\left(h\left(x;\omega_k^i\right), y\right) \tag{14}$$

where the local dataset calculates the first part of the loss. With the generated virtual features, each local model gradually learns feature information of other ends without any raw data. For ease of understanding, the specific process is illustrated in Algorithm 1. Ultimately, the edge within each silo $k$ acquires the global model $\omega_k$. For this part, by KD-based heterogeneity-aware, the local experience loss function has a stronger generalization ability for their silo.

*2) Personalized Inter-Silo Aggregation:* The described KD method enhances the local model's ability to perceive global data inner-silo, but it does not improve the aggregation algorithm for the inter-silo. Aggregating all edge updates introduces an unavoidable bias in the target, as the highly differentiated feature space limits the target model's ability to represent local features. Furthermore, synchronized aggregation limits overall model performance due to the pace of the slowest edge. Thus, to optimize individual gains, we have devised a personalized hierarchical aggregation algorithm in which each edge can freely choose the appropriate aggregation object at any time.

To address data heterogeneity, one potential solution is to select updates with similar data distributions, which can significantly improve model performance [12], [13]. This method is typically accomplished using cluster-based methods, though these may compromise privacy. Fortunately, the generator provides a privacy-safe approximation of the silo distribution. To assess which model is suitable for aggregation, each edge initially uploads the generated feature distribution $\mathcal{Z}_i$ with a consensus label space $\mathbf{Y}$. For two distributions $\mathcal{Z}_i^*$ and $\mathcal{Z}_j$,

**Algorithm 1** Inner-Silo Aggregation Algorithm

**Input:** $N, \omega_{1\sim K}^{1\sim N}, \omega_{1\sim K}, \hat{\omega}_{1\sim K}, \boldsymbol{G}_{1\sim K}, \eta, \eta'$
**Output:** $\omega_k$

→ **Knowledge Reinforcement**
**For** each end in parallel **do**:
  Receive global model $\omega_k$ and generator $\boldsymbol{G_k}$
  Generate $\mathcal{Z}_k \leftarrow \boldsymbol{G}_{\hat{\omega}_k}(z|y)$
  Calculate $\mathcal{L}, \mathcal{L}^*$ for $x \sim D_k^i$, $z \sim \mathcal{Z}_k$ respectively
  $F(\omega_k^i) \leftarrow \mathbb{E}_{x,z}[\mathcal{L} + B\mathcal{L}^*]$
  $\omega_k^i \leftarrow \omega_k^i - \eta' \nabla F(\omega_k^i)$
  Update to edge $k$
**End For**
→ **Knowledge Extraction**
For each edge in parallel do:
  → **generator Learning**
  Calculate $\hat{\mathcal{L}}$ with $z \sim G(z|y)$ for all update model
  $F(\hat{\omega}_k) \leftarrow \mathbb{E}_z[\hat{\mathcal{L}}]$
  $\hat{\omega}_k \leftarrow \hat{\omega}_k - \hat{\eta} \nabla F(\hat{\omega}_k)$
  → **Aggregation**
  $\omega_k \leftarrow \omega_k - \eta \sum_i^N \frac{|D_k^i|}{|D_k|} \omega_k^i$
  Broadcast to end in silo
**End For**

**Algorithm 2** Inter-Silo Aggregation Algorithm

**Input:** $K, \mathcal{K}, \mathcal{K}', \omega_{1\sim K}, \boldsymbol{G}_{1\sim K}, \gamma, \eta$
**Output:** $\omega_i$

Each edge finish Algorithm 1 to get model parameter $\omega_i$ and generator $\boldsymbol{G}_i$
→ **Personalization**
**For** each edge in parallel **do**:
  Sample label set $\mathbf{Y}$ from $\mathbb{R}$
  Generate latent space $\mathcal{Z}_i^* \leftarrow \boldsymbol{G}_{\hat{\omega}_i}(z|y)$
  Broadcast $(\omega_i, \mathcal{Z}_i^*)$ to other edge
  **For** each edge **do**:
    Randomly sample a subset of $\mathcal{K}'$ edges for updating weights queue
    **If** selected edge $j$ not in $M_i'$:
      append $(\mathcal{Z}_j^*, \upsilon_j)$ to $M_i'$
    **Else**:
      update $(\mathcal{Z}_j^*, \upsilon_j)$ in $M_i'$
    Calculate two sub-weights:
    $\sigma_j^{KL} \leftarrow KL(\mathcal{Z}_i^*, \mathcal{Z}_j^*)$, $\sigma_j^{\upsilon} \leftarrow \Delta\upsilon_j^{-b}$
    Update weight: $\sigma_j = \gamma\sigma_j^{KL} + C(1-\gamma)\sigma_j^{\upsilon}$
  **End For**
  → **Aggregation**
    Select top-$\mathcal{K}$ from the sorted weights queue:
      $\omega_i = \omega_i - \eta \sum_j^{\mathcal{K}} \frac{\sigma_j}{\sum[\sigma_j]} \omega_j$
**End For**

we calculate the dissimilarity using Kullback–Leibler (KL) divergence as follows:

$$\mathrm{KL}\left(\mathcal{Z}_i \middle|\middle| \mathcal{Z}_j\right) = \sum_{x\in X} \mathcal{Z}_i(x) \log \frac{\mathcal{Z}_i(x)}{\mathcal{Z}_j(x)}. \tag{15}$$

For edge $i$, we preserve the dissimilarity with a *dynamic queue* $\boldsymbol{M}_i = [\mathcal{Z}_1, \ldots, \mathcal{Z}_j, \ldots], i \neq j$. We calculate $\sigma_j^{\mathrm{KL}} = \mathrm{KL}(\mathcal{Z}_i \| \mathcal{Z}_j)$, denoting the dissimilarity of training data between edge $i$ and edge $j$. A smaller $\sigma_j^{\mathrm{KL}}$ indicates a higher benefit for utility. In real time, the queue $\boldsymbol{M}_i$ is dynamically updated, including new edges that are allowed to join the scheduling process. However, opting to update all edges leads to communication overload. Hence, in our approach, we opt to randomly select a subset of edges ($\mathcal{K}' \in K$) for updating the local dynamic queue in a round of inter-silo aggregation.

Considering the variations in computational capacity among different silos, we optimize an asynchronous collaborative aggregation via staleness-aware. For inter-silo communication, we propose asynchronous aggregation to avoid unnecessary waiting. Specifically, we reform the dynamic queue $M$ as follows: $\boldsymbol{M}_i' = [(\mathcal{Z}_1, \upsilon_1), \ldots, (\mathcal{Z}_j, \upsilon_j), \ldots]$, $\upsilon_j > 0$. The $\upsilon_j$ denotes the staleness of edge $j$ for edge $i$ which can be tagged with the latest timestamp.

Then, we set a weight $\sigma_j^{\upsilon}$ for $\omega_j$ as follows:

$$\sigma_j^{\upsilon} = \Delta\upsilon_j^{-\phi} \tag{16}$$

where $\Delta\upsilon_j$ is the time disjunction from last update. $\phi$ is the penalty factor, which is preset as 0.8. As a result, edge $i$, which updates more frequently, enjoys a higher weight. Note that for both the generated feature space and timestamps, adversaries cannot deduce raw information about the end and edge.

Based on the above two constraints, we calculate the composite weight $\sigma_j$ as follows:

$$\sigma_j = \gamma\sigma_j^{\mathrm{KL}} + C(1-\gamma)\sigma_j^{\upsilon} \tag{17}$$

where $C$ is a constant preset to 0.1 and $\gamma$ is a balance factor set to 0.5. According to that, edge $i$ selects the best subset from available edges for global aggregation. Specifically, we select $\mathcal{K}$ edges with the highest weights and obtain the new edge model $\omega_i$ with (4). We elaborate on this part in Algorithm 2.

### D. Workflow of FEELPGen

Based on the designed algorithm above, we present the global algorithm of the system in Algorithm 3.

*1) Synchronous Local Training:* In local epochs $e$, end device $i$ initializes its local model $\omega_k^i$ and generator $G_i$ obtained from edge node $k$. For each epoch in $E$, the end device learns from the local dataset $D_k^i$, which consists of real data input. To strengthen the generalization ability, $G_i$ generates the approximate feature distribution of the ensemble $\tau_k$. The empirical loss for latent space can be calculated as (12). The model employs a backpropagation algorithm to minimize the enhanced loss defined by (13). Notably, the generator extends the optimization boundary of the local model (Algorithm 3 line 6). Ultimately, the ends transmit their local model parameters to the central edge within the silo and await the commencement of the next training round. So far, the local training process has been enhanced with a heterogeneity-aware generator that learns from historical updates. Here, the training process is synchronous to maintain consistency in updating the generator.

*2) Heterogeneity-Aware Inner-Silo Aggregation:* In our approach, if the local timestamp $t_{\mathrm{end}}$ has not reached aggregation interval $T$, edges conduct inner-silo training between edge and end $i$. In this process, each edge node completes

**Algorithm 3** Global Algorithm

**Init:** $E, T, \mathcal{T}, \omega_{1\sim K}, \omega_{1\sim K}^{1\sim N}, \boldsymbol{G}_{1\sim K}, \hat{\omega}_{1\sim K}, \gamma, \eta. \eta'$
1 → **Synchronous Local Training in ends:**
2 **For** each end in parallel **do**:
3 $\omega_k^i \leftarrow \omega_k$
4 $\boldsymbol{G} \leftarrow \hat{\omega}_k$
5 **For** each local epochs **do**:
6 $\omega_k^i \leftarrow \mathbf{KR}(D_k^i, \boldsymbol{G})$ in Algorithm 1
7 **End For**
8 $t_{end} \leftarrow t_{end} + 1$
9 update local model to edge
10 **End For**
11 → Asynchronous aggregation in edges:
12 **While** $t_{edge} < \mathcal{T}$:
13 **For** each edge in parallel **do**:
14 **If** $t_{end} \% T = 0$:
15 → **Inter-silo aggregation**
16 $\mathcal{Z}_k \leftarrow \boldsymbol{G}_{\hat{\omega}_k}(z|y)$
17 → **privacy-preserving PCA for FEELPGen$^+$**
18 $\tilde{\mathcal{B}}_k \rightarrow sup_{d'}(M^T M + E)$
19 $\sigma_j^{KL} \leftarrow KL(\tilde{\mathcal{B}}_i, \tilde{\mathcal{B}}_j)$
20 aggregate $\omega_k$ with Algorithm 2
21 $t_{edge} \leftarrow t_{edge} + 1$
22 **Else**:
23 → **Inner-silo aggregation**
24 $\omega_k, \hat{\omega}_k \leftarrow$ **KE** $(\omega_k^i, \hat{\omega}_k)$ in Algorithm 1
25 Broadcast $\omega_k$ to end in silo
26 **end For**

two updates: one for the global model aggregation and one for updating the generator (Algorithm 3, line 24). First, the edges initiate the next aggregation step upon receiving updates from all silo nodes, leading to the formation of the global model $\omega_k$. Subsequently, the generator conducts optimization [see (10)] to enhance its perceptual power by utilizing regularized knowledge extracted from local updates. Finally, the parameters of the global models and the generator will be broadcast for the next local training.

*3) Asynchronous Interact Silo Aggregation:* If the local timestamp $t_{\text{end}}$ has reached aggregation interval $T$, the edge node updates it with other edge models individually. This selection strategy is detailed in Algorithm 2. Upon completion of the inner-silo update, the edge node proceeds by uploading the generated feature space $\mathcal{Z}_k$ (Algorithm 3, line 16) and randomly sampling $(\mathcal{Z}_k, \upsilon_k)$ responsive edges in timestamp $t_{\text{edge}}$. Subsequently, the weights queue $\boldsymbol{M}_k'$ is updated (Algorithm 3, line 19), and the top-$\mathcal{K}$ edges' updated models are selected and aggregated (Algorithm 3, line 20). Finally, the newest global model is distributed to each end node within silo $k$. This process operates asynchronously, allowing the edge server to conduct inter-silo updates without waiting for the completion of other edges. The dynamic queues store historical weight memories, thus avoiding additional computation and storage.

### E. Optimized FEELPGen$^+$

Revisiting FEELPGen, our optimizations all rely on the generated virtual feature distributions $\mathcal{Z}^*$, which have a dimension bounded by $d$. However, frequent updates to the high-dimensional data distributions are bound to incur huge communication costs. Transmitting raw feature spaces is also insecure. Therefore, we address this concern with privacy-preserving principal component analysis (PPCA) [36]. Like PCA, we aim to find a low-dimensional subspace $\mathcal{B}$, which preserves more important information. However, PCA is mathematically reversible and thus cannot ensure privacy. Therefore, we specify the generated feature space as a matrix $M$ $[d * d]$. Under the calculated covariance matrix $M^T M$, we release a noise $E$ which is bounded by privacy budget $\varepsilon$

$$\tilde{A} = M^T M + E \quad \left(E \sim \mathcal{N}\left(0, \Delta_{\varepsilon,\delta}\right)\right) \tag{18}$$

where the introduced performance-neutral noise will guarantee its privacy. Dwork et al. [36] proved that by fine-tuning the parameters, noise-added PCA can extract better performance from original algorithms when the inherent dimensionality of the input is much lower than the ambient dimension. Therefore, the injection of random perturbations severs the link between original data and postprocessed data.

We then conduct feature decomposition with singular value decomposition for $\tilde{A}$: $\text{SVD}(\tilde{A}) = U\Sigma V^T$, where $U$ is the left singular subspace and $V$ is the right. $\Sigma$ is a diagonal metric whose diagonals are singular values $\sigma_{1\sim d}$. Finally, the released top-$d'$ singular subspace of $U$ is $\tilde{\mathcal{B}}$ (Algorithm 3, line 18), which contains the main $d'$-dimensional features for $\mathcal{Z}^*$, including the most features. Based on that, we design an irreversible feature compression scheme that can enhance privacy and reduce communication overhead.

## IV. Evaluation

In this section, we analyze the convergence ability of FEELPGen, as stated in Theorem 1, and then evaluate its computational complexity.

### A. Convergence Analysis

We will next demonstrate that FEELPGen can still converge after the introduction of KD. For analytical convenience, we refine the local optimization objective as follows: $\mathcal{L}(D^i, \omega^i) = \mathcal{L}_{\omega^i}(f(h(x), y)) + p_g \hat{\mathcal{L}}_{\hat{\omega}^i}(G(z|y), y)$. The loss function $\mathcal{L}$ is *nonconvex.* In FEELPGen, the generator is a special feature extractor $z \sim G(x, \hat{\omega}_k)$ for heterogeneity-aware. Assuming that loss minimization is achieved through $E$ rounds of stochastic gradient descent, where $e \in \{1/2, 1, \ldots, E, \ldots\}$ denotes the local training epoch and $t$ denotes the global round. $e = 1/2$ means the intermediate epoch between the previous round of aggregation and the next local training. We short the average weight of client $i$ as $p_i = |D_k^i|/|D_k|$ and the average weight of generator as $p_g = b^*/B^i$. First, we establish the following assumptions.

*Assumption 1 (L-Smoothness):* The gradient of the loss function $\mathcal{L}$ is $L_1$-smooth and the loss function of the generator satisfies $L_2$-smooth

$$\begin{aligned} &\left|\left|\nabla\mathcal{L}\left(\omega^{t_1}\right) - \nabla\mathcal{L}\left(\omega^{t_2}\right)\right|\right| \le L_1 \left|\left|\omega^{t_1} - \omega^{t_2}\right|\right| \quad \forall t_1, t_2 > 0 \\ &\left|\left|\nabla\mathcal{L}\left(\hat{\omega}^{t_1}\right) - \nabla\mathcal{L}\left(\hat{\omega}^{t_2}\right)\right|\right| \le L_2 \left|\left|\hat{\omega}^{t_1} - \hat{\omega}^{t_2}\right|\right| \quad \forall t_1, t_2 > 0. \end{aligned} \tag{19}$$

*Assumption 2 (Unbiased Gradient):* The stochastic gradient $g^t$ on a batch $\xi$ is an unbiased estimator of the local gradient

$$\mathbb{E}_{\xi}\left[g^t\right] = \nabla\mathcal{L}\left(\omega^t\right). \tag{20}$$

*Assumption 3 (Bounded Variance and Bounded Expectation of Gradients):* The variance of the local gradient is bounded by

$$\mathbb{E}\left[\left\|g^t - \nabla\mathcal{L}\left(\omega^t\right)\right\|^2\right] \le \varphi^2, \quad \mathbb{E}\left[\left\|\hat{g}^t - \nabla\mathcal{L}\left(\hat{\omega}^t\right)\right\|^2\right] \le \hat{\varphi}^2 \tag{21}$$

and its expectation is bounded by $V$

$$\mathbb{E}\left[\left\|g^t\right\|\right] \le V^2, \quad \mathbb{E}\left[\left\|\hat{g}^t\right\|\right] \le \hat{V}^2. \tag{22}$$

Similarly, the bound $\hat{\varphi}, \hat{V}$ for the gradient of generator $\hat{g}$.

*Lemma 1:* If the loss function satisfies Assumption 1, the upper bound for the linear approximation at $\omega$ is

$$\underbrace{\mathcal{L}\left(\omega^{t_1}\right) - \mathcal{L}\left(\omega^{t_2}\right) - \langle\nabla\mathcal{L}\left(\omega^{t_2}\right), \omega^{t_1} - \omega^{t_2}\rangle}_{(1)} \le \frac{L_1}{2}\left|\left|\omega^{t_1} - \omega^{t_2}\right|\right|^2.$$

*Proof:* Here, an auxiliary integral variable $\zeta$ is introduced to compute the average inner product between the gradient variation and the direction

$$\begin{aligned}
(1) &= \int_0^1 \langle\nabla\mathcal{L}\left(\omega^{t_1} + \zeta\left(\omega^{t_2} - \omega^{t_1}\right)\right) - \nabla\mathcal{L}\left(\omega^{t_1}\right),\ \omega^{t_2} - \omega^{t_1}\rangle\, d\zeta \\
&\overset{(1)}{\le} \int_0^1 \left\|\mathcal{L}\left(\omega^{t_1} + \zeta\left(\omega^{t_2} - \omega^{t_1}\right)\right) - \nabla\mathcal{L}\left(\omega^{t_1}\right)\right\| \left\|\omega^{t_2} - \omega^{t_1}\right\| d\zeta \\
&\overset{(2)}{\le} \int_0^1 L_1 \left\|\omega^{t_1} + \zeta\left(\omega^{t_2} - \omega^{t_1}\right) - \omega^{t_1}\right\| \left\|\omega^{t_2} - \omega_{t_1}\right\| d\zeta \\
&\le \int_0^1 L_1\zeta \left\|\omega^{t_2} - \omega^{t_1}\right\|^2 d\zeta \\
&\le \frac{L_1}{2}\left\|\omega^{t_2} - \omega^{t_1}\right\|^2.
\end{aligned} \tag{23}$$

The inequality (1) satisfies the Cauchy–Schwarz inequality: $\langle a, b\rangle \le \|a\|\|b\|$. And the inequality (2) follows the Assumption 1.

*Lemma 2:* By taking the expectation and applying scaling to Lemma 1, the loss function after $E$ rounds of local training is bounded by

$$\mathbb{E}\left[\mathcal{L}^{E,t+1}\right] \le \mathcal{L}^{\frac{1}{2},t+1} - \left(\eta' - \frac{\eta'^2 L_1}{2}\right) EV^2 + \frac{L_1}{2}\eta'^2 E\varphi^2 \tag{24}$$

where $\eta'$ denotes the local learning rate.

*Proof:* Following the assumption, the expectation of the two sides of Lemma 1 is given by

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}^{e+1,t+1}\right] \\
&\overset{(1)}{\le} \mathbb{E}\left[\mathcal{L}^{e,t+1} + \langle\nabla\mathcal{L}^{e,t+1}, \omega^{e+1,t+1} - \omega^{e,t+1}\rangle + \frac{L_1}{2}\left\|\omega^{e+1,t+1} - \omega^{e,t+1}\right\|^2\right] \\
&= \mathbb{E}\left[\mathcal{L}^{e,t+1} - \eta'\langle\nabla\mathcal{L}^{e,t+1}, g^{e,t+1}\rangle + \frac{L_1}{2}\eta'^2\varphi^2\right] \\
&\overset{(2)}{\le} \mathcal{L}^{e,t+1} - \eta'\left\|\nabla\mathcal{L}^{e,t+1}\right\|^2 + \frac{L_1}{2}\eta'^2\mathbb{E}\left[\left\|g^{e,t+1}\right\|^2\right] \\
&= \mathcal{L}^{e,t+1} - \eta'\left\|\nabla\mathcal{L}^{e,t+1}\right\|^2 + \frac{L_1}{2}\eta'^2\left(\left\|\nabla\mathcal{L}^{e,t+1}\right\| + \mathbb{E}\left[\left\|g^{e,t}\right\|^2\right] - \mathbb{E}\left[g^{e,t}\right]^2\right) \\
&\overset{(3)}{\le} \mathcal{L}^{\frac{1}{2},t+1} - \sum_{\frac{1}{2}}^{E-1}\left(\eta' - \frac{\eta'^2 L_1}{2}\right)\left\|\nabla\mathcal{L}^{e,t+1}\right\|^2 + \frac{L_1}{2}\eta'^2 E\varphi^2 \\
&\le \mathcal{L}^{\frac{1}{2},t+1} - \left(\eta' - \frac{\eta'^2 L_1}{2}\right) EV^2 + \frac{L_1}{2}\eta'^2 E\varphi^2
\end{aligned} \tag{25}$$

where the inequalities (1)–(3) respectively follow Lemma 1, Assumption 2, and Assumptions 2–3.

*Lemma 3:* Following the second part of the target loss, since the generated knowledge $z$ is simultaneously aggregated in the server, the loss function of client $i$ at global round $t+1$ can be bounded as follows:

$$\mathbb{E}\left[\mathcal{L}^{t+1}\right] \le \mathcal{L}^{E,t} + \frac{\eta'^2 L_1}{2}E^2V^2 + 2p_g\eta' L_2 E\hat{V}. \tag{26}$$

*Proof:* Taking the generated variable $z$ into the proof on loss reduction

$$\begin{aligned}
\mathcal{L}^{\frac{1}{2},t+1} - \mathcal{L}^{E,t} &= \mathcal{L}\left(\omega^{\frac{1}{2},t+1}, z^{t+1}\right) - \mathcal{L}\left(\omega^{E,t}, z^t\right) \\
&\le \frac{L_1}{2}\left\|\sum_i^N p_i\omega_i^{E,t} - \omega_i^{\frac{1}{2},t}\right\|^2 + \mathcal{L}\left(\omega^{E,t+1}, z^{t+1}\right) \\
&\quad - \mathcal{L}\left(\omega^{E,t}, z^t\right).
\end{aligned} \tag{27}$$

Taking expectation of both sides

$$\begin{aligned}
\mathbb{E}\left(\mathcal{L}^{\frac{1}{2},t+1} - \mathcal{L}^{E,t}\right) &\overset{(1)}{\le} \frac{L_1}{2}\mathbb{E}\left\|\omega_i^{E,t} - \omega_i^{\frac{1}{2},t}\right\|^2 + \mathcal{L}\left(\omega^{E,t+1}, z^{t+1}\right) \\
&\quad - \mathcal{L}\left(\omega^{E,t}, z^t\right) \\
&= \frac{L_1}{2}\mathbb{E}\left\|\sum_{e=\frac{1}{2}}^{E-1}\eta' g_i^{e,t}\right\|^2 + \mathcal{L}\left(\omega^{E,t+1}, z^{t+1}\right) \\
&\quad - \mathcal{L}\left(\omega^{E,t}, z^t\right) \\
&\overset{(2)}{\le} \frac{L_1\eta'^2}{2}E^2V^2 \\
&\quad + \underbrace{\mathcal{L}\left(\omega^{E,t+1}, z^{t+1}\right) - \mathcal{L}\left(\omega^{E,t}, z^t\right)}_{(2)}
\end{aligned} \tag{28}$$

where inequality (1) follows from Jensen's inequality: $\mathbb{E}\|\mathbb{E}(a) - a\|^2 \le \mathbb{E}\|a\|^2$ as $a = \omega_i^{E,t} - \omega_i^{1/2,t}$. Inequality (2) follows Assumption 3.

Next, the proof for the latter part (2) is provided below

$$\begin{aligned}
\mathbb{E}\left[(2)\right] &= p_g\mathbb{E}\left\|\hat{\mathcal{L}}\left(\hat{\omega}_k^{E,t+1}\right) + \hat{\mathcal{L}}\left(\hat{\omega}_k^{E,t}\right)\right\| \\
&\le p_g L_2 \sum_{k=1}^K \mathbb{E}\left[\left\|\hat{\omega}_k^{E,t} - \hat{\omega}_k^{E,t-1}\right\|\right] \\
&\overset{(1)}{\le} p_g L_2 \sum_{k=1}^K \mathbb{E}\left[\left\|\hat{\omega}_k^{E,t} - \hat{\omega}_i^{\frac{1}{2},t}\right\| + \left\|\hat{\omega}_k^{\frac{1}{2},t} - \hat{\omega}_k^{E,t-1}\right\|\right] \\
&\le p_g L_2 \sum_{k=1}^K \mathbb{E}\left[\eta' E\hat{V} + \left\|\hat{\omega}_k^{\frac{1}{2},t} - \hat{\omega}_k^{E,t-1}\right\|\right] \\
&\overset{(2)}{\le} p_g L_2 \sum_{k=1}^K \mathbb{E}\left[\eta' E\hat{V} + \sqrt{\mathbb{E}\left[\left\|\sum_{e=\frac{1}{2}}^{E-1}\eta'\hat{g}_k^{e,t-1}\right\|\right]}\right]
\end{aligned}$$

$$\leq p_g L_2 \sum_{k=1}^{K} \left(\eta' E\hat{V} + \eta' E\hat{V}\right) = 2\eta' L_2 E\hat{V} \tag{29}$$

where equality (1) follows the triangle inequality. Inequality (2) satisfies Jensen's inequality. Lemma 3 can be calculated by combining (28) and (29).

*Theorem 1 (FEELPGen Convergence):* For each end, the loss of the training model after each communication round is bounded by

$$\mathbb{E}\left[\mathcal{L}^{\frac{1}{2},t+1}\right] \leq \mathcal{L}^{\frac{1}{2},t} - \left(\eta' - (E+1)\frac{\eta'^2 L_1}{2}\right) EV^2 + \frac{L_1\eta'^2}{2} E\varphi^2 + 2p_g\eta' L_2 E\hat{V}. \tag{30}$$

When $\eta' < (2EV^2 - 4p_g L_2 E\hat{V})/(L_1 E\varphi^2 + L_1 E^2 V^2 + L_1 EV^2)$, the loss decreases with each round, thereby meeting the conditions for gradient descent. It provides a convergence guarantee, ensuring that the FEELPGen algorithm reaches a near-optimal solution.

*Proof:* By combining Lemma 2 and Lemma 3, we obtain the following inequality:

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}^{\frac{1}{2},t+1}\right] &\leq \mathcal{L}^{E,t} + \frac{\eta'^2 L_1}{2} E^2 V^2 + 2p_g\eta' L_2 E\hat{V} \\ &\leq \mathcal{L}^{\frac{1}{2},t} - \left(\eta' - \frac{\eta'^2 L_1}{2}\right) EV^2 + \frac{L_1\eta'^2}{2} E\varphi^2 \\ &\quad \times \frac{\eta'^2 L_1}{2} E^2 V^2 + 2p_g\eta' L_2 E\hat{V} \\ &= \mathcal{L}^{\frac{1}{2},t} - \left(\eta' - (E+1)\frac{\eta'^2 L_1}{2}\right) EV^2 + \frac{L_1\eta'^2}{2} E\varphi^2 \\ &\quad + 2p_g\eta' L_2 E\hat{V}. \end{aligned} \tag{31}$$

To satisfy SGD, the value of the latter part should be less than 0

$$\begin{aligned}&\eta'\left(\eta'\frac{L_1 E\varphi^2}{2} + 2p_g L_2 E\hat{V} - \left(1 - \eta'\frac{(E+1)L_1}{2}\right) EV^2\right) < 0 \\ &\eta'\left(\frac{L_1 E\varphi^2 + L_1 E^2 V^2 + L_1 EV^2}{2}\right) < EV^2 - 2p_g L_2 E\hat{V} \\ &\eta' < \frac{2EV^2 - 4p_g L_2 E\hat{V}}{L_1 E\varphi^2 + L_1 E^2 V^2 + L_1 EV^2} \end{aligned} \tag{32}$$

where the local learning rate $\eta' > 0$.

### B. Privacy Bounds for PPCA

*Lemma 4 (Worst-Case Utility Guarantee [36]):* We set $\mathcal{B}_{d'}$ as the rank $-d'$ feature matrix which is extracted from the original $d$-dimensional matrix $A_d$. And $\tilde{\mathcal{B}}_{d'}$ is the rank-$d'$ subspace of the perturbed $\tilde{A}_d$. With high probability

$$\|M\mathcal{B}\|_F^2 - \|M\tilde{\mathcal{B}}\|_F^2 \leq O\left(d'\sqrt{d}\Delta_{\varepsilon,\delta}\right) \tag{33}$$

where $\Delta_{\varepsilon,\delta} = (2{*}\ln(1.25/\delta))^{1/2}/\varepsilon$. From that, suppose the compress rate is fixed as $d, d'$, and the error bound is proportional to the noise level. Although Lemma 4 has demonstrated an exponential relationship between the error and noise bounds, specific privacy budget bounds $\varepsilon$ have not been determined.

*Theorem 2 (Lower Bounds for Privacy Budget):* For perturbation algorithm $\mathcal{R} : \mathcal{D} \to \tilde{\mathcal{D}}$, letting a fingerprinting code $(d+1, d, f, \beta, \xi)$ satisfy weakly robust [37]. If $\xi \leq 1/2$ and $\delta = O(1/n^2)$, then $\mathcal{R}$ satisfies $(\varepsilon, \delta)$ differential privacy

$$\varepsilon \geq \ln\frac{p(1-\xi)/d - \delta}{\xi} \quad \left(p = \Pr\left[\mathcal{R} \text{ is } \beta\text{-vaild for } S\right]\right) \tag{34}$$

where $p$ denotes the $\beta$-valid for subset $S \subseteq [d]$ vectors.

*Proof:* Following the proof of fingerprinting codes [38], we set $\mathcal{R}' = \text{Trace}(\mathcal{R})$. As the first property of weakly robust in [36, Definition 10], $\Pr(\text{Trace}(R) \mid \beta - \text{ vaild for } S) \geq 1 - \xi$.. Let $p$ donate $\Pr[\mathcal{R}] < 1$. By combining and scaling, we can calculate that $\Pr(\mathcal{R}') \geq p(1-\xi)$. As the second property of weakly robust—$\Pr(\mathcal{R}') \leq \xi$ and $\mathcal{R}'$ satisfies $(\varepsilon, \delta)$-DP—we can derive that

$$\frac{p(1-\xi)}{d} \leq e^{\varepsilon}\xi + \delta. \tag{35}$$

Through rearrangement and simplification, we can obtain $\varepsilon \geq \ln(p(1-\xi)/d - \delta)/\xi$. We analyze the upper bounds of utility guarantee and the lower bounds of the privacy budget $\varepsilon$. The privacy-guaranteed level of noise increases directly with the original dimension of the matrix $d$. This implies that a larger space necessitates more noise, but not more than the threshold that destroys utility, which needs to be carefully chosen. Therefore, in Theorem 2, we offer a lower bound for privacy budgets at the worst utility for reference.

### C. Complexity Analysis

We analyze the complexity of the proposed scheme from both *computation* and *communication* perspectives. The computational overhead of FEELPGen primarily stems from two components: the distillation process and the personalization. The computational complexity of the distillation process is $O(b^*|G|)$, which arises from the training of the generator. $|G|$ denotes the number of parameters in the generator model. $b^*$ denotes the number of labels sampled per round. The complexity of personalization is $O(\mathcal{K}' + \mathcal{K})$, which is driven from sampling-based state update $O(\mathcal{K}')$ and Top-$\mathcal{K}$ client selection. Therefore, for FEELPGen, the total additional computational overhead is $O(\mathcal{T}K(TNb^*G + (\mathcal{K}' + \mathcal{K})))$. The additional communication overhead comes from two parts: inner-silo and inter-silo communications. In inner-silo learning, $N$ ends transmit $d$-dimensional feature distribution per round. Therefore, the communication overhead for this part is $O(\mathcal{T}TNKd)$. For $\mathcal{T}$ personalized aggregation rounds, $K$ edge collaboratively aggregates their private global models. In each round, $\mathcal{K}'$ edges' features are sampled to update the state queue $M$. Therefore, the communication overhead is $O(\mathcal{T}K\mathcal{K}'d)$. Each edge owns a queue whose size is $O(d)$. Therefore, the total additional communication overhead is $O(\mathcal{T}K(TNd + \mathcal{K}'d) + Kd)$. Furthermore, FEELPGen$^+$ can reduce the communication overhead to $O(\mathcal{T}K(TNd' + \mathcal{K}'d') + Kd')$ by reducing feature dimensions ($d' \ll d$). Notably, while FEELPGen offers superior performance compared to a basic hierarchical FEEL strategy, these gains come at the cost of additional computational and communication overhead. This trade-off may limit its applicability in certain practical scenarios. Consequently, potential adopters should evaluate this additional overhead before deployment.

## V. Experiments

In this section, we conduct experimental evaluations of our work from four aspects: utility, efficiency, heterogeneity, and hyperparameters.

### A. Evaluation Methodology

*1) Experimental Setup:* The experiments are run on a server with Ubuntu 18.04 as the operating system, an Intel[1] Xeon[1] Silver 4210 CPU @ 2.20 GHz, 128 GB of RAM, and 4 NVIDIA GeForce RTX 3090 GPUs.

*2) Datasets and Model Architectures:* In our experiments, we utilize four real datasets: *MNIST* [39] and *EMNIST* [40], which are widely used for image classification tasks, and *CelebA* [41], a real face recognition dataset. To simulate edge computing scenarios, we evaluate our method on the *UCI_HAR* [42], an IoT dataset for human activity recognition using smartphone sensor data. We split the dataset into 200 ends while ensuring that their sample distribution adheres to a $\alpha$-bounded Dirichlet distribution [43] for heterogeneity. In the case of CelebA, due to its inherent data heterogeneity, we divide the data by the number of celebrity classes $|D_c|$. We use a two-layer convolutional neural network with 26 434 trainable parameters for MNIST and EMNIST. A fully connected neural network with 664 646 trainable parameters is set for the HAR dataset. For the CelebA dataset, we additionally added a convolutional layer, resulting in a total of 25 954 trainable parameters. Finally, we use a lightweight two-layer multilayer perception as the generator $\boldsymbol{G}$, which will not incur excessive computational and communication overhead.

*3) Hyperparameters:* We deploy the FEELPGen framework with ten edges ($K = 10$), each overseeing 20 ends ($N = 20$). Half of the end updates were used for inner-silo aggregation, which conducts five intervals ($T = 5$) for each inter-silo aggregation ($\mathcal{T} = 50$). Each local node trains for $E$ epochs ($E = 20$). We set the learning rate to 0.01 for both $\eta$ and $\eta'$. The learning rate for the generator is set as 0.0001 with a decay rate of 0.98. During inter-silo aggregation, each edge node initially selected $\mathcal{K}' = 4$ updates for updating its dynamic queue $M$. Next, $\mathcal{K} = 5$ updates will be selected individually for aggregation. To measure the robustness of heterogeneity, data will be split into different levels: $\alpha = [0.03, 0.05, 0.1, 0.5]$. A smaller $\alpha$ indicates a higher degree of heterogeneity. Finally, we applied PPCA to compress the 32-D approximated feature to 2 dimensions with $(1, 0.001)$-DP.

*4) Benchmarks:* In this article, we set eight representative schemes as benchmarks.

*FedAvg-c* [18]: FedAvg stands out as the classic FL architecture, characterized by a two-layer structure centered around a single server.

*FedProx* [26]: A popular heterogeneous optimization algorithm that adds a local regularization term. We employ it in both centralized structures, FedProx-c, and hierarchical structures, FedProx-h.

*HierFAVG [1]:* It is the first to offer an cloud–edge–end FEEL solution.

*FedALA [24]:* A personalized FL algorithm with adaptive local aggregation.

*SDFEEL [5]:* With a similar semidecentralized architecture, SDFEEL has introduced a hierarchical aggregation algorithm RFL-HA with intra/intercluster for FEEL.

*HPFL [15]:* It introduces personalized operations into edge learning based on FuerFAVG and achieves the balance between accuracy and efficiency through edge scheduling.

*ESPerHFL [17]:* A personalized client–edge–cloud hierarchical FL algorithm.

*FEELPGen*$^-$: To evaluate the performance of personalization, we reduce the optimized inter-silo aggregation algorithm with the normal FedAvg.

### B. Evaluation of Utility

In Table III, we perform comprehensive experiments to assess the accuracy performance of various benchmarks. In summary, FEELPGen generally outperforms other benchmarks. Specifically, for MNIST, the accuracy of our work exceeds 93% under highly heterogeneous data distribution. The same is also observed for EMNIST and CelebA, where our scheme exhibits better performance. Subsequently, we will further demonstrate the superiority of our approach through comparisons involving different architectures (Group1), similar methods (Group2), and self-comparisons (Group3).

*1) Group1:* In FL approach, both FedAvg-c and FedProx-c demonstrate strong performance due to centralized data distribution. When we distribute the data to ends, the performance of FedProx-h significantly deteriorates. Specifically, for the MNIST task, we observe that FedProx-h experiences a substantial accuracy drop, with errors increasing from 9.07% to 54.18%. This decline is primarily due to an increase in the number of ends, which results in unbalanced data features within individual domains and limits the representational capacity of a single model. Nevertheless, our proposed scheme effectively addresses this issue. By employing a continuously trained generator, a single model can expand its feature boundaries, ultimately enhancing its generalization ability. Thus, this explains the usability of our approach in edge environments.

*2) Group2:* FEELPGen achieves state-of-the-art accuracy compared to similar FEEL approaches. We observe that HierFAVG, which adopts an end–edge–cloud structure with a basic aggregation algorithm, struggles to converge at lower values of $\alpha$. At $\alpha = 0.05$, predictive accuracy drops to 8.56%. Among the latest semidecentralized schemes, SDFEEL also experiences accuracy degradation. Under the same structure, FEELPGen outperforms SDFEEL, achieving improvements ranging from 3.64% to 37.83% on MNIST and 13.11% to 15.02% on EMNIST. Among personalized schemes, HPFL outperforms FedALA and ESPerHFL, and FEELPGen achieves 2.43% to 6.7% improvement for MNIST and 7.93% to 14.62% improvement for EMNIST compared to HPFL. Intuitively, the optimization effect becomes more pronounced as heterogeneity increases. Consequently, our proposed solutions demonstrate optimal performance relative to the latest benchmarks.

*3) Group3:* To validate the effectiveness of the proposed two-layer algorithm, we conduct a comparison experiment

[1]Registered trademark.

TABLE II
AVERAGE ACCURACY OF VARIOUS BENCHMARKS UNDER DIFFERENT LEVELS OF DATA HETEROGENEITY CONSTRAINTS

| | | FedAvg-c | FedProx-c | FedProx-h | HierFAVG | FedALA | SDFEEL | HPFL | ESPerHFL | FEELPGen$^-$ | FEELPGen | FEELPGen$^+$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **MNIST** | **$\alpha$=0.05** | 83.61±2.24 | 85.05±1.39 | 30.87±1.77 | 32.19±4.04 | 81.06±0.63 | 55.85±1.07 | 86.98±0.22 | 85.3±0.22 | 84.12±0.92 | 93.68±0.53 | **93.84±0.42** |
| | **$\alpha$=0.1** | 88.27±0.63 | 87.98±0.81 | 58.99±1.62 | 60±3.3 | 84.69±0.78 | 88.61±0.45 | 89.94±0.24 | 88.58±0.29 | 87.72±0.98 | **95.17±0.1** | 95.15±0.1 |
| | **$\alpha$=0.5** | 92.41±0.18 | 92.26±0.16 | 83.19±0.85 | 83.44±0.63 | 92.66±1.33 | 93.36±0.2 | 94.62±0.18 | 93.52±0.25 | 93.79±0.56 | **97.05±0.03** | 97.05±0.06 |
| **EMNIST** | **$\alpha$=0.05** | 57.06±1.43 | 53.86±2.03 | 8.46±0.32 | 8.56±0.25 | 63.17±1.15 | 62.35±1.24 | 62.41±0.65 | 65.64±0.78 | 47.06±0.43 | **77.03±0.23** | 76.93±0.87 |
| | **$\alpha$=0.1** | 61.87±1.11 | 60.3±1.54 | 6.17±0.26 | 12.14±0.33 | 67.9±1.65 | 67.24±0.76 | 66.76±0.78 | 68.22±0.83 | 52.18±1.02 | **80.26±0.28** | 80.11±0.31 |
| | **$\alpha$=0.5** | 71.26±0.52 | 70.33±0.52 | 25.03±0.63 | 40.65±1.02 | 76.24±0.44 | 71.47±0.51 | 76.65±0.4 | 74.71±0.63 | 61.5±0.74 | **84.58±0.19** | 84.34±0.1 |
| **HAR** | **$\alpha$=0.1** | 82.43±4.24 | 82.47±3.93 | 81.51±0.51 | 85.28±0.78 | 78.34±0.67 | 63.02±0.93 | 86.03±0.65 | 75.91±0.38 | 79.65±0.63 | 85.78±0.55 | **86.64±0.92** |
| | **$\alpha$=0.5** | 88.1±1.56 | 87.34±0.6 | 85.58±0.75 | 85.77±0.82 | 83.49±0.51 | 62.07±0.48 | 87.16±0.47 | 72.55±0.73 | 81.22±0.32 | 87.82±0.32 | **88.69±0.24** |
| **CelebA** | **$\lvert D_c\rvert$=50** | 88.17±0.48 | 88.37±0.49 | 84.77±0.81 | 85.64±0.4 | 86.84±0.41 | 86.19±0.56 | 86.09±1.09 | 86.95±0.14 | 86.23±0.44 | **88.62±0.19** | 88.84±0.15 |
| | **$\lvert D_c\rvert$=100** | 88.98±0.19 | 88.99±0.24 | 84.34±0.62 | 85.06±0.48 | 86.77±0.18 | 86.67±0.31 | 88.95 ±0.21 | 87.84±0.43 | 86.01±0.59 | **90.12±0.12** | 90.08±0.09 |

TABLE III
AVERAGE COMPUTATION CONSUMPTION AND COMMUNICATION CONSUMPTION OF DIFFERENT SCHEMES UNDER DIFFERENT PHASES, DATASET

| Dataset | Scheme | Compute Consumption (s) | | Communication Consumption (B) | |
|---|---|---|---|---|---|
| | | end-edge | edge-edge | end-edge | edge-edge |
| **MNIST** | HierFAVG | 0.677 | 0.008 | \ | |
| | FEELPGen | 1.263 | 0.131 | 40720 | 41072 |
| | FEELPGen$^+$ | 1.265 | 0.088 | | 2672 |
| **CelebA** | HierFAVG | 1.872 | 0.011 | \ | |
| | FEELPGen | 2.411 | 0.196 | 40688 | 41072 |
| | FEELPGen$^+$ | 2.446 | 0.111 | | 2672 |

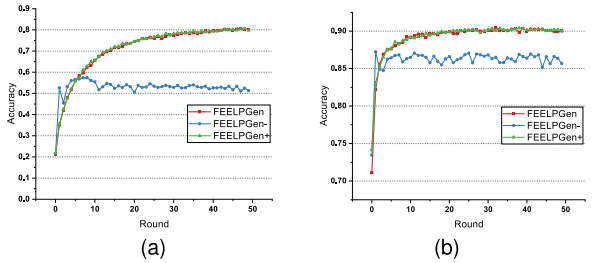



Fig. 2. Convergence performance of different FEELPGen variants on the EMNIST and CelebA datasets. (a) EMNIST ($\alpha$ = 0.1). (b) CelebA ($|D_c|$ = 100).

for our work in Table II and Fig. 2. As shown in Fig. 2(a) and (b), the three schemes all tend to converge, but FEELPGen$^-$ shows the weakest performance. It focuses solely on KD within the inner silo. In Table II, it outperforms the baseline HierFAVG but falls slightly behind SDFEEL. When combined with personalized aggregation, FEELPGen shows a significant accuracy improvement ranging from 3.26% to 9.56% for MNIST and from 23.08% to 51% for EMNIST. It demonstrates the effectiveness of our two-layer aggregation algorithm, FEELPGen, providing substantial performance improvements. On the other hand, we tested the impact of PPCA with FEELPGen$^+$. Compared to FEELPGen, the addition of PPCA in FEELPGen$^+$ has minimal impact on model performance, and even leads to a slight performance improvement in the HAR task. This satisfies the fine-tuning optimization objective of Dework et al. [36]. Our optimized solutions for privacy protection and efficiency enhancement do not negatively affect overall performance.

### C. Evaluation of Efficiency

To assess the efficiency of our scheme, we analyze both computational and communication overheads. We select HierFAVG as the baseline due to its simple architecture and algorithmic design, making it the most efficient among related works. Resource consumption at various stages is measured and detailed in Table III, facilitating a fine-grained analysis of the algorithm's trade-offs. Compared to the baseline, the introduction of the generator within each silo results in an additional computational cost of approximately 0.6 s, albeit with an improvement in accuracy. During inter-silo aggregation, computations addressing heterogeneity incur a cost ranging from 0.12 to 0.18 s. Simultaneously, FEELPGen leads to an additional communication overhead of approximately 40 000 bytes due to the inclusion of auxiliary information. Consequently, although the scheme enhances accuracy, it inevitably incurs additional overhead. However, there is potential for optimization in these areas. In FEELPGen$^+$, we compress the auxiliary distribution using PPCA to enhance efficiency. In comparison to FEELPGen, the optimized scheme reduces computational overhead by approximately 32%–43% during edge–edge aggregation. Simultaneously, the communication cost is reduced to 2672B, corresponding to an approximate 16× compression ratio.

Considering the computational overhead in large-scale deployment scenarios, we test the time consumption of one round of personalization computation in Fig. 3. When we linearly increase the number of silos $K$ to 100, the time overhead increases linearly, reaching 0.2311 s. It means that the computational overhead of personalization is proportional to the number of silos, which remains acceptable for large-scale scenarios. Furthermore, as shown in Fig. 3(b), when the number of samples per round increases, the time overhead increases linearly. When updating with 80% of the states

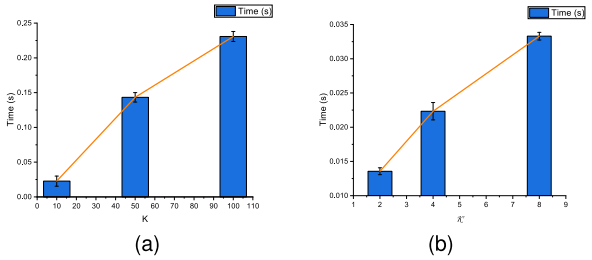



Fig. 3. Time consumption of one round of personalization under large-scale deployment scenarios. (a) Impact of $K$. (b) Impact of $\mathcal{K}'$.

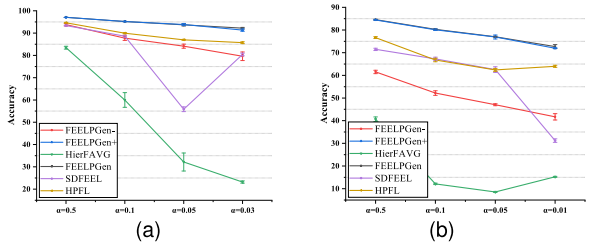



Fig. 4. Under MNIST/EMNIST, the performance of benchmarks varies under different levels of heterogeneity $\alpha$. (a) MNIST. (b) EMNIST.

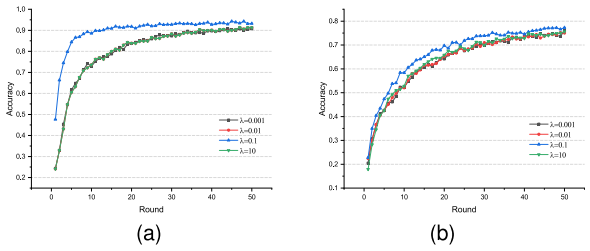



Fig. 5. Performance of FEELPGen under different balance factor $\lambda$. (a) MNIST ($\alpha = 0.05$). (b) EMNIST ($\alpha = 0.05$).

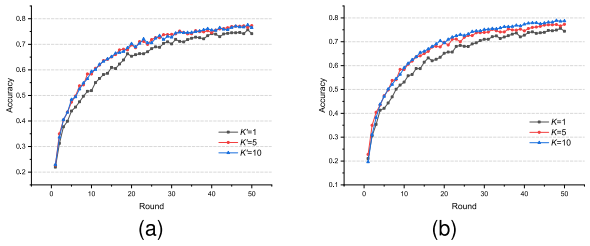



Fig. 6. For EMNIST, the impact of the number of state update samples $\mathcal{K}'$ and the number of aggregated models $\mathcal{K}$ on performance. (a) Impact of $\mathcal{K}'$. (b) Impact of $\mathcal{K}$.

sampled, the time overhead is only 0.0332 s. It means that sampling can further enhance the efficiency of personalization.

In summary, although FEELPGen introduces additional computational and communication overhead, these overheads are justified by the corresponding gains in accuracy, with the overall overhead remaining within acceptable limits.

### D. Evaluation of Heterogeneity

To assess the algorithm's robustness in highly heterogeneous scenarios, we conducted controlled experiments at various levels of $\alpha$, as outlined in Table II. For the handwriting recognition task, the algorithm's performance does decline with increasing heterogeneity. However, our decay rate was comparatively slower. We show the appearance in Fig. 4. Notably, even under extreme heterogeneity, our scheme demonstrates superior usability and outperforms other benchmarks. Under the highly challenging condition of $\alpha = 0.01$, FEELPGen maintained an accuracy of 72.68%, demonstrating a remarkable improvement of 30.96% compared to the suboptimal scheme. The personalized optimization scheme HPFL also achieved the second-best performance. In contrast, SDFEEL performs well at low intensities but exhibits visible utility collapse at strong threats. In comparison, HPFL is more robust. HierFAVG performs the worst because it focuses only on efficiency but lacks heterogeneous optimization algorithms. Here, we notice that SDFEEL achieves 87.02% in the condition of $\alpha = 0.03$ [as shown in Fig. 4(a)]. We attribute this to a loss explosion, as the loss reached up to 598.4 in the results. Furthermore, it is noteworthy that FEELPGen and FEELPGen$^-$ display comparable robust performance. This can be attributed to the complementary feature information generated through KD and privately transmitted by the generator. In the naturally heterogeneous CelebA dataset, we also tested the effect of local data size. Model performance shows a minor enhancement with an increase in the volume of training data.

### E. Ablation Study

In this section, we will evaluate the impact of several important hyperparameters.

*1) Impact of Balance Factor $\lambda$:* In FEELPGen, $\lambda$ is used to control the regularization term, constraining the errors introduced by noise and local optimization. In Fig. 5, we evaluate the accuracy for different $\lambda = \{0.001, 0.01, 0.1, 10\}$ on the MNIST and EMNIST datasets. From the results, when $\lambda$ is set to 0.1, the performance is optimal. A reasonable value of $\lambda$ yields an improvement of up to 2.57% on EMNIST and 3.31% on MNIST. This is because when $\lambda$ is too small, it fails to constrain the bias introduced by noise. Conversely, when $\lambda$ is too large, the regularization term limits the step size of model updates, thereby reducing the model's generalization ability. Therefore, the balance factor has an empirically optimal solution.

*2) Impact of Edge Sampling Factors $\mathcal{K}$ and $\mathcal{K}'$:* On the EMNIST dataset, we evaluated the impact of personalized edge aggregations on model performance. We compared two critical parameters: the number of state updates per round $\mathcal{K}'$ and the number of aggregations per round $\mathcal{K}$. The former determines the precision of the state queue $M$ at each edge, while the latter determines the number of knowledge aggregated by each edge node in each round. As shown in Fig. 6(a), when only one target feature ($\mathcal{K}' = 1$) is sampled per round to update the dynamic queue $M$, a noticeable drop in accuracy is observed. However, when more than 50% of the states ($\mathcal{K}' = \{5, 10\}$) are sampled for updates, the accuracy tends to stabilize and converge. This implies that a larger $\mathcal{K}'$ can

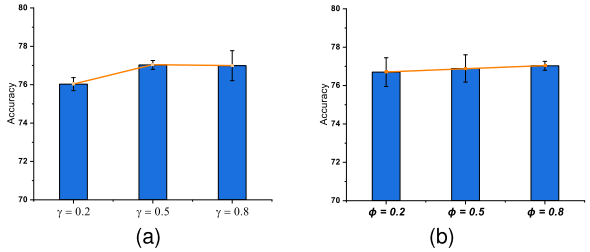



Fig. 7. Impact of asynchronous optimization parameters on model performance. $\gamma$ is a balance factor and $\phi$ is a penalty factor for staleness. (a) Impact of $\gamma$. (b) Impact of $\phi$.

TABLE IV

KL DIVERGENCE OF THE GENERATED FEATURE DISTRIBUTION. CLASSES REPRESENT LABELS: C0 (DIGIT "0"), C7 (DIGIT "7"), AND C9 (DIGIT "9")

| Round | 0 | 100 | 150 |
|---|---|---|---|
| **C0** | 0.052 | 0.026 | 0.001 |
| **C7** | 0.059 | 0.095 | 0.009 |
| **C9** | 0.069 | 0.037 | 0.004 |

enhance model performance, but there also exists an optimal threshold. Additionally, as shown in Fig. 6(b), the more the edge models are aggregated, the better the model's performance. However, it is noteworthy that more detailed perception and aggregation imply greater communication overhead, which needs to be balanced in specific scenarios.

*3) Impact of Asynchronous Optimization:* In asynchronous optimization, we introduced several parameters, including a penalty factor $\phi$ to evaluate staleness and a balance factor $\gamma$ to adjust the relative weight of personalization. Since $C$ is a constant that needs to be set according to the specific scenario, we exclude it from consideration. In personalized inter-silo aggregation, $\gamma$ is used to balance the weights between personalization optimization and staleness in (17). Intuitively, a larger value of $\gamma$ implies a greater emphasis on personalization optimization, which benefits accuracy improvement. The results, as shown in Fig. 7(a), indicate that when $\gamma$ is set to 0.8, the performance is 1.1% better than when it is set to 0.2. Additionally, we introduce the penalty factor $\phi$ to reduce the weight of outdated updates. In Fig. 7(b), this parameter has no significant impact on the model's accuracy. However, a larger $\phi$ reduces the error of the performance. Therefore, for FEELPGen, it is necessary to determine appropriate hyperparameters for different application scenarios.

*4) Convergence of the Generator:* In this section, we analyze the performance of the generator in FEELPGen. For EMNIST, a random batch of labels is sampled for the generator to approximate the feature distribution. We select a set of classes $\{0, 7, 9\}$ with different batches $\{6, 9, 4\}$, and two input samples are randomly sampled for each class. The KL divergence is introduced to measure the similarity between generated feature distributions across training rounds, as shown in Table IV. With the progression of training rounds, the similarity between the generated distributions improves, indicating that the generator can gradually approximate the global feature distribution with increasing precision. Overall, our results demonstrate that a lightweight, low-dimensional generator can effectively capture the ensemble feature distribution of the target silo.

TABLE V

UNDER EMNIST, $\alpha = 0.1$, THE PERFORMANCE OF FEELPGEN UNDER DIFFERENT DROP RATE

| Drop rate | 0 | 0.1 | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|---|
| **Inner-silo** | 80.26±0.28 | 79.85±0.11 | 78.47±0.34 | 78.02±0.28 | 76.98±0.47 |
| **Inter-silo** | | 80.13±0.51 | 80.3±0.31 | 80.38±0.38 | 80.21±0.45 |

*5) Impact of Dynamic Communication:* This communication bottleneck severely limits the scalability of large-scale distributed training on server-based architectures [44]. To model the communication failures resulting from an unstable network environment, we employ a dropout rate that is applied to both the inner-silo and inter-silo communication phases. These experiments are evaluated on the EMNIST dataset, with data partitioned to create a heterogeneity level of $\alpha = 0.1$. The results presented in Table V indicate that the performance degradation is more pronounced due to inner-silo dropouts. Specifically, at a dropout rate of 40%, we observe a 3.28% decrease in accuracy. Conversely, the personalized aggregation stage demonstrates greater resilience to such losses. We attribute this disparity to the following reasons: a reduction in the number of participating end devices directly affects the model's convergence capability. In contrast, during personalized aggregation, the dynamic weight and dynamic queue we designed adaptive compensates for these dropouts, thereby mitigating their adverse effects.

## VI. CONCLUSION

To better leverage the capabilities of edge terminals in heterogeneous edge networks, we focus on cross-silo FEEL, a paradigm suitable for realistic collaboration training among diverse organizations. We introduce FEELPGen, an advanced edge learning framework. The optimized two-layer aggregation algorithm enables edge nodes to privately assess data distributions within their silos and select appropriate cross-silo aggregation strategies based on the generated feature distributions. With a small amount of additional overhead, our work ensures enhanced utility improvements under heterogeneous data distributions. Extensive experiments and theoretical analysis demonstrate that our design outperforms similar approaches in both utility and privacy. In future work, we aim to implement a fully asynchronous training paradigm to avoid the delays caused by synchronous inner-silo training. Additionally, we seek fine-grained personalization that can extend to the user level. Optimizing broadcast communication costs in decentralized networks with numerous edges is also imperative.

## ACKNOWLEDGMENT

The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this article.

## REFERENCES


[1] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[2] J. Mu, Y. Cui, W. Ouyang, Z. Yang, W. Yuan, and X. Jing, "Federated learning in 6G non-terrestrial network for IoT services: From the perspective of perceptive mobile network," *IEEE Netw.*, vol. 38, no. 4, pp. 72–79, Jul. 2024.

[3] Y. J. Ahn, M. Kim, J. Lee, Y. Shen, and J. P. Jeong, "IoT edge-cloud: An Internet-of-Things edge-empowered cloud system for device management in smart spaces," *IEEE Netw.*, vol. 38, no. 3, pp. 109–117, May 2024.

[4] 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.

[5] Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Semi-decentralized federated edge learning with data and device heterogeneity," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 2, pp. 1487–1501, Jun. 2023.

[6] W. Y. B. Lim et al., "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022.

[7] C. Huang, M. Tang, Q. Ma, J. Huang, and X. Liu, "Promoting collaboration in cross-silo federated learning: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 62, no. 4, pp. 82–88, Apr. 2024.

[8] Z. Zhou et al., "Safeguarding privacy and integrity of federated learning in heterogeneous cross-silo IoRT environments: A moving target defense approach," *IEEE Netw.*, vol. 38, no. 3, pp. 25–32, May 2024.

[9] Y. Huang et al., "Personalized cross-silo federated learning on non-IID data," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 9, pp. 7865–7873.

[10] K. Liu, S. Hu, S. Z. Wu, and V. Smith, "On privacy and personalization in cross-silo federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 5925–5940.

[11] X. Zhou, W. Liang, J. She, Z. Yan, and K. I. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5308–5317, Jun. 2021.

[12] Q. Wu et al., "HiFlash: Communication-efficient hierarchical federated learning with adaptive staleness control and heterogeneity-aware client-edge association," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1560–1579, May 2023.

[13] 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.

[14] S. Chen, Y. Wang, D. Yu, J. Ren, C. Xu, and Y. Zheng, "Privacy-enhanced decentralized federated learning at dynamic edge," *IEEE Trans. Comput.*, vol. 72, no. 8, pp. 2165–2180, Aug. 2023.

[15] C. You, K. Guo, H. H. Yang, and T. Q. S. Quek, "Hierarchical personalized federated learning over massive mobile edge computing networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8141–8157, Nov. 2023.

[16] B. Tang, Y. Xiao, L. Zhang, B. Cao, M. Tang, and Q. Yang, "AFL-HCS: Asynchronous federated learning based on heterogeneous edge client selection," *Cluster Comput.*, vol. 27, no. 5, pp. 6247–6264, Aug. 2024.

[17] C. Ma, X. Li, B. Huang, G. Li, and F. Li, "Personalized client-edge-cloud hierarchical federated learning in mobile edge computing," *J. Cloud Comput.*, vol. 13, no. 1, p. 161, Dec. 2024.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[19] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[20] B. Xu, H. Zhao, H. Cao, S. Garg, G. Kaddoum, and M. M. Hassan, "Edge aggregation placement for semi-decentralized federated learning in industrial Internet of Things," *Future Gener. Comput. Syst.*, vol. 150, pp. 160–170, Jan. 2024.

[21] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2021, pp. 1–10.

[22] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019, *arXiv:1909.12488*.

[23] M. Tang et al., "FedCor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10092–10101.

[24] J. Zhang et al., "FedALA: Adaptive local aggregation for personalized federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 9, pp. 11237–11244.

[25] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized federated learning using hypernetworks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9489–9502.

[26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[27] J. Luo and S. Wu, "Adapt to adaptation: Learning personalization for cross-silo federated learning," in *Proc. IJCAI: Conf.*, 2022, p. 2166.

[28] Z. Qin, S. Deng, M. Zhao, and X. Yan, "FedAPEN: Personalized cross-silo federated learning with adaptability to statistical heterogeneity," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2023, pp. 1954–1964.

[29] L. Xie et al., "pFLFE: Cross-silo personalized federated learning via feature enhancement on medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2024, pp. 599–610.

[30] J. Chen, H. Yan, Z. Liu, M. Zhang, H. Xiong, and S. Yu, "When federated learning meets privacy-preserving computation," *ACM Comput. Surv.*, vol. 56, no. 12, pp. 1–36, Dec. 2024.

[31] Q. Xie et al., "Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey," *IEEE Internet Things J.*, vol. 11, no. 14, pp. 24569–24580, Jul. 2024.

[32] H. Kaminaga, F. M. Awaysheh, S. Alawadi, and L. Kamm, "MPCFL: Towards multi-party computation for secure federated learning aggregation," in *Proc. IEEE/ACM 16th Int. Conf. Utility Cloud Comput.*, Dec. 2023, pp. 1–10.

[33] X. Tang, C. Guo, K.-K. R. Choo, and Y. Liu, "An efficient and dynamic privacy-preserving federated learning system for edge computing," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 207–220, 2023.

[34] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 12878–12889.

[35] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116,Jan. 1995.

[36] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze Gauss: Optimal bounds for privacy-preserving principal component analysis," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, May 2014, pp. 11–20.

[37] G. Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, vol. 55, no. 2, pp. 1–24, May 2008.

[38] M. Bun, J. Ullman, and S. Vadhan, "Fingerprinting codes and the price of approximate differential privacy," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2014, pp. 1–10.

[39] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

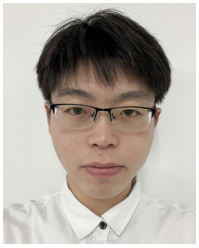
[40] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2921–2926.

[41] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celebA) dataset," *Retrieved August*, vol. 15, p. 11, Aug. 2018.

[42] D. Anguita et al., "A public domain dataset for human activity recognition using smartphones," in *Proc. Esann*, vol. 3, no. 1, 2013, pp. 3–4.

[43] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 965–978.

[44] M. Fang et al., "Byzantine-robust decentralized federated learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dec. 2024, pp. 2874–2888.



**Xiao Jiang** is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His research interests include the security of artificial intelligence, federated learning, and edge computing.

**Hong Zhong** was born in Anhui, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2005.

She is currently a Professor and the Ph.D. Supervisor at the School of Computer Science and Technology, Anhui University, Hefei. She has over 200 scientific publications in reputable journals (e.g., IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON BIG DATA), academic books, and international conferences. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN).

**Depeng Chen** received the Ph.D. degree in computer science from the Universitat Autónoma de Barcelona, Bellaterra, Spain, in 2020.

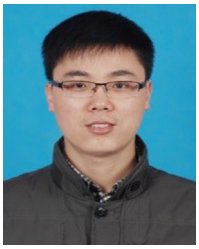
He is currently a Lecturer at the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests include trustworthy artificial intelligence systems, communication anonymity, and privacy.

**Jing Zhang** (Member, IEEE) is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

She has nearly 20 scientific publications in reputable journals (e.g., IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Information Sciences*, and *Science China Information Sciences and Vehicular Communications*) and international conferences. Her research interests include vehicular ad hoc network, IoT security, and applied cryptography.

**Qingyang Zhang** (Member, IEEE) was born in Anhui, China, in 1992. He received the B.Eng. and Ph.D. degrees in computer science from Anhui University, Hefei, China, in 2014 and 2021, respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. He has over 30 scientific publications in reputable journals including *Secure Computing*, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and IEEE TRANSACTIONS ON COMPUTERS and international conferences. His research interests include edge computing, computer systems, and security.



**Jie Cui** (Senior Member, IEEE) was born in Henan, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2012.

He is currently a Professor and the Ph.D. Supervisor at the School of Computer Science and Technology, Anhui University, Hefei. He has over 200 scientific publications in reputable journals (e.g., IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON MULTIMEDIA), academic books, and international conferences. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN).

Dr. Cui is on the Editorial Board of several international journals, such as *IET Communications*, *Security and Communication Networks, and Sensors*.