# HPFL: Hybrid Privacy-Preserving Federated Learning Framework Based on Homomorphic Encryption and Differential Privacy

**Anonymous authors**
Paper under double-blind review

## Abstract

Federated Learning (FL) enables model training on distributed data without the disclosure of private information. However, it is still vulnerable to malicious aggregators in FL. Existing Differential Privacy (DP) methods suffer from accuracy degradation due to noise injections. While Homomorphic Encryption (HE) schemes have no model accuracy loss, they need tremendous computation overheads, especially for a large model, which slows down FL convergence time. Moreover, existing hybrid DP and HE schemes suffers from low accuracy and high computation complexity. To this end, this paper proposes a hybrid privacy-preserving FL (HPFL) framework that dynamically integrates HE with DP at each device's side. We formulate this issue as a utility maximization problem under the consideration of time consumption for both model training and model protections within each global training round. A closed-form optimal HE ratio for each device is theoretically derived, and a model gradient magnitude-based mask is further designed to determine the HE-based model parameters. Extensive experiments on diverse data setting demonstrate that proposed HPFL significantly outperforms DP method only with accuracy improvements of up to 30%. Moreover, our proposed HPFL can significantly reduce the overall training time of FL by approximately 80%, while suffering from 3% accuracy loss, compared with HE only.

## 1 Introduction

Federated Learning (FL) enables collaborative training on edge devices with their local data, which has been widely adopted in multimedia applications, such as personalized recommendation systems Di et al. (2025). In FL, some malicious entities, such as the eavesdropper or an unreliable model aggregator, can still infer the raw data of edge devices. Therefore, differential privacy (DP) and homomorphic encryption (HE) methods are often adopted to protect privacy during FL training process. However, existing works on DP or HE in FL still face the following issues. 1) Although DP provides privacy guarantees, it perturbs local model gradients, leading to slow global model convergence and model training accuracy, especially under non-Independently and Identically Distributed (non-IID) data scenarios He et al. (2024), Yang et al. (2025), Zhao et al. (2025a); 2) Although HE can preserve data privacy without the loss of model accuracy, it suffers from significant high computational overheads, prolonging the FL model training time Jalali and Chen (2024), Zhao et al. (2023); 3) Some recent works, such as Ahmar et al. (2025), Negoya et al. (2025), start to study hybrid HE and DP schemes, which suffer from either high computational overhead or significant accuracy degradation Negoya et al. (2025), Jia et al. (2022). This is because they all use DP and HE to a full local model. Therefore, the goal of this paper is to jointly consider DP and HE to design a hybrid low-complexity model protection scheme, while suffering from limited model accuracy loss. However, designing such an integrated scheme faces the following challenges.

- **The integrations of HE and DP methods:** This involves the design of which local model parameters are preserved by HE or DP. The challenge of this design lies in 1) on one hand, we need the design to be a low computation complexity; 2) On the other hand, the designed scheme should also attain a sound training performance, e.g., less training time, and high training accuracy. It is not easy to achieve both, as they are naturally contradictory with



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

each other. For example, if the integration is too complex, it will fail to meet the real-time requirements in FL. Otherwise, it will be unable to obtain high model accuracy.

- **Heterogeneous local model aggregations:** Due to devices' heterogeneities, some devices with strong computation capacities would apply HE more for local model protections, while others may prefer to DP. This results in heterogeneous uploaded local models at the server's side, though the structure of their local models are the same. Consequently, the server should implement a new aggregation scheme for these heterogeneous models, which fundamentally beyond the scope of traditional federated aggregation schemes McMahan et al. (2023).

To address above challenges, this paper proposes a hybrid privacy-preserving FL (HPFL) scheme, which firstly determines a HE ratio for each well-trained local model. Then, a model gradient magnitude-based mask is generated to determine which specific model parameters are protected by HE or DP in a local model. The main contributions of this paper are summarized as follows:

- We propose a novel adaptive HPFL scheme that jointly applies HE for some model parameters and DP for others. Moreover, a hybrid secure aggregation method is devised to address heterogeneous models aggregations.
- We mathematically formulate our considered problem as a training accuracy maximization problem. Then, an optimal closed-form HE ratio is derived, and we further design a model gradient magnitude-based mask to determine the specific model parameters by HE and DP.
- Experiments show that compared to DP solution only, our proposed HPFL can achieve a higher model accuracy up to 30%. Besides, with almost the same accuracy as HE-only solutions, the proposed method overall FL training time by approximately 80%.

## 2 Related Work

Privacy protection received widespread attention in recent years. In follows, we discuss the related works from three aspects, i.e., DP in FL, HE in FL , and Hybrid Schemes in FL.

**DP in FL:** DP provides privacy protection for FL by injecting noise during the training process. However, stronger privacy generally needs larger noise injection, which severely degrades model accuracy. To alleviate this issue, two methods were adopted, which are 1) adding noising to different model layers. For example, He et al. (2024) proposed ACS-FL, which performed adaptive perturbation based on the range radius of weights in each layer to reduce redundant noise. Zhao et al. (2025b) further introduced mutual information to allocate privacy budgets according to the weight correlation of the same layer between adjacent training rounds; 2) adding noise to devices based on their different privacy protection preferences. For instant, Qian et al. (2026) integrated DP with knowledge distillation, and a non-uniform sampling mechanism for clients to self-select privacy budgets is designed to add Gaussian noise to the gradients. Moreover, Yang et al. (2025) proposed DPGAN-IT, which applied DP-SGD to the discriminator's adversarial updates to achieve privacy-preserving generation of multimodal data. However, the above methods employed uniform noise injection across all parameters, without distinguishing between critical and non-critical parameters in the model, ultimately slowing down convergence and degrading model accuracy.

**HE in FL:** HE provided lossless-accuracy privacy protection for FL by performing aggregation operations directly on ciphertexts. However, this method consumes substantial computational resources, making the time overhead of a single training round far exceed that of plaintext FL. Typical works in this field are Jalali and Chen (2024) tackled the encryption computation efficiency problem by employing the CKKS scheme supplemented with a lightweight iTLS protocol to reduce per-encryption and transmission time. Zhao et al. (2023) addressed the risk of raw data leakage by applying threshold Paillier HE, so that clients only uploaded encrypted local model parameters. Yang et al. (2024) reduced computation energy consumption by adaptively adjusting each node's CPU frequency and the global aggregation period under energy and time constraints, Hussein et al. (2025) selected participating devices based on model update quality and computation efficiency, Liu et al. (2023) quantified the contribution of neighboring nodes and transmitted only high-value data to reduce traffic, and Hu and Li (2025) maximized the loss value of exposed model weights to enable gradients and model updates to screen out the parameters most critical for obfuscating training traces and defending against privacy attacks, thereby reducing the impact of encryption. Although these

HE-based schemes could maintain model accuracy, they need powerful computation resources on devices, which could be unrealistic in practice, as devices are usually heterogeneity in computations.

**Hybrid Schemes in FL:** Existing hybrid schemes can be classified into three categories. The first category adopted DP to raw data on the client side and HE to model parameters on the server side. For example, Sébert et al. (2022) designed a Poisson random quantization operator to convert continuous noisy gradients into discrete integers that could be processed by HE, and the server performed model aggregation only in the ciphertext domain, without ever accessing plaintext gradients. Ullah et al. (2024) adopted a strategy where the device added Laplacian noise to the original feature data locally and used quantum HE (QHE) to encrypt model parameters for transmission. Ahmar et al. (2025) proposed a temporally separated hybrid strategy, in which DP was used at the client side during training and Cheon–Kim–Kim–Song (CKKS) was applied at the server during aggregation. The second category applied HE and DP independently to a full model, where devices first added DP noise and then HE encryption. For instance, in order to alleviate communication congestion in 5G networks, Mahmood et al. (2025) first added DP noise to model updates, and HE encryption was then adopted. Jia et al. (2022) combined threshold Paillier HE with DP, where devices first added Gaussian noise and then used threshold HE encryption. To achieve stronger privacy protection Hu and Zhang (2026), the model weight gradients were first clipped and perturbed with noise, and then the noisy weights were encrypted using Paillier HE for integer ciphertext transmission. The third category aims to use DP or HE to different devices based on their computation capacities. Specifically, Negoya et al. (2025) achieved privacy protection by balancing accuracy and computational overhead. However, although the above hybrid schemes can achieve stronger privacy protection, they still could cause significant accuracy loss, which ultimately slows down convergence and degrades model accuracy.

In summary, existing research has not addressed the trade-off among privacy, computational complexity, and model training accuracy. Therefore, this paper proposes an adaptive HPFL method that uses HE to encrypt some model parameters while encrypting the remaining ones with differential privacy, so as to maximize the training performance of HPFL.

## 3 Preliminaries

### 3.1 FL

A typical FL system often consists of a parameter server (PS) and $K$ devices, denoted by the set $\mathcal{K} = \{1, 2, \ldots, K\}$. Each device $k$ holds a private local dataset $D_k = \{(x_j, y_j)\}_{j=1}^{\mathcal{D}_k}$, where $D_k = |\mathcal{D}_k|$ is the number of samples. The global model parameter vector is denoted as $w \in \mathbb{R}^z$, where $z$ is the total number of model parameters. where $f_k(w)$ is the local loss function of device $i$, $\lambda_k = |\mathcal{D}_k|/|D|$ is the weighted coefficient, $|D|$ is the total number of data samples across all devices.

To minimize $F(w)$, FL usually involves $T$ global aggregation rounds with the time duration in a round of $\Delta T$, where each round is indexed by $t \in \{1, 2, \ldots, T\}$. In each $t$, the server firstly broadcasts the current global model $w^t$ to all devices. Then, each device $k$ trains the received global model on its local data with $s$ numbers of local iterations. After that, the server will apply $w^{t+1} = \sum_{k=1}^{K} \lambda_k w_k^t$ to obtain an aggregated model $w^{t+1}$ right after the receipt of all well-trained local models from devices.

### 3.2 DP

In FL, DP is often leveraged to protect local well-trained models through adding noises. In this paper, the widely-used Renyi DP is adopted, which is defined as follows.

**Definition 1 ($(\alpha, \rho)$-Rényi DP) Fu et al. (2022):** A randomized mechanism $\mathcal{A}$ satisfies $(\alpha, \rho)$-RDP if for all adjacent datasets $D \sim D'$, it satisfies $D_\alpha(\mathcal{A}(D) \| \mathcal{A}(D')) \leq \rho$, where $\rho$ represents the privacy attenuation coefficient, and $D_\alpha(P\|Q) = \frac{1}{\alpha-1} \log \mathbb{E}_{x\sim Q} \left[ \left( \frac{P(x)}{Q(x)} \right)^\alpha \right]$ is the Rényi divergence of order $\alpha > 1$.

To achieve the RDP in **Definition 1**, we adopt the Gaussian mechanism to perturb the local model updates. Specifically, in each training round, device $k$ first calculates its local model update $\Delta \mathbf{w}_k^t =$

$\mathbf{w}_k^t - \mathbf{w}^{t-1}$. Then, we clip $L_2$-norm of $\Delta w_k^t$ to a threshold $C$ as $\tilde{\Delta}\mathbf{w}_k^t = \Delta\mathbf{w}_k^t \cdot \frac{C}{\max(1, \|\Delta\mathbf{w}_k^t\|_2 / C)}$, where clipped model update satisfies $\|\tilde{\Delta}\mathbf{w}_k^t\|_2 \leq C$ Wei et al. (2020).

### 3.3 HE AND THE CKKS SCHEME

In FL, HE is another privacy-preserving method, where the server can aggregate encrypted well-trained models without decryptions. Compared to other HE schemes, such as BGV/BFV Albrecht et al. (2018), the CKKS scheme can reduce the computation and communication overheads Cheon et al. (2017). The implementation of CKKS can be summarized as follows.

1) Key Generation: The secret key $(sk)$ is a short polynomial, typically sampled from a discrete Gaussian distribution. The public key $(pk)$ consists of two parts, denoted as $(a, b)$. Here, $a$ is a polynomial uniformly sampled from $\mathcal{R}_q$, where $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ is the ring of polynomials whose coefficients are reduced modulo $q$ (the symbol "/" represents the quotient operation, $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ is the integer polynomial ring), $b = a \cdot sk + e \bmod q$, where $e$ is a small error polynomial sampled from a discrete Gaussian distribution Cheon et al. (2017), and mod denotes the modulo operation.

2) Encoding and Encryption: Vector $\mathbf{m} \in \mathbb{R}^{N/2}$ is the original information, which is first encoded into a plaintext based on $\mathcal{R}$. Based on this encoded plaintext, each device will encrypt its local model parameter using the common public key, thereby generating a ciphertext $c = (c_0, c_1) \in \mathcal{R}^2$. It should be noted that due to the introduction of random masking and error terms, the ciphertext size will be significantly expanded compared to the original plaintext Cheon et al. (2017).

3) Decryption and Decoding: In the CKKS scheme, the ciphertext $\acute{m}$ is decrypted with the $sk$ as $\acute{m} = (c_0 + c_1 \cdot sk) \mod q$. Then, an inverse complex embedding map Cheon et al. (2017) is used to recover the original information from the encoded plaintext.

## 4 SYSTEM MODEL

In this section, we first introduce the design rationale of the proposed HPFL framework, and then mathematically formulate the research problem.

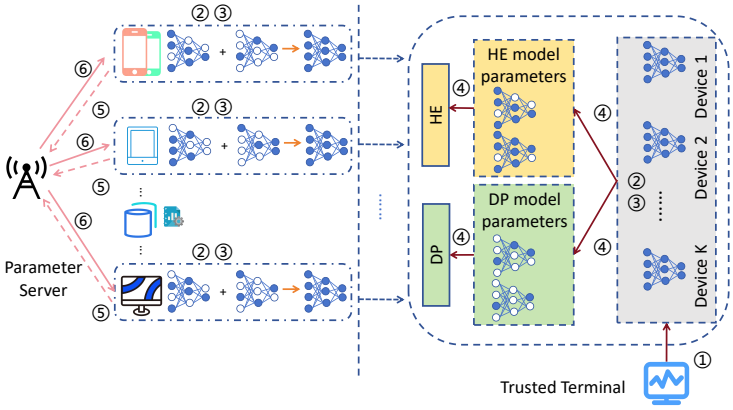



Figure 1: System model of HPFL.

### 4.1 FRAMEWORK OF PROPOSED HPFL

As illustrated in Fig. 1, the proposed HPFL consists of a parameter server (PS) and $K = |\mathcal{K}|$ devices, with index $k \in \mathcal{K}$. Similar to existing work, such as Bugshan et al. (2023), Hu et al. (2024), we assume that all devices are honest, and the PS is honest, but curious. The proposed HPFL scheme can be described with the following steps.

1) Model initialization (①): The trusted terminal generates a $pk$ and a $sk$. The $pk$ is distributed to all devices, while the $sk$ is securely injected into devices via a secure channel.

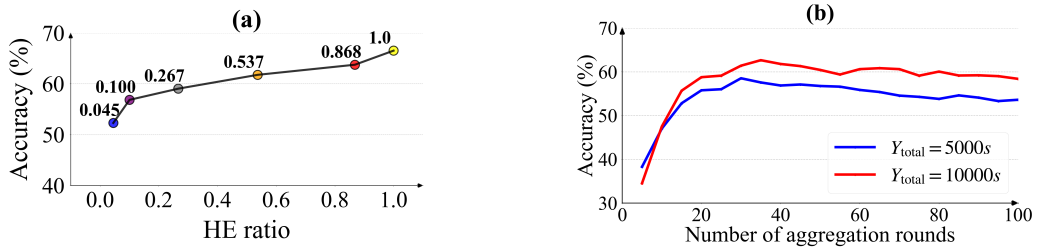



Figure 2: Experimental results on the training accuracy of the global model under different HE ratios and different $T$. Fig. 2(a) are obtained under $Y_{total} \in \{1000, 2000, 5000, 10000, 20000\}$, while the data in Fig. 2(b) are collected under $Y_{total} \in \{5000, 10000\}$.

2) Local Model Training (②): Each device $k$ initializes with the global model $w^t$, and performs $\tau$ steps of stochastic gradient descent (SGD) on its local data to obtain an updated local model $w_k^t$. Then, the total local training time for device $k$ in the $t$-th global training round is given by $Y_k^{t,C} = \frac{|\mathcal{D}_k| \cdot \Omega_k \cdot \tau}{C_k^t}$. Here, $\Omega_k$ represents the number of CPU cycles required by device $k$ to process one data sample, and $C_k^t$ denotes its computational capacity.

3) Determination of the HE ratio $h_{k,p}^t$ (③): For any device $k$ in each global training round $t$, the PS will determine a HE ratio of its local model based on its computational resource and communication capability. Then, the rest ratio of local model, denoted as $1 - h_{k,p}^t$, is protected by DP. So, we have

$$h_{k,p}^t + h_{k,d}^t = 1. \tag{1}$$

4) Evaluations of local model parameters (④): In the proposed HPFL, we can build a gradient magnitude-based binary mask $\mathcal{M}_k^t \in \{0,1\}^z$ based on the HE ratio $h_{k,p}^t$. Based on this mask $\mathcal{M}_k^t$, local model $\mathbf{w}_k^t$ can be split into two parts, i.e., $\mathbf{w}_{k,p}^t = \mathcal{M}_k^t \odot \mathbf{w}_k^t, \quad \mathbf{w}_{k,d}^t = \neg\mathcal{M}_k^t \odot \mathbf{w}_k^t$, where $\neg$ is the logical NOT. Note that $\mathbf{w}_{k,p}^t$ is called the HE-enabled model, which is encrypted by the CKKS, denoted as $[\mathbf{w}_{k,p}^t]^p = \mathsf{Enc}(\mathbf{w}_{k,p}^t)$; and $\mathbf{w}_{k,d}^t$ is termed the DP-enabled model, which is obtained through DP noise additions, denoted as $[\mathbf{w}_{k,d}^t]^d = \tilde{\Delta}\mathbf{w}_{k,d}^t + \mathcal{N}(0, (\sigma_k^t)^2 \cdot C^2 \cdot \mathbf{I})$.

We will detail the determination of $\mathcal{M}_k^t$ in Section V-B. $\mathcal{M}_k^t$. Since the above procedures involve computational overhead, the overall computation time for encryption and DP noise additions is

$$Y_k^{t,\mathrm{E}} = \frac{h_{k,p}^t \cdot z \cdot \alpha_{\mathrm{HE}} \cdot \Omega_k + (1 - h_{k,p}^t) \cdot z \cdot \alpha_{\mathrm{DP}} \cdot \Omega_k}{C_k^t}, \tag{2}$$

where $\alpha_{\mathrm{HE}}$ and $\alpha_{\mathrm{DP}}$ denote the average computational time per parameter for HE and DP operations, respectively. Moreover, by considering that the HE-enabled model will be expanded with a coefficient $s_{HE}$ Cheon et al. (2017), the local model upload time of device $k$ is

$$Y_k^{t,U} = \frac{h_{k,p}^t \cdot z \cdot s_{HE} + (1 - h_{k,p}^t) \cdot z}{R_k^t}, \tag{3}$$

where $R_k^t = B_k^t \log_2(1 + G_k^t)$ is the uplink transmission rate with the allocated bandwidth $B_k^t$ and the Signal-to-Noise Ratio (SNR) $G_k^t$.

Similar to most works Jalali and Chen (2024); Yang et al. (2024), we also omit the downlink transmission time from the PS to devices, as the corresponding bandwidth is much larger than that of the upload link Hayek et al. (2025). Thus, the total time for device $k$ in the $t$-th global round is the sum of local training, privacy-preserving computation, and upload time, which is

$$Y_k^t = Y_k^{t,C} + Y_k^{t,E} + Y_k^{t,U}. \tag{4}$$

To ensure that model training and uploading are completed within the duration of each global aggregation round, i.e., $\Delta T$, we impose the following constraint.

$$\max_{k \in \mathcal{K}} Y_k^t \leq \Delta T. \tag{5}$$

5) Hybrid Secure model aggregations (⑤): Upon the receipt of all well-trained local protected models from devices, the PS performs the following weighted model aggregation.

$$[w_p^{t+1}]^p = \bigoplus_{k=1}^{K}(\lambda_k \cdot [w_{k,p}^t]^p), [w_d^{t+1}]^d = \sum_{k=1}^{K} \lambda_k [w_{k,d}^t]^d, \tag{6}$$

where $\bigoplus$ denotes homomorphic ciphertext addition.

6) Global model distribution and combination (⑥): The PS broadcasts the aggregated models $[w_p^{t+1}]^p$ and $[w_d^{t+1}]^d$ to devices. Each device will locally decrypt HE-enabled global model as $w_p^{t+1} = \mathsf{Dec}([w_p^{t+1}]^p)$. After that, at each device side, this decrypted HE-enabled model will be merged with DP-enabled model as $w_k^{t+1} = \mathcal{M}_k^t \odot w_p^{t+1} + \neg\mathcal{M}_k^t \odot [w_d^{t+1}]^d$. Note that the above process will continue until a predefined number of global aggregation rounds $T^*$ is reached. As stated in **Theorem** 2 in Wei et al. (2020), there exists an optimal number of global aggregation rounds in FL with privacy protection. For the determination of $T^*$, please refer to Wei et al. (2020). Moreover, in this paper, in each global aggregation round $t$, the HE ratio $h_{k,p}^t$ and the mask $\mathcal{M}_k^t$ will be determined as follows.

### 4.2 Problem Formulation

The objective of this paper is to train a global model with higher model accuracy. According to **Theorem** 2 in Wei et al. (2020), model accuracy is positively related to the privacy budget $\varepsilon$, i.e., the larger $\epsilon$, the higher model accuracy. Moreover, it is shown in Fig. 2 (a) that the model training accuracy improves as the HE ratio increases $h_{k,p}^t$. In addition, Fig. 2(b) shows that there exists an optimal number of global aggregation rounds $T^*$ to obtain the highest model accuracy. Based on these results, we formulate a global accuracy model that is jointly determined by the privacy budget $\varepsilon$, the HE ratio $h_{k,p}^t$, and the number of aggregation rounds.

$$Acc(w^T) = \Phi_0 - \underbrace{\Phi_1(\frac{T^*}{\varepsilon} + \frac{T^*}{\varepsilon^2})(1 - h_{k,p}^t)}_{A_1} - \underbrace{\Phi_2(\frac{Y_{\text{total}}}{Y_k^t} - T^*)}_{A_2}, \tag{7}$$

where $Y_{total} = T^* \cdot \Delta T$ denotes the total training time across all rounds, and $\Phi_0$ represents the highest achievable accuracy without any privacy mechanism, $\Phi_1$ quantifies the accuracy degradation because of DP (i.e., when $h_{k,d}^t > 0$), $\Phi_2$ denotes the average utility loss per aggregation round. Hence, in this paper, we aim to maximize the model training performance by optimizing the ratio between HE and DP under privacy and latency constraints, which can be formally formulated as

$$\mathcal{P}1: \quad \max_{h_{k,p}^t} \mathbb{E}[\text{Acc}(w^{T^*})] \quad \textit{s.t.}\ (1), (5).$$

## 5 The Proposed Optimal Solution

In this section, to solve $\mathcal{P}1$, we first convert it into a one-dimensional unconstrained non-convex optimization problem, and derive an optimal HE ratio $h_{k,p}^{t,*}$. Then, based on $h_{k,p}^{t,*}$, we determine the gradient magnitude-based mask $\mathcal{M}_k^t$.

### 5.1 Optimal HE Ratio $h_{k,p}^{t,*}$

To solve $\mathcal{P}1$, we first introduce the following lemma, which simplifies the original constrained optimization problem into a one-dimensional optimization problem.

**Lemma 1.** *Under the constraint of time duration within the global round $\Delta T$, the optimal HE ratio $h_{k,p}^{t,*}$ for any device $k$ satisfies $h_{k,p}^{t,*}(\Delta T) = \min(1, \frac{\Delta T - b_k^t}{a_k^t})$, where $a_k^t$ and $b_k^t$ are positive constants, which are $a_k^t = z \cdot \left(\frac{\alpha_{\text{HE}} - \alpha_{\text{DP}}}{C_k^t} \cdot \Omega_k + \frac{s_{\text{HE}} - 1}{R_k^t}\right) > 0$, and $b_k^t = z \cdot \left(\frac{\alpha_{\text{DP}}}{C_k^t} \cdot \Omega_k + \frac{1}{R_k^t}\right) + \frac{|\mathcal{D}_k| \cdot \Omega_k \cdot \tau}{C_k^t} > 0$.*

*Proof:* The proof can be found in the Appendix A. ■

Based on **Lemma** 1, we have the following reformulated objective function

$$E[\mathrm{Acc}(w^{T^*})] = \Phi_0 - \Phi_1(\frac{T^*}{\varepsilon} + \frac{T^*}{\varepsilon^2})(1 - \min(1, \frac{\Delta T - b_k^t}{a_k^t})) - \Phi_2(\frac{Y_{\text{total}}}{Y_k^t} - T). \tag{8}$$

Moreover, based on (8), we can define an average HE ratio function as $\bar{h}(\Delta T) = (\frac{T^*}{\varepsilon} + \frac{T^*}{\varepsilon^2}) \min(1, \frac{\Delta T - b_k^t}{a_k^t})$. It should be noted that $\bar{h}(\Delta T)$ is a piecewise linear convex increasing function. Moreover, as demonstrated in Lemma 1, we should have $Y_k^t = \Delta T$. Then, the $A_2$ term in (1 can be changed to $\frac{Y_{total}}{\Delta T} - T^* = T^*(\frac{Y_{total}}{T^* \Delta T} - 1)$. Therefore, the original constrained optimization problem $\mathcal{P}1$ can be equivalently transformed into an unconstrained unimodal optimization problem.

$$\Delta T^* = \max_{\Delta T} f(\Delta T) \triangleq \bar{h}(\Delta T) - \psi \cdot (\frac{Y_{total}}{T^* \Delta T} - 1), \tag{9}$$

where $\psi$ is a trade-off factor used to balance between accuracy and efficiency. Since $\bar{h}(\Delta T)$ is a piecewise linear convex increasing function and $\frac{1}{\Delta T}$ is a convex decreasing function, their difference, i.e., $f(\Delta T)$, is therefore unimodal. This property guarantees that $f(\Delta T)$ possesses a unique global maximum point. Once $\Delta T^*$ is obtained, the closed-form optimal HE ratio for each device can be calculated accordingly.

$$h_k^{t,*} = \frac{(\Delta T^* C_k^t - |\mathcal{D}_k|\Omega_k \tau - z\alpha_{DP}\Omega_k) R_k^t - z C_k^t}{z(\alpha_{HE} - \alpha_{DP})\Omega_k R_k^t + z(s_{HE} - 1) C_k^t}. \tag{10}$$

Finally, by considering the constraint (5), the optimal HE ratio is changed to $h_{k,p}^{t,*} = \min(1, h_k^{t,*})$.

### 5.2 Determinations of the Mask $\mathcal{M}_k^t$

Based on the optimal HE ratio $h_{k,p}^{t,*}$, we still need to choose a number of specific model parameters that are protected by the CKKS method, while others are preserved by the DP method. To this end, we propose a dynamic gradient magnitude-based binary mask scheme.

The main idea of this gradient magnitude-based binary mask method is that each local model parameter is firstly evaluated by an importance score with the exponential moving average (EMA) over its historical gradient magnitudes. Then, a HE-enabled model can be generated according to the optimal HE ratio. Specifically, once the completion of local updating in the $t$-th global round, device $k$ can obtain its current gradient $\nabla f_k(w^t) \in \mathbb{R}^z$ and then update its importance score $v_k^t[i]$ for each model parameter $i \in \{1, 2, \ldots, z\}$ as

$$v_k^t[i] = \gamma v_k^{t-1}[i] + (1 - \gamma)(\nabla f_k(w^t)_i)^2, \tag{11}$$

where $\gamma \in (0, 1)$ is the momentum coefficient, and $v_k^0[i] = 0$. Then, based on $v_k^t[i]$, we define the average gradient energy (AGE) at $t$-th global round as $s_k^t[i] = v_k^t[i]$. Note that a higher value of $s_k^t[i]$ indicates more contributions to the global model. To determine the mask $\mathcal{M}_k^t$, we first compute the quantile of the AGE $s_k^t[i]$ at the optimal HE ratio, i.e., $h_{k,p}^{t,*}$, as Tenison et al. (2023) $s_k^{t,*} = \text{quantile}(s_k^t, h_{k,p}^{t,*})$. Then, based on this $s_k^{t,*}$, we apply the following criterion to generate a binary mask $\mathcal{M}_k^t$.

$$\mathcal{M}_k^t[i] = \begin{cases} 1, & \text{if } s_k^t[i] \geq s_k^{t,*}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Normally, with this mask $\mathcal{M}_k^t$, the local model $w_k^t$ can be separated into two parts based on $\mathbf{w}_{k,p}^t = \mathcal{M}_k^t \odot \mathbf{w}_k^t, \quad \mathbf{w}_{k,d}^t = \neg\mathcal{M}_k^t \odot \mathbf{w}_k^t$. For more clarity, we summarize the procedures of the proposed HPFL in **Algorithm 1** of **Appendix A**.

## 6 Experimental Setup

In this section, we conduct experiments on real-world datasets to validate the effectiveness of HPFL.

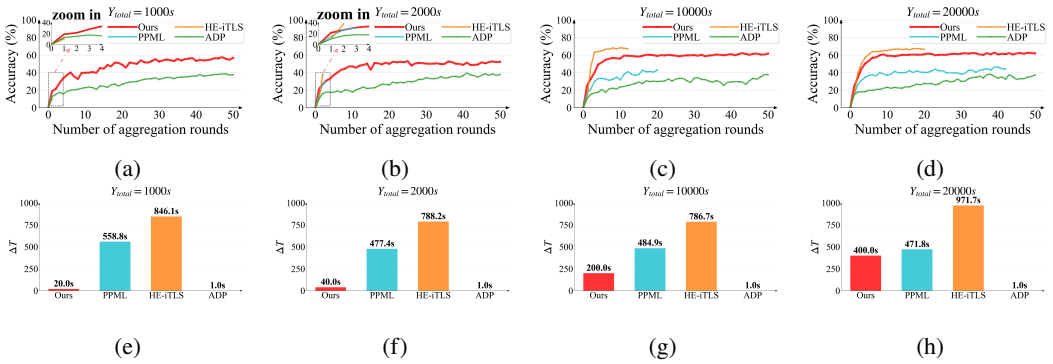



Figure 3: Training accuracy and time duration per training round with different $Y_{total}$.

**Datasets and Models:** Following Zhao et al. (2025b), Hussein et al. (2025), we also employ the FMNIST, CIFAR-10, and STL-10 datasets, with convolutional neural networks (CNNs). For the detailed CNN architecture, please refer to the appendix C.2.4.

**Parameter Settings:** We conduct experiments on Python 3.9, PyTorch 2.4.1, cuda 12.4 and NVIDIA GeForce RTX 5060. The central processing unit (CPU) frequency is uniformly sampled from the interval $[1.0, 4.2]$ GHz, the upload link bandwidth is uniformly sampled from $[0.5, 35.0]$ MHz, the signal-to-noise ratio (SNR) is uniformly sampled from $[3.0, 32.0]$ dB, and the number of CPU cycles required to process a single sample is uniformly sampled from $[800, 2500]$. The privacy budget is set to $\varepsilon = 70$ Li et al. (2024); Huang et al. (2024). $T$ and$\tau$ are set to 50 and 5, respectively. The learning rate is set to 0.01 for the FMNIST datasets, and 0.05 for the others.

To further validate the effectiveness of our proposed method, we compare it with FedAvg McMahan et al. (2023) and the following two state-of-the-art privacy-preserving FL schemes.

- **ADP** Tariq et al. (2026): An adaptive DP FL scheme that dynamically adjusts the injected noise in different global rounds.
- **HE-iTLS** Wang et al. (2025): A HE-based FL framework that employs the CKKS scheme.
- **PPML** Negoya et al. (2025): Each device autonomously selects either HE or DP based on its own computational capability and privacy protection requirements.
- **KDH** Hu and Zhang (2026): The gradients of the global mode is first clipped and perturbed with DP noise. Then, the perturbed model parameters are encrypted with Paillier HE.
- **Our-FRH**: The HE ratio is fixed at 50% in each global round, with the remaining 50% of parameters protected by DP.
- **Our-RM**: The HE ratio $h_{k,p}^t$ is determined by our proposed method, while a random binary mask scheme is adopted to generate the mask $\mathcal{M}_k^t$.
- **Our-FM**: The HE ratio $h_{k,p}^t$ is obtained based on our method, but the mask is determined only in the first global round and remains fixed throughout the entire training process.

### 6.1 Evaluations of Model Training Accuracy

Fig. 3(a)-(d) show the training accuracy of the global model of HPFL, ADP, HE-iTLS, and PPML on dataset CIFAR-10 under 10 devices with different total training times $Y_{\text{total}}$. It can be observed that our method achieves much higher accuracy than ADP and PPML. This is because the DP noise is added to all devices in ADP adds, while PPML applies DP noise to a fraction of devices that are computationally weak. However, when the total training time is large (e.g., $Y_{total} = 2 \times 10^4 s$), the accuracy of our method is slightly lower than that of the HE-only scheme HE-iTLS. This is because our method still applies both DP and HE to balance the overall computation overhead. On the other hand, as shown in Fig. 3(e)-(h), the average total time per global training round of HE-iTLS and PPML is much higher than that of our proposed method, which validates that our approach is computational efficiency.

Table 1: Test accuracy (%) on different datasets.

| $Y_{total}$ ($10^3$s) | FMNIST | | | | CIFAR-10 | | | | STL-10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.100 | 0.200 | 1 | 2 | 1 | 2 | 10 | 20 | 1 | 2 | 10 | 20 |
| FedAvg | **84.20** | **84.10** | **82.80** | **81.26** | **68.88** | **67.54** | **66.92** | **66.53** | **59.35** | **60.12** | **60.34** | **60.08** |
| HE-iTLS | 82.29 | 82.46 | 83.43 | 83.34 | 10.01 | 37.91 | 60.65 | 67.46 | 22.31 | 47.15 | 58.16 | 60.65 |
| ADP | 72.43 | 77.94 | 74.61 | 74.89 | 37.52 | 38.36 | 38.32 | 39.34 | 32.01 | 31.40 | 31.95 | 33.10 |
| PPML | 61.71 | 68.53 | 79.68 | 79.50 | 25.42 | 29.91 | 42.76 | 46.34 | 23.78 | 31.76 | 42.02 | 44.46 |
| **Ours** | **80.33** | **83.52** | **81.88** | **81.93** | **51.11** | **57.59** | **62.24** | **63.84** | **53.97** | **55.32** | **58.23** | **59.02** |
| Our-FRH | 57.71 | 76.05 | 82.79 | 82.33 | 33.54 | 39.81 | 48.22 | 56.32 | 36.28 | 38.42 | 53.52 | 57.11 |
| Our-RM | 77.19 | 79.70 | 82.93 | 82.50 | 28.62 | 41.03 | 55.02 | 57.92 | 22.08 | 27.26 | 49.00 | 56.73 |
| Our-FM | 75.00 | 77.17 | 81.26 | 82.20 | 36.00 | 47.78 | 52.10 | 63.89 | 31.85 | 35.20 | 54.86 | 58.90 |

Table 1 presents the test accuracies of eight FL methods on four datasets under different total training times $Y_{total}$ with 10 clients. Note that since the FedAvg method has no privacy protections, the achieved accuracy should be the highest, and it is used as the baseline in this paper. As shown in this table, regardless of $Y_{total}$, our proposed method always outperforms ADP and PPML on all datasets and has a marginal performance gap with FedAvg. For instance, on CIFAR-10 with $Y_{total} = 10^4 s$, our method achieves 62.24% accuracy, while FedAvg attains 66.92%. Compared with HE-iTLS, our method exhibits markedly superior performance under small $Y_{total}$ scenarios. This is because when the time budget $Y_{total}$ is small, HE-iTLS cannot complete the entire encryption process due to the heavy computational overhead of the HE scheme; in contrast, our method can locally finish the privacy protection operations within such a short time duration. Moreover, the hybrid scheme PPML performs much worse on complex datasets, such as CIFAR-10 and STL-10, than that on datasets FMNIST. This is because PPML only chooses the privacy protection scheme (either DP or HE) based on devices' computation capacities, rather than based on the importance of the model parameters, which could lead to slow convergence.

### 6.2 Ablation Studies

Table 1 also shows the ablation studies, which further confirms the necessity of each proposed component. We can observe that 1) The test accuracy by Our-FRH is inferior to our proposed, which means that the HE ratio for a global model must be considered; 2) Although Our-RM adopts the same strategy to get the HE ratio in a model, its testing accuracy is still lower than our proposed, which demonstrates the necessity of choosing the "right" model parameters for HE or DP; 3) Due to device heterogeneity, such as differences in computational power and communication bandwidth, Our-FM demonstrates that the gradient magnitude-based mask, i.e., $\mathcal{M}_k^t$, should be updated throughout the FL training process. These results collectively validate the effectiveness of our overall design, which is to dynamically adjust the HE ratio. In addition, due to space limitations, the detailed privacy analysis is provided in **Appendix B**, the additional experimental setup, simulation results, defenses under different attack models, and extended model experiments are provided in **Appendix C**.

## 7 Conclusion

This paper proposes a hybrid FL framework that combines HE and DP. The framework dynamically partitions parameters based on gradients. A closed-form solution for the optimal encryption ratio is derived through optimization. Experiments show that the framework significantly outperforms DP only on different datasets, and can achieve comparable accuracy to HE method only. Future work may extend to consider more optimization factors, such as device selection or dynamic resource allocation, under asynchronous learning.

## 8 AI USE STATEMENT

In this work, we used generative AI tools to assist with paper polish and code writing. We did not use generative AI tools for paper content generation. Additionally, we have reviewed all AI-assisted work. For example, LLM-generated code was verified and tested for correctness by at least 2 authors. We take responsibility for the final content of this work, including text, claims, or content produced with the aid of generative AI.

## 9 ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. Our research does not involve human subjects, animal experiments, or the collection of sensitive personal data. The datasets used are publicly available benchmarks. We have considered the potential societal impacts and do not foresee risks of misuse, privacy violations, or discrimination.

## 10 REPRODUCIBILITY STATEMENT

All datasets used in this work are publicly available. The datasets, model hyperparameters, computational resources, and other experimental settings are described in the Experimental Setup section. The complete algorithm is provided in Appendix A, and the complete proofs are provided in Appendix B. We believe that these materials are sufficient to reproduce the main findings of this paper.

## REFERENCES

Jawaad Ahmar, Iqra Batool, Mostafa M. Fouda, Mohamed I. Ibrahem, and Zubair Md Fadlullah. 2025. Privacy-Utility Trade-offs in Federated Learning for 6G Networks: A Systematic Evaluation of Software-based Privacy Mechanisms. In *2025 IEEE Smart World Congress (SWC)*. IEEE, Calgary, Canada, 1914–1919. doi:10.1109/SWC65939.2025.00296

Martin Albrecht, Melissa Chase, Hao Chen, et al. 2018. *Homomorphic Encryption Standard*. Technical Report. HomomorphicEncryption.org, Toronto, Canada.

Neda Bugshan, Ibrahim Khalil, Mohammad Saidur Rahman, Mohammed Atiquzzaman, Xun Yi, and Shahriar Badsha. 2023. Toward Trustworthy and Privacy-Preserving Federated Deep Learning Service Framework for Industrial Internet of Things. *IEEE Transactions on Industrial Informatics* 19, 2 (2023), 1535–1547. doi:10.1109/TII.2022.3209200

Jung Hee Cheon, Andrey Kim, et al. 2017. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*. Springer, Springer, Hong Kong, China, 409–437.

Yicheng Di, Jiansong Fan, et al. 2025. Global Perception Federated Recommender System for Click-Through Rate Prediction. In *2025 IEEE ICME*. IEEE, Nantes, France, 1–6. doi:10.1109/ICME59968.2025.11209581

Jie Fu, Zhili Chen, and Xiao Han. 2022. Adap DP-FL: Differentially Private Federated Learning with Adaptive Noise. In *2022 IEEE TrustCom*. IEEE, Wuhan, China, 656–663. doi:10.1109/TrustCom56396.2022.00094

Robert J. Hayek, Joaquin Chung, et al. 2025. Federated Learning over 5G, WiFi, and Ethernet: Measurements and Evaluation. arXiv:2504.04678 [cs.NI] https://arxiv.org/abs/2504.04678

Zaobo He, Lintao Wang, and Zhipeng Cai. 2024. Clustered Federated Learning With Adaptive Local Differential Privacy on Heterogeneous IoT Data. *IEEE Internet Things J.* 11, 1 (2024), 137–146. doi:10.1109/JIOT.2023.3299947

Chenghao Hu and Baochun Li. 2025. MaskCrypt: Federated Learning With Selective Homomorphic Encryption. *IEEE Transactions on Dependable and Secure Computing* 22, 1 (2025), 221–233. doi:10.1109/TDSC.2024.3392424

Jiahui Hu, Zhibo Wang, Yongsheng Shen, et al. 2024. Shield Against Gradient Leakage Attacks: Adaptive Privacy-Preserving Federated Learning. *IEEE/ACM Transactions on Networking* 32, 2 (2024), 1407–1422. doi:10.1109/TNET.2023.3317870

Junxu Hu and Ying Zhang. 2026. HHB-FL: Privacy-Preserving Federated Learning via Hierarchical Encryption and Blockchain for Maritime IoT. *IEEE Internet of Things Journal* 13, 5 (2026), 8436–8451. doi:10.1109/JIOT.2025.3641841

Guangjing Huang, Qiong Wu, Peng Sun, Qian Ma, and Xu Chen. 2024. Collaboration in Federated Learning With Differential Privacy: A Stackelberg Game Analysis. *IEEE Transactions on Parallel and Distributed Systems* 35, 3 (2024), 455–469. doi:10.1109/TPDS.2024.3354713

Suzanne Hussein, Jingling Yuan, Zhishu Shen, et al. 2025. FedFlex: Privacy-Aware Homomorphic Encryption Federated Learning Incorporating Dual-Factor Thompson Sampling. *IEEE Internet Things J.* 12, 15 (2025), 30098–30113. doi:10.1109/JIOT.2025.3569657

Nasir Ahmad Jalali and Hongsong Chen. 2024. Federated Learning Security and Privacy-Preserving Algorithm and Experiments Research Under Internet of Things Critical Infrastructure. *Tsinghua Sci. Technol.* 29, 2 (2024), 400–414. doi:10.26599/TST.2023.9010007

Bin Jia, Xiaosong Zhang, Jiewen Liu, Yang Zhang, Ke Huang, and Yongquan Liang. 2022. Blockchain-Enabled Federated Learning Data Protection Aggregation Scheme With Differential Privacy and Homomorphic Encryption in IIoT. *IEEE Transactions on Industrial Informatics* 18, 6 (2022), 4049–4058. doi:10.1109/TII.2021.3085960

Yanan Li, Shusen Yang, Xuebin Ren, Liang Shi, and Cong Zhao. 2024. Multi-Stage Asynchronous Federated Learning With Adaptive Differential Privacy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 2 (2024), 1243–1256. doi:10.1109/TPAMI.2023.3332428

Tao Liu, Peng Li, Yu Gu, et al. 2023. S-Glint: Secure Federated Graph Learning With Traffic Throttling and Flow Scheduling. *IEEE Trans Green Commun Netw.* 7, 2 (2023), 894–903. doi:10.1109/TGCN.2022.3187149

Abuthar Mahmood, Sasikumar Gurumoorthy, Meghana A, Yogesh Ramaswamy, and Gayathiri B. 2025. Privacy Preserving Analytics in IoT Systems using Federated Learning with Homomorphic Encryption and Differential Privacy. In *2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, Ballari, Karnataka, India, 1–5. doi:10.1109/ICDCECE65353.2025.11035884

H. Brendan McMahan, Eider Moore, et al. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629 [cs.LG] https://arxiv.org/abs/1602.05629

Ilya Mironov. 2017. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, IEEE Computer Society, Santa Barbara, CA, USA, 263–275. doi:10.1109/csf.2017.11

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 739–753. doi:10.1109/SP.2019.00065

Yusaku Negoya, Feifei Cui, Zilong Zhang, Miao Pan, Tomoaki Ohtsuki, and Aohan Li. 2025. A Privacy-Preserving Federated Learning Method with Homomorphic Encryption in Omics Data. arXiv:2511.06064 [cs.CR] https://arxiv.org/abs/2511.06064

Wenjun Qian, Wei Yu, Xin Ge, Cong Li, Lianyuan Li, and Zheng Li. 2026. PDP-FedKD: Personalized Differential Privacy With Adaptive Budget Selection in Heterogeneous Federated Learning. *IEEE Signal Processing Letters* 33 (2026), 166–170. doi:10.1109/LSP.2025.3631383

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. arXiv:1806.01246 [cs.CR]

Arnaud Grivet Sébert, Renaud Sirdey, Oana Stan, and Cédric Gouy-Pailler. 2022. Protecting Data from all Parties: Combining FHE and DP in Federated Learning. arXiv:2205.04330 [cs.CR] https://arxiv.org/abs/2205.04330

Omer Tariq, Muhammad Bilal Akram Dastagir, and Dongsoo Han. 2026. ADP-QFed: Privacy-Preserving Quantized Federated Learning for Intelligent Edge Sensing IoT Systems. *IEEE Internet of Things Journal* 13, 5 (2026), 8335–8347. doi:10.1109/JIOT.2025.3634475

Irene Tenison, Sai Aravind Sreeramadas, et al. 2023. Gradient Masked Averaging for Federated Learning. arXiv:2201.11986 [cs.LG] https://arxiv.org/abs/2201.11986

Shoaib Ullah, Madam Hussain Shah, and Adeel Anjum. 2024. Quantum Enhanced Federated Learning with Differential Privacy. In *2024 International Conference on Frontiers of Information Technology (FIT)*. IEEE, Islamabad, Pakistan, 1–6. doi:10.1109/FIT63703.2024.10838402

Yehui Wang, Baoxian Zhang, Jinkai Zhang, and Cheng Li. 2025. Efficient Privacy-Preserving Federated Learning via Homomorphic Encryption-Enabled Over-the-Air Computation. *IEEE Transactions on Mobile Computing* 24, 12 (2025), 13743–13759. doi:10.1109/TMC.2025.3610887

Kang Wei, Jun Li, and Ming. Ding. 2020. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* 15 (2020), 3454–3469. doi:10.1109/TIFS.2020.2988575

Jiming Yang, Xu Wang, Yi Jin, et al. 2025. Generative Adversarial Network-based Image and Tabular Data Generation with Differential Privacy. In *2025 IEEE ICME*. IEEE, Nantes, France, 1–6. doi:10.1109/ICME59968.2025.11209493

Ruizhe Yang, Tonghui Zhao, F. Richard Yu, et al. 2024. Blockchain-Based Federated Learning With Enhanced Privacy and Security Using Homomorphic Encryption and Reputation. *IEEE Internet Things J.* 11, 12 (2024), 21674–21688. doi:10.1109/JIOT.2024.3379395

Bowen Zhao, Ximeng Liu, Wei-Neng Chen, et al. 2023. CrowdFL: Privacy-Preserving Mobile Crowdsensing System Via Federated Learning. *IEEE Trans. Mob. Comput.* 22, 8 (2023), 4607–4619. doi:10.1109/TMC.2022.3157603

Xu Zhao, Gang Li, and Jun Cai. 2025a. A Novel Differential Privacy Federated Learning Framework: An Adaptive Budget Allocation and Reversion Method. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, Nantes, France, 1–6. doi:10.1109/ICME59968.2025.11209553

Xijun Zhao, Gang Li, and Hongming Chen. 2025b. MIPP-FL: Personalized Layer Privacy Protection Federated Learning Based on Mutual Information. In *2025 IEEE ICME*. IEEE, Nantes, France, 1–6. doi:10.1109/ICME59968.2025.11209201

# Appendix

## A Proof

*Proof:* From (4) and the ciphertext expansion property of CKKS Cheon et al. (2017), the total time consumption of device $k$, i.e., $Y_k^t$, in $t$-th global round can be expressed as a linear function of $h_{k,p}^t$, which is

$$Y_k^t(h_{k,p}^t) = \frac{|\mathcal{D}_k| \cdot C_t \cdot \tau}{C_k^t} + \frac{h_{k,p}^t \cdot z \cdot \alpha_{HE} \cdot \Omega_k}{C_k^t} + \frac{(1-h_{k,p}^t) \cdot z \cdot \alpha_{DP} \cdot \Omega_k}{C_k^t} + \frac{h_{k,p}^t \cdot z \cdot s_{HE} + (1-h_{k,p}^t) \cdot z}{R_k^t} \triangleq a_k^t \cdot h_{k,p}^t + b_k^t, \tag{A.1}$$

where $a_k^t > 0$ and $b_k^t > 0$. Therefore, $Y_k^t$ increases monotonically with $h_{k,p}^t$. From (7), the expected accuracy $\mathbb{E}[\mathrm{Acc}(w^T)]$ depends on $h_{k,p}^t$ through $A_1$ and $A_2$. The first term $A_1$ increases linearly with

**Algorithm 1:** HPFL Algorithm

**Input:** Device set $\mathcal{K}$, local datasets $\{\mathcal{D}_k\}_{k=1}^{K}$, initial model $w^0$, maximum number of aggregation rounds $T^*$, number of local training steps $\tau$, delay constraint $\Delta T$, privacy budget $\epsilon$, clipping threshold $C$, EMA coefficient $\gamma$.

**Output:** Trained global model $w^{T^*}$.

The trusted terminal generates the CKKS key pair $(pk, sk)$, distributes $pk$ to all devices, and securely injects $sk$ into the device side. The server initializes the global model $w^0$.

**for** $t = 1, 2, \ldots, T^*$ **do**

- The server broadcasts the current global model $w^{t-1}$ to all devices.
- **foreach** *device* $k \in \mathcal{K}$ *executes in parallel* **do**
  - Perform $\tau$ steps of local SGD on $w^{t-1}$ based on $\mathcal{D}_k$ to obtain $w_k^t$.
  - According to the device's computational capability $C_k^t$, upload rate $R_k^t$, and delay constraint $\Delta T$, compute the optimal HE ratio $h_{k,p}^{t,*}$, and set $h_{k,d}^{t,*} = 1 - h_{k,p}^{t,*}$.
  - Compute the current gradient $\nabla f_k(w^t)$, and update the historical gradient importance of each parameter according to (11).
  - Based on $h_{k,p}^{t,*}$, compute the importance threshold and generate the binary mask $\mathcal{M}_k^t$.
  - Partition the local model into the HE part and the DP part according to (4)
  - Perform CKKS encryption on $w_{k,p}^t$ to obtain $[w_{k,p}^t]_p$.
  - Perform norm clipping and add Gaussian noise to $w_{k,d}^t$ to obtain $[w_{k,d}^t]_d$.
  - Upload $([w_{k,p}^t]_p, [w_{k,d}^t]_d$ to the server.
- The server separately performs weighted aggregation for the HE and DP parameters according to $w_k^{t+1} = \mathcal{M}_k^t \odot w_p^{t+1} + \neg\mathcal{M}_k^t \odot [w_d^{t+1}]^d$, and broadcasts $([w_p^{t+1}]^p, [w_d^{t+1}]^d)$ to devices.
- Each device decrypts the HE aggregated model parameters, and performs mask merging according to $\mathbf{w}_{k,p}^t = \mathcal{M}_k^t \odot \mathbf{w}_k^t, \quad \mathbf{w}_{k,d}^t = \neg\mathcal{M}_k^t \odot \mathbf{w}_k^t$ to obtain the next-round model.

**return** $w^{T^*}$;

$h_{k,p}^t$, as higher HE ratio reduces the cumulative DP noise. The second term $A_2$ decreases linearly with $h_{k,p}^t$, because $Y_k^t$ is a linear increasing function of $h_{k,p}^t$. Besides, a larger $Y_k^t$ reduces the number of communication rounds within the total time $Y_{\text{total}}$. Furthermore, by considering the constraint (5), each device must satisfy $a_k^t \cdot h_{k,p}^t + b_k^t \leq \Delta T$. Then, we have

$$h_{k,p}^t \leq \frac{\Delta T - b_k^t}{a_k^t}. \tag{A.2}$$

Considering that $h_{k,p}^t \in [0, 1]$, the optimal HE ratio for device $k$ is given by $h_{k,p}^{t,*}(\Delta T) = \min(1, \frac{\Delta T - b_k^t}{a_k^t})$. Note that with $h_{k,p}^{t,*}(\Delta T)$, the time consumption of all devices equals $\Delta T$ exactly, which means the constraint (5) holds. This completes the proof. ■

# B HPFL Algorithm

# C Analysis of Privacy and Security

## C.1 Privacy Analysis by HE in HPFL

In the proposed HPFL, the CKKS HE scheme is adopted, whose security strictly depends on the ability to solve the post-quantum hard problem of Ring Learning with Errors (RLWE) Cheon et al. (2017). For an adversary without the decryption key, to solve this RLWE is impossible, as it is a well-recognized core hard problem in the field of lattice-based cryptography Cheon et al. (2017). Moreover, with both classical and quantum computing models, there is no efficient solving algorithm with polynomial time complexity for this problem. Therefore, the PS without a key cannot recover any plaintext information from the ciphertexts. Besides, according to the properties of HE Cheon

et al. (2017), the PS doesn't need to know plaintexts for model aggregations. Therefore, the PS cannot decrypt the model information by HE method, but can actually aggregate those model parameters without performance loss.

### C.2 Privacy Analysis by DP in HPFL

Since the proposed HPFL framework also adopts the $(\alpha, \rho)$-Rényi differential privacy framework for privacy protection, based on Mironov (2017), it also satisfies $(\varepsilon, \delta)$-differential privacy, where $\delta$ is the noise scale. Moreover, according to the definition of global sensitivity and the Gaussian mechanism in DP Wei et al. (2020), the scale of the noise must be calibrated to the global sensitivity of the query function. Therefore, we first conduct a sensitivity analysis on the DP-enabled model parameters.

#### C.2.1 Sensitivity Analysis

Note that the query function $s_{\mathcal{D}_k}$ is used to denote the mapping from the local dataset $\mathcal{D}_k$ to the DP-enabled model parameters, i.e., $s_{\mathcal{D}_k} \triangleq \mathbf{w}_{k,d}^t$, where $\mathbf{w}_{k,d}^t$ are model parameters by DP protection on dataset $\mathcal{D}_k$. Combined with (4), its specific expression is given as

$$\begin{aligned} s_{\mathcal{D}_k} \triangleq w_{k,d}^t =& h_{k,d}^t \cdot \arg\min_{\mathbf{w}} F_k\left(\mathbf{w}, \mathcal{D}_k\right) \\ &= \frac{h_{k,d}^t}{|\mathcal{D}_k|} \sum_{j=1}^{|\mathcal{D}_k|} \arg\min_{\mathbf{w}} F_k(\mathbf{w}, \mathcal{D}_{k,j}). \end{aligned} \tag{C.1}$$

With **Definition 1**, if we have two adjacent datasets $\mathcal{D}_k$ and $\mathcal{D}_k'$, the model sensitivity of any two datasets $\mathcal{D}_k$ and $\mathcal{D}_k^{'}$ by DP is

$$\begin{aligned} \Delta s_{\mathcal{D}_k} &= \max_{\mathcal{D}_k, \mathcal{D}_k'} \| s_{\mathcal{D}_k} - s_{\mathcal{D}_k'} \| \\ &= \max_{\mathcal{D}_k, \mathcal{D}_k'} \| \frac{h_{k,d}^t}{|\mathcal{D}_k|} \sum_{j=1}^{|\mathcal{D}_k|} \arg\min_{\mathbf{w}} F_k(\mathbf{w}, \mathcal{D}_{k,j}) \\ &- \frac{h_{k,d}^t}{|\mathcal{D}_k'|} \sum_{j=1}^{|\mathcal{D}_k'|} \arg\min_{\mathbf{w}} F_k(\mathbf{w}, \mathcal{D}_{k,j}') \| = \frac{2 h_{k,d}^t C}{|\mathcal{D}_k|}. \end{aligned} \tag{C.2}$$

Then, the global sensitivity is defined as the maximum sensitivity across all devices, i.e., $\Delta s = \max_k \{\Delta s_{\mathcal{D}_k}\}$.

#### C.2.2 Noise Scale and Privacy Budget

According to **Definition 1**, to achieve $(\epsilon, \delta)$-DP, the Gaussian mechanism adds zero-mean noise with scale $\sigma$ to DP-enabled model parameters. Moreover, based on the conversion relationship of RDP Mironov (2017), the noise scale must satisfy

$$\sigma^t \geq \frac{\Delta s \sqrt{2 \ln(1.25/\delta_k^t)}}{\epsilon_k^t} = \frac{2 h_{k,d}^t C \sqrt{2 \ln(1.25/\delta_k^t)}}{|\mathcal{D}_k| \epsilon_k^t}. \tag{C.3}$$

Then, the lower bound of the privacy budget at each round can be expressed as

$$\epsilon_k^t \geq \frac{2 C h_{k,d}^t \sqrt{2 \ln(1.25/\delta_k^t)}}{|\mathcal{D}_k| \sigma_{max}}. \tag{C.4}$$

Note that in (C.4), the smaller the proportion of parameters protected by DP $h_{k,d}^t$, the lower the model sensitivity, which in turn results in a smaller added noise $\sigma$ under the same privacy budget, and a less adverse impact on model accuracy.

### C.2.3 Privacy Preservation of DP

According to **Definition 1**, for any adjacent datasets $\mathcal{D}_k$ and $\mathcal{D}'_k$, if we add noise to the model parameters based on (C.4), an attacker cannot distinguish outputs $s_{\mathcal{D}_k}$ and $s_{\mathcal{D}'_k}$. Therefore, even if the PS can access the DP-enabled model parameters, it still cannot distinguish whether true information exists, thereby protecting individual privacy.

In summary, under our HPFL framework, since the PS does not possess the decryption key, it cannot recover the plaintext, which protects the HE-enabled model parameters. Furthermore, if the DP-enabled model is injected with Gaussian additive noise based on (C.4), according to **Definition** 1, the PS cannot obtain devices' gradients, thereby ensuring privacy mathematically.

### C.2.4 CNN architectures

For FMNIST, the model consists of 2 convolutional layers, 2 pooling layers, and 2 fully connected layers. For CIFAR-10, the model consists of 6 convolutional layers, 3 pooling layers, and 2 fully connected layers. For STL-10, the model consists of 3 convolutional layers, 3 pooling layers, and 2 fully connected layers.

## D Extended Experiments

**Attack models:** To further validate the robustness and the performance of privacy protection, we consider the following two mainstream threat models.

- **White-box** Nasr et al. (2019): Adversaries train attack models on both the training and testing datasets to learn the differences for inference attacks (i.e., determining whether a specific data sample was involved in a device's local training). Moreover, adversaries capture the correlations between model parameters across different training rounds. The success rate of the attack is formulated as

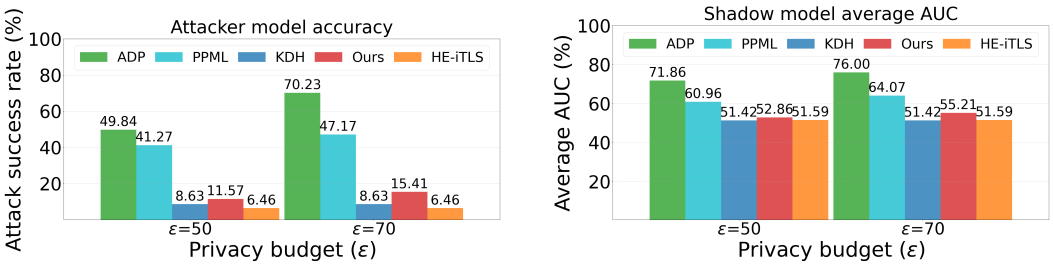
$$\text{SR} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{test}}} \mathbb{I}(\hat{y} = y), \quad y \in \{0, 1\}, \tag{D.1}$$

  where $|\mathcal{D}_{\text{test}}|$ is the total number of samples in the attack testing set; $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside the parentheses is true and 0 otherwise; $\hat{y}$ is the membership prediction output by the attack model for a sample; $y$ is the true membership label of the sample, where $y = 1$ indicates that the sample is a member participating in the model training, and $y = 0$ indicates that the sample is a non-member.

- **ML-Leaks** Salem et al. (2018): Adversaries divide the shadow dataset $\mathcal{D}_{shadow}$ into training datasets $\mathcal{D}_{train}^{shadow}$ and testing datasets $\mathcal{D}_{test}^{shadow}$. The shadow dataset has a similar distribution to the target model's training data, which is used to train shadow models that mimic the behavior of the target model. Then, a shadow model $\mathcal{M}_{shadow}$ is trained on $\mathcal{D}_{train}^{shadow}$. For each record in $\mathcal{D}_{shadow}$, adversaries select the three largest posterior values from the output of $\mathcal{M}_{shadow}$ and label them as 1 or 0 to train the attack model. Finally, adversaries feed the three largest posterior values into the attack model to obtain predictions about whether a data sample is a member of the target model's training set. In order to evaluate the effectiveness of the membership inference attack, we define a metric, i.e., AUC, as

$$\text{AUC} = \frac{\sum_{i\in\mathcal{P}} \text{rank}_i - \frac{|\mathcal{P}|(|\mathcal{P}|+1)}{2}}{|\mathcal{P}| \cdot |\mathcal{N}|}, \tag{D.2}$$

  where $\mathcal{P}$ is the set of all member samples participating in the model training, also denoted as positive samples; $\mathcal{N}$ is the set of the set of non-member samples that did not participate in training; $\text{rank}_i$ is the rank of the $i$-th positive sample when all samples are sorted in ascending order according to the attack model's prediction scores. Note that the closer the attack AUC value is to 0.5, the more indistinguishable the attacker's membership inference capability is from random guessing, which means a better privacy protection effect of the scheme.




(a) White-box (b) ML-Leaks

Figure 4: Performance comparisons under two attack models

### D.1 Performance of Privacy Preservation by Attack Models

Fig. 4 evaluates the privacy protection performance of different privacy preservation schemes, i.e., ADP, HE-iTLS, KDH, PPML, and our method. The model is attacked by both White-box and ML-Leaks. From Fig. 4(a), it can be observed that as the privacy budget $\epsilon$ increases, the attack success rate of white-box shows a significant upward trend on both the DP scheme (i.e., green bars) and the PPML scheme (i.e., cyan bars), which demonstrates that a higher privacy budget $\epsilon$ implies lower level noise injected into gradients, resulting in easier privacy information leakage in model updates. However, for the attack success rate of the proposed HPFL framework (i.e., red bars), although slightly higher than those of the HE and KDH schemes, it is significantly lower than those of the ADP and PPML schemes regardless of the privacy budget $\epsilon$, demonstrating its superior privacy protection performance. Furthermore, Fig. 4(b) demonstrates the relationship between different privacy budgets and the average AUC of the two attack models. The results show that the privacy protection performance of HPFL is very close to that of the HE scheme (i.e., HE-iTLS) and KDH, with only a 2%-4% difference in attack AUC values. In contrast, the attack AUC values of the ADP and PPML schemes are significantly higher, indicating that their privacy protection ability is much weaker than ours, which is consistent with the previous conclusions. Moreover, by combining Figs. 3 and 4, we can conclude that in comparison to ADP and PPML, our proposed HPFL achieves higher training accuracy and less overall time spent per global training round, while obtaining comparable privacy protection performance with HE-iTLS and KDH, and similar accuracy to HE-iTLS. For instance, in Fig. 3(d) and Fig. 4(b), our method achieves 63% model accuracy and 55% average AUC, which are almost the same as HE-iTLS. However, from Fig. 3(h), our method only spends 400s to complete the entire process of training, privacy protections, and uploading, which is less than half of that of HE-iTLS, and is nearly on par with the PPML scheme, yet with much higher accuracy than PPML.

### D.2 Extension to other Neural Networks

To verify the effectiveness and feasibility of proposed HPFL framework on other neural networks, such as Deep Neural Network (DNN), we reevaluate the model training accuracy on FMNIST dataset with a DNN model, comprising 12 convolutional layers, 2 pooling layers, 2 fully connected layers, and 4 residual blocks. As shown in Fig. 5, the total time is at $Y_{total} = 10^4$s, with other simulation parameters unchanged. Fig. 5 shows that the ADP scheme exhibits the lowest accuracy. In contrast, the HPFL scheme achieves faster convergence and higher accuracy, which can be comparable to that of both FedAvg and HE-iTLS. These conclusions are consistent with the results observed in Fig. 3. Overall, our proposed HPFL scheme can be extended to other deep neural networks.

### D.3 Extension to More Devices

Fig. 6 evaluates the accuracies on CIFAR-10 by our proposed HPFL under different numbers of devices ($K \in \{10, 20, 50\}$) with a fixed total training time budget of $Y_{total} = 10^4$. Experimental results demonstrate that regardless of the number of devices, the proposed HPFL can always converge within a few global aggregation rounds. This demonstrates the robustness of our method and its applicability to large-scale device settings. It should also be noted that a smaller number of devices

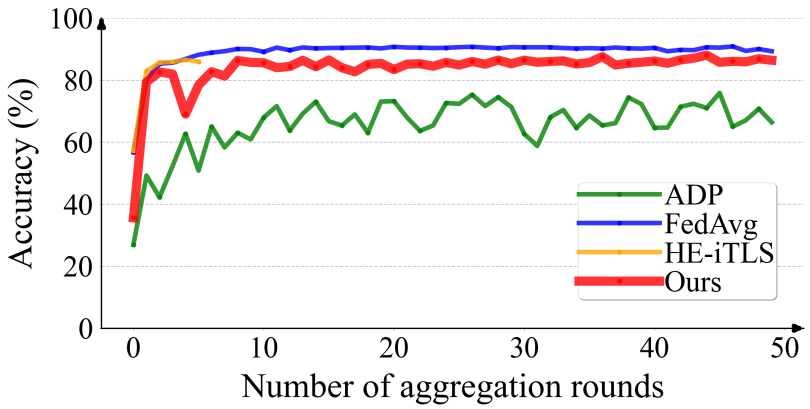



Figure 5: Accuracy comparison under DNN model.

generally leads to faster convergence. This is probably because devices protect their model parameters using different DP and HE settings, which is referred to as privacy heterogeneity in this paper. When more devices participate in FL, this heterogeneity becomes more severe, thereby slowing down convergence.

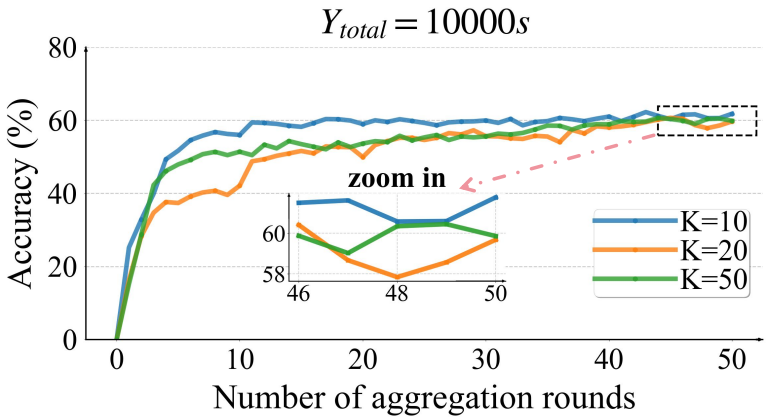



Figure 6: Accuracy for different device counts K.