PROTECT WHAT MATTERS, HIDE WHERE: SELECTIVE DIFFERENTIAL PRIVACY WITH ENCRYPTED MASKS FOR FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning (FL) enables collaborative model training without sharing raw data, yet exchanging model updates still poses privacy leakage risks. Existing privacy-preserving methods based on Homomorphic Encryption (HE) or Differential Privacy (DP) often entail substantial computational and communication overhead or result in degraded model utility, respectively. To address these challenges, we propose FedHPP, an efficient hybrid privacy-preserving FL framework that combines model sparsity-aware DP protection with HE-encrypted privacy mask techniques. Generally, FedHPP identifies a block of critical parameters using a Top-K selection mechanism based on local model update magnitudes and applies DP noise exclusively to this block. Simultaneously, it uses HE to protect compact binary masks that pinpoint DP-injected and Non DP-injected parameter positions, concealing clients' private parameter selection patterns while substantially reducing encryption overhead. Aggregated masks guide clients in denoising, mitigating the impact of perturbations on model convergence. Extensive experiments conducted on real-world datasets under varying degrees of data heterogeneity and comparison against 10 baselines, demonstrating FedHPP's effectiveness in balancing three-way tensions among FL privacy, utility, and efficiency.

1 INTRODUCTION

Federated learning (FL) enables clients to collaboratively train a shared model by exchanging model updates with a server while keeping their private data local (McMahan et al., 2017; Yang et al., 2019; Gao et al., 2020; Yang et al., 2026). This characteristic of separating data from models has motivated its adoption in privacy-sensitive applications, including healthcare (Xu et al., 2022; Nguyen et al., 2023), finance (Long et al., 2021), and recommender systems (Yang et al., 2020; Yu et al., 2023). However, keeping raw data local does not fully eliminate privacy risks: information encoded in shared model updates can still enable an adversary to infer or reconstruct private training data (Zhu et al., 2019; Zhao et al., 2020; Geiping et al., 2020). For example, a curious central server can reconstruct users' private data through gradient inversion attacks (GIAs) (Zhu et al., 2019; Zhao et al., 2020). To mitigate these risks, homomorphic encryption (HE) (Zhang et al., 2020) and differential privacy (DP) (Abadi et al., 2016) are two primary privacy-preserving types incorporated in FL.

In DP-based FL (DPFL), clipping and random perturbation are commonly employed to limit client information leakage within shared model updates (Wang et al., 2022; Zhu & Ling, 2022; Li et al., 2025a; Shi et al., 2025). Early works primarily focused on centralized scenarios, assuming a trusted central entity responsible for aggregation (Abadi et al., 2016; Gu et al., 2025). In contrast, Local Differential Privacy (LDP) eliminates the need for a trusted central server, making it more suitable for distributed scenarios (Zhu & Ling, 2022). Under LDP-FL, each client independently perturbs local model updates before uploading them, ensuring the server receives only noisy results. However, LDP-FL's robust privacy relies on substantial random noise, which inevitably degrades model performance (Zhang et al., 2022; Sun et al., 2026). Conversely, reducing the noise magnitude improves model performance but weakens privacy defenses and raises vulnerability to gradient inversion attacks (Hu et al., 2024; You et al., 2025). This dilemma demonstrates that **applying uniform-intensity, indiscriminate noise protection to all model parameters is an inefficient privacy allocation strategy**. Consequently, a natural design approach is to shift toward selective DP

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.

perturbation, applying noise strictly to critical parameters of high importance to safeguard sensitive information while reducing total noise energy (Xu et al., 2025). We note that (Xu et al., 2025) has to change the standard one-FL-server or coordinator setting into a multi-party setting, which does not naturally fit the default FL environment—our proposal does fit.

However, selective DP faces new challenges in FL. Under non-independent and identically distributed (Non-IID) data, parameter importance evaluated locally varies across clients, leading to discrepancies in noise-injection locations. This inconsistency creates non-uniform perturbation states for identical parameter dimensions across clients, severely compromising global model convergence stability (Liu et al., 2020). Consequently, clients must precisely record and lock their noise-injection locations to enable accurate denoising and global alignment in later stages. Nevertheless, transmitting these indices in plaintext exposes local data distributions and training dynamics, weakening overall privacy. HE-based FL (HEFL) provides a viable security solution by allowing the server direct aggregating ciphertext; however, HE ciphertexts can expand to tens of times the size of plaintext models or more. For clients with limited computing power and uplink bandwidth, processing and transmitting the ciphertexts impose severe overhead (Bonawitz et al., 2017; Jiang et al., 2021). Even with batching techniques that pack multiple values into a single ciphertext (Zhang et al., 2020; Hu & Li, 2025), **existing full-model HE overhead still scales sharply with growing model sizes**.

Based on this analysis, we derive a key insight: **Selective DP and HE need not be applied to the entire model simultaneously; instead, they should undertake complementary defensive roles protecting different aspects of the data.** Specifically, the model parameters can undergo lightweight perturbation via selective DP, while the masks specifying the perturbation locations can be encrypted using HE. Since the dimensionality overhead of the masks constitutes only a negligible fraction (e.g., $\sim$1%) of the full model, this elegantly circumvents the bottlenecks associated with full-model HE, protecting sensitive positional information at a minimal cryptographic cost.

Accordingly, we propose FedHPP, an efficient hybrid privacy-preserving FL framework that combines block-wise and selective DP perturbation with HE-protected masks. Specifically, each client flattens the parameters and partitions them into parameter blocks. It then selects the Top-K blocks according to their average update magnitude and applies clipping and noise perturbation to the selected blocks. A binary privacy mask records the selected and unselected blocks. To conceal each client's mask from the server during aggregation, FedHPP encrypts the masks using HE. Clients decrypt the aggregated masks to identify the parameter blocks associated with perturbation across participating clients and perform mask-guided noise mitigation. By combining selective DP perturbation, encrypted mask aggregation, and client-side noise mitigation, FedHPP offers a lightweight and flexibly adjustable trade-off among privacy protection, model utility, and efficiency. We summarize the contributions of our paper as follows:

- We propose FedHPP, a hybrid privacy-preserving FL framework that combines selective DP with HE, which balances three-way tensions among privacy, utility, and efficiency.
- We design an efficient protection mechanism based on Top-K selection and encrypted privacy masks. Selective perturbation limits the scope of noise injection, while encrypting masks instead of the full model reduces ciphertext computation and communication. In terms of the number of encrypted, FedHPP changes the workload from $\mathcal{O}(d)$ to $\mathcal{O}(J)$, where d is the number of model parameters and J is the number of entries in the privacy mask. FedHPP further uses aggregated masks to support client denoising and reduce the impact of perturbation on model training.
- We evaluate FedHPP on real-world datasets under varying degrees of data heterogeneity and compare it with ten state-of-the-art baselines. The experimental results demonstrate a tunable trade-off among privacy protection, model utility, and efficiency.

2 Related Work

DPFL. Early research primarily focused on centralized scenarios, where a trusted central server aggregates model updates from individual clients (Abadi et al., 2016). However, a fundamental limitation of this approach is its reliance on a trusted server to aggregate raw client model updates—a condition that is difficult to satisfy in real-world applications due to escalating privacy concerns and increasingly stringent data regulations. In contrast, local DP (LDP) (Gursoy et al., 2021; Zhang et al., 2022) has emerged as a framework better suited for FL (Wang et al., 2022; Gu et al., 2025;

Liao et al., 2026), as it offers privacy protection without relying on a trusted central server. Under LDP mechanisms, clients add noise to their parameters before uploading them to an untrusted server; however, this is known to cause a significant degradation in model utility.

HEFL. Phong et al. (2018) improved plaintext space utilization by concatenating multiple quantized gradient values into a single plaintext before applying Paillier encryption (Paillier, 1999). However, this approach required converting all model updates into non-negative integers, which significantly expanded the range of aggregated sums and led to encoding overflows. To address this limitation, Zhang et al. (2020) and Zheng et al. (2023) employed quantization-based model compression to reduce communication overhead. More recently, Li et al. (2025b) and Hu & Li (2025) leveraged CKKS (Cheon et al., 2017) to improve encryption and decryption efficiency. Nevertheless, as the number of model parameters grows rapidly, these schemes still incur lengthy encryption and decryption times, and their communication efficiency remains insufficient for practical deployment.

3 Background and System Settings

3.1 Privacy Preserving Federated Learning

FL. Consider an FL system consisting of a central server S and N clients. Each client i maintains a local dataset $\mathcal{D}_i$ of size $|\mathcal{D}_i|$. Let $W^t \in \mathbb{R}^d$ represent the global model parameters at communication round t, where d is the parameter dimensionality. In round t, client i receives W^t and initializes its local model as $W_i^t = W^t$ for local training $W_i^t \leftarrow W_i^t - \eta \nabla \mathcal{L}_i(W_i^{t,0}; \mathcal{D}_i)$, where $\mathcal{L}_i(\cdot)$ represents the loss function, and η is the local learning rate. Upon completion, client i calculates the model parameter update $\Delta W_i^t = W_i^t - W^t$ and uploads it to the server. The server aggregates these updates to obtain the next global model $W^{t+1} = W^t + \sum_{i=1}^N \alpha_i \Delta W_i^t$, where $\alpha_i = |\mathcal{D}_i| / \sum_{i=1}^N |\mathcal{D}_i|$ is the aggregation weight for client i, satisfying $\sum_{i=1}^N \alpha_i = 1$.

Since model parameters inherently carry privacy information derived from local training data, uploading them directly can still lead to serious privacy breaches (Zhu et al., 2019; Zhao et al., 2020). To address the challenges, HE and DP are widely adopted in FL (for information regarding DP and HE, please refer to Appendix A). We briefly summarize these approaches below.

DPFL. Keeping FL protocol unchanged, client i adds Gaussian noise Z_i^t, satisfying DP requirements, to ΔW_i^t before uploading it to the server; here, each dimension independently follows the distribution $\mathcal{N}(0, \sigma^2 I_d)$, such that $\overline{\Delta W_i^t} \leftarrow \Delta W_i^t + Z_i^t$. This method entails lower computational and communication overhead, but the introduction of noise can easily lead to an imbalance between model performance and privacy protection.

HEFL. Keeping FL protocol unchanged, client i encrypts ΔW_i^t and uploads $\mathrm{Enc}_{pk}(\Delta W_i^t)$ to the server. After aggregation, client i decrypts $\mathrm{Dec}_{sk}(\mathrm{Enc}_{pk}(\Delta W_i^t))$ and completes the local training. Here, $\mathrm{Enc}_{pk}(\cdot)$ and $\mathrm{Dec}_{sk}(\cdot)$ denote encryption with public key pk and decryption with private key sk, respectively. After encryption, $\mathrm{Enc}_{pk}(\Delta W_i^t)$ is transmitted as ciphertext, effectively resisting GIA attacks; however, frequent large-scale ciphertext transmission and computationally intensive ciphertext operations pose severe challenges for the processing power and storage capacity of local devices, as well as the efficiency of interaction with cloud servers.

3.2 System Model

Threat Model. In this study, we consider a threat model widely used in DP and HE (Li et al., 2025b; You et al., 2025; Du et al., 2025). Specifically, the server is assumed to be "honest-but-curious"; it strictly executes the aggregation protocol and has full access to client-uploaded model parameters, potentially employing GIA techniques (Geiping et al., 2020; Fan et al., 2025; Zhu et al., 2026) to reconstruct local training data. However, the server lacks access to auxiliary information like batch sizes, regularization statistics, or data distributions—a reasonable constraint as clients do not need to disclose such data. Meanwhile, clients do not collude with the server.

The previous analysis shows that HE and DP each offer unique advantages in protecting model parameters, but both face inherent bottlenecks that are difficult to coordinate. Under the above threat model, we summarize the following challenges:

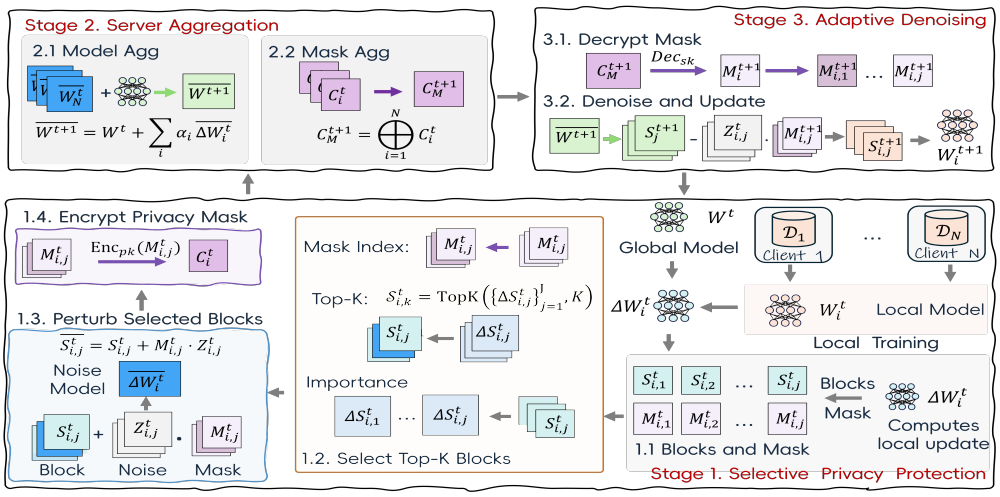


Figure 1: Framework and workflow of the FedHPP.

1) Computational and Communication Challenges. HEFL enables ciphertext-domain aggregation and prevents semi-honest servers from observing raw gradients, yet incurs significant ciphertext expansion and computational intensity. These overheads hinder large-scale FL deployments with limited computational resources and uplink bandwidth. Reducing ciphertext size is thus needed to alleviate edge-device storage, computing, and communication bottlenecks.

2) Privacy-utility Challenges. DPFL's noise introduces a utility-privacy trade-off. Strict privacy budgets (e.g., (ϵ, δ)-DP) require noise variance proportional to parameter dimensionality, slowing convergence or even collapsing accuracy. Conversely, reducing noise degrades privacy rapidly over rounds. A selective noise mechanism is necessary to achieve a privacy-utility balance.

3) Dilemma of Hybrid Privacy. Privacy strength, communication efficiency, and model accuracy are inherently in tension. HE incurs high ciphertext overhead; DP is lightweight but degrades model performance through noise accumulation. Single techniques optimize only one dimension. A hybrid paradigm is urgently needed, synergizing HE and DP mechanisms for joint optimization.

4 Method Design

4.1 FedHPP Framework Design

By integrating HE and DP, **FedHPP** enhances privacy protection while mitigating the communication overhead of HE schemes and the performance degradation of DP schemes. The FedHPP framework is illustrated in Fig 1, and its workflow comprises the following three stages:

Stage 1: Top-K Selective Privacy Protection. After local training, client i computes the local update $\Delta W_i^t = W_i^t - W^t$ and partitions it into J parameter blocks $S_{i,j}^t$. Meanwhile, a binary privacy mask $M_i^t \in \{0,1\}^J$ is generated, where $M_{i,j}^t = 1$ indicates a perturbed block and $M_{i,j}^t = 0$ otherwise. The client evaluates the sensitivity of each block and selects the Top-K sensitive blocks for Gaussian perturbation, i.e., $\overline{S_{i,j}^t} = S_{i,j}^t + M_{i,j}^t \cdot Z_{i,j}^t, Z_{i,j}^t \sim \mathcal{N}(0, \sigma^2 I_B), j \in [J]$. Then, FedHPP encrypts privacy mask $C_i^t = \mathrm{Enc}_{pk}(M_i^t)$, while the reconstructed perturbed update $\overline{\Delta W}_i^t \leftarrow Concat(\overline{S_{i,1}^t}, \ldots, \overline{S_{i,J}^t})$. Finally, client i uploads $\overline{\Delta W}_i^t$ and C_i^t to the server.

Stage 2: Aggregation. After receiving $\overline{\Delta W}_i^t$ and C_i^t from clients, the server aggregates the perturbed model updates as $\overline{W^{t+1}} = W^t + \sum_{i=1}^N \alpha_i \overline{\Delta W_i^t}$ and the encrypted masks $C_M^{t+1} = \bigoplus_{i=1}^N C_i^t$. The server then broadcasts the $\overline{W}^{t+1}$ and C_M^{t+1} to the clients.

Stage 3: Mask-Guided Adaptive Denoising. Each client decrypts the aggregated mask as $M^{t+1} = \mathrm{Dec}_{sk}(C_M^{t+1})$, and identifies the parameter blocks affected by privacy perturbation. For a perturbed

block whose noise can be reconstructed from the locally stored random seed, the client removes the corresponding noise $\overline{W_{i,j}^{t+1}} = \overline{W_j^{t+1}} - \alpha_i Z_{i,j}^t$. Unperturbed blocks are retained directly.

4.2 TOP-K BASED SELECTIVE PRIVACY PROTECTION

Parameter Partitioning and Mask Initialization. After completing local training, client i obtains the model parameter update $\Delta W_i^t = W_i^t - W^t$ and flattens it into the vector $\Delta W_i^t = [\Delta W_{i,1}^t, \Delta W_{i,2}^t, \ldots, \Delta W_{i,d}^t]$, where d denotes the total number of trainable parameters in the model. To overcome the limitations imposed by specific network architectures, FedHPP maps the trainable parameters of various models into a unified parameter vector space, thereby enabling seamless application to convolutional neural networks, fully connected networks, and other parameterized models. Subsequently, the flattened parameters are partitioned based on size into multiple blocks $S_i^t = [S_{i,1}^t, S_{i,2}^t, \ldots, S_{i,J}^t], J = d/B$ (where B represents the number of model parameters per block; e.g., $S_{i,1}^t = [\Delta W_{i,1}^t, \Delta W_{i,2}^t, \ldots, \Delta W_{i,B}^t]$ is the first block). To precisely anchor the parameter blocks designated for privacy perturbation, the client simultaneously constructs a binary privacy mask $M_i^t \in \{0,1\}^J$ that corresponds to the number of parameter blocks.

Importance Measurement and Top-K Selection. The Top-K method is widely adopted to reduce communication overhead in FL by sharing only a small fraction of parameters while maintaining performance comparable to full dense updates. However, conventional Top-K selects the top K parameters based on absolute magnitude (Li et al., 2025b; Hu & Li, 2025; Xu et al., 2025), a fine-grained strategy that overlooks the collective contribution of parameter regions. Moreover, for large-scale models, combining such fine-grained selection with DP can lead to performance degradation due to fixed Top-K noise injection. To address this, FedHPP selects K parameter blocks according to the average magnitude of parameter blocks. Specifically, let the j-th block of client i contain B parameters, and denote the sum of average updates within this block as $S_{i,j}^t$. The block importance is then defined as $\Delta S_{i,j}^t = \left|sum(S_{i,j}^t)/B\right|$. Based on this metric, the K blocks with the largest $\Delta S_{i,j}^t$ are selected as critical indices $\mathcal{S}_{i,k}^t$, and the binary privacy mask $M_{i,j}^t$ is assigned accordingly:

$$M_{i,j}^t = \begin{cases} 1, & j \in \mathcal{S}_{i,k}^t, \\ 0, & j \notin \mathcal{S}_{i,k}^t, \end{cases}, \quad \mathcal{S}_{i,k}^t = \mathrm{TopK}\left(\{\Delta S_{i,j}^t\}_{j=1}^J, K\right), \tag{1}$$

where $M_{i,j}^t = 1$ indicates that the block falls within the scope of privacy protection, whereas 0 indicates the opposite.

Selective Perturbation. To strictly limit the sensitivity of client model updates, FedHPP first performs ℓ_2-norm clipping: $\Delta W_i^t = \Delta W_i^t \cdot \min\left(1, \frac{C}{\|\Delta W_i^t\|_2}\right)$, where C is a preset clipping threshold. Subsequently, Gaussian random noise $Z_{i,j}^t \sim \mathcal{N}(0, \sigma^2 I_B)$ is added only to the selected parameter blocks $S_{i,k}^t$ (corresponding to $M_{i,j}^t = 1$). The locally uploaded model after selective perturbation can be expressed as

$$\overline{S_{i,j}^t} = S_{i,j}^t + M_{i,j}^t \cdot Z_{i,j}^t, \qquad \overline{\Delta W_i^t} = [\overline{S_{i,1}^t}, \ldots, \overline{S_{i,J}^t}]. \tag{2}$$

Under the target privacy budget (ϵ, δ)-DP, the noise scale for the Gaussian mechanism is set to $\sigma \geq C\sqrt{2\ln(1.25/\delta)}/\epsilon$. Since perturbation is applied to K blocks, the expected perturbation energy is $\mathbb{E}\left[\|M_i^t \odot Z_i^t\|_2^2\right] = KB\sigma^2$ (in contrast, the energy for full-parameter perturbation is $\mathbb{E}\left[\|Z_i^t\|_2^2\right] = d\sigma^2$). Therefore, the ratio of the selective perturbation energy in FedHPP to the total parameter perturbation energy is $KB\sigma^2/d\sigma^2 = KB/d = \rho$. For a given noise scale, FedHPP effectively reduces the perturbation energy to a fraction ρ of the original level, thereby significantly mitigating the negative impact of privacy protection on model performance.

Privacy Mask Encryption. FedHPP strategically focuses HE solely on the privacy mask itself, by computing $C_i^t = \mathrm{Enc}_{pk}(M_i^t)$. Let b_c be the average HE ciphertext size (in bits); while traditional full-parameter encryption results in a ciphertext expansion volume as high as db_c, FedHPP encrypts only the mask (which has a dimensionality of J), thereby drastically reducing the extent of ciphertext expansion Jb_c. This design ensures security while significantly alleviating the communication burden and improving computational efficiency regarding the ciphertext. Finally, the client shares $\overline{\Delta W_i^t}$ and C_i^t with the server.

4.3 Model Aggregation

The server performs weighted aggregation directly on $\overline{\Delta W_i^t}$, using the aggregation formula $\overline{W^{t+1}} = W^t + \sum_i \alpha_i \overline{\Delta W_i^t}$. Based on the aforementioned aggregation rule, the server-side weighted aggregation of the client perturbation model can be further expanded to yield:

$$\overline{W^{t+1}} = W^t + \sum_i \alpha_i \overline{\Delta W_i^t} = W^t + \sum_i \alpha_i [S_{i,1}^t, \ldots, S_{i,J}^t] + \sum_i \sum_j^J \alpha_i M_{i,j}^t \cdot Z_{i,j}^t \tag{3}$$

Meanwhile, the server performs aggregation directly in the ciphertext $C_M^{t+1} = \bigoplus_{i=1}^N C_i^t = \mathrm{Enc}_{pk}(\sum_i M_i^t)$. It should be noted that C_M^{t+1} is not used to recover the clients' original random noise; rather, it primarily serves to identify the parameter locations in the current $\overline{W^{t+1}}$ affected by privacy perturbations, thereby providing indexing information for subsequent client model updates and de-biasing processes. Finally, the server broadcasts C_M^{t+1} and $\overline{W^{t+1}}$ to clients.

4.4 Mask-Guided Adaptive Denoising

After receiving C_M^{t+1} and $\overline{W^{t+1}}$, the client first decrypts C_M^{t+1} using private key sk ($M^{t+1} = \mathrm{Dec}_{sk}\left(C_M^{t+1}\right)$). Based on the M^{t+1}, the client identifies the parameter indices affected by privacy perturbation $\mathcal{S}^{t+1} = \left\{j : M_j^{t+1} > 0\right\}$. For parameters where $M_j^{t+1} = 0$, the parameters remain unaffected by the current round's privacy perturbation, so the aggregated result is retained directly; For parameters where $M_j^{t+1} > 1$, this does not imply exact cancellation; independent zero-mean noise is reduced by averaging only in a second-moment sense; For parameters where $M_j^{t+1} = M_{i,j}^t = 1$, the updated model parameter can thus be expressed as $W_{i,j}^{t+1} = \overline{W_j^{t+1}} - \alpha_i Z_{i,j}^t$. Therefore, the local model update can be expressed as

$$W_{i,j}^{t+1} = \overline{W_j^{t+1}} - \alpha_i M_{i,j}^t \mathbf{1}\{M_j^{t+1} = 1\} Z_{i,j}^t \tag{4}$$

Notably, FedHPP does not use the aggregation mask to recover client noise on an element-wise basis; instead, the mask only identifies the protected parameter regions, while noise impact is mitigated through noise cancellation, thereby ensuring model performance. The overall process is summarized in Algorithm 1 (Appendix B).

5 Theoretical Analysis

In the following, we state the necessary theorems. Full theoretical details appear in Appendix C.

Theorem 1 (Selective DP Guarantee) *Let $D_i \simeq D_i'$ denote two neighboring local datasets differing in one training sample. Let $\epsilon > 0$ and $0 < \delta < 1$. Assume that $\sigma_j > 0$ for every block $j = 1, \ldots, J$, so that the fixed covariance Σ_i is invertible. Define the noise-normalized model sensitivity as $\beta_i = \sup_{D_i \simeq D_i'} \left\| \Sigma_i^{-1/2} \left(\bar{\Delta}_i^t(D_i) - \bar{\Delta}_i^t(D_i') \right) \right\|_2$. A sufficient upper bound is $\beta_i^2 \leq \sum_{j=1}^J \frac{\Delta_{2,j}^2}{\sigma_j^2}$. If $\beta_i \leq \epsilon / \sqrt{2\ln(1.25/\delta)}$, then the complete uploaded update satisfies (ϵ, δ)-differential privacy.*

Theorem 2 (Client's Privacy Mask) *Assume the employed HE scheme satisfies semantic security and the server does not possess the corresponding private decryption key. For a client i's encrypted privacy mask $C_i^t = \mathrm{Enc}_{pk}(M_i^t)$, the server cannot effectively recover the corresponding plaintext privacy mask M_i^{t+1} with non-negligible probability.*

Communication Efficiency Analysis. Let b_p be the plaintext parameter size (in bits), and b_c be the average HE ciphertext size (in bits), where $b_p \ll b_c$. Per exchange, full-model DPFL incurs a communication overhead of $C_{\mathrm{DPFL}} = db_p$, while full-model HEFL requires $C_{\mathrm{HEFL}} \approx db_c$. For FedHPP, the overhead is $C_{\mathrm{FedHPP}} \approx db_p + Jb_c$ with $J = d/B$. Accounting for HE ciphertext packing with capacity s, the payloads simplify to $C_{\mathrm{HEFL}} = \lceil d/s \rceil b_c$ and $C_{\mathrm{FedHPP}} = db_p + \lceil J/s \rceil b_c$. Although increasing B reduces the mask count J, the serialized ciphertext payload remains unchanged unless J drops sufficiently to decrease $\lceil J/s \rceil$.

Table 1: The impact of different data distributions on test accuracy(%). γ indicates the Dirichlet distribution parameter. FedHPP is bolded. Baselines' best results are bolded.

Methods	FMNIST			CIFAR-10			CIFAR-100			Tiny-ImageNet		
	IID	$\gamma=0.3$	$\gamma=0.5$	IID	$\gamma=0.3$	$\gamma=0.5$	IID	$\gamma=0.3$	$\gamma=0.5$	IID	$\gamma=0.3$	$\gamma=0.5$
FedAvg	**90.47**	89.01	89.32	**75.03**	71.24	**72.13**	**36.89**	29.96	**34.49**	**34.01**	29.46	**34.14**
DP-SGD	85.63	83.27	84.47	72.56	67.41	68.73	35.34	28.52	33.67	33.81	28.97	32.46
DP-RSA	84.30	80.87	81.64	71.35	63.79	64.43	35.08	28.62	32.68	33.52	29.03	32.68
DP-BREM	86.77	85.52	85.74	72.61	68.71	69.23	36.23	28.46	33.05	32.86	28.75	32.79
FedCEO	88.63	84.41	85.76	72.55	66.93	67.02	36.08	28.56	33.45	33.13	**29.73**	32.41
DP-FedSAM	88.91	85.77	86.94	72.35	68.39	68.53	36.20	27.14	32.73	32.14	28.64	33.12
CLOVER	87.88	84.01	85.18	72.43	68.57	68.82	35.87	26.38	33.24	33.45	28.58	33.45
FedSPA	90.17	87.46	88.79	72.81	69.63	70.25	36.43	**30.01**	33.51	33.63	29.02	34.06
BatchCrypt	90.36	**89.02**	**89.51**	74.91	**71.39**	72.04	36.72	29.76	33.86	33.79	29.37	33.31
FedPHE	89.85	88.96	89.23	74.89	70.27	71.78	36.74	29.95	33.63	33.87	29.28	33.53
FedHPP	**90.41**	**88.57**	**89.26**	**74.93**	**70.92**	**71.55**	**36.57**	**31.04**	**33.76**	**34.13**	**29.87**	**34.37**

6 EXPERIMENTS

6.1 EXPERIMENTAL SETTINGS

Datasets: We use FashionMNIST (FMNIST), CIFAR-10, CIFAR-100, Tiny-ImageNet, DomainNet, Office-Home, and AG News for our experiments. To evaluate the performance of FedHPP, we partition the datasets using the independent and identically distributed (IID) setting and the Dirichlet distribution (Non-IID) setting, respectively (Caldas et al., 2018; Yang et al., 2026). For FMNIST and CIFAR-10, we employ a CNN model consisting of three convolutional layers and two fully connected layers. We use ResNet-18 for CIFAR-100 and ViT-Small for Tiny-ImageNet..

Baselines: To comprehensively evaluate our framework, we compare it with three main categories: FedAvg (McMahan et al., 2017); DPFL (DP-SGD (Abadi et al., 2016), DP-RSA (Zhu & Ling, 2022), DP-BREM (Gu et al., 2025), FedCEO (Li et al., 2025a), DP-FedSAM (Shi et al., 2025), CLOVER (Xu et al., 2025), and FedSPA (Liao et al., 2026)), and HEFL (BatchCrypt (Zhang et al., 2020), FedPHE (Li et al., 2025b)).

Metrics: To test the model performance, communication load, and computational efficiency of FedHPP, we use test accuracy, communication traffic, and time consumption to represent them respectively. To test how well FedHPP resists privacy attacks (e.g., iDLG (Zhao et al., 2020), InvG (Geiping et al., 2020), FedLeak (Fan et al., 2025), and CFGIA (Zhu et al., 2026)), we use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) as evaluation metrics. For more details, see the appendix D.1.

6.2 METHOD PERFORMANCE

Accuracy. Table 1 compares FL performance under DP and HE mechanisms. FedAvg and HEFL show nearly identical accuracy across datasets (e.g., 75.03%, 74.91%, 74.89% on CIFAR-10 IID), confirming HE hardly causes any loss of accuracy. DPFL suffers notable accuracy drops: over 6.17% (DP-RSA) on FMNIST IID versus FedAvg, and DP-SGD falls to 35.34% on CIFAR-100 IID (baseline 36.89%), underscoring noise-induced utility erosion. FedHPP clearly outperforms DPFL and closely matches FedAvg/HEFL: 90.41% on FMNIST IID (vs. 90.47%), 70.92% on CIFAR-10 Non-IID (γ=0.3) (vs. DP-SGD 67.41%), and 34.37% on Tiny-ImageNet Non-IID (γ=0.5), leading similar privacy methods by nearly 2%(DP-SGD). This indicates that FedHPP's Top-K block selection and selective perturbation can effectively reduce the impact of DP noise on accuracy.

Computational and Communication Efficiency. Table 2 shows the performance of FedHPP in terms of computation time and communication overhead (/ indicates no data). Compared to DPFL (DP-SGD, CLOVER), the DP time of FedHPP is slightly higher (e.g., 0.04s vs 0.003s under CNN), but the overall training time is similar. Compared to HEFL (BatchCrypt, FedPHE), FedHPP's advantages are more pronounced in ResNet-18 and ViT-Small. Its HE time is only 0.004-0.012s, whereas BatchCrypt reaches 5.43-102.45s and FedPHE 0.94-19.84s; HE communication overhead is only

Table 2: In different models, impact of DP and HE on computational efficiency and communication traffic. Time: computation time of a client, including training, DP, and HE (encryption, decryption); Traffic: traffic load (MB) of a client, including the model and ciphertext; / indicates no data.

Methods	CNN(CIFAR-10)					ResNet-18(CIFAR-100)					ViT-Small(Tiny-ImageNet)				
	Time (s)			Traffic		Time (s)			Traffic		Time (s)			Traffic	
	Tra	DP	HE	Tra/DP	HE	Tra	DP	HE	Tra/DP	HE	Tra	DP	HE	Tra/DP	HE
FedAvg	0.72	/	/	2.41	/	2.82	/	/	105.08	/	40.89	/	/	75.12	/
DP-SGD	0.72	0.003	/	2.41	/	2.85	0.02	/	105.08	/	41.15	0.23	/	75.12	/
CLOVER	0.95	0.003	/	2.41	/	2.95	0.04	/	105.08	/	41.32	0.04	/	75.12	/
BatchCrypt	0.72	/	5.43	/	5.42	2.97	/	102.45	/	206.77	40.57	/	51.37	/	126.95
FedPHE	1.42	/	0.94	/	15.94	2.85	/	19.84	/	1748.04	40.78	/	9.59	/	846.46
FedHPP	0.89	0.04	0.004	2.41	0.32	2.82	0.82	0.012	105.08	0.96	41.17	0.37	0.008	75.12	0.64

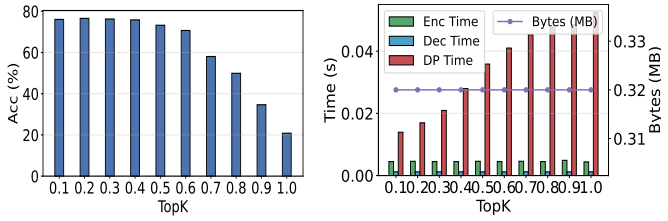
Table 3: The performance of DP and HE against different GIA attacks on CIAFR-10 under IID.

Methods	iDLG			InvG			FedLeak			CFGIA		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
FedAvg	25.49	0.77	0.07	22.65	0.78	0.06	15.89	0.28	0.24	20.23	0.29	0.03
DP-SGD	6.30	0.05	0.44	6.86	0.07	0.37	6.75	0.07	0.33	7.47	0.08	0.35
FedCEO	6.72	0.06	0.45	14.89	0.10	0.33	14.99	0.11	0.34	17.60	0.16	0.34
CLOVER	4.47	0.07	0.52	6.02	0.02	0.55	5.78	0.04	0.54	6.55	0.06	0.55
HE-FL	/	/	/	/	/	/	/	/	/	/	/	/
FedHPP	5.42	0.05	0.40	6.67	0.02	0.49	6.41	0.02	0.50	6.68	0.02	0.52

0.32-0.96 MB, much lower than BatchCrypt's 5.42-206.77MB and FedPHE's 15.94-1748.04MB. Overall, FedHPP's selective perturbation strategy slightly increases the computational overhead during the DP phase, but its mechanism of performing HE only on the mask significantly shortens HE processing time and greatly reduces the amount of ciphertext transmission, thereby achieving an overall improvement in computational and communication efficiency.

6.3 Robustness Against Attacks

Table 3 shows the performance of different methods in resisting GIA attacks. In Table 3, when defending against iDLG, InvG, FedLeak, and CFGIA attacks, FedHPP's PSNR (5.42-6.68) and SSIM (0.02-0.05) are significantly lower than unprotected FedAvg, while LPIPS (0.40-0.52) is notably higher, indicating that the effectiveness of attack reconstruction is effectively suppressed. Compared with DPFL methods, FedHPP's privacy protection ability is comparable to DP-SGD and CLOVER, and overall better than FedCEO. Compared with HEFL, FedHPP fails to completely block attacks and slightly compromises security; it effectively avoids the high computational and communication costs brought by traditional HE schemes while maintaining the predetermined level of privacy.



(a) Accuracy (b) Time/Traffic

Figure 2: The impact of Different Top-K Ratios on Model Accuracy, Time, and Communication Traffic in CIFAR-10 (IID).

6.4 Ablation Study

Top-K Ratio. In Fig 2, as the Top-K ratio increases from 0.1 to 1.0, the model accuracy of

Table 4: Impact of different Top-K mechanisms on model accuracy on CIFAR-10.

Methods	IID	$\gamma = 0.3$	$\gamma = 0.5$
Top-K_{Rand}	73.71	66.73	67.93
Top-K_{Fix}	73.35	67.08	68.27
FedHPP	74.93	70.92	71.55

Table 5: The impact of different ϵ on model performance on CIFAR-100 (IID) and CFGIA.

Metrics	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	$\epsilon = 7$	$\epsilon = 10$
Acc(%)	26.53	36.42	36.89	37.28	43.46
PSNR	5.47	5.94	6.22	6.82	7.94
SSIM	0.008	0.008	0.01	0.01	0.02
LPIPS	0.86	0.85	0.83	0.79	0.75

FedHPP gradually decreases from $\sim$76% to $\sim$20%, while the DP time rises from around 0.01s to approximately 0.05s. This occurs because larger ratios protect more blocks, injecting more noise and leading to reduced accuracy and increased noise generation. In contrast, the enc time, dec time, and traffic remain basically constant and at a low level across the entire Top-K. This is because the number and scale of masks do not change with the Top-K ratio, so the overhead of enc/dec and the ciphertext communication volume remain stable. This indicates that FedHPP can flexibly balance privacy protection and model utility by adjusting the Top-K ratio, while maintaining light computing and communication overheads.

Parameter Selection Mechanism. Table 4 shows that FedHPP consistently achieves the highest model accuracy. Compared to Top-K_{Rand}, which randomly selects K parameters, FedHPP improves accuracy by 1.22%, 4.19%, and 3.62%, respectively; compared to Top-K_{Fix}, which selects a fixed set of K parameters, it improves accuracy by 1.58%, 3.84%, and 3.28%, respectively. This shows that FedHPP's importance metric based on average block update magnitude identifies optimization-critical blocks and better preserves model utility.

Different ϵ. Table 5 shows that as the privacy budget ϵ increases from 1 to 10, FedHPP's accuracy on CIFAR-100 (IID) improves from 26.53% to 43.46%, though defensive capability correspondingly weakens: PSNR rises from 5.47 to 7.94, SSIM increases from 0.008 to 0.02, and LPIPS drops from 0.86 to 0.75, indicating that the quality of attack-based reconstruction gradually improves. Conversely, a smaller ϵ introduces stronger noise perturbations, resulting in poorer reconstruction quality for CFGIA and stronger privacy protection, but inevitably at the cost of sacrificing accuracy. This shows that after selecting a block of key parameters, FedHPP can flexibly balance privacy and utility by adjusting ϵ, thereby meeting diverse requirements across different scenarios.

Across Datasets and Architectures. Table 6 shows that FedHPP performs well across both ViT-Small and Transformer architectures and on DomainNet, Office-Home, and AG News. Specifically, it achieves accuracies of 58.20% and 61.81% on DomainNet and Office-Home, respectively, and 57.39% and 60.57% on the same datasets; on AG News, it reaches accuracies of 88.67% and 87.32%. These results demonstrate that FedHPP possesses effective generalization capabilities across architectures and datasets, maintaining stable performance and robustness. The appendix D.2 contains further ablation results (e.g., Parameter Block Dimension B).

Table 6: The generalization performance of FedHPP on different architectures and datasets. / indicates no data.

Dataset	ViT-Small		Transformer	
	IID	$\gamma = 0.3$	IID	$\gamma = 0.3$
DomainNet	58.20	57.39	/	/
Office-Home	61.81	60.57	/	/
AG News	/	/	88.67	87.32

7 Conclusion

In this paper, we propose FedHPP, a hybrid privacy-preserving FL framework that combines DP with HE. FedHPP performs Top-K selection and noise injection based on the average magnitude of local parameter block updates and encrypts the privacy masks, thereby reducing the computational dimension of HE from $\mathcal{O}(d)$ to $\mathcal{O}(J)$ and significantly cutting computational and communication overhead. Experiments demonstrate that FedHPP effectively balances privacy, utility, and efficiency across diverse datasets and heterogeneity settings; its excellent generalization capability further confirms its effectiveness and robustness. Future work will focus on defense mechanisms against strong threat models (e.g., collusion attacks), comprehensively enhancing the framework's security boundaries while maintaining existing levels of privacy and utility.

AI USE STATEMENT

In this work, we used generative AI tools to help with writing and to correct grammar and spelling mistakes.

ETHICS STATEMENT

This paper adheres to the ICLR Code of Ethics. The research presented herein does not involve human subjects, animals, or any other ethical issues. We obtained all data used in this study from publicly available sources. Our approach raises no issues regarding discrimination, bias, or fairness.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of the research results. For more details, see the appendix D.

REFERENCES

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 308–318, 2016.

Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pp. 1175–1191, 2017.

Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin (eds.), *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pp. 409–437, 2017.

Jiacheng Du, Jiahui Hu, Zhibo Wang, Peng Sun, Neil Gong, Kui Ren, and Chun Chen. Sok: On gradient leakage in federated learning. In *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pp. 3045–3064. USENIX Association, 2025.

Mingyuan Fan, Fuyi Wang, Cen Chen, and Jianying Zhou. Boosting gradient leakage attacks: Data reconstruction in realistic FL settings. In *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pp. 2985–3004, 2025.

Yansong Gao, Minki Kim, Sharif Abuadbba, Yeonjae Kim, Chandra Thapa, Kyuyeon Kim, Seyit Ahmet Çamtepe, Hyoungshick Kim, and Surya Nepal. End-to-end evaluation of federated learning and split learning for internet of things. In *International Symposium on Reliable Distributed Systems, SRDS 2020, Shanghai, China, September 21-24, 2020*, pp. 91–100, 2020.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Xiaolan Gu, Ming Li, and Li Xiong. DP-BREM: differentially-private and byzantine-robust federated learning with client momentum. In *34th USENIX Security Symposium, USENIX Security 2025, Seattle, WA, USA, August 13-15, 2025*, pp. 3065–3082, 2025.

Mehmet Emre Gursoy, Acar Tamersoy, Stacey Truex, Wenqi Wei, and Ling Liu. Secure and utility-aware data collection with condensed local differential privacy. *IEEE Trans. Dependable Secur. Comput.*, 18(5):2365–2378, 2021.

Chenghao Hu and Baochun Li. Maskcrypt: Federated learning with selective homomorphic encryption. *IEEE Trans. Dependable Secur. Comput.*, 22(1):221–233, 2025.

Jiahui Hu, Jiacheng Du, Zhibo Wang, Xiaoyi Pang, Yajie Zhou, Peng Sun, and Kui Ren. Does differential privacy really protect federated learning from gradient leakage attacks? *IEEE Trans. Mob. Comput.*, 23(12):12635–12649, 2024.

Zhifeng Jiang, Wei Wang, and Yang Liu. FLASHE: additively symmetric homomorphic encryption for cross-silo federated learning. *CoRR*, abs/2109.00675, 2021.

Yuecheng Li, Lele Fu, Tong Wang, Jian Lou, Bin Chen, Lei Yang, Jian Shen, Zibin Zheng, and Chuan Chen. Clients collaborate: Flexible differentially private federated learning with guaranteed improvement of utility-privacy trade-off. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, 2025a.

Yuqing Li, Nan Yan, Jing Chen, Xiong Wang, Jianan Hong, Kun He, Wei Wang, and Bo Li. Fedphe: A secure and efficient federated learning via packed homomorphic encryption. *IEEE Trans. Dependable Secur. Comput.*, 22(5):5448–5463, 2025b.

Tianchi Liao, Xiaojun Deng, Lele Fu, Sheng Huang, Bowen Deng, Hong-Ning Dai, and Chuan Chen. Learning in the right subspace: Personalized differential private federated learning with noise filtering. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD 2026, Jeju Island, Korea, August 9-13, 2026*, pp. 3039–3050. ACM, 2026.

Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated SGD under local differential privacy with top-k dimension selection. In *Database Systems for Advanced Applications - 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24-27, 2020, Proceedings, Part I*, volume 12112, pp. 485–501, 2020.

Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. *CoRR*, abs/2108.10749, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017.

Dinh C. Nguyen, Quoc-Viet Pham, Pubudu N. Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia A. Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Comput. Surv.*, 55(3):60:1–60:37, 2023.

Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern (ed.), *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pp. 223–238, 1999.

Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.*, 13(5):1333–1345, 2018.

Yifan Shi, Kang Wei, Li Shen, Yingqi Liu, Xueqian Wang, Bo Yuan, and Dacheng Tao. Toward the flatter landscape and better generalization in federated learning under client-level differential privacy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(12):11632–11643, 2025.

Yan Sun, Qixin ZHANG, Li Shen, and Dacheng Tao. Convergent differential privacy analysis for general federated learning. In *International Conference on Learning Representations*, volume 2026, pp. 116665–116696, 2026.

Fayao Wang, Yuanyuan He, Yunchuan Guo, Peizhi Li, and Xinyu Wei. Privacy-preserving robust federated learning with distributed differential privacy. In *IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2022, Wuhan, China, December 9-11, 2022*, pp. 598–605, 2022.

An Xu, Wenqi Li, Pengfei Guo, Dong Yang, Holger Roth, Ali Hatamizadeh, Can Zhao, Daguang Xu, Heng Huang, and Ziyue Xu. Closing the generalization gap of cross-silo federated medical image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 20834–20843, 2022.

Shuangqing Xu, Yifeng Zheng, and Zhongyun Hua. Harnessing sparsification in federated learning: A secure, efficient, and differentially private realization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS 2025, Taipei, Taiwan, October 13-17, 2025*, pp. 2354–2368, 2025.

Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. Federated recommendation systems. In Qiang Yang, Lixin Fan, and Han Yu (eds.), *Federated Learning - Privacy and Incentive*, volume 12500 of *Lecture Notes in Computer Science*, pp. 225–239. 2020.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, 2019.

Qiantao Yang, Liquan Chen, Mingfu Xue, and Songze Li. Tackling resource-constrained and data-heterogeneity in federated learning with double-weight sparse pack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27592–27600, Mar. 2026.

Zhichao You, Xuewen Dong, Shujun Li, Ximeng Liu, Siqi Ma, and Yulong Shen. Local differential privacy is not enough: A sample reconstruction attack against federated learning with local differential privacy. *IEEE Trans. Inf. Forensics Secur.*, 20:1519–1534, 2025.

Yang Yu, Qi Liu, Likang Wu, Runlong Yu, Sanshi Lei Yu, and Zaixi Zhang. Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 4854–4863, 2023.

Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference, USENIX ATC 2020, July 15-17, 2020*, pp. 493–506, 2020.

Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26048–26067, 2022.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *CoRR*, abs/2001.02610, 2020.

Yifeng Zheng, Shangqi Lai, Yi Liu, Xingliang Yuan, Xun Yi, and Cong Wang. Aggregation service for federated learning: An efficient, secure, and more resilient realization. *IEEE Trans. Dependable Secur. Comput.*, 20(2):988–1001, 2023.

Heng Zhu and Qing Ling. Bridging differential privacy and byzantine-robustness via model aggregation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 2427–2433, 2022.

Kaiming Zhu, Jinsheng Yang, Siyang Guo, Huaqian Qin, Taiyu Wang, Junbo Wang, Yuhong Nan, and Zibin Zheng. Cracking federated privacy: Initialization-Resilient gradient inversion with Fine-Grained reconstruction. In *35th USENIX Security Symposium (USENIX Security 26)*, 2026.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14747–14756, 2019.

A PRELIMINARIES

A.1 DIFFERENTIAL PRIVACY

This section introduces the foundational concepts and formal definitions of DP, a rigorous framework for quantifying privacy guarantees in data analysis. We begin with central differential privacy (CDP), which assumes the presence of a trusted curator who collects the entire dataset and applies a randomized mechanism to protect individual records.

Two datasets $\mathcal{D} = \{d_1, \ldots, d_n\}$ and $\mathcal{D}' = \{d'_1, \ldots, d'_n\}$ are termed *neighboring* if they differ by exactly one record: that is, there exists an index $i \in [n]$ such that $d_i \neq d'_i$, while $d_j = d'_j$ for all $j \in [n] \setminus \{i\}$. Here, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ for any $n \in \mathbb{N}$.

Definition 1 (Central Differential Privacy) *A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy if, for every pair of neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ and every measurable set $\mathcal{S} \subseteq \mathcal{Y}$, the following inequality holds:*

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta.$$

In contrast, local differential privacy (LDP) operates under a stronger threat model where no trusted server exists. Each client independently perturbs their own data before transmission, thereby ensuring privacy at the source.

Definition 2 (Local Differential Privacy) *A randomized mechanism $\mathcal{R} : \mathcal{X} \to \mathcal{Y}$ satisfies ε_0-local differential privacy if, for any two inputs $d, d' \in \mathcal{X}$ and any output setminus $\mathcal{S} \subseteq \mathcal{Y}$, the following condition is satisfied:*

$$\Pr[\mathcal{R}(d) \in \mathcal{S}] \leq e^{\varepsilon_0} \cdot \Pr[\mathcal{R}(d') \in \mathcal{S}].$$

A.2 HOMOMORPHIC ENCRYPTION

HE can be computed directly on encrypted data without prior decryption, and the results of the encrypted computation are consistent with the plaintext computation. It allows special functions to perform operations such as addition and multiplication on encrypted ciphertexts, and only the authorized party holding the decryption key can decrypt the result. For example, for any plaintext message $m1$ and $m2$, $\mathsf{Enc}(m1) \oplus \mathsf{Enc}(m2) = \mathsf{Enc}(m1 + m2)$, where $\oplus$ is an additive operation. Similarly, if $\mathsf{Enc}(m1) \odot \mathsf{Enc}(m2) = \mathsf{Enc}(m1 \cdot m2)$, where $\odot$ is a multiplicative operation.

B FEDHPP ALGORITHM IMPLEMENTATION

Based on the above mechanism design, we formalize FedHPP as the pseudocode shown in the Algorithm 1. Specifically, before training begins, the server sets the global model W^0, privacy budget (ϵ, δ), Top-K ratio ρ, clipping threshold C, and total number of rounds T.

Local training and update computation: In each round t, clients i receive W^t in parallel and perform gradient descent based on local data $\mathcal{D}_i$ to obtain W_i^t; the update is computed and clipped to C, i.e., $\Delta W_i^t = \text{Clip}(W_i^t - W^t, C)$ (Lines 3-4). Block selection and mask generation: ΔW_i^t is partitioned into J blocks; the indices of the top K critical blocks, $\mathcal{K}_i$, are selected based on an importance metric derived from update magnitude, and a binary privacy mask M_i^t is generated accordingly (Lines 5-8). Selective noise addition and encrypted upload: Sampled Gaussian noise Z_i^t is added only at positions indicated by the mask to construct the protected update $\overline{\Delta W_i^t} = \Delta W_i^t + M_i^t \odot Z_i^t$; the mask is encrypted using a public key to obtain C_i^t, and both are uploaded to the server (Lines 9-12).

Server aggregation: The server performs weighted aggregation of the protected updates, $W^{t+1} = W^t + \sum \alpha_i \overline{\Delta W_i^t}$, and executes homomorphic addition on the encrypted masks, $C_M^{t+1} = \bigoplus C_i^t$, subsequently broadcasting the aggregated results (Lines 13-15).

Decryption and Denoising Update: The authorized client uses the private key to decrypt the aggregated mask $M^{t+1} = \text{Dec}_{sk}(C_M^{t+1})$, utilizes M^{t+1} to identify the positions for denoising, and updates the local model $W_{i,j}^{t+1} = \overline{W_j^{t+1}} - \alpha_i M_{i,j}^t \mathbf{1}\{M_j^{t+1} = 1\} Z_{i,j}^t$ (Lines 16-17).

Algorithm 1: FedHPP

Input: Initial global model W^0, privacy budget (ϵ, δ), Top-K ratio ρ, clipping bound C, number of communication rounds T.
Output: Global model W^T.

1 **for** $t = 0$ **to** $T - 1$ **do**
2 **foreach** *client* $i \in [\mathcal{N}]$ **do // in parallel**
3 Perform local training: $W_i^t = W_i^t - \eta \nabla \mathcal{L}_i(W_i^{t,0}; \mathcal{D}_i)$;
4 Compute local parameters update: $\Delta W_i^t = W_i^t - W^t$;
5 Clip: $\Delta W_i^t = \mathrm{Clip}(\Delta W_i^t, C)$;
6 Parameter block $S_i^t = [S_{i,1}^t, S_{i,2}^t, \ldots, S_{i,J}^t], S_{i,1}^t = [\Delta W_{i,1}^t, \Delta W_{i,2}^t, \ldots, \Delta W_{i,B}^t]$;
7 Compute block importance scores and select the top-K blocks: $\mathcal{S}_{i,k}^t = \mathrm{TopK}\left(\{\Delta S_{i,j}^t\}_{j=1}^J, K\right)$;
8 Generate binary privacy mask M_i^t;
9 Generate selective DP noise: $Z_i^t \sim \mathcal{N}(0, \sigma^2 I_B)$;
10 Construct protected parameter blocks: $\overline{S_{i,j}^t} = S_{i,j}^t + M_{i,j}^t \cdot Z_{i,j}^t$, $\overline{\Delta W_i^t} = [\overline{S_{i,1}^t}, \ldots, \overline{S_{i,J}^t}]$;
11 Encrypt the privacy mask: $C_i^t = \mathrm{Enc}_{pk}(M_i^t)$;
12 Upload $\overline{\Delta W_i^t}$ and C_i^t;
13 Server aggregates model parameters: See Eq 3;
14 Server aggregates encrypted masks: $C_M^{t+1} = \bigoplus_{i=1}^N C_i^t$;
15 Server broadcasts $\overline{W^{t+1}}$ and C_M^{t+1};
16 Authorized clients decrypt the aggregated mask: $M^{t+1} = \mathrm{Dec}_{sk}(C_M^{t+1})$;
17 Apply mask-guided debiasing/noise cancellation and update the local model; See Eq 4;
18 **return** W^T;

Iteration termination: Lines 1-17 are repeated until T communication rounds are completed, returning the final global model W^T.

C Theoretical Analysis

C.1 Proof of Theorem 1

We now consider the complete model consisting of all J parameter blocks. Define the clipped model update as

$$\bar{\Delta}_i^t = [\bar{\Delta}_{i,1}^t; \bar{\Delta}_{i,2}^t; \ldots; \bar{\Delta}_{i,J}^t] \in \mathbb{R}^d. \tag{5}$$

Its global ℓ_2 sensitivity is

$$\Delta_{2,\mathrm{model}} = \sup_{D_i \simeq D_i'} \left\| \bar{\Delta}_i^t(D_i) - \bar{\Delta}_i^t(D_i') \right\|_2 \leq \sqrt{\sum_{j=1}^J \Delta_{2,j}^2}. \tag{6}$$

If the same clipping threshold C is used for all blocks, then

$$\Delta_{2,\mathrm{model}} \leq 2C\sqrt{J}. \tag{7}$$

For a deterministic update query perturbed by Gaussian noise with a fixed covariance, every neighboring-query difference must lie in the noise-supported subspace to permit a full-output guarantee with $\delta < 1$. As a sufficient construction for the *complete* uploaded update, we therefore consider the full-support block-wise Gaussian mechanism

$$\mathcal{M}_i^t(D_i) = \bar{\Delta}_i^t(D_i) + Z_i^t, \tag{8}$$

where

$$Z_i^t = [Z_{i,1}^t; \ldots; Z_{i,J}^t] \tag{9}$$

and

$$Z_{i,j}^t \sim \mathcal{N}(0, \sigma_j^2 I_B). \tag{10}$$

Hence,

$$Z_i^t \sim \mathcal{N}(0, \Sigma_i), \tag{11}$$

with the following block-diagonal covariance, fixed independently of the private dataset:

$$\Sigma_i = \mathrm{diag}(\sigma_1^2 I_B, \ldots, \sigma_J^2 I_B). \tag{12}$$

Let $\epsilon > 0$ and $0 < \delta < 1$. Assume that $\sigma_j > 0$ for every block $j = 1, \ldots, J$, so that the fixed covariance Σ_i is invertible. Define the noise-normalized model sensitivity as

$$\beta_i = \sup_{D_i \simeq D_i'} \left\| \Sigma_i^{-1/2} \left(\bar{\Delta}_i^t(D_i) - \bar{\Delta}_i^t(D_i') \right) \right\|_2. \tag{13}$$

A sufficient upper bound is

$$\beta_i^2 \leq \sum_{j=1}^{J} \frac{\Delta_{2,j}^2}{\sigma_j^2}. \tag{14}$$

If

$$\beta_i \leq \frac{\epsilon}{\sqrt{2\ln(1.25/\delta)}}, \tag{15}$$

Proof 1 *Consider the whitening transformation*

$$\mathcal{T}(x) = \Sigma_i^{-1/2} x. \tag{16}$$

Applying it to Eq. (8) gives

$$\mathcal{T}(\mathcal{M}_i^t(D_i)) = \Sigma_i^{-1/2} \bar{\Delta}_i^t(D_i) + \Sigma_i^{-1/2} Z_i^t. \tag{17}$$

Since

$$\Sigma_i^{-1/2} Z_i^t \sim \mathcal{N}(0, I_d), \tag{18}$$

the transformed mechanism is a standard isotropic Gaussian mechanism. Its sensitivity is exactly β_i in Eq. (13). Therefore, the Gaussian mechanism guarantees (ϵ, δ)-DP whenever Eq. (15) holds. Since an invertible deterministic transformation does not alter the privacy relation between the two output distributions, the original mechanism also satisfies (ϵ, δ)-DP.

C.2 Proof of Theorem 2

Proof 2 *Suppose there exists an efficient server-side algorithm $\mathcal{A}$ capable of recovering the corresponding plaintext mask from the ciphertext c_i^t with non-negligible advantage. Then, algorithm $\mathcal{A}$ could further distinguish between the ciphertexts of two candidate masks, M_0 and M_1, thereby determining with non-negligible probability whether a given ciphertext corresponds to M_0 or M_1. This would contradict the semantic security of the employed HE scheme. Therefore, provided the server does not hold the private key, a single client's privacy mask remains confidential.*

C.3 Convergence Analysis

To analyze the influence of selective privacy perturbation on the optimization process, we study the convergence behavior of FedHPP for general non-convex objective functions. Different from conventional federated optimization, FedHPP introduces Gaussian perturbation only into the Top-K sensitive parameter blocks and subsequently performs mask-guided adaptive debiasing. Therefore, the optimization process is jointly affected by stochastic gradient noise, client heterogeneity, and the residual privacy perturbation after debiasing. To ensure the rigor of the theoretical analysis, we adopted the assumptions widely used in DPFL works (Zhang et al., 2022; Sun et al., 2026).

Assumption 1 (Smoothness) *The global objective function F is L-smooth. For any x and y, we have*

$$\|\nabla F(x) - \nabla F(y)\|_2 \leq L\|x - y\|_2. \tag{19}$$

It follows that,

$$F(y) \leq F(x) + \langle \nabla F(x), y - x \rangle + \frac{L}{2}\|y - x\|_2^2. \tag{20}$$

Assumption 2 (Lower-Bounded Objective) *There exists a finite constant $F^\star$ such that*

$$F(w) \geq F^\star, \qquad \forall w. \tag{21}$$

Assumption 3 (Bounded Stochastic Gradient Variance) *The stochastic gradient g^t satisfies*

$$\mathbb{E}\left[\|g^t - \nabla F(w^t)\|_2^2\right] \leq V_t, \qquad v_t^2 := \rho d \sigma_t^2 \sum_{i=1}^N \alpha_i^2. \tag{22}$$

Assumption 4 (Fresh Gaussian Noise) *Let $\mathcal{F}_t$ contain all past randomness and the current local training results, clipped updates, and selected masks, immediately before the current noise is generated. Conditional on $\mathcal{F}_t$, the vectors Z_i^t are mutually independent and satisfy*

$$Z_i^t \mid \mathcal{F}_t \sim \mathcal{N}(0, \sigma_t^2 I_d), \qquad i = 1, \ldots, N, \tag{23}$$

where σ_t is a deterministic parameter-noise standard deviation.

Consider N clients, all participating in every round with weights $\alpha_i \geq 0, \sum_{i=1}^N \alpha_i = 1$. The global objective is

$$F(W) = \sum_{i=1}^N \alpha_i F_i(W), \qquad \sum_{i=1}^N \alpha_i = 1. \tag{24}$$

Let ΔW_i^t denote the actual clipped parameter increment produced by client i in round t, before adding the new privacy noise.

The local block mask is $M_{i,j}^t \in \{0, 1\}$, whereas $M_j^t = \sum_{i=1}^N M_{i,j}^t$ denotes the aggregate count. There are $J = d/B$ equal-sized blocks, indexed by $j = 1, \ldots, J$, and each client selects K blocks, so that $KB = \rho d$. Whenever M_i^t is multiplied elementwise with a parameter vector, each block indicator is repeated B times to obtain a d-dimensional mask. The uploaded update and aggregate parameter noise are

$$\overline{\Delta W_i^t} = \Delta W_i^t + M_i^t \odot Z_i^t, \qquad \xi_{\text{raw}}^t = \sum_{i=1}^N \alpha_i (M_i^t \odot Z_i^t). \tag{25}$$

Define the effective direction associated with the complete local update as

$$g^t = -\frac{1}{\eta_t} \sum_{i=1}^N \alpha_i \Delta W_i^t. \tag{26}$$

This is a definition of the direction induced by the actual round update, not an assumption that a multistep local update is a single stochastic gradient. Consequently, the server recursion is exactly

$$W^{t+1} = W^t + \sum_{i=1}^N \alpha_i \overline{\Delta W}_i^t = W^t - \eta_t g^t + \xi_{\text{raw}}^t. \tag{27}$$

After receiving this new server model and correctly recovering the aggregate mask, client i removes its own current-round noise only from uniquely perturbed blocks:

$$W_{i,j}^{t+1} = W_j^{t+1} - \alpha_i M_{i,j}^t \mathbf{1}\{M_j^t = 1\} Z_{i,j}^t \tag{28}$$

The next round of local training begins with W_i^{t+1}. For parameter blocks satisfying $M_j^t > 1$, the noise within them is retained in the model; for parameter blocks perturbed by only a single client, the perturbation noise is also retained in the models of all clients except the one that generated it. We assume that the required local noise samples can be precisely reconstructed without introducing an additional noise-free global recursion step, given that masks can depend on local updates. Under this assumption, the server noise satisfies:

$$\mathbb{E}[\xi_{\text{raw}}^t \mid \mathcal{F}_t] = 0, \qquad \mathbb{E}[\|\xi_{\text{raw}}^t\|_2^2 \mid \mathcal{F}_t] = \rho d \sigma_t^2 \sum_{i=1}^N \alpha_i^2 =: v_t^2 \tag{29}$$

The second relation holds with equality when each client perturbs exactly KB coordinates. The effect of client-side denoising is reflected in the subsequent update direction g^{t+1} and its error bound, rather than as a multiplicative reduction of the current server noise.

Under these assumptions, we have arrived at the following theorem.

Theorem 3 (Finite-Round Stationarity Bound) *Under Assumptions 1–4 hold, give constant $\eta_t = \eta$, $\sigma_t = \sigma$, $V_t \le V$, If the deterministic step sizes satisfy $0 < \eta_t \le 1/L$, then, for every integer $T \ge 1$,the theorem gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(W^t)\|_2^2 \le \frac{2(F(W^0)-F^\star)}{\eta T} + V + \frac{L\rho d\sigma^2}{\eta}\sum_{i=1}^{N}\alpha_i^2 \tag{30}$$

The expectation includes local training randomness and privacy noise across rounds. The conclusion bounds the step-size-weighted average squared gradient norm of the server iterates and does not require g^t to be unbiased.

Proof 3 *Conditioned on $\mathcal{F}_t$, the selected masks are fixed. Since each mask selects KB scalar coordinates,*

$$\mathbb{E}[M_i^t \odot Z_i^t \mid \mathcal{F}_t] = 0, \qquad \mathbb{E}[\|M_i^t \odot Z_i^t\|_2^2 \mid \mathcal{F}_t] = KB\sigma_t^2. \tag{31}$$

For distinct clients, conditional independence and zero means imply

$$\mathbb{E}[\langle M_i^t \odot Z_i^t, M_k^t \odot Z_k^t\rangle \mid \mathcal{F}_t] = 0, \qquad i \ne k. \tag{32}$$

Expanding the squared norm of the aggregate gives

$$\mathbb{E}[\|\xi_{\text{raw}}^t\|_2^2 \mid \mathcal{F}_t] = KB\sigma_t^2\sum_{i=1}^{N}\alpha_i^2 = \rho d\sigma_t^2\sum_{i=1}^{N}\alpha_i^2 = v_t^2. \tag{33}$$

The same upper bound applies when at most KB coordinates are selected. Overlap between clients' masks does not invalidate Eq. 32; the cross terms vanish in expectation, while the noise on shared blocks remains in the realized model.

Applying Eq. 20 to consecutive server iterates yields

$$F(W^{t+1}) \le F(W^t) + \langle \nabla F(W^t), W^{t+1} - W^t\rangle + \frac{L}{2}\|W^{t+1} - W^t\|_2^2. \tag{34}$$

Substitute $W^{t+1} - W^t = -\eta_t g^t + \xi_{\text{raw}}^t$. The linear and quadratic terms become

$$\langle \nabla F(W^t), W^{t+1} - W^t\rangle = -\eta_t\langle \nabla F(W^t), g^t\rangle + \langle \nabla F(W^t), \xi_{\text{raw}}^t\rangle. \tag{35}$$

$$\|W^{t+1} - W^t\|_2^2 = \eta_t^2\|g^t\|_2^2 - 2\eta_t\langle g^t, \xi_{\text{raw}}^t\rangle + \|\xi_{\text{raw}}^t\|_2^2. \tag{36}$$

By construction, W^t and g^t are $\mathcal{F}_t$-measurable and the fresh noise has conditional mean zero. Hence,

$$\mathbb{E}[\langle \nabla F(W^t), \xi_{\text{raw}}^t\rangle \mid \mathcal{F}_t] = \langle \nabla F(W^t), \mathbb{E}[\xi_{\text{raw}}^t \mid \mathcal{F}_t]\rangle = 0. \tag{37}$$

$$\mathbb{E}[\langle g^t, \xi_{\text{raw}}^t\rangle \mid \mathcal{F}_t] = \langle g^t, \mathbb{E}[\xi_{\text{raw}}^t \mid \mathcal{F}_t]\rangle = 0. \tag{38}$$

Taking conditional expectation in Eq. 34 therefore gives

$$\mathbb{E}[F(W^{t+1}) \mid \mathcal{F}_t] \le F(W^t) - \eta_t\langle \nabla F(W^t), g^t\rangle + \frac{L\eta_t^2}{2}\|g^t\|_2^2 + \frac{L}{2}v_t^2. \tag{39}$$

Applying the law of total expectation yields

$$\mathbb{E}F(W^{t+1}) \le \mathbb{E}F(W^t) - \eta_t\mathbb{E}\langle \nabla F(W^t), g^t\rangle + \frac{L\eta_t^2}{2}\mathbb{E}\|g^t\|_2^2 + \frac{L}{2}v_t^2. \tag{40}$$

Only cross terms involving the fresh noise have been eliminated. Previous noise, including noise retained on shared blocks, remains in the current model and training direction.

For every realization of g^t, expansion of its squared distance from the true gradient gives the exact identity

$$-2\langle \nabla F(W^t), g^t \rangle = -\|\nabla F(W^t)\|_2^2 - \|g^t\|_2^2 + \|g^t - \nabla F(W^t)\|_2^2. \tag{41}$$

Substituting this identity into Eq. 40 and collecting terms yields

$$\frac{\eta_t}{2}\mathbb{E}\|\nabla F(W^t)\|_2^2 + \frac{\eta_t(1-L\eta_t)}{2}\mathbb{E}\|g^t\|_2^2 \le \mathbb{E}F(W^t) - \mathbb{E}F(W^{t+1}) + \frac{\eta_t}{2}\mathbb{E}\|g^t - \nabla F(W^t)\|_2^2 + \frac{L}{2}v_t^2. \tag{42}$$

The coefficient of $\mathbb{E}\|g^t\|_2^2$ is $\eta_t/2 - L\eta_t^2/2$.

Since $0 < \eta_t \le 1/L$, the second term on the left of Eq. 42 is nonnegative and can be dropped. Applying Assumption 3 gives

$$\frac{\eta_t}{2}\mathbb{E}\|\nabla F(W^t)\|_2^2 \le \mathbb{E}F(W^t) - \mathbb{E}F(W^{t+1}) + \frac{\eta_t}{2}V_t + \frac{L}{2}v_t^2. \tag{43}$$

The direction error contributes $\eta_t V_t/2$. The fresh parameter noise contributes $Lv_t^2/2$, with no additional factor η_t^2, because it is added directly to the model in Eq. 27.

Sum Eq. 43 over $t = 0, \dots, T-1$:

$$\frac{1}{2}\sum_{t=0}^{T-1}\eta_t\mathbb{E}\|\nabla F(W^t)\|_2^2 \le \sum_{t=0}^{T-1}\big(\mathbb{E}F(W^t) - \mathbb{E}F(W^{t+1})\big) + \frac{1}{2}\sum_{t=0}^{T-1}\eta_t V_t + \frac{L}{2}\sum_{t=0}^{T-1}v_t^2. \tag{44}$$

The objective differences telescope. Because W^0 is deterministic and $F(W^T) \ge F^\star$,

$$\sum_{t=0}^{T-1}\big(\mathbb{E}F(W^t) - \mathbb{E}F(W^{t+1})\big) = F(W^0) - \mathbb{E}F(W^T) \le F(W^0) - F^\star. \tag{45}$$

Substitution into Eq. 44 followed by multiplication by two gives

$$\sum_{t=0}^{T-1}\eta_t\mathbb{E}\|\nabla F(W^t)\|_2^2 \le 2(F(W^0) - F^\star) + \sum_{t=0}^{T-1}\eta_t V_t + L\sum_{t=0}^{T-1}v_t^2. \tag{46}$$

Finally, Substituting $\eta_t = \eta$, $\sum_{t=0}^{T-1}\eta_t V_t \le \eta T V$, and $\sum_{t=0}^{T-1}v_t^2 = T\rho d\sigma^2\sum_{i=1}^{N}\alpha_i^2$ into Eq. 46 , and dividing both sides by ηT, the expression simplifies to:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(W^t)\|_2^2 \le \frac{2(F(W^0) - F^\star)}{\eta T} + V + \frac{L\rho d\sigma^2}{\eta}\sum_{i=1}^{N}\alpha_i^2 \tag{47}$$

D Experiments Details

D.1 Experiments Setup

Dataset: In this study, we employed multiple benchmark datasets to evaluate model performance comprehensively. For image classification tasks, we utilized FashionMNIST (FMNIST), CIFAR-10, CIFAR-100, and Tiny-ImageNet. Specifically, FMNIST comprises 70,000 grayscale images across 10 classes; CIFAR-10 and CIFAR-100 each contain 60,000 RGB images, spanning 10 and 100 classes, respectively; and Tiny-ImageNet consists of 120,000 RGB images across 200 classes. To validate FedHPP in more realistic scenarios, we also incorporated DomainNet (345 classes across 6 domains, approximately 600,000 RGB images), Office-Home (65 shared object categories across 4 domains, totaling 260 classes, approximately 15,500 RGB images), and the text dataset AG News (4 classes, 127,600 text samples). Regarding data partitioning, we considered both Independent and Identically Distributed (IID) and Non-IID settings. The Non-IID scenarios were simulated using the

Dirichlet distribution (Dir(γ)) (Caldas et al., 2018; Yang et al., 2026), with the parameter γ used to precisely control the degree of data heterogeneity across clients.

Models: To comprehensively evaluate the performance of FedHPP across varying task complexities and data modalities, we selected a range of benchmark datasets covering both image and text classification, and configured appropriate deep learning model architectures tailored to the characteristics and scale of each dataset. The specific experimental setups are as follows:

1. For the relatively lightweight image classification tasks of Fashion-MNIST (FMNIST) and CIFAR-10, we employed a classic Convolutional Neural Network (CNN) architecture. This network consists of three convolutional layers—each followed by a batch normalization layer and a ReLU activation function—and two fully connected layers.
2. For the CIFAR-100 and Tiny-ImageNet datasets, which feature a larger number of classes and higher image resolutions, we deployed ResNet-18 and Vision Transformer (ViT-Small), respectively, as the backbone networks.
3. For the DomainNet and Office-Home datasets—used to assess domain generalization and cross-domain transfer—we consistently utilized ViT-Small to evaluate FedHPP's generalization capability and robustness under conditions of data distribution shift.
4. For the AG News text classification task, we adopted a Transformer-based encoder architecture. The specific configuration of this model includes 12 attention heads, a word embedding dimension of 384, and a Transformer encoder module with a depth of 6 layers.

Parameter Settings: We run T=200 communication rounds with N=100 clients. Each client has 3/5 local training epochs. The learning rate is set to $0.001 - 0.05$ and $0.05 - 0.1$ respectively for FMNIST, CIFAR-10, CIFAR-100, Tiny-ImageNet, DomainNet, Office-Home, and AG News. The batch size is set to 16 for FMNIST, CIFAR-10, and CIFAR-100, and 32/64 for Tiny-ImageNet, DomainNet, and Office-Home. The target privacy parameters are setted $\epsilon = 2, \delta = 1e-6, C = 0.8$; Top-$K$ ratio is 0.5. We implement CKKS with TenSEAL, and their poly modulus degrees are set to 8192.

Baselines: To comprehensively evaluate the performance of the proposed method, the following representative federated learning baselines were selected for comparative experiments:

1. **Non-DP and Non-HE:** FedAvg (McMahan et al., 2017), the classic federated learning algorithm, achieves collaborative model updates through multiple rounds of local training and server-side weighted averaging, serving as the standard benchmark for assessing the fundamental performance of federated learning systems.
2. **DPFL:** DP-SGD (Abadi et al., 2016), DP-RSA (Zhu & Ling, 2022), and DP-BREM (Gu et al., 2025) are gradient-clipping-based DP mechanisms. DP-FedSAM (Shi et al., 2025) incorporates the SAM optimizer; by simultaneously minimizing loss value and loss sharpness, it generates locally smooth models that exhibit greater stability and robustness against weight perturbations. FedCEO (Li et al., 2025a) employs low-rank matrix factorization to compress gradient updates, effectively mitigating model performance degradation caused by noise injection while strictly adhering to the privacy budget, thereby achieving an optimal trade-off between privacy and utility. CLOVER (Xu et al., 2025) introduces a Top-K mechanism that reduces the overall privacy cost by adding noise only to a block of critical parameters. FedSPA (Liao et al., 2026) constructs a personalized DPFL framework based on subspace projection aggregation, enabling model personalization via low-dimensional subspace projection while simultaneously protecting client privacy.
3. **HEFL:** BatchCrypt (Zhang et al., 2020) proposes an aggregation strategy that encodes multiple plaintext gradients into a single ciphertext, effectively enhancing communication and computational efficiency under HE. FedPHE (Li et al., 2025b) combines the CKKS approximate HE with a Top-K parameter selection mechanism, reducing ciphertext operational overhead while ensuring the confidentiality of gradient transmission.

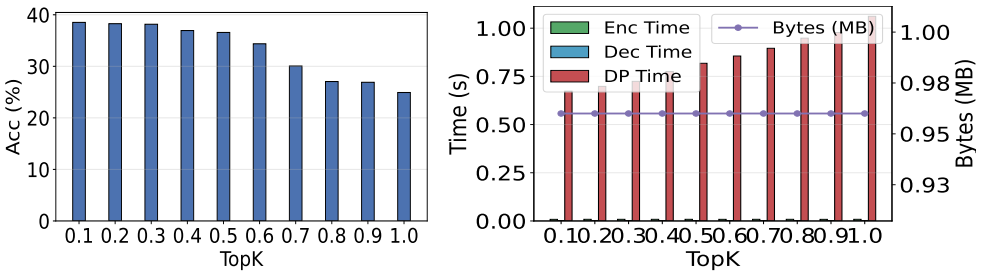
Metrics: To test the model performance, communication load, and computational efficiency of FedHPP, we use test accuracy, communication traffic, and time consumption to represent them, respectively. To test how well FedHPP resists privacy attacks (e.g., iDLG (Zhao et al., 2020),

InvG (Geiping et al., 2020), FedLeak (Fan et al., 2025), and CFGIA (Zhu et al., 2026); for the attack settings, we are keeping them similar to those of CFGIA.), we use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) as evaluation metrics. PSNR measures pixel-level distortion; a higher value indicates better reconstruction quality. SSIM assesses the structural similarity of an image by considering luminance, contrast, and structural components; a value closer to 1 signifies better reconstruction quality. LPIPS is a perceptual similarity metric that measures the distance between the deep feature representations of two images; a lower value indicates higher perceptual similarity.

D.2 Ablation Results

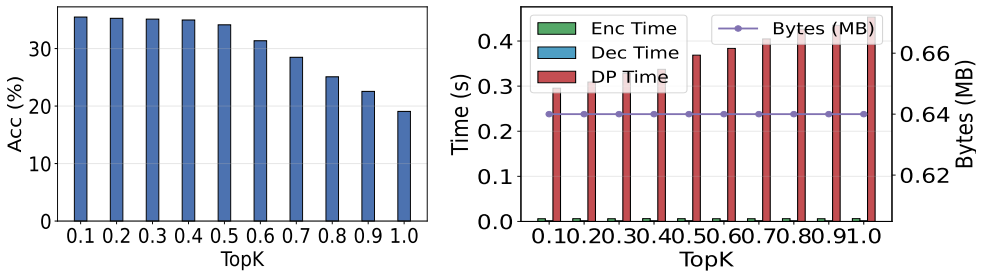
D.2.1 Top-K Ratio

As shown in Figs 3 and 4, FedHPP's model accuracy on both CIFAR-100 (IID) and Tiny-ImageNet (IID) exhibits a continuous decline as the Top-K ratio increases from 0.1 to 1.0. Specifically, in Fig 3a, the accuracy for CIFAR-100 drops from approximately 38.52% to 24.92%, while in Fig 4a, the accuracy for Tiny-ImageNet falls from about 35.48% to 19.07%. Concurrently, the DP time in both figures rises significantly with the Top-K ratio: increasing from roughly 0.67s to 1.02s for CIFAR-100 and from about 0.30s to 0.45s for Tiny-ImageNet.



(a) Accuracy (b) Time/Traffic

Figure 3: Impact of Different Top-K Ratios on Model Accuracy, Time, and Communication Traffic in CIFAR-100 (IID).



(a) Accuracy (b) Time/Traffic

Figure 4: Impact of Different Top-K Ratios on Model Accuracy, Time, and Communication Traffic in Tiny-ImageNet (IID).

In contrast, the Enc time, Dec time, and communication volume remain virtually constant and low across the entire Top-K range. The underlying reason is that a higher Top-K ratio results in a larger block of parameters being selected for privacy protection; this necessitates the injection of a greater total amount of DP noise, which exacerbates interference with model optimization, thereby reducing accuracy and increasing the time overhead for noise generation. Conversely, Enc time, Dec time, and communication volume depend solely on the scale of the encryption mask. Since FedHPP consistently performs HE on a fixed number of privacy masks, with neither the count nor the size of the masks changing with the Top-K ratio, the overhead for homomorphic encryption/decryption and the volume of ciphertext communication remain stable. This demonstrates that, under a fixed

differential privacy budget, FedHPP allows for a flexible trade-off between the scope of privacy protection and model utility by adjusting the Top-K ratio, while maintaining predictable encryption and communication overheads.

D.2.2 Parameter Selection Mechanism

Table 7 compares the model accuracy of FedHPP against two Top-K mechanisms—Top-K_{Rand} and Top-K_{Fix}—across three datasets (FMNIST, CIFAR-100, and Tiny-ImageNet) under both IID settings and varying levels of data heterogeneity ($\gamma = 0.3$ and 0.5). The results demonstrate that FedHPP achieves the highest accuracy across all configurations. For instance, on FMNIST, FedHPP attains accuracies of 90.41%, 88.57%, and 89.26% for the IID, $\gamma = 0.3$, and $\gamma = 0.5$ settings, respectively—outperforming Top-K_{Rand} and Top-K_{Fix} by ∼0.83%-2.35%.

Table 7: Impact of different Top-K mechanisms on model accuracy.

Methods	FMNIST			CIFAR-100			Tiny-ImageNet		
	IID	$\gamma = 0.3$	$\gamma = 0.5$	IID	$\gamma = 0.3$	$\gamma = 0.5$	IID	$\gamma = 0.3$	$\gamma = 0.5$
Top-K_{Rand}	88.06	87.11	88.27	33.52	29.58	31.46	32.21	27.57	31.98
Top-K_{Fix}	88.13	86.89	88.43	33.68	29.63	31.37	32.09	27.53	31.87
FedHPP	90.41	88.57	89.26	36.57	31.04	33.76	34.13	29.87	34.37

On the more challenging CIFAR-100, FedHPP achieves an IID accuracy of 36.57% (exceeding the baselines by ∼3%) and 33.76% at $\gamma = 0.5$ (leading by ∼2.3%). Similarly, on Tiny-ImageNet, FedHPP maintains a consistent lead, reaching 34.37% accuracy at $\gamma = 0.5$, which is roughly ∼2.4% higher than the baselines. These results clearly demonstrate that the advantages of FedHPP vary depending on the dataset and partitioning setup. The underlying reason is that FedHPP employs an importance metric based on the average magnitude of change in parameter blocks; this allows it to effectively identify and select blocks that are critical to model optimization for noise injection, thereby avoiding the ineffective perturbations associated with random or fixed selection strategies and better preserving model utility while ensuring privacy.

D.2.3 Different ϵ

Table 8 illustrates the performance of FedHPP in defending against CFGIA attacks and maintaining model utility across CIFAR-10 and Tiny-ImageNet datasets (under both IID and non-IID settings with $\gamma = 0.3$) at various privacy budgets (ϵ).

Table 8: The impact of different ϵ on model performance under CFGIA (Zhu et al., 2026).

Datasets	Metrics	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 5$	$\epsilon = 7$	$\epsilon = 10$
CIFAR-10 (IID)	Acc(%)	72.51	74.93	75.78	77.42	79.23
	PSNR	5.27	6.68	6.95	7.25	7.36
	SSIM	0.01	0.02	0.02	0.02	0.02
	LPIPS	0.81	0.52	0.48	0.48	0.48
CIFAR-10 ($\gamma = 0.3$)	Acc(%)	67.76	70.92	72.45	75.98	77.05
	PSNR	5.69	6.65	6.98	7.20	7.68
	SSIM	0.02	0.03	0.03	0.03	0.03
	LPIPS	0.84	0.77	0.76	0.73	0.73
Tiny-ImageNet (IID)	Acc(%)	31.27	34.13	35.23	35.78	36.02
	PSNR	5.41	6.43	6.74	6.95	7.17
	SSIM	0.01	0.01	0.01	0.02	0.02
	LPIPS	0.93	0.88	0.86	0.77	0.76
Tiny-ImageNet ($\gamma = 0.3$)	Acc(%)	26.69	29.87	30.57	31.73	32.49
	PSNR	6.14	6.40	6.54	6.67	6.51
	SSIM	0.01	0.01	0.01	0.01	0.02
	LPIPS	0.98	0.91	0.87	0.87	0.81

As ϵ increases from 1 to 10, model accuracy generally improves: rising from 72.51% to 79.23% for CIFAR-10 (IID) and from 26.69% to 32.49% for Tiny-ImageNet ($\gamma = 0.3$). Concurrently, reconstruction quality generally improves as ϵ increases, indicating weaker empirical protection against the evaluated attacks (e.g., for CIFAR-10 IID, PSNR rises from 5.27 to 7.36 and SSIM from 0.01 to 0.02), while LPIPS decreases (from 0.81 to 0.48). This indicates that reconstruction quality improves as ϵ increases, while privacy protection decreases. Conversely, at lower ϵ values (e.g., $\epsilon = 1$), the PSNR is merely 5.27, SSIM is as low as 0.01, and LPIPS is as high as 0.81; the attack fails to reconstruct meaningful information, signifying robust privacy protection, yet model accuracy remains at 72.51%, demonstrating that FedHPP preserves acceptable utility even under strict privacy constraints. These trends persist under data heterogeneity ($\gamma = 0.3$) and on the more complex Tiny-ImageNet dataset, confirming the method's robustness. The underlying reason is that FedHPP first selects a block of critical parameters using a Top-K approach and then adds noise specifically to these blocks; consequently, the amount of injected noise can be flexibly controlled by adjusting ϵ. A smaller ϵ results in more noise and stronger defense, whereas a larger ϵ yields less noise and superior model performance. This selective noise-injection mechanism effectively limits noise interference with the global model, enabling FedHPP to achieve an adjustable trade-off between privacy protection and model utility while maintaining stable defense and performance across diverse datasets and heterogeneity conditions.

D.2.4 Parameter Block Dimension B

Table 9 illustrates the impact of the block dimension B on FedHPP's performance. As B increases from 512 to 2048, the number of blocks J decreases, and the block granularity becomes progressively coarser. Observations show that Tra Time, DP Time, HE Time, Tra/DP (MB), and HE (MB) remain virtually unchanged across all models and datasets; this indicates that FedHPP's computational and communication overheads are insensitive to B, a result of the fact that only compact binary masks undergo HE, with the total volume of masks remaining relatively stable regardless of B. Regarding model accuracy, the CNN (CIFAR-10) achieves its peak at $B = 512$ (75.67%) and drops to 73.91% at $B = 2048$, whereas ResNet-18 (CIFAR-100) and ViT-Small (Tiny-ImageNet) attain their highest accuracies at $B = 2048$ (38.75% and 34.45%, respectively).

Table 9: Impact of different parameter block dimensions B (resulting in different generated block number/mask dimensions J) on FedHPP. IID and CFGIA (Zhu et al., 2026).

Metrics	**CNN (CIFAR-10)**			**ResNet-18 (CIFAR-100)**			**ViT-Small (Tiny-ImageNet)**		
	B=512	**B=1024**	**B=2048**	**B=512**	**B=1024**	**B=2048**	**B=512**	**B=1024**	**B=2048**
Acc(%)	75.67	74.93	73.91	36.87	36.57	38.75	33.89	34.13	34.45
Tra Time(s)	0.89	0.89	0.91	2.86	2.82	2.92	42.07	41.17	41.78
DP Time(s)	0.04	0.04	0.04	0.83	0.82	0.81	0.38	0.37	0.38
HE Time(s)(Enc/Dec)	0.004	0.004	0.004	0.012	0.012	0.012	0.008	0.008	0.008
Tra/DP(MB)	2.41	2.41	2.41	105	105	105	75.12	75.12	75.12
HE(MB)	0.32	0.32	0.32	0.96	0.96	0.96	0.64	0.64	0.64
PSNR	6.95	6.68	7.25	4.13	3.89	4.19	6.52	6.43	6.74
SSIM	0.02	0.02	0.01	0.02	0.02	0.02	0.01	0.01	0.01
LPIPS	0.75	0.52	0.81	0.78	0.81	0.74	0.86	0.88	0.87

In terms of privacy metrics, PSNR, SSIM, and LPIPS fluctuate with B; for instance, the CNN exhibits the highest PSNR (7.25) but also the highest LPIPS (0.81) at $B = 2048$, highlighting discrepancies among reconstruction quality metrics. Overall, the choice of B influences block granularity and the scope of parameters protected by Top-K selection—thereby affecting model utility and privacy strength—without significantly increasing system overhead. However, as long as the underlying HE scheme provides a sufficiently large slot capacity (e.g., CKKS with a polynomial modulus degree of $poly_modulus_degree = 8192$, slots $N_{\text{slots}} = 4096$) and the total packed data volume stays within the limit of a fixed number of ciphertexts, the final serialized ciphertext size remains unchanged after layer-by-layer padding or alignment rounding. Consequently, FedHPP allows for a flexible trade-off between privacy, utility, and efficiency through the adjustment of B.

E MORE DISCUSSIONS AND FUTURE DIRECTIONS

A core motivation for federated learning is enabling collaborative modeling without sharing raw data; however, existing privacy-preserving schemes struggle to optimize model security, utility, and communication efficiency simultaneously. We explore a collaborative mechanism—FedHPP—that combines selective differential privacy with lightweight homomorphic encryption, offering an effective way to balance these three factors. In subsequent research, we will further investigate this hybrid privacy-preserving mechanism. First, we will address more complex, realistic threat models—including malicious servers, multi-client collusion, and adaptive gradient inversion—and analyze the combined security boundaries formed by integrating selective perturbation and encryption masking under varying attack capabilities. Second, we will refine the unified theoretical framework for differential privacy and homomorphic encryption by integrating factors such as dynamic privacy budget allocation, noise magnitude, and ciphertext overhead, and exploring the quantitative relationships between privacy, utility, and efficiency across protection levels. Finally, future research will incorporate mechanisms such as decaying step sizes and dynamic noise scheduling to establish tighter finite-time convergence bounds and privacy accounting methods; simultaneously, we will extend this approach to practical deployment environments (e.g., scenarios involving large-scale clients and dynamic participation) to systematically evaluate the security, scalability, and long-term training stability of hybrid privacy-preserving federated learning.