Communication-Efficient Federated Learning for Edge Computing with Gradient Leakage Defense

Xihong Yang, Haixia Cui, *Senior Member, IEEE*, Feipeng Dai, Bo Xie, Yejun He, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Federated learning (FL) has emerged as a promising paradigm for privacy-preserving model training across distributed edge devices, enabling local data utilization without explicit sharing. However, in edge computing environments characterized by heterogeneous resources and intermittent connectivity, FL remains vulnerable to gradient leakage attacks (GLA), where adversaries reconstruct private data from shared model updates. Although the existing defenses, such as differential privacy (DP) and gradient compression, offer partial mitigation, they often result in significant performance degradation or increased communication overhead. In this paper, we analyze that the risk of privacy leakage is highly sensitive to the client-side training configurations and gradient magnitudes. Based on this, we propose a risk-aware FL framework tailored for the edge scenarios, which not only performs per-device privacy risk assessment but also introduces subtractive dithering quantization to the inject controllable Gaussian noise into local models. Additionally, a noise-aware aggregation strategy is presented by adjusting each client's contribution to preserve the global model utility. Experimental results on FashionMNIST and CIFAR-10 demonstrate that the proposed framework achieves strong defense against the GLA, reduces the communication costs by over 50%, and maintains the competitive accuracy.**

***Index Terms*—Gradient leakage attack, communication efficiency, dithering quantization, federated learning.**

I. Introduction

The integration of edge computing and federated learning (FL) provides essential technical support for enabling continual learning in edge network architectures [1]. Edge computing can mitigate latency and bandwidth bottlenecks by offloading computational tasks to devices located closer to data sources while FL can enable collaborative model training across distributed devices without requiring data centralization, thereby enhancing data privacy protection. However, deploying FL in edge environments has several significant challenges, among which the communication constraints, limited computational resources, and privacy preservation are the most critical [2]–[4].

This work was supported in part by the GuangDong Basic and Applied Basic Research Foundation under Grant 2024A1515012052, in part by the National Key Research and Development Program of China under Grant 2023YFE0107900, in part by the National Natural Science Foundation of China under grants 61871433, 61828103, 61201255, and 62071306.

X. Yang, H. Cui, F. Dai, and B. Xie are with the School of Electronic Science and Engineering (School of Microelectronics), South China Normal University, Foshan 528225, China (e-mail: 2023025108@m.scnu.edu.cn, cuihaixia@m.scnu.edu.cn, daifeipeng@126.com, bo.xie@m.scnu.edu.cn).

Yejun He is with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: heyejun@126.com).

Mohsen Guizani is with the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi 99163, UAE (email: mguizani@ieee.org).

Edge devices typically communicate with servers via wireless networks, which are inherently limited in bandwidth and prone to instability. In FL, the frequent data transmission, such as model parameters or gradients, is required throughout the training process. When the model size is large or the number of participating edge devices is high, the resulting communication overhead can become substantial, potentially leading to the dropout of bandwidth-constrained clients. To alleviate these communication bottlenecks, a variety of optimization strategies have been proposed in recent years to enhance the practicality and efficiency of FL in edge environments. The sparsification technology can reduce the model's size by either randomly selecting [5] or retaining the top k parameters with the largest magnitudes [6], [7]. Due to the high parameter redundancy in deep neural networks (DNNs), these methods can reduce the communication overhead by two to three orders of magnitude without severely affecting the training performance.

In contrast to the sparsification, the quantization approximates parameters with lower precision, reducing the number of bytes required for transmission. Although the model compression significantly enhances the transmission efficiency, it inevitably introduces noise. Introducing artificial noise is an effective way to enhance the privacy protection [8]. Some existing studies have investigated the impact of sparsification on privacy protection in [9], [10]. Their findings suggest that even with a pruning rate of 90%, the effective privacy protection cannot be achieved. Besides, the statistical analysis in [11] reveals that the subtractive dithering quantization is equivalent to injecting Gaussian noise into the original gradient, thereby eliminating the need for additional artificial noise for privacy protection. However, the fixed noise intensity introduced by this method often leads to significant degradation in model performance.

Moreover, edge devices are typically constrained in computational capacity and exhibit significant hardware heterogeneity, which brings out lots of difficulties to the training efficiency and stability of FL systems. Therefore, how to alleviate the local computational burden is critical for the successful deployment of FL in edge environments. One common approach is to adopt compact models or apply model compression techniques, such as pruning and quantization, to reduce the computational and storage demands. Additionally, allowing devices to select local model architectures according to their computational capabilities has emerged as another effective strategy. The core idea is to enable the resource-limited devices to train only a subset of the global model

parameters, while more powerful devices train the full model. Yet, aggregating heterogeneous model structures is non-trivial and often requires knowledge distillation for the effective model fusion. Similarly, the models can be partitioned into two segments: the lightweight components (e.g., classification layers) are trained on edge devices, while the computationally intensive parts (e.g., feature extractors) are trained on nearby edge servers or in the cloud followed by recombination and update. Finally, to address the disparities in device training speeds, asynchronous FL or intelligent scheduling strategies can be employed to improve the system throughput and robustness. Nevertheless, asynchronous updates may introduce stale gradients, thereby complicating the theoretical convergence analysis of the learning process.

In addition, in edge computing environments, the large number and wide distribution of participating entities introduce substantial risks of sensitive information leakage during the communication and coordination. Some studies in [12]–[16] have shown that even without directly sharing the raw data, merely exchanging locally computed gradients or model parameters can enable adversaries to reconstruct the private training data, thereby compromising the user privacy. To address these privacy and security problems, several existing defense mechanisms have been proposed to mitigate the aforementioned attacks, including the homomorphic encryption [17], secure multi-party computation [18], differential privacy (DP) [19], and gradient compression [20], [21]. The first two methods, based on cryptographic principles, can prevent the privacy leakage while preserving the model performance. However, they are computationally intensive and memory-demanding which makes them impractical for the resource-constrained IoT devices.

Therefore, the DP-FL has emerged as the predominant privacy-preserving technique. By converting the random quantization noise into the controllable artificial one, it can help to address the training efficiency challenge in differential privacy. There typically involves injecting artificial noise, such as Gaussian or Laplacian noise, during local training or server aggregation to obscure sensitive information. Although some existing studies in [22], [23] argued that DP-FL could not withstand the generative adversarial network(GAN)-based data reconstruction attacks, the attack settings and privacy budgets adopted in these works were often overly idealized, making them insufficient to faithfully reflect the privacy leakage in realistic conditions. Further, the effectiveness of DP-FL also depends on the noise magnitude, offering simplicity in implementation and scalability for the large-scale training. It is crucial to highlight that the privacy protection often comes at the expense of training performance. The noise introduced to safeguard privacy induces biases in the gradient updates, which can result in convergence issues and a decline in model accuracy. An ideal privacy-preserving algorithm for distributed learning should not only provide strong privacy guarantees but also ensure minimal impact on model performance. Since the DP-FL adjusts noise intensity based on the user's privacy budget to meet the DP requirements, this approach fails to adequately assess the actual risk when faced with some novel attacks, such as gradient leakage attack (GLA). The authors indicate that the heterogeneous local settings of users (e.g., batch size and local epochs) contribute to varying the levels of information leakage risk in [24]. Additionally, the studies in [15], [25] indicate that the models at different stages of training demonstrate varying levels of resistance to GLA. Specifically, the difficulty of launching an attack evolves with the number of communication rounds, underscoring the importance of designing defense mechanisms that are adaptive to the model's training state.

To address the above concerns, we analyze the fundamental mechanisms by which the gradient leakage occurs in FL and uncover a critical relationship between the data reconstruction vulnerability and local training configurations. Our study reveals that the clients with small batch sizes and few local epochs tend to generate gradient updates with higher signal-to-noise ratios, making them substantially more susceptible to the gradient inversion attacks. Notably, this vulnerability persists even in later stages of training when the global model is near the convergence. These findings suggest that the leakage potential is not merely a byproduct of the model architecture or dataset characteristics, but is also deeply rooted in how the training is scheduled on each device. The existing defense strategies, which apply fixed noise levels [11] or uniform quantization schemes across clients [26], fail to account for this variability and are therefore insufficient in protecting users with more vulnerable configurations. This motivates the development of a more nuanced defense mechanism that dynamically adapts to the privacy risk associated with each client's training context. Specifically, we first estimate the potential leakage risk on each client by analyzing the gradient norms in relation to its local training parameters. Based on this risk, we employ the subtractive dithering quantization to inject the controllable Gaussian noise into local models, ensuring both the privacy protection and communication efficiency. To further reduce the adverse effects of noise on model performance, we present an aggregation strategy that adjusts each client's contribution according to its noise level.

In summary, our main contributions in this paper are as follows:

- We analyze that the risk of gradient-based data leakage in FL is closely tied to the local training configurations and the model's convergence state, revealing non-uniform privacy vulnerabilities across clients.
- We propose a novel FL algorithm that integrates the privacy risk assessment with subtractive dithering quantization. Specifically, our method dynamically calibrates per-device noise scales based on the gradient norms and training settings, and incorporates a noise-aware aggregation scheme to mitigate the negative impact of noisy updates. This framework achieves a joint improvement in the privacy, communication efficiency, and model accuracy.
- We evaluate the performance of the proposed algorithm via extensive experiments on benchmark datasets which demonstrate its effectiveness in enhancing the privacy protection, preserving accuracy, and significantly reducing the communication costs.

The remainder of this paper is organized as follows. Section II introduces the related work on GLA defense and communication efficiency optimization. Section III presents the mechanisms of FL and GLA, as well as the theorem for subtractive dithering quantization. Section IV analyzes the factors influencing GLA, and Section V provides the optimization objectives and algorithm design. Simulation results are presented in Section VI. Finally, Section VII is the conclusion drawn from this work.

II. Related Work

A. Defenses Against Gradient Leakage Attacks

In FL, GLA typically assumes an honest-but-curious adversary, such as a central server or participating devices, which follows training protocol but attempts to reconstruct user data from exchanged gradients or model parameters. GLAs can be broadly categorized into two classes: optimization-based attacks [27], which iteratively minimize the discrepancy between synthetic and real gradients, and linear equation-based attacks [28], which exploit the deterministic relationship between model updates and input data.

To counter such threats, various defense mechanisms have been proposed. A prominent category involves secure multi-party computation (SMPC). For instance, Li et al. in [29] introduced a chain-based SMPC protocol utilizing a single-mask and sequential communication scheme, achieving a promising balance between privacy and model utility. However, the computational overhead introduced by SMPC protocols remains prohibitive for edge devices with limited resources.

Another major line of defense involves DP-FL. Hu et al. in [25] proposed a dynamic privacy budgeting scheme, using the model's performance gain per round as a proxy for privacy risk. This approach adjusts the noise variance and clipping thresholds over time based on DP theory. While this method captures the evolving nature of privacy risk, it relies heavily on a predefined global privacy budget, limiting its adaptability to diverse client behaviors and non-IID data distributions. Li et al. in [30] proposed injecting noise selectively into critical model layers, determined by feature importance scores, thereby enhancing interpretability while reducing unnecessary perturbation. However, this layer-wise noise strategy introduces trade-offs in training efficiency and often increases communication overhead. The authors in [31] implemented the per-example-based client DP and proposed a dynamic noise decay strategy to optimize the privacy-utility trade-off. However, they ignored accounting for the impact of heterogeneous training settings on leakage risks. Additionally, the per-example noise addition introduced significant computational overhead.

B. Model Compression for Communication-Efficient FL

Given the high dimensionality of DNNs, the communication efficiency remains a critical bottleneck in the large-scale FL deployments. Some model compression methods, particularly sparsification and quantization, aim to reduce the size of transmitted updates without compromising the model performance.

The gradient sparsification techniques [32], [33] often transmit only the most significant gradients, assuming the error feedback mechanisms can recover the remaining information over time. This approach can drastically reduce the communication volume. But its impact on privacy is ambiguous. For example, one study in [10] suggested that the sparsity levels above 30% can effectively obfuscate the sensitive data, whereas some others [34], [35] found that even with 80% sparsity, the data reconstruction remains feasible. These conflicting results may arise from variations in model complexity and data heterogeneity, which influence the parameter redundancy. Furthermore, the sparsity is often treated as a fixed hyperparameter, determined empirically rather than adaptively.

The model quantization reduces the precision of each parameter, thereby lowering the total number of bits required per update. In [36], the differentiated quantization precision was jointly optimized based on gradient norms and heterogeneous communication capabilities, effectively mitigating the straggler effect. While, the optimization focuses solely on communication efficiency, without analyzing the impact of quantization on privacy leakage risks. In [35], an autoencoder-based method was used to perform the learned quantization, adaptively injecting perturbations based on latent representations. Yet, due to the inherent complexity and non-determinism of encoder-decoder architecture, controlling the induced noise is difficult, often resulting in unstable convergence. In contrast, the authors in [11] introduced a subtractive dithering quantization scheme that analytically transforms quantization noise into zero-mean Gaussian noise with controllable variance. This method not only reduces the communication cost but also aligns with the established DP techniques. Despite its promise, the subtractive dithering quantization has been shown to degrade the accuracy by up to 30% on certain datasets, particularly when the uniform quantization levels are applied across heterogeneous devices. This highlights the necessity of personalized quantization strategies that take into account the individual device conditions, privacy risks, and communication constraints.

Although the existing defense and compression strategies have made substantial progress, they still tend to address either privacy or communication efficiency in isolation. The SMPC offers strong protection but is often impractical in real-world deployments. The DP-FL methods can adaptively protect privacy but struggle with static budgets and limited scalability. The sparsification and quantization reduce the communication costs, but their privacy-preserving effects are inconsistent or require costly tuning. In contrast, our proposed approach integrates the privacy risk assessment directly into the quantization process. By analyzing the gradient norms and an intrinsic indicator of privacy vulnerability, we dynamically adjust the quantization noise level for each client and assign the noise-aware aggregation weights.

III. Preliminary

In this section, we introduce some background knowledge relevant to the workflow of FL, the principles of GLA, and the subtractive dithering quantization algorithm. Figure 1 illustrates the process of FL training and GLA, and demonstrates the protective effect of model quantization on data.

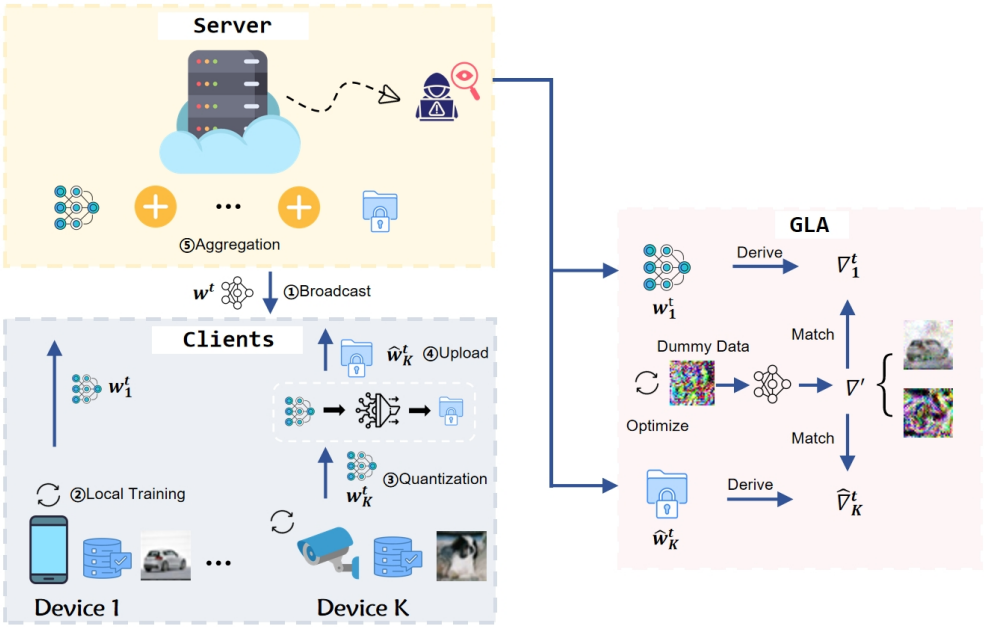

Fig. 1: Framework of FL and comparison of effects of GLA. Device 1 directly uploads local model while Device K uploads model after quantization.

A. FL System

Our FL considers a system composed of a set of clients indexed by $\mathcal{K} = 1, 2, \ldots, k, \ldots, K$, where each client k possesses a local dataset denoted as $\mathcal{D}_k = \{(\boldsymbol{x}, \boldsymbol{y})\}$, with $\boldsymbol{x}$ representing the input features and $\boldsymbol{y}$ denoting the corresponding labels. Under the orchestration of a central server, these clients collaboratively train a shared global model $f(\boldsymbol{w})$, where the model parameter $\boldsymbol{w}$ resides in the parameter space $\mathbb{R}^m$.

To enable the collaborative model training across the distributed clients, the FL leverages the additive structure of global loss function, which permits decomposing the optimization objective into client-local subproblems. Specifically, the overall goal is to minimize a global empirical risk function defined as:

$$\min_{\boldsymbol{w}\in\mathbb{R}^m} \mathcal{L}(\boldsymbol{w}) := \frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_k(\boldsymbol{w}, \mathcal{D}_k), \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^m$ denotes the model parameters to be optimized, and $\mathcal{L}_k(\boldsymbol{w}, \mathcal{D}_k)$ represents the local loss evaluated over the client k's dataset $\mathcal{D}_k$. In the classification tasks, this local loss is typically instantiated as cross-entropy.

The additive form of the objective function in (1) allows the global optimization to be achieved via decentralized computation. That is, each client minimizes its own local loss function and contributes to the overall objective through aggregation. During each communication round t, the central server disseminates the current global model $\boldsymbol{w}^{t-1}$ to the selected clients. Each client then performs the local updates using the stochastic gradient descent (SGD) over mini-batches of its local data. The gradient of local loss is computed as:

$$\boldsymbol{G}_k^{t-1} = \nabla_{\boldsymbol{w}} \left(\frac{1}{B_k} \sum_{i=1}^{B_k} \mathcal{L}_i(\boldsymbol{w}^{t-1}, \mathcal{B}_k) \right), \tag{2}$$

where $\mathcal{B}_k \subset \mathcal{D}_k$ is a mini-batch sampled from client k's dataset, and B_k is the batch size. The local model is then updated as:

$$\boldsymbol{w}_k^t = \boldsymbol{w}^{t-1} - \eta \boldsymbol{G}_k^{t-1}, \tag{3}$$

where η is the local learning rate. After completing the local updates, each client transmits the updated model $\boldsymbol{w}_k^t$ to the central server. Exploiting the linearity of loss function, the server aggregates the updates typically by weighted averaging to construct the new global model $\boldsymbol{w}^t$, which serves as the initialization for the next round of training.

The key of aggregation lies in computing a weighted average of the locally updated model parameters received from participating clients. It can be formally expressed as:

$$\boldsymbol{w}^t = \sum_{k=1}^{K} \gamma_k \boldsymbol{w}_k^t, \tag{4}$$

where γ_k denotes the aggregation weight assigned to client k. In the canonical Federated Averaging (FedAvg) algorithm [2], these weights are typically set in proportion to the relative size of each local dataset, i.e., $\gamma_k = |\mathcal{D}_k| / |\mathcal{D}|$, where $|\mathcal{D}|$ represents the total number of training samples across all clients.

This sequence of broadcasting, local updating, and centralized aggregation constitutes a single communication round. Through repeated execution of this process over T communication rounds, the global model can be iteratively refined. Under some appropriate conditions on the loss function and

learning rates, the aggregated model $\boldsymbol{w}^T$ is guaranteed to converge to a stationary point of the global objective.

B. Privacy Threats Model

Recent studies have shown that even when raw data are not directly exposed, adversaries can still infer private information from shared model updates through optimization-based techniques. One of the most representative threats is the GLA. We illustrate the process that GLA in the following.

In this setting, the server is modeled as an honest-but-curious adversary that attempts to reconstruct the private input data based on the observed gradients. The attacker begins by randomly initializing a pair of dummy inputs (x', y') and computes a corresponding dummy gradient using the same loss function as the client:

$$\boldsymbol{G'} = \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, x', y'). \tag{5}$$

To approximate the true client gradient $\boldsymbol{G}$, the attacker optimizes the dummy input by minimizing the discrepancy between $\boldsymbol{G'}$ and $\boldsymbol{G}$:

$$x', y' = \arg\min_{x', y'} \mathbb{J}(\boldsymbol{G'}, \boldsymbol{G}), \tag{6}$$

where $\mathbb{J}(\cdot,\cdot)$ denotes a gradient distance metric, typically instantiated as the squared ℓ_2 norm $\|\boldsymbol{G'} - \boldsymbol{G}\|^2$ or the cosine similarity loss $1 - \frac{\langle \boldsymbol{G'}, \boldsymbol{G} \rangle}{\|\boldsymbol{G'}\| \|\boldsymbol{G}\|}$. The optimization algorithms, such as L-BFGS and Adam, have been widely adopted to solve this inverse problem effectively [37].

The empirical results suggest that the high-fidelity reconstructions can be achieved even for the deep models, such as LeNet and ResNet-18. Moreover, the difficulty of reconstructing private data does not necessarily scale with the model complexity, highlighting the pervasive risk of gradient leakage in FL. These findings underscore the necessity of integrating robust privacy-preserving mechanisms to safeguard against such attacks.

C. Quantization Method

To address the privacy risks associated with directly transmitting model parameters in FL, we propose a subtractive dithering quantization algorithm to perturb the local model updates which leverages carefully the designed random variables to transform the quantization noise into the controllable Gaussian noise [11]. The following two lemmas present the theoretical foundations for the quantization process [38]–[40].

Lemma 1. *Let $V \sim \Gamma\left(\frac{3}{2}, \frac{1}{2}\right)$ be a gamma-distributed random variable with shape parameter $\frac{3}{2}$ and rate parameter $\frac{1}{2}$. If the conditional distribution of X is given by $(X \mid V = v) \sim$ Unif $(\mu - \sigma\sqrt{v}, \mu + \sigma\sqrt{v})$, then the marginal distribution of X satisfies $X \sim \mathcal{N}(\mu, \sigma^2)$.*

Proof: The probability density function of the random variable V is given by:

$$f_V(v) = \frac{v^{\frac{1}{2}}}{\Gamma\left(\frac{3}{2}\right)} \left(\frac{1}{2}\right)^{\frac{3}{2}} e^{-\frac{v}{2}} = \sqrt{2v} e^{-\frac{v}{2}} \quad \text{for } v > 0. \tag{7}$$

The conditional distribution $(X|V = v) \sim$ *Unif*$(\mu - \sigma\sqrt{v}, \mu + \sigma\sqrt{v})$ implies that for a given value of $V = v$, X is uniformly distributed between $\mu - \sigma\sqrt{v}$ and $\mu + \sigma\sqrt{v}$. Thus, we have:

$$f_{X|V}(x|v) = \frac{1}{2\sigma\sqrt{v}} \quad \text{for } \mu - \sigma\sqrt{v} \le x \le \mu + \sigma\sqrt{v}. \tag{8}$$

To find the marginal distribution of X, we integrate over the distribution of V:

$$f_X(x) = \int_0^\infty f_{X|V}(x|v) f_V(v) dv = \frac{1}{\sigma} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \tag{9}$$

Then, we obtain a Gaussian distribution:

$$X \sim \mathcal{N}(\mu, \sigma^2).$$

■

Lemma 2. *Given a quantization function $Q(\cdot)$ with step size Δ, and a random variable $U \sim$ Unif$(-\frac{\Delta}{2}, \frac{\Delta}{2})$, for any scalar Y, define $\hat{Y} = Q(Y + U) - U$. Then, $\hat{Y} = Y + U'$, where U' is an independent random variable with $U' \sim$ Unif$(-\frac{\Delta}{2}, \frac{\Delta}{2})$.*

Proof: For any scalar Y, let $U \sim$ *Unif*$(-\Delta/2, \Delta/2)$ and define $Z = Y + U$. Then Z is uniformly distributed over an interval of length Δ. The quantization error $U' = Q(Z) - Z$ is uniformly distributed on $(-\Delta/2, \Delta/2)$ because the quantization function Q is periodic with period Δ, and the input Z is uniform over an interval of length Δ. Now, $\hat{Y} = Q(Y + U) - U = Q(Z) - U$. Since $Q(Z) = Z + U'$, we have:

$$\hat{Y} = (Z + U') - U = (Y + U + U') - U = Y + U'. \tag{10}$$

Thus, $\hat{Y} = Y + U'$ where $U' \sim$ *Unif*$(-\Delta/2, \Delta/2)$. Moreover, U' is independent of Y. ■

Based on these principles, the implementation of subtractive dither quantization proceeds as follows. Let w_j denote the j-th component of the model parameter vector $\boldsymbol{w}$, where $j \in [m]$:

1) Each client samples $V_j \sim \Gamma\left(\frac{3}{2}, \frac{1}{2}\right)$ and computes the quantization step size $\Delta_j = 2\sigma\sqrt{V_j}$, where σ is a tunable noise parameter.
2) The client adds uniform noise $U_j \sim$ *Unif*$(-\frac{\Delta_j}{2}, \frac{\Delta_j}{2})$ to w_j to obtain $\tilde{w}_j = w_j + U_j$. By **Lemma** 1, this effectively introduces zero-mean Gaussian noise with variance σ^2.
3) The perturbed value is quantized using:
$$Q(x) \triangleq \left\lceil \frac{x - \Delta_j/2}{\Delta_j} \right\rfloor \Delta_j + \frac{\Delta_j}{2}, \tag{11}$$
where $\lceil \cdot \rfloor$ denotes rounding to the nearest integer.
4) Using a shared random seed, the server reconstructs U_j and removes the dither by computing $\hat{w}_j = Q(w_j + U_j) - U_j$. According to **Lemma** 2, this yields $\hat{w}_j = w_j + U'_j$, where U'_j is an independent uniform variable with zero mean and variance σ^2. Repeating this process for all $j \in [m]$ results in a quantized model $\hat{\boldsymbol{w}}$ that is statistically equivalent to the original $\boldsymbol{w}$ plus IID Gaussian noise.

In the typical edge computing settings, the model parameters are stored and transmitted using 32-bit floating-point representations, which can lead to excessive communication overhead in bandwidth-constrained environments. The gradient quantization can significantly reduce this cost. For the parameters

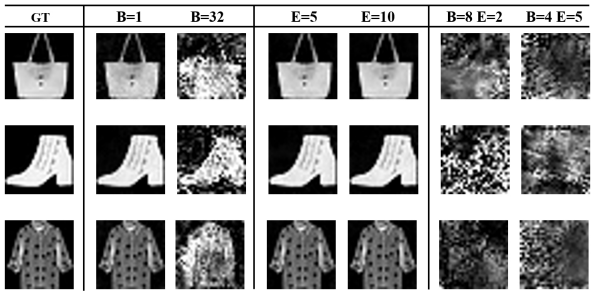

Fig. 2: Reconstruction results of different local training configurations. GT denotes the original ground truth image.

bounded by $|w_j| \leq C$, the number of bits required to transmit w_j is:

$$b_j = \left\lceil \log_2 \left(2 \cdot \left\lceil \frac{C}{\Delta_j} \right\rceil + 1 \right) \right\rceil, \tag{12}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. Compared to the traditional differential privacy methods that rely on the Gaussian noise injection, the proposed quantization framework offers superior communication efficiency while preserving privacy, which is particularly valuable in resource-constrained edge computing applications.

IV. Assessment of Privacy Leakage Risk

A. The Pivotal Role of Training Configurations

The core principle of GLA lies in inferring the private data from the gradient information embedded in locally trained models. The amount of gradient information exposed directly affects the success rate of such attacks. In the FedAvg algorithm, each communication round entails $E \cdot |\mathcal{D}_k| / B$ local SGD steps, where E is the number of local epochs and B denotes the batch size. Typically, clients upload either their local models $\boldsymbol{w}_k$ or the model updates $\Delta \boldsymbol{w}_k$ rather than raw gradients. However, given that the server generally has access to local training hyperparameters (e.g., learning rate and batch size), it can reconstruct the gradient information using (3).

Notably, the most widely adopted GLA methods, such as InvertGrad [15] and DLG [10], achieve their best reconstruction performance under the configuration setting $|\mathcal{D}_k| = E = B = 1$, which maximizes the privacy leakage risk. However, such configurations are rarely used in practice due to their inefficiency and high communication cost. Consequently, the attack results under these idealized assumptions do not accurately reflect the practical vulnerability of FL systems, necessitating a reassessment under the realistic training settings.

To systematically evaluate the impact of local training configurations on GLA effectiveness, we conduct experiments on the FashionMNIST dataset using a convolutional neural network(CNN). Each attack is run for 20,000 optimization steps to ensure convergence. Due to its superior performance in terms of reconstruction speed and image fidelity, InvertGrad is adopted as the default attack method. We follow the original InvertGrad open-source implementation, where $|\mathcal{D}_k| = B$ is assumed.

Figure 2 presents the reconstruction results under varying the local configurations, with the default setting being $B = E = 1$. As shown, when only a single image is involved, the high-quality reconstructions are possible even with a large number of local epochs—an observation consistent with prior work. As the batch size B increases, the quality of reconstructed images declines though salient features and object contours often remain identifiable, indicating a residual privacy risk under practical settings. However, when both $B > 1$ and $E > 1$, as is common in real-world FL scenarios, the reconstruction attempts generally fail.

In the federated edge computing deployments, the device-specific constraints often lead to personalized training configurations. This heterogeneity implies that the risk of privacy leakage varies across devices. Our analysis results suggest that the severe leakage primarily occurs under some specific constrained conditions. Therefore, the global application of defense mechanisms may be unnecessary. Instead, the targeted protection strategies can be applied selectively to the high-risk devices, optimizing both security and system efficiency.

B. The Security of Converged Model

As the model approaches convergence, the magnitudes of its gradients diminish, thereby reducing the effectiveness of gradient-based data reconstruction attacks [25]. This observation appears consistent with the underlying mechanism of GLA, which relies on the gradient information to infer the private data. However, our prior experiments revealed the noticeable variation in the quality of reconstructed images, suggesting that both image characteristics and gradient magnitudes may influence the success of such attacks.

To investigate the impact of model convergence and image features on the GLA performance, we conduct some experiments across different stages of training. Specifically, we select some input images and perform the GLA using gradients extracted from the models at various training rounds. Here, R0 denotes the untrained model, while R200 corresponds to a fully converged model after 200 rounds of FL. For each case, we computed the ℓ_2 norm of the extracted gradients and evaluated the quality of reconstructed images.

As shown in Figure 3, the ℓ_2 norms of gradients significantly decrease as the training progresses. Nevertheless, the high-quality image reconstruction remains feasible even from a converged model. On the relatively simple MNIST dataset, the GLA performance appears largely unaffected by the model convergence. In contrast, for the more complex FashionMNIST dataset, the reconstruction failures tend to occur on the inputs with particularly low-gradient magnitudes. These results suggest a positive correlation between the gradient norm and the privacy leakage risk.

Since the data distributions across clients are typically non-identical and independent, the privacy risk assessments should be performed individually based on each client's gradient characteristics. In the traditional DP-FL frameworks [41], it is common to allocate a global privacy budget across rounds and devices, adjusting the noise levels accordingly. However, such static parameter-driven strategies may misestimate the privacy

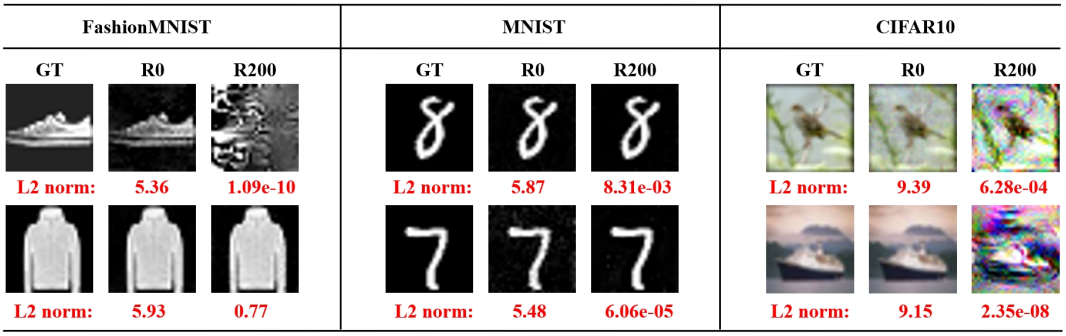

Fig. 3: The performance of GLA on different model states. Test images are sampled from the benchmark datasets (FashionMNIST, MNIST and CIFAR10).

risks when facing the adaptive threats like GLA. Moreover, the excessive noise injection can severely compromise the model utility, which highlights the need for the gradient-aware fine-grained privacy mechanisms tailored to the actual leakage risks.

V. Algorithm Design

A. Design Objectives

The previous sections have discussed the privacy leakage risks in FL and analyzed how the local training configurations affect the performance of GLA. The empirical results indicate that increasing batch size and number of local iterations can substantially reduce the effectiveness of such attacks. We also observe that the privacy assessment strategies based solely on the model convergence fail to accurately capture the actual data leakage risk. These findings motivate the development of a more reliable privacy risk evaluation mechanism grounded in the operational principles of GLA.

While the various noise-based defense strategies have demonstrated effectiveness [20], [26], they often rely on the high noise intensities which significantly delay the model convergence, leading to excessive communication overhead—particularly detrimental in the bandwidth-constrained FL environments. To balance the privacy preservation with the communication efficiency, we propose a lightweight quantized FL framework.

Specifically, we employ subtractive dither quantization on the local model updates, transforming the quantization noise into the controllable Gaussian noise [11]. This not only enhances the privacy protection but also reduces the volume of transmitted data, thereby lowering the communication costs. Furthermore, by incorporating the gradient-based privacy risk assessment, the framework can adaptively defend against the GLA with the minimal noise injection, ensuring both efficiency and robustness.

B. Leakage Risk Quantification Mechanism

Figure 4 illustrates the evolution of gradient magnitudes across communication rounds for three different batch sizes. When $B > 1$, the gradient norms generally decrease as the model converges, reducing the amount of information available

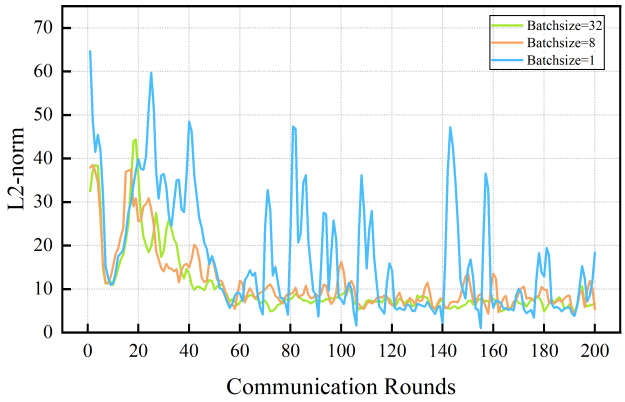

Fig. 4: Gradient norms for different batch sizes.

to potential adversaries and thereby hindering the gradient-based reconstruction. In contrast, when $B = 1$, the gradient norm exhibits greater fluctuations and remains close to its initial value even after convergence. As previously noted, under the idealized attack conditions, a converged model does not necessarily prevent the effective gradient leakage.

Given the strong correlation between gradient magnitude and privacy risk, we define a normalized privacy risk score as:

$$R_k = \frac{\|\boldsymbol{G}_k\|}{\|\boldsymbol{G}_{\max}\|} \cdot \frac{1}{B^E}, \tag{13}$$

where $R_k \in [0, 1]$ represents the estimated privacy risk of client k, and $\|\boldsymbol{G}_{\max}\|$ is the maximum observed gradient norm during training, reflecting the highest potential leakage. Although the gradient norms from different configurations may occasionally appear similar, their actual privacy risks differ significantly, especially when the batch size and local epoch count vary. To account for this discrepancy without overestimating the required noise level, we introduce a decay factor based on the batch and epoch settings.

The noise scale for client k is then computed as:

$$\sigma_k = R_k \cdot \sigma_{\max}, \tag{14}$$

where $\sigma_{\max}$ is a predefined upper bound on noise magnitude. σ is closely related to the quantization precision. The lower quantization precision corresponds to a larger σ, which in

turn indicates the stronger privacy protection capabilities. Therefore, we take σ corresponding to the minimum quantization precision, i.e., 1-bit, as the maximum noise scale, approximately equal to 0.01. In practical DLG attacks, it has been shown that at this noise intensity, the recovered data is nearly indistinguishable, indicating that this noise is both sufficient and overly strong. In scenarios where a client requires substantial noise to preserve the privacy, the naive averaging during aggregation may degrade the global model performance.

To address this issue, we propose a noise-aware aggregation scheme that adaptively adjusts weights to mitigate the impact of noisy updates:

$$\gamma_k = \frac{1}{\sigma_k + \epsilon} \Big/ \sum_k \frac{1}{\sigma_k + \epsilon}, \tag{15}$$

where ϵ is a small constant to ensure numerical stability. This weighting mechanism prioritizes updates with lower noise, thereby enhancing the robustness while preserving the privacy-utility trade-off.

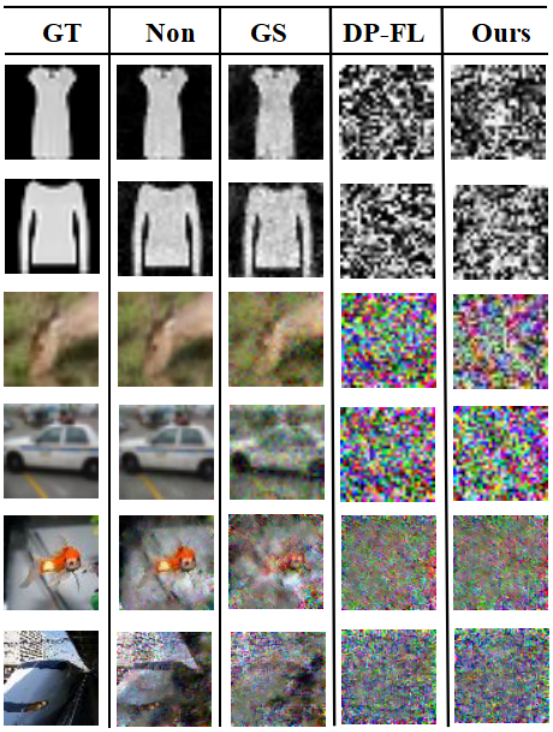

Fig. 5: Resistance of different defense mechanisms, with $B = 1$, $E = 1$.

C. Convergence Analysis

From the above, we have analyzed the noise addition mechanism and proposed an aggregation weight distribution based on noise intensity. Next, we will analyze the convergence of the proposed algorithm. First, we introduce the assumptions 1-4 [42], [43], which are widely used in the theoretical analysis of federated learning.

Assumption 1: $\mathcal{L}_1, \cdots, \mathcal{L}_K$ are all L-smooth: for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $\mathcal{L}_k(\boldsymbol{v}) \leq \mathcal{L}_k(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla \mathcal{L}_k(\boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{v} - \boldsymbol{w}\|^2$.

Assumption 2: The variance of local gradient and global gradient is bounded: $\mathbb{E}[\|\nabla \mathcal{L}_k(\boldsymbol{w}) - \nabla \mathcal{L}(\boldsymbol{w})\|^2] \leq M^2$, where $M < \infty$.

Assumption 3: The gradient of $\mathcal{L}_k(\boldsymbol{w})$ is bounded: $\mathbb{E}[\|\nabla \mathcal{L}_k(\boldsymbol{w})\|^2] \leq G^2$, where $G < \infty$.

Assumption 4: The global objective function $\mathcal{L}(\boldsymbol{w})$ has a lower bound: $\mathcal{L}(\boldsymbol{w}) \geq \mathcal{L}^*$, $\forall \boldsymbol{w} \in \mathbb{R}^m$.

Theorem 1: Based on the above assumptions, the convergence bound of the proposed algorithm can be derived as follows:

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{w}^t)\|^2\right] \leq \frac{2[\mathcal{L}(\boldsymbol{w}^0) - \mathcal{L}^*]}{\eta T} + \sum_k \gamma_k^2(\sigma_k^2 + M^2). \tag{16}$$

Proof: By the *assumption 1*, we have:

$$\mathcal{L}(\boldsymbol{w}^{t+1}) \leq \mathcal{L}(\boldsymbol{w}^t) + \langle \nabla \mathcal{L}(\boldsymbol{w}^t), \boldsymbol{w}^{t+1} - \boldsymbol{w}^t \rangle + \frac{L}{2}\|\boldsymbol{w}^{t+1} - \boldsymbol{w}^t\|^2. \tag{17}$$

Substituting the update rule:

$$\boldsymbol{w}^{t+1} - \boldsymbol{w}^t = -\eta \sum_{k=1}^{K} \gamma_k (G_k^t + U_k). \tag{18}$$

Taking expectation on both sides:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{w}^{t+1}) - \mathcal{L}(\boldsymbol{w}^t)] \leq \underbrace{-\eta \left\langle \nabla \mathcal{L}(\boldsymbol{w}^t), \mathbb{E}\left[\sum_{k=1}^{K} \gamma_k (G_k^t + U_k)\right] \right\rangle}_{A} + \underbrace{\frac{L\eta^2}{2} \mathbb{E}\left\| \sum_{k=1}^{K} \gamma_k (G_k^t + U_k) \right\|^2}_{B}. \tag{19}$$

Note that $\mathbb{E}[U_k] = 0$, which is guaranteed by the zero-mean property of Gaussian noise, and $\langle \mathbf{a}, \mathbf{b} \rangle = \frac{1}{2}(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a} - \mathbf{b}\|^2)$, we obtain:

$$\begin{aligned} A = -\frac{\eta}{2}\|\nabla \mathcal{L}(w^t)\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla \mathcal{L}(w^t) - \sum_k \gamma_k G_k^t\|^2 \\ - \frac{\eta}{2}\mathbb{E}\|\sum_k \gamma_k G_k^t\|^2, \end{aligned} \tag{20}$$

and B can be calculated by:

$$\begin{aligned} B &\leq \frac{L\eta^2}{2}\mathbb{E}\Big\|\sum_{k=1}^{K} \gamma_k G_k^t\Big\|^2 + \frac{L\eta^2}{2}\mathbb{E}\Big\|\sum_{k=1}^{K} \gamma_k U_i\Big\|^2 \\ &\leq \frac{L\eta^2}{2}\mathbb{E}\Big\|\sum_{k=1}^{K} \gamma_k G_k^t\Big\|^2 + \frac{L\eta^2 \sigma_k^2 \gamma_k^2}{2}. \end{aligned} \tag{21}$$

Combining A and B and make ensure $L\alpha \leq 1$, we have:

$$\|\nabla \mathcal{L}(\boldsymbol{w}^t)\| \leq \frac{2\mathbb{E}[\mathcal{L}(\boldsymbol{w}^t) - \mathcal{L}(\boldsymbol{w}^{t+1})]}{\eta} + \sum_k \gamma_k^2(\sigma_k^2 + M^2). \tag{22}$$

Algorithm 1: Gradient Leakage-Resistant and Communication-Efficient Federated Learning.

Input: Initial global model $\boldsymbol{w}^0$, batch size B, the number of epochs E, noise scale σ_{max}, learning rate η, the upper bound of gradient $\|\boldsymbol{G}_{\max}\|$

Output: The converged model $\boldsymbol{w}*$

for *Round* $t = 1 : T$ **do**
  Server sends the global model $\boldsymbol{w}^t$, σ_{max} and $\|\boldsymbol{G}_{\max}\|$to the selected devices
  for each selected client in parallel **do**
    for *Epoch* $e = 1 : E$ **do**
      for *Batch* $\mathcal{B}_l \in \mathcal{D}_k$ **do**
        Compute gradient $\boldsymbol{G}_{k,l}^{t,e}$ from Eq.(2).
        Update local model from Eq.(3).
      end for
    end for
    Accumulate the gradient $\boldsymbol{G}_k^t = \sum_e \sum_l \boldsymbol{G}_{k,l}^{t,e}$
    Obtain the noise scale $\sigma_k \leftarrow R_k \sigma_{max}$
    Sample a set of uniform random variables $\boldsymbol{U}_k$ and quantize the model: $\boldsymbol{q}_k^t \leftarrow Q(\boldsymbol{w}_k^t + \boldsymbol{U}_k)$
    Upload the σ_k, $\boldsymbol{q}_k^t$ and the random seed s_k
  end for
  Server performs dither subtraction $\hat{\boldsymbol{w}}_k^t \leftarrow \boldsymbol{q}_k^t - \boldsymbol{U}_k$
  Assigns aggregation weights γ_k form Eq.(15).
  Updates the global model $\boldsymbol{w}^{t+1} \leftarrow \sum_k \gamma_k \hat{\boldsymbol{w}}_k^t$
end for

Taking the expectation over t from 0 to $T-1$ yields *Theorem 1*. ■

D. Workflow

Algorithm 1 outlines the proposed Gradient Leakage-Resistant and Communication-Efficient FL framework. The key stages are described as follows:

i) **Initialization:** Prior to training, the server simulates a standard client-side SGD procedure with $E = 1$ and $B = 1$ to estimate the upper bound $\|\boldsymbol{G}_{\max}\|$ of the gradient norm. Subsequently, a series of GLAs are conducted on the quantized models to determine the maximum tolerable noise scale σ_{max}, ensuring a balance between the privacy protection and convergence feasibility.

ii) **Local Training and Privacy Risk Estimation:** In each communication round, the server randomly selects a subset of clients to participate in the training. Each selected client performs the local model updates using the mini-batch SGD. During the training, all per-batch gradients $\boldsymbol{G}_{k,l}^{t,e}$ are accumulated to compute the full gradient $\boldsymbol{G}_k^t$, from which the privacy risk score R_k and corresponding noise level σ_k are derived according to (13) and (14), respectively.

iii) **Subtractive Dither Quantization:** Each client perturbs its model parameters with the uniform noise $\boldsymbol{U}_k$ and applies the quantization as $\boldsymbol{q}_k^t = Q(\boldsymbol{w}_k^t + \boldsymbol{U}_k)$. Upon receiving $\boldsymbol{q}_k^t$ and the shared seed s_k, the server reconstructs the effective model as $\hat{\boldsymbol{w}}_k^t = \boldsymbol{q}_k^t - \boldsymbol{U}_k$. According to Lemma 2, this reconstruction yields $\hat{\boldsymbol{w}}_k^t = \boldsymbol{w}_k^t + \boldsymbol{U}_k'$, where $\boldsymbol{U}_k'$ follows a zero-mean Gaussian distribution with variance σ_k^2.

iv) **Noise-Aware Aggregation:** The server computes aggregation weights γ_k using the inverse-noise weighting strategy defined in (13). This ensures that the contributions from clients with higher noise are down-weighted, improving the robustness of the global model update. The global model is then updated as $\boldsymbol{w}^{t+1} = \sum_k \gamma_k \hat{\boldsymbol{w}}_k^t$, and the process proceeds to the next round.

TABLE I: COMPUTATIONAL COMPLEXITY AND RUNNING TIME OF DIFFERENT ALGORITHMS

Algorithms	Complexity	Time(s)
FedAvg	$\mathcal{O}(s)$	4.92
GS	$\mathcal{O}(s + m\log(km))$	4.99
DP-FL	$\mathcal{O}(s + m)$	5.14
Proposed	$\mathcal{O}(s + m)$	5.52

E. Computation Complexity Analysis

We analyze the computational complexity incurred during the local training phase on a single client. Depending on the algorithm, the client k may perform some operations including SGD, sparsification, gradient clipping, noise injection, and quantization. Let $\mathcal{O}(s)$ denote the computational complexity of local SGD, which depends on the local training configurations, such as batch size, number of epochs, and model architecture.

Assuming a total of T communication rounds and m model parameters, we summarize the per-round computational complexity of representative methods in Table I. In sparsified methods, such as Gradient Sparsification (GS), the additional cost arises from identifying and processing the top-k fraction of parameters. In differential privacy-based methods (e.g., DP-FL), the extra operations include gradient clipping and noise generation, contributing an the overhead of $\mathcal{O}(m)$.

Our proposed method introduces similar overhead due to the application of subtractive dither quantization and privacy risk evaluation. Specifically, the quantization process and the calculation of gradient norms contribute a linear cost in the number of model parameters. As a result, the overall complexity of our method is $\mathcal{O}(s+m)$, which is comparable to DP-FL and lower than that of sparsification-based approaches that require additional sorting operations.

Compared to the baseline FedAvg method, which incurs only $\mathcal{O}(s)$ complexity, our method introduces a moderate additional overhead of $\mathcal{O}(m)$. This increase stems from the integration of privacy-preserving components, including gradient norm computation, adaptive noise calibration, and quantization. Such overhead is the necessary trade-off for achieving the enhanced privacy protection, and is significantly more lightweight than the sparsification-based methods that involve sorting or selection processes of complexity $\mathcal{O}(s + m\log(km))$.

We employ ResNet-18 for training on the Tiny-ImageNet dataset with a batch size of 32 and epochs set to 3. The

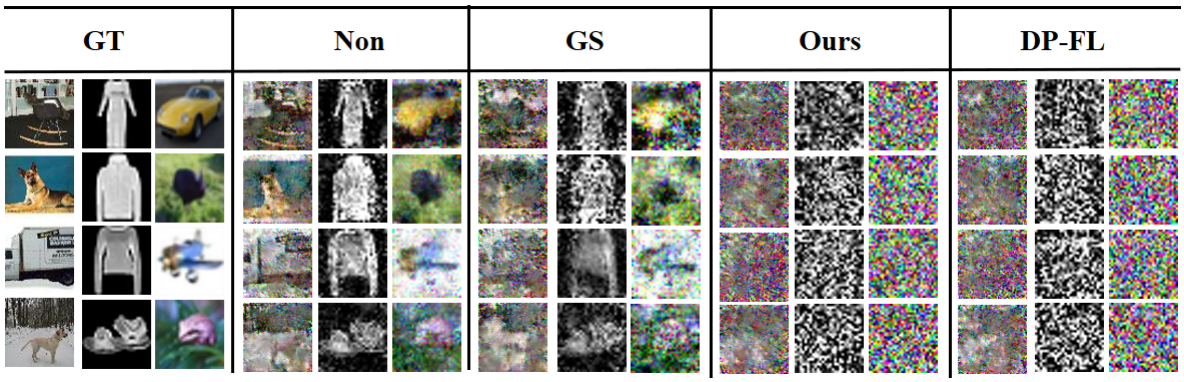

Fig. 6: Resistance of different defense mechanisms, with $B = 8$, $E = 1$.

experimental platform configuration is specified in Section VI. The average local training time per communication round required by different algorithms is presented in Table I.

VI. Experimental Results

A. Experiment Settings

All experiments are implemented in PyTorch and conducted on a workstation equipped with an Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz and two NVIDIA RTX 3090 GPUs, each with 24 GB of CUDA memory.

Datasets and model. We evaluate the proposed method on two widely used image classification benchmarks:

- **FashionMNIST** [44]: Comprising 60,000 grayscale training images and 10,000 test images of casual clothing items, each image is resized to 28×28 pixels.
- **CIFAR-10** [45]: Containing 60,000 color images from 10 classes, each of size 32×32 pixels.
- **Tiny-ImageNet**: The dataset comprises 200 categories with a resolution of 64×64, including 100,000 images in the training set and 10,000 images in the test set.

To simulate the non-IID data distributions, we partition the datasets across clients using a Dirichlet distribution $\mathrm{Dir}(\alpha)$, following the approach in [46]. A smaller α corresponds to a higher degree of heterogeneity. For experiments on CIFAR10 and FashionMNIST, the employed model architecture is a CNN consisting of five convolutional blocks, two fully connected layers, and ReLU activations. For the Tiny-ImageNet dataset, we used a ResNet-18 network [47].

To evaluate the proposed algorithm in terms of privacy protection and communication efficiency, we compare it with the following representative baselines:

- **DP-FL** [48]: Implements the Gaussian mechanism by injecting noise during local SGD updates. The noise standard deviation is set to $\sigma = 0.01$.
- **Gradient Sparsification (GS)** [10]: Only transmits the gradients with the top-k magnitudes, while the rest are zeroed out. To highlight the method's vulnerability to GLA, we set the sparsity ratio to $p = 0.9$.

Evaluation Criteria. We assess the proposed algorithm from three key perspectives:

1) **Privacy protection**: Resistance to GLA under varying training conditions.
2) **Training efficiency**: Measured by model convergence speed and final accuracy.
3) **Communication overhead**: Quantified by the total volume of data transmitted.

B. Evaluation of Defense Effectiveness

TABLE II: The Defense Performance Of related algorithms on different datasets, where $E = 1$, $B = 1$.

Dataset	Algorithm	MSE ↑	PSNR ↓	SSIM ↓
FashionMNIST	Non-private	0.0061	31.19	0.9547
	GS	0.0508	21.98	0.8185
	DP-FL	1.8190	6.45	0.0512
	Ours	**2.3956**	**5.32**	**0.0357**
CIFAR10	Non-private	0.0072	33.47	0.9375
	GS	0.1894	19.25	0.4956
	DP-FL	1.9962	9.02	0.0285
	Ours	**2.4069**	**8.21**	**0.0144**
Tiny-ImageNet	Non-private	0.1280	21.84	0.5441
	GS	0.7595	14.11	0.1241
	DP-FL	1.3807	11.52	0.0192
	Ours	**1.3750**	**11.54**	**0.0183**

TABLE III: The Defense Performance Of related algorithms on different datasets, where $E = 1$, $B = 8$.

Dataset	Algorithm	MSE ↑	PSNR ↓	SSIM ↓
FashionMNIST	Non-private	0.3861	13.59	0.3935
	GS	0.6100	11.58	0.2662
	DP-FL	**2.0540**	**5.96**	**0.0291**
	Proposed	1.8630	6.39	0.0358
CIFAR10	Non-private	0.6709	14.02	0.2761
	GS	1.2178	11.36	0.1513
	DP-FL	**2.8207**	**7.61**	0.0172
	Proposed	2.7737	7.67	**0.0164**
Tiny-ImageNet	Non-private	0.8197	14.02	0.1963
	GS	1.5490	11.22	0.0581
	DP-FL	2.3943	9.29	0.0191
	Ours	**2.3684**	**9.37**	**0.0193**

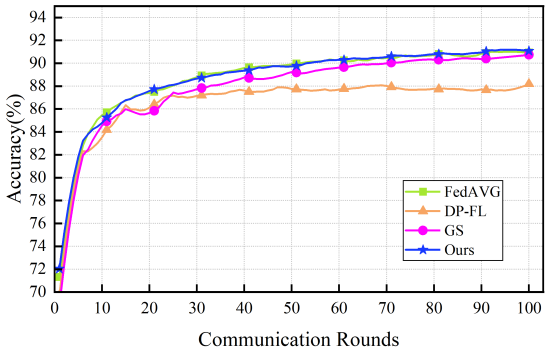

Fig. 7: The test accuracy with respect to communication rounds on FashionMNIST dataset, $B = 32$, $E = 3$.

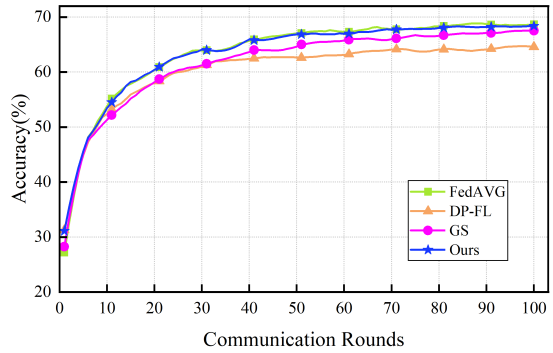

Fig. 8: The test accuracy with respect to communication rounds on CIFAR10 dataset, $B = 32$, $E = 3$.

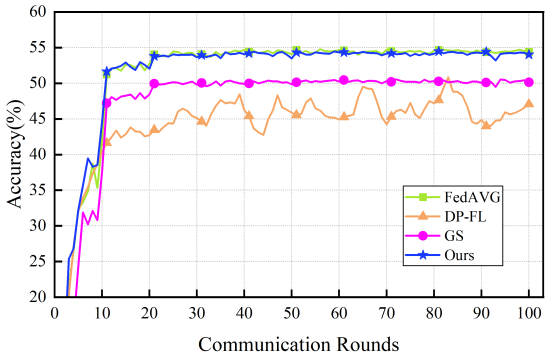

Fig. 9: The test accuracy with respect to communication rounds on Tiny-ImageNet dataset, $B = 32$, $E = 3$.

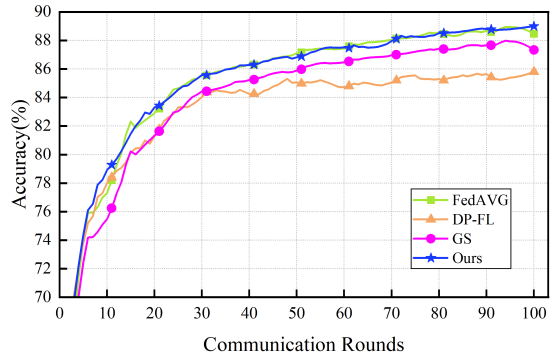

Fig. 10: The test accuracy with respect to communication rounds on FashionMNIST dataset, $B = 32$, $E = 1$.

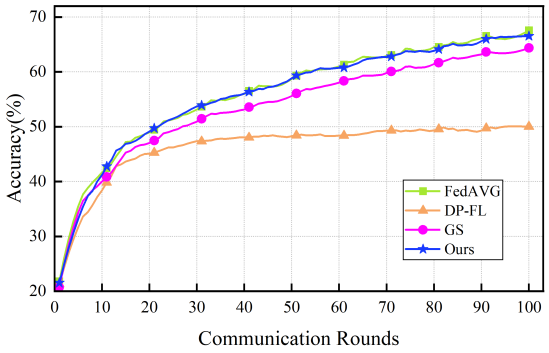

Fig. 11: The test accuracy with respect to communication rounds on CIFAR10 dataset, $B = 32$, $E = 1$.

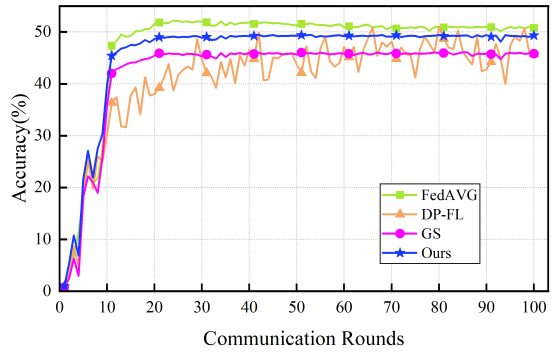

Fig. 12: The test accuracy with respect to communication rounds on Tiny-ImageNet dataset, $B = 32$, $E = 1$.

InvertGrad [15] leverages the cosine similarity to assess the discrepancy between virtual and real gradients, demonstrating the superior performance over other similar algorithms, particularly when the gradient norm is small. In the following experiments, we adopt it as the baseline attack. The quality of data reconstruction in GLA serves as a crucial measure of privacy protection. To evaluate the clarity of the reconstructed images, we use three widely adopted metrics in image processing: mean square error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM). Smaller PSNR and SSIM values, or larger MSE values, indicate more robust privacy preservation. To illustrate the defense capability in scenarios with high privacy leakage risks, we assume that the data size of the target client k is equal to the batch size, i.e., $|\mathcal{D}_k| = B$. Although this is a relatively rare condition, in IoT settings with constrained storage resources, it is often unfeasible to guarantee that all devices possess ample data. Consequently, evaluating privacy risks under extreme conditions is both relevant and necessary.

We first evaluate the performance of each defense algorithm under the optimal attack settings, where the image is fed into the initial global model, resulting in the largest gradient norm and the highest risk of data leakage. As shown in Figure 5, "Non" refers to the scenario without any defense mechanism, where the attacker can almost perfectly reconstruct the image. In the case of GS, the model sparsification removes some

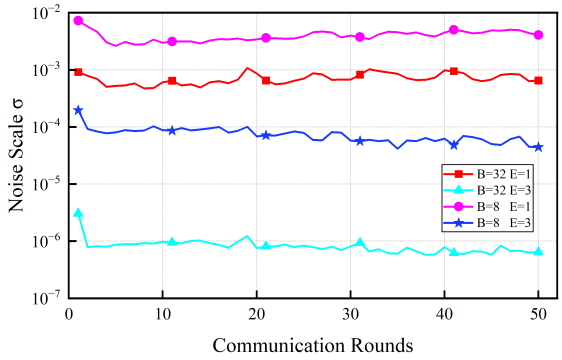

Fig. 13: Variation of Gaussian noise scale σ with communication rounds under different local settings.

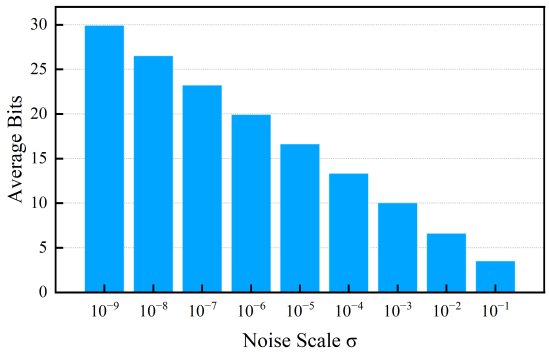

Fig. 14: The average number of bits required to transmit each parameter under different noise scales.

sensitive information from the gradient. However, since the gradients with larger norms are still retained, the image still is reconstructed quite effectively, albeit with some noise. Both DP-FL and the algorithm proposed in this paper successfully thwart the GLA attacks. As seen in Table II, our proposed algorithm outperforms all others across the evaluation metrics.

To further assess the privacy risks in practical FL training, we input a batch ($B = 8$) into the same global model for both training and attack, and select four images with the highest reconstruction quality from each dataset's batch for reference. As shown in Figure 6, as previously analyzed, increasing the training parameters reduces the effectiveness of the GLA attack. Even without any defense mechanisms, some noise still appears in the reconstructed images, but the majority of the information remains identifiable, leaving data security unprotected. In this setup, although the sparsification introduces additional noise, the key privacy information, such as the color and contours of objects, is still discernible. This highlights that GS is inadequate in providing reliable defense.

In Table III, we observe that the DP-FL outperforms our algorithm in terms of MSE and PSNR, as we reduce the noise intensity based on the declining privacy risk of the devices. However, the small quantity of noise intensity can not impact the effectiveness of our defense and all the reconstructed images remain completely unrecognizable.

C. Model Performance Evaluation

Model prediction accuracy is a critical metric for assessing the model performance. In this section, we train the model on the FashionMNIST and CIFAR10 datasets, evaluating the accuracy using their respective test datasets. To simulate a FL task in an IoT environment, we distribute the data across 50 devices in a non-IID fashion and set the local training parameters for each device to $B = 32$ and $E = 3$. In each communication round, the server randomly selects 10 devices to participate in the training, and after each global model update, the model's performance is evaluated. After 100 communication rounds, the model has largely converged, and the training process concludes.

Figure 7 illustrates the training performance of three privacy-preserving algorithms on the FashionMNIST dataset. During the initial stages of training, the training speeds of all four algorithms are nearly identical, indicating that the model demonstrates strong robustness to noise and sparsification at this point. However, as training progresses, the accuracy of DP-FL significantly lags behind the other algorithms. At convergence, its accuracy is 3% lower than that of the original FedAVG algorithm. This decline can be attributed to the excessive Gaussian noise added during training to safeguard privacy, which hampers the model's ability to learn subtle gradient changes. In contrast, GS maintains a robust training performance even with high sparsity, thanks to the model's large parameter redundancy. Overall, our algorithm achieves accuracy almost identical to that of FedAVG, outperforming all other defense algorithms. This is made possible by the risk evaluation mechanism introduced earlier, which ensures that device data is protected from GLA attacks while minimizing the Gaussian noise intensity.

Figure 8 and Figure 9 evaluate the defense performance when handling some color images, a scenario that better reflects the real-world training data and increases the difficulty of the task. On the CIFAR10 and Tiny-ImageNet datasets, the gradient norms fluctuate more significantly, creating the illusion of a higher privacy leakage risk. However, as our analysis shows, the difficulty of GLA attacks under the same gradient norm varies substantially based on the user's batch size and epoch settings. By tailoring the noise intensity to match the user's training configuration, our algorithm consistently performs optimally in both training speed and accuracy, regardless of the training stage.

Up until now, the evaluation of FL training has been confined to a single local setting. Given the profound impact of this setting on the defense capability and training performance of our algorithm, we will adjust the epoch to 1 and conduct a new round of simulation experiments. Figure 10 and Figure 11 illustrate the impact of various defense algorithms on FL training under conditions of heightened privacy leakage risk. In this scenario, the noise scale σ_k added by our algorithm remains around 0.001, yet the training speed and accuracy still maintain a leading position. This can be explained as follows: due to the additivity of Gaussian noise, the server effectively combines the noise disturbances from multiple models at the aggregation phase. Our algorithm collects the noise intensity from all devices and appropriately reduces the aggregation

weight for high-risk devices, which helps mitigate the accuracy loss. Particularly in non-IID scenarios, where the distribution of data quantity and types is unbalanced, certain categories of images may lack sufficient training, leading to excessively large gradient norms. In such cases, larger noise is necessary to ensure data security.

When $E = 1$, the performance of the DP-FL model with CIFAR10 declines even further, with accuracy dropping by 18%. As can be seen in Figure 12, the training fails to converge on Tiny-ImageNet, suggesting that adopting DP-FL in more complex neural networks may prove to be counterproductive. While it performs exceptionally well in terms of privacy protection, it appears to struggle with balancing data security and model utility, especially when faced with attacks like GLA.

At last, we have consistently highlighted the critical fact that the local training settings play in influencing privacy leakage risks. Figure 13 provides a clear visual depiction of the Gaussian noise intensity added to the device model under various configurations. It can be observed that as the batch size and epoch increase, the standard deviation of the Gaussian noise decreases sharply. This is the primary factor contributing to the superior performance of our algorithm. The increase in these two parameters helps to obscure the gradient characteristics of individual data points, thereby preventing the attacker from accessing more original data information. As a result, the quality of image reconstruction deteriorates significantly. This also explains why noise with vastly different intensities (up to three orders of magnitude) can be added under the same gradient norm without leading to privacy leakage.

D. Communication Overhead Analysis

Communication efficiency of devices in FL is also one of the core issues we focus on. With the rapid expansion of DNNs, the quantization and sparsification techniques, which effectively reduce the amount of data transmitted, have been widely adopted. However, as demonstrated in previous experiments, even at high sparsity rates, device data remains vulnerable to leakage. To address this, we employ subtractive dither quantization in this work to balance the privacy protection and communication efficiency. During the quantization process, the devices apply different quantization steps to each model parameter, meaning that the number of bits required for transmitting each parameter varies. Figure 14 illustrates the average number of bits required to transmit a single parameter at different noise scales. Regardless of the noise level, the number of bits transmitted is always lower than the commonly used 32-bit floating-point format. Notably, under higher noise conditions, the communication overhead can be reduced by nearly 90%.

Table IV presents the total transmission data volume for each algorithm under different training configurations. "Others" refers to the FedAVG and DP-FL algorithms, which do not optimize for communication efficiency, resulting in the same communication overhead under the same number of training rounds. Due to the differences in the number of image channels between the FashionMNIST and CIFAR10 datasets, the CNN parameters used for training are 2,715,402 and 3,210,122, respectively. As shown in the table, the GS algorithm transmits the least amount of data, thanks to its transmission of only sparse models. The improvement in communication efficiency is most evident under high sparsity. However, even with this advantage, its performance in terms of privacy protection and model accuracy is less than ideal. Since the quantization step size is influenced by device training settings, the optimization effect of our proposed algorithm on communication overhead varies accordingly. When $B = 32$ and $E = 1$, the required transmission data volume is only 27.9% of that for FedAVG. Even when E is increased to 3, our model still reduces transmission data by nearly 50%, demonstrating the excellent communication efficiency of our algorithm.

TABLE IV: TOTAL TRANSMISSION DATA VOLUME UNDER DIFFERENT TRAINING CONFIGURATIONS.

Training Setting	Dataset	Algorithm	Total Data Volume (MB)
$B = 32, E = 1$	FashionMNIST	Others	10 360
		GS	1036
		Proposed	2896
	CIFAR10	Others	12 246
		GS	1225
		Proposed	3406
$B = 32, E = 3$	FashionMNIST	Others	10 360
		GS	1036
		Proposed	5629
	CIFAR10	Others	12 246
		GS	1225
		Proposed	6556

VII. CONCLUSION

In this paper, we address the fundamental trade-off issue between the privacy protection, communication efficiency, and model utility in FL for edge computing environments, where the resource-constrained heterogeneous devices and variable network conditions exacerbate the risk of privacy leakage and complicate the deployment of uniform defense mechanisms. Motivated by our observational study that the privacy risk is closely related to the client-specific training behaviors and gradient characteristics, we propose a risk-aware FL framework tailored to the edge settings. By quantifying the per-device privacy risk, our method dynamically adapts the quantization process using subtractive dithering, injecting controllable Gaussian noise to mitigate the gradient leakage. To maintain the model performance in presence of noise, we further present a noise-aware aggregation strategy that adjusts each client's contribution during the model updating. Experimental results confirm that our method achieves strong resilience to the GLA, substantial communication savings, and consistent accuracy across diverse devices. Furthermore, the proposed algorithm can be regarded as a modality-agnostic defense framework, whose theoretical foundation is compatible with differential privacy mechanisms. This property suggests that the approach has the potential to generalize beyond image-based tasks to other FL scenarios, including natural language processing, tabular data analysis, and speech recognition.

REFERENCES

[1] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in Resource Constrained Edge Computing Systems," *IEEE J. Sel. Areas Commun.,* vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

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

[3] O. Choudhury, Y. Park, T. Salonidis, A. Gkoulalas-Divanis, and A. K. Das, "Predicting adverse drug reactions on distributed health data using federated learning," *in Proc. AMIA Annu. Symp. Bethesda, MD, USA: American Medical Informatics Association*, 2019, pp. 313-322.

[4] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "FFD: A federated learning based method for credit card fraud detection," *in Proc. Int. Conf. Big Data. Cham, Switzerland: Springer*, 2019, pp. 18–32.

[5] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," *in Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1707–1718.

[6] Z. Long, Y. Chen, H. Dou, Y. Zhang, and Y. Chen, "FedSQ: Sparse-Quantized federated learning for communication efficiency," *IEEE Trans. Consumer Electron.*, vol. 70, no. 1, pp. 4050-4061, 2024.

[7] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *in Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–14.

[8] Z. Li, H. Chen, Y. Gao, Z. Ni, H. Xue, and H. Shao, "Staged noise perturbation for privacy-preserving federated learning," *IEEE Trans. Sustain. Comput.*, vol. 9, no. 6, pp. 936-947, Dec. 2024.

[9] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," *in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10132–10142.

[10] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *in Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[11] B. Hasircioglu and D. Gunduz, "Communication efficient private federated learning using dithering," 2023, arXiv:2309.07809.

[12] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning," *in Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 1–15.

[13] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, "Practical blind membership inference attack via differential comparisons," 2021, arXiv:2101.01341.

[14] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," *in Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 691–706.

[15] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–How easy is it to break privacy in federated learning?" *in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 16937–16947.

[16] Y. Yang, Z. Ma, B. Xiao, Y. Liu, T. Li, and J. Zhang, "Reveal Your Images: Gradient leakage attack against unbiased sampling-based secure aggregation," *IEEE Trans. Knowledge Data Eng*., vol. 35, no. 12, pp. 12958-12971, Dec. 2023.

[17] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," *in Proc. USENIX Annu. Tech. Conf.*, vol. 33, 2020, pp. 493–506.

[18] Y. Li, H. Li, G. Xu, T. Xiang, and R. Lu, "Practical privacy-preserving federated learning in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4692–4705, 2022.

[19] K. Wei et al., "Federated learning with differential privacy: algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[20] G. Wang, Q. Qi, R. Han, L. Bai, and J. Choi, "P2CEFL: Privacy-preserving and communication efficient federated learning with sparse gradient and dithering quantization," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 14722-14736, 2024.

[21] R. Hu, Y. Guo, and Y. Gong, "Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy," *IEEE Trans. Mobile Comput.*, early access, Dec. 2023.

[22] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," *in Proc. ACM CCS*, 2017, pp. 603–618.

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

[24] W. Wei et al., "A framework for evaluating client privacy leakages in federated learning," *in Proc. Eur. Symp. Res. Comput. Secur.*, Springer, 2020, pp. 545–566.

[25] J. Hu et al., "Shield against gradient leakage attacks: adaptive privacy-preserving federated learning," *IEEE/ACM Trans. Networking* , vol. 32, no. 2, pp. 1407-1422, 2024.

[26] N. Lang, E. Sofer, T. Shaked, and N. Shlezinger, "Joint privacy enhancement and quantization in federated learning," *IEEE Trans. Signal Processing*, vol. 71, pp. 295-310, 2023.

[27] H. Yang, M. Ge, K. Xiang, and J. Li, "Using highly compressed gradients in federated learning for data reconstruction attacks," *IEEE Trans.Inf. Forensics Security.*, vol. 18, pp. 818-830, 2023.

[28] C. Chen and N. Campbell, "Understanding training-data leakage from gradients in neural networks for image classification," *in Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2021.

[29] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multi-party computing," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6178–6186, 2021.

[30] Z. Li, H. Chen, Z. Ni, Y. Gao, and W. Lou, "Towards adaptive privacy protection for interpretable federated learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 14471-14483, 2024.

[31] W. Wei, L. Liu, Y. Wu, G. Su and A. Iyengar, "Gradient-Leakage Resilient Federated Learning," *in Proc. IEEE ICDCS*, DC, USA, Jul. 2021, pp. 797–807.

[32] A. Aji and K. Heafield, "Sparse communication for distributed gradient descent," *in Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2017, pp. 1–6.

[33] Z. Ye, W. Luo, Q. Zhou, Z. Zhu, Y. Shi, and Y. Jia, "Gradient inversion Attacks: impact factors analyses and privacy enhancement," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 46, no. 12, 2024.

[34] F. Wang, E. Hugh, and B. Li, "More than enough is too much: adaptive defenses against gradient leakage in production federated learning," *IEEE/ACM Trans. Networking* , vol. 32, no. 4, pp. 3061-3075, 2024.

[35] Y. Chen, L. Abrahamyan, H. Sahli, and N. Deligiannis, "Learned parameter compression for efficient and privacy-preserving federated learning," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 3503-3516, 2024.

[36] H. Liu, F. He, and G. Cao, "Communication-efficient federated learning for heterogeneous edge devices based on adaptive gradient quantization." *in Proc. IEEE INFOCOM*, New York City, NY, USA, May 2023, pp. 1-10.

[37] Diederik P. Kingma and Jimmy Ba. Adam, "A method for stochastic optimization," *in Proc. Int. Conf. Learn. Representations*, San Diego, May 2015, pp.1-15.

[38] L. Roberts, "Picture coding using pseudo-random noise," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp.145–154, 1962.

[39] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and dither: A theoretical survey," J. Audio Eng. Soc., vol. 40, no. 5, pp. 355–375, 1992.

[40] S. Walker, "The uniform power distribution," *J. Appl. Statist.*, vol. 26, no. 4, pp. 509–517, 1999.

[41] C. Gu, X. Cui, X. Zhu, and D. Hu, "FL2DP: Privacy-Preserving federated learning via differential privacy for artificial IoT," *IEEE Trans Ind. Informat.*, vol. 20, no. 4, pp. 5100-5111, April 2024.

[42] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, and F. Farokhi, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[43] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," *in Proc. Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, 2020, pp. 1–26.

[44] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, arXiv:1708.07747.

[45] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, May 2012.

[46] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 2351–2363.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[48] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *in Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–14.