Securing Model Weights Against Eavesdropping Adversaries in Federated Learning Using Quantization

Kushal Chakrabarti
Data and Decision Sciences
Tata Consultancy Services Research

Dipankar Maity
Department of Electrical and Computer Engineering
University of North Carolina at Charlotte

Abstract

While security research in Federated Learning (FL) has predominantly focused on protecting client data, the *confidentiality of the model parameters* themselves represents a critical and underexplored vulnerability. This work addresses model reconstruction attacks by passive eavesdroppers, a threat present in common update strategies like transmitting full models or model increments. To our knowledge, we are the first to repurpose dynamic uniform quantization as a dedicated defense for model confidentiality. Our lightweight, architecture-agnostic approach combines low-bit quantization with an adaptive clipping rule to thwart reconstruction attacks, even under warm adversary initialization. We provide theoretical guarantees establishing that our defense offers persistent, non-zero protection in both protocols. Across extensive experiments on CIFAR-10 and CIFAR-100, with up to 1000 clients in heterogeneous settings, our method reduces the adversary's test accuracy to near-random levels while maintaining global accuracy within 4% of the unquantized baseline. Our findings establish that repurposing quantization is a simple yet highly effective strategy for securing the largely overlooked area of model confidentiality in FL.

1 INTRODUCTION

While FL enables decentralized training without exposing raw client data (McMahan et al., 2017), it remains vulnerable to information leakage through the shared model updates. However, despite avoiding direct data exposure, FL remains vulnerable to information leakage through shared model updates. A growing body of work has focused on data security in FL, particularly in protecting raw data against inference attacks or gradient leakage (Zhu et al., 2019; Geiping et al., 2020; Wei et al., 2020; Li et al., 2021; Wen et al., 2023; Zhang et al., 2022; Nasr et al., 2019). These efforts primarily assume that an adversary aims to reconstruct or infer properties of the client's data. By contrast, model security—the confidentiality of the client's model parameters themselves—has received comparatively little attention (Jagielski et al., 2020; Truong et al., 2021; Kariyappa et al., 2021; Pal et al., 2020; Juuti et al., 2019; Zhu et al., 2024).

The model parameters encapsulate the solution to a specific decision-making problem and may reflect sensitive patterns tied to training data. In collaborative training among institutions, each participant may use proprietary architectures. If an adversary can recover the client's model, it may gain access to sensitive domain knowledge, internal model representations, or competitive business intelligence (Tramèr et al., 2016). In on-device personalization, client-side models may reflect user preferences or behaviors encoded in weights. The adversary may also mount model poisoning attacks to degrade the FL performance (Bhagoji et al., 2019). Moreover, reconstructing a near-optimal client model might be cheaper for the adversary than expensive training process (Jagielski et al., 2020). As such, defending model integrity from passive eavesdropping is a key step toward more robust and trustworthy FL.

We emphasize that protecting model confidentiality is an orthogonal objective to data privacy. In data privacy problem, the goal is to hide information about the client's dataset D_i contained in updates. Mathematically, the adversary tries to infer some property of the dataset $\phi(D_i)$ from the updates $U_{i,t} = \mathcal{A}_t(D_i, w_{i,t})$, where $\mathcal{A}_t$ denotes the FL algorithm and $w_{i,t}$ the local model weights. The notion of differential privacy (DP) works in this context because we can bound

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

$\sup_{D_i \sim D_i'} \left\| \Pr[\mathcal{A}_t(D_i) = z] - \Pr[\mathcal{A}_t(D_i') = z] \right\|$ with the standard adjacency notion between D_i, D_i'. Techniques such as additive noise, randomized quantization, gradient sparsification work because the privacy definition is *distributional*, i.e., based on indistinguishability of outputs under neighboring datasets. In contrast, model security considers an adversary observing a sequence of messages $\{m_t, m_{t+1}, ...\}$ sent by the client. The goal is to create an accurate estimate $\hat{w}_{a,t}$ of the client's evolving model $w_{i,t}$. This is a *longitudinal state estimation* problem, i.e., the adversary maintains an estimate and updates it over time $\hat{w}_{a,t+1} = f(\hat{w}_{a,t}, m_t)$. The key difference is that data privacy bounds the adversary's ability to distinguish neighboring training datasets based on update distribution, whereas model security bounds the adversary's ability to track a hidden dynamical system across time, using noisy, partial observations.

Since these two objectives differ both mathematically and operationally, standard defense tools (e.g., DP) tailored for data-privacy do not automatically guarantee model security. DP noise ensures $U_{i,t}(D_i) \approx U_{i,t}(D_i')$ for neighboring datasets. But an attacker attempting to reconstruct $w_{i,t}$ does not need to distinguish two datasets. They only need to solve the filtering problem $w_{a,t+1} = f(w_{a,t}, U_{i,t})$. DP noise does not ensure $\mathbb{E}\|w_{a,t} - w_{i,t}\|^2 \not\to 0$. In fact, standard Gaussian DP noise can be averaged out by the attacker over many rounds, causing the attacker's estimate to converge to the true model. Hence, DP does not guarantee a lower bound on state reconstruction error, as numerically studied in Maity and Chakrabarti (2025). Since model-security goal requires persistent non-vanishing reconstruction error, it warrants a formal investigation of methods that can protect FL models against adversarial reconstruction attacks.

A few existing works have demonstrated the feasibility of model reconstruction attacks (Jagielski et al., 2020; Truong et al., 2021; Kariyappa et al., 2021; Pal et al., 2020; Juuti et al., 2019). Among them, the attack in Jagielski et al. (2020) is based on repeated interaction between adversary and target oracle, which reveals labels, raw logits etc. The knowledge distillation methods in Truong et al. (2021); Kariyappa et al. (2021) and active learning method in Pal et al. (2020) require black-box predictions of target model. All these model extraction methods are for generic machine learning, not specific to FL. Moreover, the attacker setting in such works are inference-time and fundamentally different from passively observing a sequence of noisy, partial training updates from a client in training-time. In this work, we consider a passive, third-party adversary eavesdropping on uplink communications, aiming to reconstruct a client's full local model over time by simply listening to the exchanged messages. One such setting has recently been explored in Xu et al. (2025); Maity and Chakrabarti (2025). Their model reconstruction attack is especially powerful when the adversary has strong initialization, e.g., prior access to client model from earlier rounds, pretraining on similar data, or exploiting the downlink channel at initial rounds.

FL protocols vary in how clients communicate updates to the server: some share the full local model at each round, while others transmit only the model increment, i.e., the difference from the global model. Following the nomenclature of Maity and Chakrabarti (2025), we refer to the latter class of FL algorithms as Federated Learning with Innovation Process (FLIP), i.e., when the client returns only the innovation relative to the global model, and the former as Federated Learning withOut innovation Process (FLOP), i.e., the client returns its full local model. Prior work Maity and Chakrabarti (2025) has shown that FLIP can offer a degree of obfuscation of client's model, especially when the eavesdropping adversary lacks a good initial estimate of client model. However, we observe from our experiments that this protection may not be enough: well-initialized adversaries can still track the client's model trajectory upto a certain extent. FLOP, on the other hand, offers no such obfuscation and is completely vulnerable to model reconstruction by the adversary. These observations motivate a central question: *Can we enhance client model confidentiality in standard FL settings without modifying the core training or communication protocol of either FLIP or FLOP?*

Quantization is widely used for efficient communication, and recently utilized in enhancing data privacy (Kang et al., 2024b; Youn et al., 2023; Lang et al., 2023; Zhu et al., 2024). While some of these studies leverage (lossy) quantization to achieve DP for client data, the role of quantization in protecting client model parameters has remained largely unexplored. The existing works use quantization to bound dataset leakage. We use quantization to guarantee a dynamically persistent lower bound on model reconstruction accuracy, which is mathematically different. Specifically, prior works in data-privacy quantization overwhelmingly use static or unbiased quantization schemes that add randomization to satisfy DP bounds, where the typical role is to achieve (ϵ, δ)-DP via random discretization. These schemes are known to incur noticeable accuracy loss if the quantization is coarse, or remain statistically recoverable over time if the quantization is unbiased. On the other hand, our method uses deterministic quantization with a dynamic clipping, whose role is introducing non-vanishing, structured noise that accumulates through the adversary's recursive estimation, leading to provable lower bound on reconstruction error. Unlike DP distributional guarantee, this is a state

evolution guarantee which static quantization and unbiased compression fundamentally cannot provide. This distinction is mathematically essential and previously unexplored. A related work Zhu et al. (2024) used server-side quantization for model privacy in LLMs, but targets protection from clients rather than for them. In contrast, we consider third-party eavesdroppers and apply client-side quantization, jointly coordinated with the server.

We propose a simple, lightweight, and architecture-agnostic defense, aimed at enhancing model security in FL, leveraging dynamic uniform quantization. By combining standard uniform quantization with an adaptive clipping strategy, this method provides a delicate *zoom-in-zoom-out* effect that reduces quantization-induced accuracy loss while enhancing model confidentiality. Using quantization for data privacy or communication reduction is not new, but using quantization to create a persistent error floor for adversarial model reconstruction in both FLIP- and FLOP-type algorithms is new. Thus, our goal is *not* to develop new quantization techniques, but to repurpose existing ones, such as vanilla uniform quantization, for improved model protection in FL. Importantly, our contribution is identifying a previously unexplored security objective and showing that quantization, combined with careful dynamic adaptation, yields provable guarantees for it, while limiting utility degradation (unlike DP). Our main contributions are as follows.

- Our key insight is that low-bit quantization (e.g., 8 or 16 bits), combined with adaptive range control, injects structured noise into model updates, hindering adversarial reconstruction. Despite the challenge posed by the stochasticity of FL in estimating the bounds of the quantization domain, the server adjusts the quantization range using a simple update rule based on a scalar binary flag sent by each client. This zoom-in-zoom-out adaptation preserves model utility despite aggressive quantization.
- We present a theoretical lower bound on model protection in FLIP, showing that quantization and client drift together induce a persistent protection effect that grows over time. For the otherwise vulnerable FLOP protocol, we prove that quantization provides a provable, non-zero protection guarantee, perpetually resetting the adversary's error.
- While client models in FLOP algorithms remain unprotected (Maity and Chakrabarti, 2025), we show that quantization suppresses adversarial reconstruction to near-random behavior. For FLIP algorithms, when the adversary warm-starts with initial estimate close to the client's initial model, we show that quantization provides enhanced protection, from much earlier than unquantized FLIP.
- We demonstrate the effectiveness and scalability of our approach through extensive experiments. In both FLIP and FLOP paradigms, we show that even under challenging conditions with up to 1000 clients, heterogeneous data distributions, and persistent eavesdropping, 8-bit quantization with dynamic clipping can suppress adversarial model reconstruction to near-random behavior, while preserving global model accuracy within 4% of the unquantized baseline.

Notation: Throughout this paper, we use the subscripts $(\cdot)_a$ and $(\cdot)_i$ to denote quantities associated to the adversary and i^{th} client, respectively. We use x^q to denote an element-wise quantized version of $x \in \mathbb{R}^n$. For any natural number N, we let $[N]$ denote the set of natural numbers $\{1, \dots, N\}$. We let $\|\cdot\|$ denote the Euclidean norm of a vector and $|\cdot|$ denote the absolute value of a scalar. Finally, we let $\lfloor x \rfloor$ denote the smallest integer less than or equal to it for $x \in \mathbb{R}$.

2 PROBLEM FORMULATION

2.1 Federated Learning Setup

We consider a federated learning system comprising a central server and N clients. Each client $i \in [N]$ possesses a local dataset $\mathcal{D}_i$ and aims to collaboratively train a global model $w \in \mathbb{R}^d$. FL proceeds in communication rounds. At each round t, a subset of clients $\mathcal{S}_t \subseteq [N]$ is selected to participate. The server broadcasts the current global model w_t to the selected clients. Each selected client $i \in \mathcal{S}_t$ performs local training on its dataset $\mathcal{D}_i$ over E local epochs to update its local model to $w_{i,t+1}$. Clients then send their updates to the server, depending on the FL variant. In FLOP, such as McMahan et al. (2017), clients send the full updated model $w_{i,t+1}$; in FLIP, such as Karimireddy et al. (2020), clients send the model innovation $\Delta_{i,t} = w_{i,t+1} - w_t$. The server aggregates these updates to refine the global model w_{t+1}.

Let the random variable $\delta_{i,t} \in \{0, 1\}$ denote whether client i is selected at round t (i.e., $\delta_{i,t} = 1$) or not (i.e., $\delta_{i,t} = 0$). Assuming clients are sampled uniform randomly, we denote $p := \mathbb{P}(\delta_{i,t} = 1) = \frac{|S_t|}{N}$. For FLIP, the client dynamics can be written as

$$w_{i,t+1} = w_{i,t} + \delta_{i,t}(\Delta_{i,t} + d_{i,t}), \quad d_{i,t} = w_t - w_{i,t}, \tag{1}$$

where the mismatch $d_{i,t}$ arises from the temporal drift between server and client model.

2.2 FL in Presence of Eavesdroppers

We consider a passive, third-party adversary capable of intermittently eavesdropping on the uplink commu-

nication from a compromised client i to the server over multiple rounds. The adversary does not interfere with the FL process but aims to reconstruct the client's local model $w_{i,t}$ over time, by accumulating information from intercepted messages. A passive adversary eavesdropping only on the uplink is well motivated in wireless FL settings where the uplink is often the more vulnerable direction, as considered in Zhang et al. (2023); Xie et al. (2022); Xu and Neglia (2021). Moreover, accurately estimating server-side operations without detailed system knowledge (e.g., number of agents, data distributions, or objectives) is highly challenging and sensitive to errors. The eavesdropping mechanism is modeled by a Bernoulli process $\mu_t \sim \text{Bernoulli}(\gamma)$, similar to Xu et al. (2025). That is, every time the client is selected the adversary can intercept the uplink communication, gaining access to the model updates, only with a probability of γ. We will show that even in the strong adversarial regime where $\gamma \to 1$, quantization provides a robust defense against such eavesdropping. The adversary has access to: whether the client is sampled at the current time or not, i.e., the outcome of $\delta_{i,t}$; and updates sent by the compromised client to the server.

2.3 Model Protection via Quantization

Quantization has been explored as a mechanism for enhancing data privacy, typically as a component within Differential Privacy (DP) frameworks, where its discrete nature aids in the analysis of privacy guarantees (Kang et al., 2024a; Nguyen et al., 2024). Our work takes a different perspective. **To our knowledge, this work is the first to repurpose quantization as a defense for model confidentiality - a longitudinal reconstruction threat.** Instead of analyzing it for communication efficiency or as a tool for DP, we use it to provide formal guarantees on the reconstruction error of the model update itself.

To enhance model confidentiality of clients, we propose applying dynamic uniform quantization to the client updates prior to transmission. Each client quantizes its update $\Delta_{i,t}$ (or, $w_{i,t+1}$ for FLOP) using a quantization function $\mathcal{Q}(\cdot, q)$, resulting in the transmitted message $\mathcal{Q}(\Delta_{i,t}, q)$ for FLIP and $\mathcal{Q}(w_{i,t+1}, q)$ for FLOP. Here, $q = \{q_{\min}, q_{\max}, b\}$ denotes the hyperparameter of the quantizers: $q_{\min}$ and $q_{\max}$ denote the quantization domain and b denotes the quantization word-length (i.e., number of bits used for representing the quantized output). More details will be provided in Section 3.

The quantization function $\mathcal{Q}(\cdot, q)$ is designed to introduce controlled noise into the updates, thereby degrading the adversary's ability to accurately reconstruct the local models, while preserving the utility of the global model. Particularly, the hyperparameter q controls the amount of noise injected through quantization, and hence, it must be dynamically adapted in each round. The specific form of $\mathcal{Q}$ and the dynamic adaptation scheme for q are presented later in Section 3. In subsequent sections, $w^q_{i,t}$ and $\Delta^q_{i,t}$ denote the quantized versions of $w_{i,t}$ and $\Delta_{i,t}$, respectively.

2.4 Adversarial Model

2.4.1 Warm-start

An adversary's reconstruction success improves significantly with access to informative model checkpoints. In our threat model, we assume the adversary has access to a model checkpoint, which it uses to initialize its estimated client model $w_{a,0}$. For example, the adversary may obtain access to the client model w_{i,t_0+1}, where t_0 denotes the first round in which client i is selected.

In FLOP, if $\mu_{t_0} = 1$, the adversary directly intercepts w_{i,t_0+1}. Otherwise, it can delay its initialization until some round $t > t_0$ where both $\delta_{i,t} = 1$ and $\mu_t = 1$, i.e., the client participates and its update is intercepted. While this leads to a delayed start, it allows the adversary to initialize with an exact snapshot of the client model, thereby improving its ability to track future updates. In FLIP, the update at round t is given by the model innovation $\Delta_{i,t} = w_{i,t+1} - w_t$, and if the adversary has access to the global model w_t, it can compute $w_{i,t+1}$. For example, if the adversary intercepts the very first innovation $\Delta_{i,0}$ and knows the global model w_0, it can exactly reconstruct $w_{i,1} = w_0 + \Delta_{i,0}$ and use it as a warm-start. Partial warm-starts are also plausible in practice. An adversary might have access to a pretrained model from the same task domain, such as a publicly available model trained on similar data (Jagielski et al., 2020). This model can serve as an approximate warm-start for $w_{a,0}$, though it may deviate from the true client model due to domain or distributional mismatch. Delayed warm-starts may arise when the adversary only gains access to a client's communication later in the FL process. In such cases, the adversary must backtrack or interpolate earlier model states using intercepted updates. This generally leads to degraded reconstruction quality, especially for FLIP, where each innovation is only a delta from the current global model and not an absolute parameter snapshot. In all cases, warm-start significantly improves the adversary's ability to track the client model across rounds. This motivates the need for defenses under strong adversarial initialization assumptions.

2.4.2 Model Reconstruction

The adversary aims to reconstruct the local model $w_{i,t}$ of a compromised client i at each round t, using the intercepted messages. The reconstruction is formalized as an estimate $w_{a,t}$ by the adversary such that the

expected reconstruction error is minimized. Specifically, the client model's *protection* at round t is quantified by $\mathbb{E}[\|w_{a,t} - w_{i,t}\|^2]$. Here, the expectation is with respect to the randomness in both the eavesdropping process and the client selection of the FL algorithm. This protection metric was originally introduced in Agarwal et al. (2020) and subsequently applied in the context of FL in Maity and Chakrabarti (2025); Xu et al. (2025).

We assume the adversary follows a similar reconstruction protocol as in Maity and Chakrabarti (2025), with the key difference that, in our case, the intercepted messages are quantized. Formally, the adversary employs the following reconstruction strategy:

$$w_{a,t+1} = w_{a,t} + \delta_{i,t} z_{a,t}. \tag{2}$$

where $z_{a,t}$ is the adversary's estimate of the quantity $\Delta_{i,t} + d_{i,t}$ in (1). To estimate $\Delta_{i,t} + d_{i,t}$, the adversary separately estimates $\Delta_{i,t}$ and $d_{i,t}$, with the estimation of $\Delta_{i,t}$ being done by

$$\Delta_{a,t} = \mu_t \Delta^q_{i,t} + m(1 - \mu_t)\Delta_{a,t'}, \tag{3}$$

where $\mu_t \sim \text{Bernoulli}(\gamma)$ denotes the eavesdropping success, t' is the most recent round before t when client i was selected, and $m > 0$ is a forgetting hyperparameter that influences how much of the last successful interception of Δ_i affects the current $\Delta_{a,t}$ when $\mu_t = 0$. $\Delta_{a,t'} = 0$ if t is the first time the client is selected.

Estimating $d_{i,t}$ remains a key challenge as the adversary cannot intercept the downlink messages. Furthermore, $d_{i,t}$ depends on the server state change, which is a random variable influenced by several factors to which the adversary may not have any control or access, including the client selection mechanism, the local objective functions, and the data distributions at other clients. These dependencies make accurately estimating d_t particularly challenging. In this work, we do not focus on estimating $d_{i,t}$, and set its estimate $d_{a,t} = 0$. Nonetheless, setting $d_{a,t} = 0$ still benefits the adversary for FLOP-type algorithms, so much so that the adversary can *perfectly* estimate the client model; see Maity and Chakrabarti (2025, Section III.A).

Combining all these components, adversary's model reconstruction mechanism can be expressed as

$$w_{a,t+1} = w_{a,t} + \delta_{i,t}(\mu_t \Delta^q_{i,t} + m(1 - \mu_t)\Delta_{a,t'}). \tag{4}$$

Remark 1. Without quantization of $\Delta_{i,t}$, FLIP offers an inherent protection, as formalized in Maity and Chakrabarti (2025, Thm. 1). In contrast, FLOP exposes the local model, making it more susceptible to reconstruction attacks (Maity and Chakrabarti, 2025, Sec. III.A). In this work, we show that quantization helps both FLIP and FLOP types of algorithm, by providing a significant boost in protection for both.

2.5 Research Objective

Our goal is to evaluate the effectiveness of dynamic uniform quantization in protecting client models in FL against passive, eavesdropping adversaries, described above in Section 2.2 and Section 2.4, in both FLIP and FLOP paradigms. We aim to assess: (a) impact of quantization on the adversary's model reconstruction accuracy, where we consider test-set accuracy of the reconstructed model as the metric; (b) trade-off between model protection and global model performance, inflicted upon quantization.

3 QUANTIZED FL

A b-bit uniform quantizer $\mathcal{Q} : \mathbb{R} \to \{0, 1, \ldots, 2^{b-1}\}$ is:

$$\mathcal{Q}(x) = \begin{cases} \left\lfloor 2^b \frac{x - q_{\min}}{q_{\max} - q_{\min}} \right\rfloor, & \text{if } x \in (q_{\min}, q_{\max}), \\ 0 & \text{if } x \le q_{\min}, \\ 2^b - 1, & \text{if } x \ge q_{\max}, \end{cases} \tag{5}$$

where the interval $[q_{\min}, q_{\max}]$ is the operating range of the quantizer. $\mathcal{Q}$ is an encoding function that maps $\mathbb{R}$ to a b-bit binary word. The decoding is done by

$$x^q = q_{\min} + \epsilon \mathcal{Q}(x) + \tfrac{\epsilon}{2}. \tag{6}$$

The quantization error is defined as $e(x) = x - x^q$, which lies in the range $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$ for all $x \in [q_{\min}, q_{\max}]$. The quantization error can be arbitrarily large for x that are outside the interval $[q_{\min}, q_{\max}]$. For a vector $x \in \mathbb{R}^n$, quantization and decoding are element-wise.

3.1 Dynamic Quantizer

In FL, model weights evolve over time, and their upper and lower bounds depend on factors such as data distribution, objective functions, and learning parameters. Moreover, the stochasticity introduced by SGD and the client selection process further complicates the theoretical and practical estimation of these bounds. We propose a dynamic and adaptive bound estimation technique that is both simple to implement and highly effective. The core idea is to begin with an initial estimate of the quantization domain and then adaptively expand or contract this domain based on the occurrence or absence of saturation, respectively. This results in a delicate *zoom-in-zoom-out* effect in which the quantizer zooms out to prevent saturation and zooms in to reduce ϵ.

For simplicity, in this work, we assume that each element of $w_{i,t}$ is quantized with the same quantizer; i.e., the same domain and the same number of bits. Furthermore, since it is equally likely for each element of $w_{i,t}$ to be positive or negative, we assume that the quantization domain is of the form $[-\beta, \beta]$, for some $\beta > 0$. Consequently, by dynamically changing β over time,

we maintain the delicate *zoom-in-zoom-out* balance. In our work, the server does the adaptive β-tuning as:

$$\beta_{t+1} = \begin{cases} \alpha_1 \beta_t, & \text{if saturation occurred in round } t, \\ \alpha_2 \beta_t, & \text{otherwise,} \end{cases} \tag{7}$$

where $\alpha_1 > 1 > \alpha_2$ are hyperparameters. Essentially, (7) ensures that the quantization domain expands (i.e., zooms out) in response to saturation, and contracts (i.e., zooms in) when no saturation is observed. At the beginning of each round, the server computes β_t and broadcasts it to the selected clients along with the server model w_t. Each client quantizes their model increments and records if saturation occurred for any of the elements of this vector. Accordingly, each client sends a binary variable $\theta_{i,t}$ to signify whether saturation was observed ($\theta_{i,t} = 1$) or not ($\theta_{i,t} = 0$). The $\theta_{i,t}$'s directly indicate which client suffered from saturation and indirectly indicate the potential degradation in model accuracy, for example, more clients suffering from saturation will result in more degradation in model accuracy. In this work, the server increases β_t (i.e., $\beta_{t+1} = \alpha_1 \beta_t$) if *any* of the $\theta_{i,t}$'s is 1. In this way, we put a stronger emphasis on avoiding saturation.

Finally, we note that the server uses (9) to decode the quantized messages. The decoding has negligible computational overhead and is model-agnostic, i.e., it does not depend on the specific neural network architecture or training algorithm. Furthermore, since quantization is only applied to the model updates and not to the full training pipeline, this quantization leaves the core FL algorithm unaltered, ensuring broad compatibility with existing frameworks and minimal implementation burden. A complete FL routine with the aforementioned dynamic quantization is summarized in Algorithm 1.

Line numbers 11 (quantization) and 12 (a single pass through the vector) in Algorithm 1 require $\mathcal{O}(d)$ cost per client per round. The server-side cost to decode and β-update is $\mathcal{O}(d\,|S_t|)$ per round (lines 14-18). Each client to server uplink communication requires $bd + 1$ bits, whereas each downlink communication requires $32(d + 1)$ bits assuming a full-precision floating point needs 32-bits. Thus, dynamic quantization introduces negligible computational overhead and moderate communication savings. The dominant cost still lies in training and full-precision downlink model broadcasts.

Algorithm 1 FL with Dynamic Uniform Quantization

1: **Initialize:** Global model w_0, clipping bound β_0, zoom-out factor $\alpha_1 > 1$, zoom-in factor $\alpha_2 < 1$, quantization bit-width b, total rounds T
2: **for** $t = 0$ to $T - 1$ **do**
3: Server samples clients $\mathcal{S}_t \subseteq \{1, \ldots, N\}$
4: Server broadcasts w_t and β_t to all $i \in \mathcal{S}_t$
5: **for each** client $i \in \mathcal{S}_t$ **in parallel do**
6: Train locally on data D_i for E epochs to get updated model $w_{i,t+1}$
7: **if** FLIP protocol **then**
8: Compute $\Delta_{i,t} = w_{i,t+1} - w_t$
9: **else if** FLOP protocol **then**
10: Set $\Delta_{i,t} = w_{i,t+1}$
11: Quantize: $\Delta^q_{i,t} = \mathcal{Q}(\Delta_{i,t}, \beta_t, b)$
12: $\theta_{i,t} = \begin{cases} 1 & \text{if any element of } \Delta_{i,t} \notin [-\beta_t, \beta_t] \\ 0 & \text{otherwise} \end{cases}$
13: Send $(\Delta^q_{i,t}, \theta_{i,t})$ to server
14: $w_{t+1} = w_t + \text{Aggregate}(\{\Delta^q_{i,t}\}_{i \in \mathcal{S}_t})$ ▷ server update
15: **if** any $\theta_{i,t} = 1$ **then**
16: $\beta_{t+1} = \alpha_1 \cdot \beta_t$ ▷ Zoom out to avoid saturation
17: **else**
18: $\beta_{t+1} = \alpha_2 \cdot \beta_t$ ▷ Zoom in to reduce quantization error
19: **Return:** Final global model w_T

4 PROTECTION ANALYSIS

4.1 Theoretical Guarantee for FLIP

Let $e_t := w_{a,t} - w_{i,t}$ and $\varepsilon_{i,t} := \Delta_{i,t} - \Delta^q_{i,t}$ respectively denote the client model's *protection* and quantization error at round t. Let $\eta \in (1, \frac{1}{p})$, $\rho > 0$ and $A := (1 - p\eta) \in (0, 1)$, and $B_t := p(1 - \frac{1}{\eta})\Big((1-\rho)\mathbb{E}[\|d_{i,t}\|^2] + (1 - \frac{1}{\rho})\mathbb{E}[\|\varepsilon_{i,t}\|^2]\Big)$.

Theorem 1. *Under perfect eavesdropping $\mu_t = 1 \quad \forall t$, the protection for FLIP is lower bounded by $\mathbb{E}[\|e_t\|^2] \geq A^t \mathbb{E}[\|e_0\|^2] + \sum_{s=0}^{t-1} A^{t-1-s} B_s$.*

Theorem 1 guarantees a minimum protection for the client model at every finite round of FLIP, explicitly quantifying how protection accumulates due to quantization and client drift via $\sum_{s=0}^{t-1} A^{t-1-s} B_s$. The second term in each B_t, contributed by quantization, is non-negative, thereby guaranteeing improved protection than unquantized baseline where $\varepsilon_{i,t} = 0$ for all t. Thus, quantization adds a persistent source of error that degrades the adversary's reconstruction ability, thereby enhancing model protection.

Corollary 1. *Under perfect eavesdropping $\mu_t = 1 \quad \forall t$, assume $\exists\, \delta_d, \delta_\varepsilon > 0$ such that $\lim_{t\to\infty} \mathbb{E}[\|d_{i,t}\|^2] = \delta_d$ and $\lim_{t\to\infty} \mathbb{E}[\|\varepsilon_{i,t}\|^2] = \delta_\varepsilon$. Then, for $\eta \in (1, \frac{1}{p})$, we have $\liminf_{t\to\infty} \mathbb{E}[\|e_t\|^2] \geq \max\{\frac{1}{4}, p(1-p)\} \left(\sqrt{\delta_d} - \sqrt{\delta_\varepsilon}\right)^2$.*

Corollary 1 simplifies Theorem 1's bound under a practically justified assumption: asymptotically, the quantization error is non-zero (due to finite-bit representation) and the client model has deviated from the global model (due to local training). It establishes that persistent, non-zero protection is guaranteed as long as the asymptotic magnitude of client drift is not perfectly canceled by the magnitude of quantization error (i.e., $\delta_d \neq \delta_\varepsilon$). This bound is tight because it accounts for the potential negative correlation between $d_{i,t}$ and $\varepsilon_{i,t}$. The term

$(\sqrt{\delta_d} - \sqrt{\delta_\varepsilon})^2$ arises from the worst-case scenario where the two vectors are perfectly anti-aligned. In any practical scenario, where client drift is a dynamic quantity dependent on data and model state, and quantization error is a controllable parameter, such a perfect and sustained match is highly improbable.

4.2 Theoretical Guarantee for FLOP

In FLOP, clients transmit their full local models $w_{i,t+1}$. Without quantization, this protocol is inherently vulnerable. An adversary who intercepts a single update can perfectly reconstruct the client's model, resetting their error to zero. This makes the unquantized protocol *0-protected* against a sufficiently patient adversary. We now prove that dynamic quantization fundamentally alters this dynamic, providing a provable, non-zero protection guarantee. We consider the challenging scenario of a perfect eavesdropper, where $\mu_t = 1$ almost surely for any round t in which the client is selected. Under perfect eavesdropping, the adversary's memory of past updates is rendered irrelevant, as they receive a fresh (albeit noisy) observation at every interception. The adversary's model $w_{a,t}$ evolves as follows:

$$w_{a,t+1} = w_{a,t} + \delta_{i,t}\Delta^q_{i,t} = w_{a,t} + \delta_{i,t}w^q_{i,t}.$$

Theorem 2. *Under perfect eavesdropping $\mu_t = 1\,\forall t$, the protection for FLOP is*

$$\mathbb{E}[\|e_t\|^2] = (1-p)^t\mathbb{E}[\|e_0\|^2] + p\sum_{s=0}^{t-1}(1-p)^{t-1-s}E[\|\varepsilon_{i,s}\|^2].$$

Corollary 2. *Let $E_s = \mathbb{E}[\|\varepsilon_{i,s}\|^2]$. Under the conditions of Theorem 2,*

$$\liminf_{s\to\infty} E_s \le \liminf_{t\to\infty} \mathbb{E}[\|e_t\|^2] \le \limsup_{t\to\infty} \mathbb{E}[\|e_t\|^2] \le \limsup_{s\to\infty} E_s.$$

Theorem 2 and Corollary 2 shows that even under perfect eavesdropping, the adversary's error does not decay to zero. Instead, their error is perpetually "reset" to the current quantization noise level upon each successful observation. In the long run, the protection level is dictated entirely by the magnitude of the quantization error, establishing a persistent and quantifiable security guarantee for the otherwise vulnerable FLOP protocol.

In Theorems 1&2, we adopted perfect eavesdropping to establish worst-case security guarantees. If protection holds when the adversary observes every message, the derived bounds provide conservative guarantees against weaker adversaries with $\gamma < 1$. Moreover, these lower bounds remain valid when eavesdropping is imperfect. While analyzing the general case $\gamma < 1$ leads to similar qualitative insights, it results in significantly more complex expressions; therefore, we focused on $\mu_t = 1$.

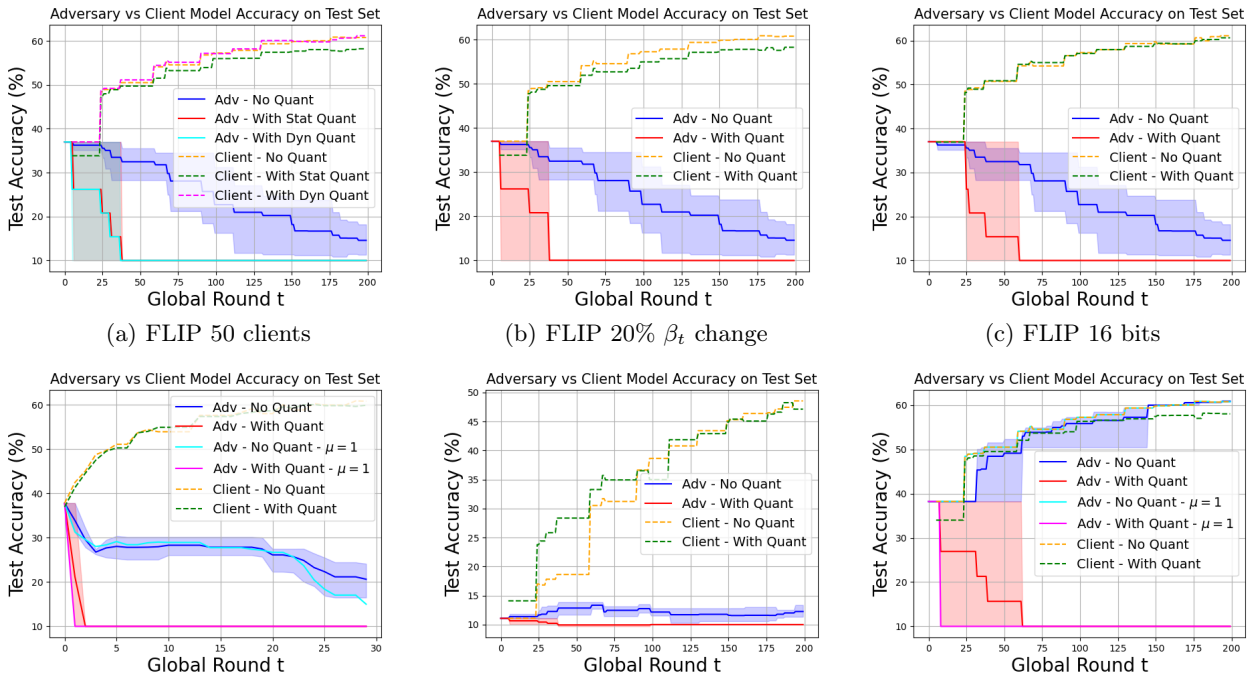
4.3 Numerical Results

We evaluate the effect of quantization on CIFAR-10 (Krizhevsky et al., 2009) using LeNet-5 (LeCun et al., 1998) and on CIFAR-100 using ResNet-18 (He et al., 2016). Our experiments explore the effect of varying the total number of clients $N \in \{50, 8\}$, quantization word-length b, dynamic clipping schedules β_t, and eavesdropping success probabilities γ. To assess the quality of $w_{a,t}$, we report its test-set accuracy, a metric aligned with prior work on task accuracy extraction in model stealing and inversion attacks (Jagielski et al., 2020; Milli et al., 2019).

The training set is equally partitioned evenly among N clients in an i.i.d. horizontal split. The server selects $|S_t| = 5$ clients uniformly at random at each round t. The adversary passively eavesdrops on the first client $i = 1$. Each selected client performs $E = 3$ local epochs using either SGD or Adam, with a batch-size 128. Learning rates are 10^{-3} for Adam and 10^{-1} for SGD. The global model w_0 is initialized randomly and identically across all experimental configurations. The β-tuning hyperparameters are set at either $\alpha_1 = 1.1, \alpha_2 = 0.9$ or $\alpha_1 = 1.2, \alpha_2 = 0.8$, with $\beta_0 = 1$. The adversary's forgetting hyperparameter is $m = 0.5$ in (3). Following Section 2.4.1, adversary's estimate is warm-started with $w_{a,0} = w_{1,t_0+1}$. For each experiment, we evaluate both the unquantized FL baseline and the proposed dynamically quantized variant, ensuring all other stochastic factors, such as client selection, $\{\mu_t\}$, and batch order are held constant. We repeat each setting with 5 independently sampled schedules and report the test accuracy of the adversary's model $w_{a,t}$ over time. The results are shown in Fig. 1. We summarize the key observations below.

- **Model protection**: Quantization consistently reduces the adversary's ability to reconstruct the client model. In the most vulnerable FLOP case, 8-bit quantization drops the adversary's test accuracy from over 60% to near-random levels of $\sim 10\%$ (Fig. 1f).
- **FLIP vs FLOP**: Without quantization, FLIP provides partial protection where adversary accuracy degrades over time (Figs. 1a-1e), but FLOP does not. Quantization closes this gap and protects both.
- **Utility preservation**: Global test accuracy is preserved within 4% of the unquantized baseline. With 16-bits, this degradation is $< 1\%$ (Table 1).

Protection timeline: Figs. 1a-1e) show adversary's test accuracy for FLIP. Without quantization, adversary accuracy gradually degrades from strong initialization and stabilizes near $13 - 15\%$. With quantization, adversary's accuracy drops sharply and stabilizes near 10% from much earlier rounds, approximating random guessing for CIFAR-10. In FLOP (Fig. 1f), unquantized adversary remains consistently accurate, but quantization suppresses reconstruction to near-random.



(a) FLIP 50 clients (b) FLIP 20% β_t change (c) FLIP 16 bits

(d) FLIP 8 clients (e) FLIP SGD (f) FLOP

Figure 1: CIFAR-10 test accuracies of $w_{a,t}$ with warm-start and client's model $w_{1,t}$, when LeNet-5 is trained, (i) without quantization (No Quant) and (ii) with dynamic quantization (With Quant). FL settings: (a) FLIP trained with Adam, $N = 50$, $\alpha_1 = 1.1, \alpha_2 = 0.9, b = 8$, (b) FLIP trained with Adam, $N = 50, \alpha_1 = 1.2, \alpha_2 = 0.8, b = 8$, (c) FLIP trained with Adam, $N = 50, \alpha_1 = 1.1, \alpha_2 = 0.9, b = 16$, (d) FLIP trained with Adam, $N = 8, \alpha_1 = 1.1, \alpha_2 = 0.9, b = 8$, (e) FLIP trained with SGD, $N = 50, \alpha_1 = 1.1, \alpha_2 = 0.9, b = 8$, (f) FLOP trained with Adam, $N = 50, \alpha_1 = 1.1, \alpha_2 = 0.9, b = 8$. For the adversary, the shaded areas represent the minimum and maximum accuracies over 5 runs and the solid line represents the mean.

Effect of bit-width increase: 16-bit quantization offers improved global utility, dropping $< 0.2\%$ compared to the unquantized baseline (see Table 1), but slighly delays protection (Fig. 1c). Specifically, the adversary's accuracy drops near 10% around $t = 60$, whereas 8-bit quantization (Fig. 1a) achieves this around $t = 37$. This illustrates a trade-off: higher precision preserves model performance but delays suppression. Thus, b can be tuned based on the specific threat model and utility-protection trade-off requirements.

Effect of frequent client participation: Fig. 1d considers the case when number of clients is reduced to $N = 8$, while keeping 5 clients active per round. As each client's participation rate increases, the adversary has more opportunities to intercept updates. As expected, the adversary's accuracy without quantization remains relatively high and degrades slower than Fig. 1a. However, quantization still succeeds in suppressing the adversary's performance even under frequent client exposure. This behavior is consistent with Theorem 1, as larger client selection frequency suggests smaller drift $d_{i,t}$, in which case quantization error $\varepsilon_{i,t}$ helps maintain a better lower bound on protection.

Effect of optimizer choice: When clients train with SGD instead of Adam, even without quantization, the adversary struggles to maintain high accuracy (Fig. 1e). Notably, there is an increase in adversary's accuracy until $t = 70$, likely due to slower model updates under SGD, which the adversary can initially track. When quantization is applied, the adversary's accuracy drops sharply instead from $t = 0$ and reaches near-random level at approximately the same t as with Adam (Fig. 1a). This suggests that dynamic quantization remains effective against adversary regardless of whether client updates are slow (SGD) or fast (Adam).

Table 1: Test accuracy (%) of global model with and without quantization. The settings are in Fig. 1-2.

Setting	**No Quant.**	**With Quant.**
FLIP, 50 clients	63.32	62.52
FLIP, 20% β change	63.32	60.69
FLIP, 16 bits	63.32	63.14
FLOP	63.14	60.47
FLIP, CIFAR-100	39.53	39.21
FLOP, CIFAR-100	39.53	35.63

Effect of persistent eavesdropping ($\mu_t = 1$): When

the eavesdropper intercepts every update of FLIP, without quantization, the adversary should have a better estimate than $\mu_t < 1$. However, the term $\frac{\gamma}{1-\gamma}$ in Maity and Chakrabarti (2025, Theorem 1) is large when $\gamma \to 1$, proving the counterintuitive result that protection is not small when $\mu_t = 1$ for FLIP, which we observe in Fig. 1d. Anyway, once quantization is applied, adversary performance drops sharply, showing efficacy of dynamic quantization even under ideal observation conditions.

Ablation study on quantization strategy: We compare our proposed dynamic quantization against a static baseline (Fig. 1a). The static approach uses a fixed $\beta_t = 0.1$ throughout the training process. While both static and dynamic quantization provides approximately the same protection, dynamic quantization has a negligible impact on the model's utility.

Large-scale evaluation: We conduct large-scale experiments on the CIFAR-100 dataset using a ResNet-18 model. The setting is made more realistic and challenging by distributing data among $N = 1000$ clients using a Dirichlet distribution with $\alpha = 0.3$ to simulate a high degree of client data heterogeneity. As shown in Fig. 2, we evaluate an adversary capable of persistent eavesdropping ($\mu_t = 1$) for both FLIP and FLOP protocols. In FLIP, the unquantized baseline shows a gradual decay in adversary accuracy, whereas our dynamic quantization immediately suppresses the adversary's performance to the level of random guessing. In FLOP protocol, the undefended adversary almost perfectly reconstructs the client model. However, when our defense is applied, the adversary's accuracy is again completely nullified from early in the training. These results affirm that our method scales to large, practical, and heterogeneous FL environments.

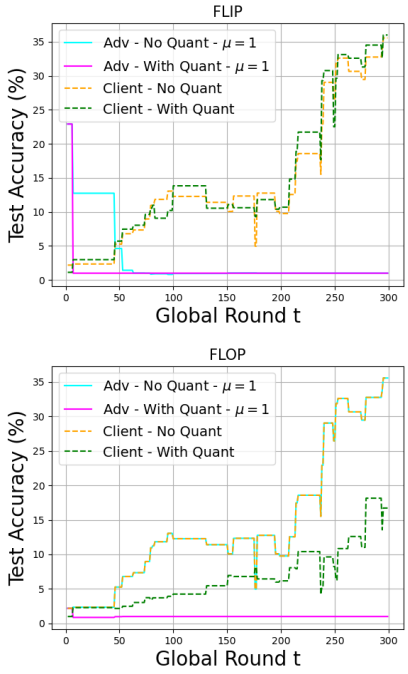


Figure 2: CIFAR-100 test accuracies of $w_{a,t}$ with warm-start and client's model $w_{1,t}$, when ResNet-18 is trained, (i) without quantization (No Quant) and (ii) with dynamic quantization (With Quant). FL settings: non-iid (Dirichlet with $\alpha = 0.3$), $N = 1000, \alpha_1 = 1.1, \alpha_2 = 0.9, b = 8, \gamma = 1$, trained with SGD, 10% clients participates per round t.

5 DISCUSSION AND CONCLUSION

Our aim was to highlight the previously overlooked security goal of protecting clients' models, explain why existing privacy tools do not apply, and propose a simple mechanism with provable protection. Furthermore, the adversary's position is crucial in a networked setting like FL. While most prior work on client data privacy assumes either the server or a client is honest but curious, less attention has been paid to a third-party adversary that simply listens to exchanged messages. We addressed this problem by introducing a lightweight, architecture-agnostic defense that uses dynamic quantization to strengthen client model security in FL against passive eavesdroppers. The most important distinction lies in the nature of the security guarantee. Unlike prior work focused on data privacy, we provided a guarantee on model reconstruction error. By injecting structured noise through low-bit quantization and adaptive clipping, we significantly hindered adversarial reconstruction of model weights with minimal utility loss, as confirmed by experiments.

Limitations: First, the quantization bound update rule is conservative, as domain contraction is halted if any client reports saturation, potentially leading to suboptimal quantization resolution. Second, we consider only passive adversaries on uplink channels; more complex threat models of coordinated multi-client eavesdropping remain unexplored.

Future work: The simplicity and generality of our approach make it a practical mechanism for model confidentiality. Several extensions are worth exploring. One is a stronger threat model with collaborative adversaries, where multiple eavesdroppers monitor different clients and an adversarial server aggregates the intercepted updates. Another is to improve attack robustness through temporal smoothing, for example by maintaining a moving average of recent model estimates to reduce quantization error and bridge long observation gaps. On the defense side, a more refined design could adapt β_t by weighting client signals according to update magnitude or local dataset size. Overall, our results provide a foundation for quantization-based model protection in FL and suggest promising directions for both adversarial modeling and defense design.

References

Agarwal, G. K., Karmoose, M., Diggavi, S., Fragouli, C., and Tabuada, P. (2020). Distortion-based lightweight security for cyber-physical systems. *IEEE Transactions on Automatic Control*, 66(4):1588–1601.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. (2019). Analyzing federated learning through an adversarial lens. In *International conference on machine learning*, pages 634–643. PMLR.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33:16937–16947.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jagielski, M., Carlini, N., Berthelot, D., Kurakin, A., and Papernot, N. (2020). High accuracy and high fidelity extraction of neural networks. In *29th USENIX security symposium (USENIX Security 20)*, pages 1345–1362.

Juuti, M., Szyller, S., Marchal, S., and Asokan, N. (2019). PRADA: protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 512–527. IEEE.

Kang, H.-S., Yun, S.-Y., and Kim, H. (2024a). Quantization for private SGD: Tighter RDP bounds and improved utility. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Kang, T., Liu, L., He, H., Zhang, J., Song, S., and Letaief, K. B. (2024b). The effect of quantization in federated learning: A Rényi differential privacy perspective. In *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 233–238. IEEE.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR.

Kariyappa, S., Prakash, A., and Qureshi, M. K. (2021). Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13814–13823.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Lang, N., Sofer, E., Shaked, T., and Shlezinger, N. (2023). Joint privacy enhancement and quantization in federated learning. *IEEE Transactions on Signal Processing*, 71:295–310.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366.

Maity, D. and Chakrabarti, K. (2025). On model protection in federated learning against eavesdropping attacks. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 8003–8008. IEEE.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Milli, S., Schmidt, L., Dragan, A. D., and Hardt, M. (2019). Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 1–9.

Nasr, M., Shokri, R., and Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE.

Nguyen, Q.-P., Tran, T.-V., and Le, T. (2024). On the privacy of quantized models: A Fourier analysis perspective. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Pal, S., Gupta, Y., Shukla, A., Kanade, A., Shevade, S., and Ganapathy, V. (2020). Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 865–872.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618.

Truong, J.-B., Maini, P., Walls, R. J., and Papernot, N. (2021). Data-free model extraction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4771–4780.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q., and Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469.

Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., and Zhang, W. (2023). A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2):513–535.

Xie, Y.-A., Kang, J., Niyato, D., Van, N. T. T., Luong, N. C., Liu, Z., and Yu, H. (2022). Securing federated learning: A covert communication-based approach. *IEEE Network*, 37(1):118–124.

Xu, C. and Neglia, G. (2021). What else is leaked when eavesdropping federated learning? In *CCS workshop Privacy Preserving Machine Learning (PPML)*.

Xu, L., Xu, D., Yi, X., Deng, C., Chai, T., and Yang, T. (2025). Decentralized federated learning algorithm under adversary eavesdropping. *IEEE/CAA Journal of Automatica Sinica*, 12(2):448–456.

Youn, Y., Hu, Z., Ziani, J., and Abernethy, J. (2023). Randomized quantization is all you need for differential privacy in federated learning. *arXiv preprint arXiv:2306.11913*.

Zhang, C., Miao, X., Huang, Y., Yang, L., and Tang, L. (2023). Performance of multi-antenna proactive eavesdropping in 5G uplink systems. *IEEE Transactions on Wireless Communications*, 22(9):6078–6091.

Zhang, J., Zhu, H., Wang, F., Zhao, J., Xu, Q., and Li, H. (2022). Security and privacy threats to federated learning: Issues, methods, and challenges. *Security and Communication Networks*, 2022(1):2886795.

Zhu, J., Lv, C., Wang, X., Wu, M., Liu, W., Li, T., Ling, Z., Zhang, C., Zheng, X., and Huang, X. (2024). Promoting data and model privacy in federated learning through quantized LoRA. *arXiv preprint arXiv:2406.10976*.

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. *Advances in neural information processing systems*, 32.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes. Section 2 formally defines the FL setup, the eavesdropping adversary, and the reconstruction attack. Section 3 and Algorithm 1 provide a clear mathematical description of the proposed dynamic quantization defense.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes. Section 3.1 explicitly analyzes the computational and communication complexity of the proposed method, detailing the costs for both clients and the server per round.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes. The source codes used to produce the figures and table in Section 4.3 are included in the supplemental material.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes. The assumptions for the theoretical guarantees, such as the perfect eavesdropping scenario and the existence of non-zero client drift for the corollaries, are explicitly stated in Section 4.]
   (b) Complete proofs of all theoretical results. [Yes. The complete mathematical proofs for Theorems 1 and 2 and their respective corollaries are provided in the Technical Appendices section.]
   (c) Clear explanations of any assumptions. [Yes. The paper provides clear justifications for its key assumptions. For instance, it explains that the non-zero error assumption in Corollary 1 is practical because finite-bit representation always induces some error and local training causes client drift.]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes. Section 4.3 describes all major hyperparameters, model architecture (LeNet-5, ResNet-18), dataset (CIFAR-10, CIFAR-100), and training details (optimizers, batch size, client selection, etc.) necessary to reproduce results. Algorithm 1 summarizes the complete FL setup with dynamic quantization for a quick look-up. Additionally, the source codes and model checkpoints used to produce the figures and table in Section 4.3 are included in the supplemental material.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes. Training/test details including optimizer, batch size, learning rate, initialization, quantization bit-width, β-update, and all other hyperparameters are reported in Section 4.3 and Figures 1-2.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes. The primary evaluation metric is clearly defined as the test-set accuracy of the reconstructed model, and the figure captions explain that shaded areas represent the min/max results over 5 independent runs.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes. Appendix E includes a description of the computing infrastructure used to conduct all the experiments.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes. The paper properly cites the creators of the datasets (CIFAR-10/100) and model architectures (LeNet-5/ResNet-18) used in the experiments.]
   (b) The license information of the assets, if applicable. [Not Applicable. The datasets and models are widely used public benchmark, and their licensing information is not required for this type of publication.]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable. The authors do not curate or release any new assets as part of this work.]
   (d) Information about consent from data providers/curators. [Not Applicable. The work uses standard, publicly available computer vision datasets that do not require new data provider consent.]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable. The CIFAR datasets do not contain personally identifiable or offensive content that would require special discussion.]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable. The research does not involve human subjects or crowdsourcing.]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable. No human participants were involved in this work.]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable. This work did not involve paid participants.]

Technical Appendices and Supplementary Materials

A PROOF OF THEOREM 1

From the adversary and client update rules:

$$w_{a,t+1} = w_{a,t} + \delta_{i,t}\Delta^q_{i,t}, \quad w_{i,t+1} = w_{i,t} + \delta_{i,t}(\Delta_{i,t} + d_{i,t}),$$

and consequently, the reconstruction error evolves as

$$e_{t+1} = e_t - \delta_{i,t}(d_{i,t} + \varepsilon_{i,t}).$$

Expanding the squared norm:

$$\|e_{t+1}\|^2 = \|e_t\|^2 - 2\delta_{i,t}\langle e_t, d_{i,t} + \varepsilon_{i,t}\rangle + \delta_{i,t}\|d_{i,t} + \varepsilon_{i,t}\|^2.$$

Apply Young's inequality with $\eta > 0$:

$$2\langle e_t, d_{i,t} + \varepsilon_{i,t}\rangle \le \eta\|e_t\|^2 + \frac{1}{\eta}\|d_{i,t} + \varepsilon_{i,t}\|^2,$$

so that

$$\|e_{t+1}\|^2 \ge (1 - \delta_{i,t}\eta)\,\|e_t\|^2 + \delta_{i,t}(1 - \frac{1}{\eta})\|d_{i,t} + \varepsilon_{i,t}\|^2.$$

Applying Young's inequality once more on the $\|d_{i,t} + \varepsilon_{i,t}\|^2$ term we obtain

$$\|e_{t+1}\|^2 \ge (1 - \delta_{i,t}\eta)\,\|e_t\|^2 + \delta_{i,t}(1 - \tfrac{1}{\eta})\big((1-\rho)\|d_{i,t}\|^2 + (1 - \tfrac{1}{\rho})\|\varepsilon_{i,t}\|^2\big).$$

Taking expectation on the randomness of $\delta_{i,t}$, we get

$$\mathbb{E}\big[\|e_{t+1}\|^2\big] \ge \underbrace{(1 - p\eta)}_{A}\,\mathbb{E}[\|e_t\|^2] + \underbrace{p(1 - \frac{1}{\eta})\big((1-\rho)\|d_{i,t}\|^2 + (1 - \tfrac{1}{\rho})\|\varepsilon_{i,t}\|^2\big)}_{B_t}.$$

Consequently, applying this inequality recursively yields

$$\mathbb{E}[\|e_t\|^2] \ge A^t\mathbb{E}[\|e_0\|^2] + \sum_{s=0}^{t-1} A^{t-1-s}B_s.$$

The proof is complete.

B PROOF OF COROLLARY 1

From Theorem 1, the recursive inequality for the expected error is:

$$\mathbb{E}[\|e_{t+1}\|^2] \ge A \cdot \mathbb{E}[\|e_t\|^2] + B_t,$$

where $A = (1 - p\eta)$ and $B_t = p(1 - \frac{1}{\eta})\big((1-\rho)\|d_{i,t}\|^2 + (1 - \frac{1}{\rho})\|\varepsilon_{i,t}\|^2\big)$. For a stable, non-trivial bound, we require $A \in (0, 1)$ and $B_t > 0$. This is achieved by selecting η such that $1 < \eta < 1/p$.

The asymptotic lower bound on the error is given by:

$$\liminf_{t\to\infty} \mathbb{E}[\|e_t\|^2] \ge \frac{\liminf_{t\to\infty} B_t}{1 - A} = \frac{\liminf_{t\to\infty} B_t}{p\eta}.$$

Using the corollary's assumptions that $\lim_{t\to\infty}\mathbb{E}[\|d_{i,t}\|^2]=\delta_d$ and $\lim_{t\to\infty}\mathbb{E}[\|\varepsilon_{i,t}\|^2]=\delta_\varepsilon$, we can choose $\rho=\sqrt{\frac{\delta_\varepsilon}{\delta_d}}$, which leads to

$$\begin{aligned}\liminf_{t\to\infty}\mathbb{E}[\|d_{i,t}+\varepsilon_{i,t}\|^2] \geq (1-\rho)\delta_d+(1-\tfrac{1}{\rho})\delta_\varepsilon &= (1-\sqrt{\frac{\delta_\varepsilon}{\delta_d}})\delta_d+(1-\sqrt{\frac{\delta_d}{\delta_\varepsilon}})\delta_\varepsilon\\ &= \delta_d-\sqrt{\delta_d\delta_\varepsilon}+\delta_\varepsilon-\sqrt{\delta_d\delta_\varepsilon}\\ &= \left(\sqrt{\delta_d}-\sqrt{\delta_\varepsilon}\right)^2.\end{aligned}$$

Now, we substitute this back into the expression for the asymptotic bound of B_t:

$$\liminf_{t\to\infty}B_t \geq p\left(1-\frac{1}{\eta}\right)\left(\sqrt{\delta_d}-\sqrt{\delta_\varepsilon}\right)^2.$$

Finally, substituting this into the overall asymptotic error bound gives:

$$\liminf_{t\to\infty}\mathbb{E}[\|e_t\|^2] \geq \frac{p(1-1/\eta)\left(\sqrt{\delta_d}-\sqrt{\delta_\varepsilon}\right)^2}{p\eta} = \frac{1-1/\eta}{\eta}\left(\sqrt{\delta_d}-\sqrt{\delta_\varepsilon}\right)^2 = \frac{\eta-1}{\eta^2}\left(\sqrt{\delta_d}-\sqrt{\delta_\varepsilon}\right)^2.$$

This completes the proof. Given that $\eta\in(1,\frac{1}{p})$, we can pick $\eta=\min\{2,\frac{1}{p}\}$ which will maximize the quantity $\frac{\eta-1}{\eta^2}$. Consequently, we may conclude $\liminf_{t\to\infty}\mathbb{E}[\|e_t\|^2] \geq \max\{\frac{1}{4},p(1-p)\}\left(\sqrt{\delta_d}-\sqrt{\delta_\varepsilon}\right)^2$.

C PROOF OF THEOREM 2

We derive the recursion for $E[\|e_{t+1}\|^2]$ using the law of total expectation, conditioning on whether client i is selected at round t.

Case 1: Client is not selected (probability $1-p$). The adversary's model is unchanged, $w_{a,t+1}=w_{a,t}$. The client's local model is also unchanged (as it did not participate in the round), so $w_{i,t+1}=w_{i,t}$. The error is $e_{t+1}=e_t$. The contribution to the expected error is $(1-p)E[\|e_t\|^2]$.

Case 2: Client is selected (probability p). The adversary intercepts the update and sets their model to $w_{a,t+1}=w^q_{i,t+1}$. The new reconstruction error is $e_{t+1}=w_{a,t+1}-w_{i,t+1}=w^q_{i,t+1}-w_{i,t+1}=-\varepsilon_{i,t+1}$. The contribution to the expected squared error is $p\cdot E[\|\varepsilon_{i,t+1}\|^2]$.

Combining the two cases gives $E[\|e_{t+1}\|^2]=(1-p)E[\|e_t\|^2]+p\cdot E[\|\varepsilon_{i,t+1}\|^2]$. Unrolling this recursion completes the proof.

D PROOF OF CORROLLARY 2

From Theorem 2, the model protection is given by:

$$E[\|e_t\|^2]=(1-p)^tE[\|e_0\|^2]+p\sum_{k=1}^{t}(1-p)^{t-k}E[\|\varepsilon_{i,k}\|^2].$$

As $t\to\infty$, the first term vanishes since $0<p\leq 1$:

$$\lim_{t\to\infty}(1-p)^tE[\|e_0\|^2]=0.$$

Thus, the asymptotic behavior of P_t is determined entirely by the summation term, which we denote $S_t=p\sum_{s=0}^{t-1}(1-p)^{t-1-s}E_s$. We will now prove the bounds on the limit superior and limit inferior of $E[\|e_t\|^2]$.

1. Proof of the $\limsup$ **inequality:** Let $L=\limsup_{s\to\infty}E_s$. By the definition of the limit superior, for any $\delta>0$, there exists a round N such that for all $s>N$, we have $E_s<L+\delta$. For any $t>N$, we can split the sum S_t into two parts:

$$S_t=p\sum_{s=0}^{N}(1-p)^{t-1-s}E_s+p\sum_{s=N+1}^{t-1}(1-p)^{t-1-s}E_s.$$

As $t \to \infty$, the first part (the "head") vanishes, as it is a finite sum of terms where $(1-p)^{t-s} \to 0$:

$$\lim_{t\to\infty} p\sum_{s=0}^{N}(1-p)^{t-1-s}E_s = 0.$$

For the second part (the "tail"), we apply our bound $E_s < L + \delta$:

$$p\sum_{s=N+1}^{t-1}(1-p)^{t-1-s}E_s < (L+\delta)\cdot p\sum_{s=N+1}^{t-1}(1-p)^{t-1-s}.$$

The sum of the weights in the tail, $p\sum_{k=N+1}^{t}(1-p)^{t-k}$, approaches 1 as $t \to \infty$. Therefore, taking the limit superior of the tail gives:

$$\limsup_{t\to\infty}\left(p\sum_{s=N+1}^{t-1}(1-p)^{t-1-s}E_s\right) \leq L+\delta.$$

Since the head of the sum vanishes, $\limsup_{t\to\infty} E[\|e_t\|^2] = \limsup_{t\to\infty} S_t \leq L + \delta$. Because this holds for any arbitrary $\delta > 0$, we must conclude that $\limsup_{t\to\infty} E[\|e_t\|^2] \leq L$, which is:

$$\limsup_{t\to\infty} E[\|e_t\|^2] \leq \limsup_{s\to\infty} E_s.$$

2. Proof of the $\liminf$ **inequality:** The argument is symmetric. Let $l = \liminf_{s\to\infty} E_s$. For any $\delta > 0$, there exists a round N such that for all $s > N$, we have $E_s > l - \delta$. We again split the sum S_t for $t > N$. The head of the sum vanishes as $t \to \infty$. For the tail, we apply the lower bound:

$$p\sum_{s=N+1}^{t-1}(1-p)^{t-1-s}E_s > (l-\delta)\cdot p\sum_{s=N+1}^{t-1}(1-p)^{t-1-s}.$$

Taking the limit inferior, we find that $\liminf_{t\to\infty} E[\|e_t\|^2] = \liminf_{t\to\infty} S_t \geq l - \delta$. Since this inequality must hold for any $\delta > 0$, we conclude that:

$$\liminf_{t\to\infty} E[\|e_t\|^2] \geq \liminf_{s\to\infty} E_s.$$

Combining the two inequalities gives the result stated in the corollary.

E ADDITIONAL EXPERIMENTAL RESULTS

Experiments were conducted on Kaggle using 2× NVIDIA Tesla T4 GPUs (16 GB each) and Intel Xeon CPUs with PyTorch 2.6.0+cu124.

β adaptation behavior: Fig. 3 shows how the dynamic clipping bound β_t evolves over rounds. Early on, the quantization domain $[-\beta_t, \beta_t]$ contracts as updates are large. In FLIP, $\Delta_{i,t}$ being locally accumulated gradients shrinks near optimality, stabilizing β_t around ~ 0.02. Since β_t is increased if at least one of the $\theta_{i,t}$'s are 1 for any t, fluctuations persist, helping maintain the saturation-resolution trade-off. A more aggressive adaptation ($\alpha_1 = 1.2, \alpha_2 = 0.8$) accelerates convergence of β_t around $t = 15$, with slightly better utility (see Table 2) but larger fluctuations. In contrast, in FLOP, since transmitted updates are full models $w_{i,t+1}$ which need not be small in norm near optimality, β_t keeps increasing over time.

Table 2: Best test accuracy (%), over the last 5 rounds, of global, client-1', and adversary's models with and without quantization. The adversary's reported accuracy is the mean over 5 runs. The setting for each row are mentioned in Figures 1-2.

Dataset	Setting	Entity	No Quant.	With Quant.
CIFAR-10, LeNet-5	FLIP, 50 clients	global	63.32	62.52
		client-1	60.77	60.37
		adversary	14.58	10.00
	FLIP, 20% β change	global	63.32	60.69
		client-1	60.77	58.26
		adversary	14.58	10.00
	FLIP, 16 bits	global	63.32	63.14
		client-1	60.77	60.61
		adversary	14.58	10.00
	FLIP, 8 clients	global	62.98	62.75
		client-1	60.85	59.94
		adversary	20.63	10.00
	FLOP	global	63.14	60.47
		client-1	60.94	57.82
		adversary	60.90	10.00
CIFAR-100, ResNet-18	FLIP	global	39.53	39.21
		client-1	35.57	36.00
		adversary	1.00	1.00
	FLOP	global	39.53	35.63
		client-1	35.57	16.72
		adversary	35.57	1.00

F QUANTIZATION BACKGROUND

A b-bit quantizer with domain $(q_{\min}, q_{\max}) \subset \mathbb{R}$ is a mapping $\mathcal{Q} : \mathbb{R} \to \{0, 1, \ldots, 2^b - 1\}$ where

$$\mathcal{Q}(x) = \begin{cases} \left\lfloor 2^b \frac{x - q_{\min}}{q_{\max} - q_{\min}} \right\rfloor, & \text{if } x \in (q_{\min}, q_{\max}), \\ 0 & \text{if } x \leq q_{\min}, \\ 2^b - 1, & \text{if } x \geq q_{\max}. \end{cases} \tag{8}$$

The region $(-\infty, q_{\min}] \cup [q_{\max}, \infty)$ is referred to as the *saturation region* for the quantizer, as every input in the region $(-\infty, q_{\min}]$ results in a saturated quantized output of 0 and those in $[q_{\max}, \infty)$ result in $2^b - 1$. The quantity $\epsilon := \frac{q_{\max} - q_{\min}}{2^b}$ is referred to as the *quantization resolution*[1]. Notice that $\mathcal{Q}(x)$ is a b-bit binary word

[1]A lower value of ϵ means better resolution.

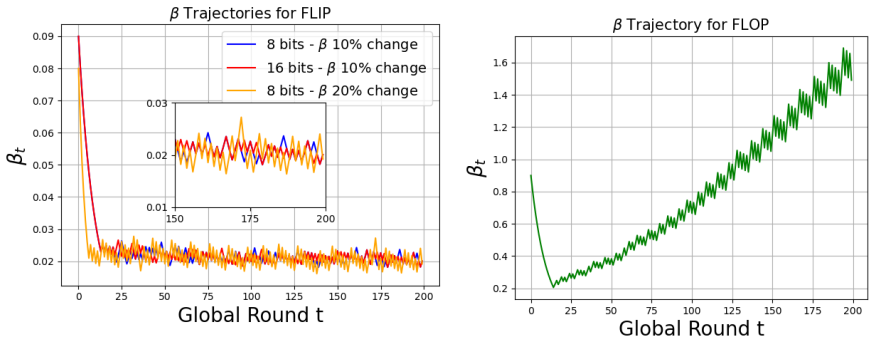


Figure 3: β_t adaptation trajectories in case of dynamic quantization, for (left) FLIP with the FL settings from Figs. 1a-1c and for (right) FLOP with the FL setting from Fig. 1f.

that is agnostic to the quantization domain. Therefore, x is decoded as:

$$x^q = q_{\min} + \epsilon \mathcal{Q}(x) + \tfrac{\epsilon}{2}. \tag{9}$$

We refer to $e(x) := x - x^q$ as the *quantization error*. When x is a vector, quantization and decoding is performed element-wise. The quantization domains $(q_{\min}, q_{\max})$ can be different for different elements of x as well as the number of bits, b, allocated to each element of x can be different. Consequently, different elements of x may have different quantization resolutions and error bounds.

For every $x \in [q_{\min}, q_{\max}]$, the quantization error is in the range $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$, providing a guaranteed upper bound on the mismatch between x and its quantized version. However, such guarantee is missing when x lies in the saturation region, in which case, the mismatch could be arbitrarily large, leading to a significant loss of information. Therefore, while designing a quantizer, one must be careful about picking the domain $[q_{\min}, q_{\max}]$ such that saturation does not occur often. In practice, if a bound on x is known (i.e., $x_{\min} \leq x \leq x_{\max}$), one may simply choose $q_{\min} = x_{\min}$ and $q_{\max} = x_{\max}$. However, it becomes challenging when x changes with time—such as the case for the client weights $w_{i,t}$—and finding a common bound that works across all time is challenging. Even when one finds a bound $[x_{\min}, x_{\max}]$ that works for all time, such a bound can be overly conservative and therefore, the quantization resolution ϵ will be coarse. This results in a natural trade-off between quantization saturation and quantization resolution: reducing the occurrences of saturation requires a wider domain $[q_{\min}, q_{\max}]$ which results in a coarser resolution (i.e., bigger ϵ) and vice-versa. A delicate balance must be maintained in order to maintain the model accuracy.