PPFLex: Securing Non-IID Optimization in Federated Learning via MPC

Nergiz Yuca[1], Christian Internò[2], Nikolay Matyunin[3], Markus Olhofer[3], Barbara Hammer[2], Stefan Katzenbeisser[1]

[1]University of Passau
[2]Bielefeld University
[3]Honda Research Institute Europe GmbH
nergiz.yuca@uni-passau.de

Abstract

Effective deployment of Federated Learning (FL) often faces the dual challenge of ensuring high model performance on heterogeneous (non-IID) data and providing strong privacy guarantees. To improve performance on non-IID data, advanced FL optimization methods have emerged that share auxiliary insights, such as client gradient behaviors. While these gradient-guided optimization FL methods, such as Federated Loss Exploration (`FedLEx`), improve the model accuracy, their reliance on sharing additional gradient information creates an unaddressed privacy vulnerability. In this work, we empirically quantify this privacy-sensitive data leakage and address it through an end-to-end Secure Multi-Party Computation (MPC)-based solution that secures `FedLEx`. Specifically, we conduct a privacy leakage experiment and show that a malicious server can indeed infer clients' label sets from shared guidance matrices in a pathological non-IID setting. `PPFLex` replaces the single server in `FedLEx` setting with three MPC servers to securely compute the global guidance matrix and perform federated averaging under semi-honest and malicious adversary assumptions across Semi2k, Replicated2k, SPDZ2k, and PSReplicated2k protocols using the MP-SPDZ framework. Finally, we quantify the practical trade-offs of using MPC and analyze `FedLEx's` robustness to noise. Our experiments over MNIST data show that `PPFLex` successfully preserves the model accuracy achieved by the unsecured `FedLEx` pipeline while providing stronger privacy guarantees.

Introduction

Machine Learning (ML) has become widely used in various application domains, such as healthcare (Qayyum et al. 2020), finance (Ahmed et al. 2022), autonomous driving (Kiran et al. 2020), and blockchain (Liu et al. 2020). However, data privacy concerns remain a major barrier to widespread adoption. Google introduced Federated Learning (FL) in 2016, enabling collaborative model training without directly sharing raw data. By performing model training locally, FL addresses the risk of model parameter exposure and supports compliance with data protection regulations such as the General Data Protection Regulation (GDPR) (Truong et al. 2020), the California Consumer Privacy Act (CCPA) (Bukaty 2019) and the Personal Data Protection Act (PDPA) (Chik 2013).

Despite its promise, the practical deployment of FL presents several challenges. One is the challenge of statistical heterogeneity, where client data are non-independent and identically distributed (non-IID) on the local devices, occurring when training data attributes are imbalanced across clients due to perturbations (Zhu et al. 2021). To tackle the limitations of standard averaging in these non-IID settings, researchers have increasingly adopted knowledge transfer techniques (Shao et al. 2023) to harmonize learning across diverse client distributions. Federated Loss Exploration (`FedLEx`) (Internò et al. 2024a) is one such method that integrates guided transfer learning into the FL framework by analyzing gradient behavior to guide the learning process. Unlike conventional FL approaches, `FedLEx` does not rely on weight-sharing alone. Instead, it focuses on aggregating the gradient information from the different loss surfaces of each client. This approach is designed to address the challenges of non-IID data, where class distribution differences between clients can create biases that hinder learning performance.

A further concern is privacy: although FL provides a layer of protection, it remains vulnerable to inference attacks, where adversaries reconstruct private training data from intermediate model updates (Chai et al. 2023; Lyu et al. 2020; Mothukuri et al. 2021). Such attacks, including gradient and model inversion, can expose sensitive client information, undermining FL's privacy guarantees (Zhu, Liu, and Han 2019). To address these concerns, several privacy-preserving technologies have been proposed. Homomorphic Encryption (HE) (Doan et al. 2023) allows computations on encrypted data, however introduces high computational overhead. Differential Privacy (DP) (Ji, Lipton, and Elkan 2014) adds noise to protect privacy, however, it may degrade model performance. Secure Multi-Party Computation (MPC) (Lindell 2020) allows a group of untrusted parties to process their data jointly without revealing individual information and is emerging as a promising solution for multi-party settings.

This creates a gap: on one hand, existing MPC-based secure aggregation protocols are mainly focused on traditional FL averaging and do not focus on providing privacy for sharing additional gradient knowledge that can be used to opti-

Accepted at the 1st Workshop on Federated Learning for Critical Applications (FLCA), in conjunction with AAAI 2026.

mize performance in non-IID settings. On the other hand, while FedLEx enhances traditional FL training by leveraging gradient analysis, it lacks privacy guarantees as gradient deviations can still reveal sensitive information, an adversary with access to model updates may reconstruct private training data, similar to traditional FL vulnerabilities.

This paper aims to close this research gap by studying FedLEx (Internò et al. 2024a) as a representative knowledge-transfer optimizer that improves global model performance under non-IID settings and by proposing PPFLex, a privacy-preserving FL framework that combines the strengths of FedLEx and MPC-based secure aggregation. Having established the need for privacy, we first provide empirical evidence that the core optimization process of FedLEx introduces an additional source of privacy leakage. In particular, we demonstrate that the additional gradient information from clients can be exploited by a malicious server to infer properties of their local data distributions.

To address this identified vulnerability, we introduce and analyze PPFLex, a privacy-preserving framework that leverages MPC to enable the private computation of FedLEx's guidance information. The main contribution of our work is an investigation of the resulting trade-offs, offering insights into the real-world costs and benefits of securing non-IID data in advanced FL systems. We use the well-known MNIST benchmark dataset to model the heterogeneous scenario, providing a reproducible base for analyzing the performance and privacy trade-offs of applying MPC. Finally, we analyze the robustness of the core FedLEx algorithm to noisy gradients, which allows us to estimate its noise tolerance in practical FL deployments.

Our contribution. The main contributions of our work are:

- We empirically demonstrate that the additional gradient information shared in FedLEx introduces a privacy vulnerability that can leak sensitive client data and quantify its severity in a realistic setting.
- We develop PPFLex, an end-to-end privacy-preserving implementation of FedLEx via MPC. Unlike existing methods that separately address either non-IID optimization or generic secure aggregation, PPFLex combines both, preserving model accuracy under non-IID data in a privacy-preserving way.
- We provide a benchmark that quantifies the trade-offs between model accuracy, privacy, and communication/-computational overhead across different MPC protocols and adapts to different security assumptions including semi-honest and malicious servers.
- We analyze the FedLEx's tolerance to noisy gradients, a common concern when server-side inspection is restricted, and propose a more resilient normalization approach RobFedLEx.

The remainder of this paper is organized as follows. First, we review related work on secure FL and optimized FL frameworks. Then, we provide background on the underlying FedLEx framework. Afterwards, we introduce our proposed PPFLex framework, describe the implementation details, and present evaluation results, analyzing PPFLex's performance and its computational overhead. Finally, we conclude the paper and suggest directions for future work.

Related Work

In this section, we review recent advancements in secure aggregation techniques for FL and optimized FL frameworks.

Secure Aggregation in FL

Various MPC-based secure aggregation protocols enable FL clients to share their locally trained models with a group of servers in a privacy-preserving manner. For example, SAFELearn (Fereidooni et al. 2021) is a generic privacy-preserving FL framework that employs fully homomorphic encryption and MPC for global model privacy. SafeFL (Gehlhar et al. 2023) is another MPC-based framework designed to evaluate the performance of FL techniques against privacy inference and poisoning attacks. ELSA (Rathee et al. 2023) is a secure aggregation protocol for FL that assumes malicious security against clients in a two-server setting. WW-FL (Marx et al. 2023) introduces a unified MPC-based framework for large-scale FL, enabling global model privacy and robustness against poisoning attacks. SPEFL (Shen et al. 2024) is a privacy-preserving, MPC-based FL framework specifically designed for the Internet of Things (IoT) to provide device privacy and robustness against poisoning attacks on resource-limited devices. ScionFL (Ben-Itzhak et al. 2022) presents an MPC-based secure quantized aggregation framework for FL to reduce server-client communication in FL while providing robustness against malicious clients. Although these works ensure secure aggregation in FL, they do not focus on addressing optimization challenges caused by non-IID data distributions, which our work specifically targets.

Optimizing FL on Non-IID Data

Recent FL optimization studies address non-IID challenges through transfer learning and knowledge distillation. In this context, transferring information from a global server or peer clients often using large pre-trained models to local environments can enhance client performance on unseen data (Li, Zhang, and Kumar 2025; Wang, Sun, and Zhao 2025; Lin, Chen, and Li 2021). Currently, adaptations of meta-learning algorithms, such as an updated variant of the Reptile method, and multitask formulations like MOCHA shows improved personalization and client adaptation. To further tackle heterogeneity, researchers apply hierarchical clustering techniques to categorize clients based on weight update dynamics, facilitating more effective grouping and tailored optimization (Sattler et al. 2019; Lee, Park, and Kim 2025). Parallel efforts aim to identify data distributions that more accurately reflect client diversity (Chen, Patel, and Nguyen 2025; Chen, Zhou, and Wang 2025). Finally, relevant to FedLEx (Internò et al. 2024a), advances in adaptive gradient methods reshape federated optimization. Modern optimizers such as AdamW (Loshchilov and Hutter 2019) and second-order methods are gradually replacing traditional stochastic gradient descent in FL. Techniques like

group normalization further accelerate convergence (Wu and He 2018; Ruiz, Garcia, and Silva 2025). Note that no existing work has yet leveraged the transfer knowledge of loss function topology to guide client gradients integrated with MPC, a gap that our research aims to fill.

Background: Federated Loss Exploration

In a traditional Federated Learning (FL) setting, clients train models on their local device decoupling training from direct access to raw training data. Each client i ($1 \leq i \leq K$) holds data distributed according to $p_i(x, y)$ with $D_i = \{(x_i^{(1)}, y_i^{(1)}), \ldots, (x_i^{(N_i)}, y_i^{(N_i)})\}$, where $x \in \mathbb{R}^d$, representing a d-dimensional feature vector and $y \in \{1, \ldots, M\}$ corresponding to a class label from M categories. The global objective function represents the optimization goal by measuring overall model performance. It is defined to minimize the aggregate loss across all clients as follows:

$$\mathcal{L}(W_{\text{global}}) = \frac{1}{K} \sum_{i=1}^{K} \mathbb{E}_{(x_i, y_i) \sim p_i} [\mathcal{L}(W_{\text{global}}, x_i, y_i)],$$

where $\mathcal{L}(W_{\text{global}}, x_i, y_i)$ denotes the global loss function for a data point (x_i, y_i) with $\mathbb{E}_{(x_i, y_i) \sim p_i}$ denoting the expected loss over each client's local data distribution $p_i(x, y)$. A well-known FL aggregation scheme Federated-Averaging (FedAvg) (McMahan et al. 2016) optimizes this objective through iterative training rounds. First, the server distributes the initial global model parameters W_{global} to clients. Then, clients perform local training on their data and send their updated model parameters back to the server. The server averages the updates to compute the global model.

`FedLEx` extends this process by introducing a knowledge transfer approach (Internò et al. 2024a) to handle the challenge of non-IID data in FL. In such settings, client data distributions differ, causing to suboptimal or slow global model convergence. To overcome this, `FedLEx` introduces a "loss exploration phase" that identifies which model parameters are most sensitive to the statistical differences in each client's data, thereby providing a consistent global model. We now briefly describe the main phases in `FedLEx`; for a detailed explanation, we refer the reader to the original paper (Internò et al. 2024a).

Federated Loss Exploration. In this phase, each client acts as an explorer of its loss landscape, identifying critical parameter directions via gradient deviations (Internò et al. 2024b; Nikolić, Andrić, and Nikolić 2023). At its core, explorer clients train the initial global model on their local data and measure the magnitude of change for each model parameter, which provides insights into each parameter's sensitivity to the client's data distribution. This exploration first applies Stochastic Gradient Descent (SGD) (Bottou 2010) to iteratively adjust model parameters and minimize loss. The update rule for the m-th model parameter $W_{i,m}$ for client i at exploration step e is:

$$W_{i,m}^{(e+1)} = W_{i,m}^{(e)} - \eta \, \nabla \mathcal{L}_i \left(W_{i,m}^{(e)}; D_i \right),$$

where η is the learning rate and $\nabla \mathcal{L}_i(\cdot)$ is the gradient of the local loss function $\mathcal{L}_i$ for client i, computed on its local dataset D_i. The cumulative update over a set number of exploration steps, E_{exp}, is:

$$W_{i,m}^{\text{Final}} - W_{i,m}^{\text{Initial}} = -\eta \sum_{t=1}^{E_{\text{exp}}} \nabla \mathcal{L}_i \left(W_{i,m}^{(t)}; D_i \right),$$

where $W_{i,m}^{\text{Initial}}$ and $W_{i,m}^{\text{Final}}$ are the parameter values at the beginning and end of the exploration phase, respectively, and t indexes the training steps within this phase. Subsequently, to identify the gradient variability of the loss landscape for each parameter m, the squared deviation is computed as: $\delta_{i,m}^2 = \left(W_{i,m}^{\text{Initial}} - W_{i,m}^{\text{Final}} \right)^2$. Finally, each client i compiles these squared deviations into a local guidance matrix, G_{local_i}, where the m-th element is defined as $G_{\text{local}_{i,m}} = \delta_{i,m}^2$.

Global Guidance Matrix Construction. After the exploration phase, the server aggregates the local guidance matrices (G_{local}) from all clients. It then normalizes them to construct the global guidance matrix, G_{global}. Each entry in this matrix indicates the average gradient deviation for a specific model parameter across the clients.

To construct G_{global_m}, we first compute $\bar{G}_m = \frac{1}{K} \sum_{i=1}^{K} G_{\text{local}_{i,m}}$, the average of local matrices across all parameters. More exactly, each entry G_{global_m} indicates the average normalized deviation of the gradient for the m^{th} parameter across clients.

Then we apply min-max normalization to compute the global guidance per parameter m across all clients:

$$G_{\text{global}_m} = \frac{\bar{G}_m - \min(\bar{G}_m)}{\max(\bar{G}_m) - \min(\bar{G}_m)}. \tag{1}$$

Finally, the guidance matrix G_{global} is sent back to the clients in order to guide their model updates in each round of FL.

Federated Learning. Subsequently, the local training process begins with selected clients computing their gradient updates. Let $\Delta W_{i,m}$ represent the gradient update for the m-th parameter of client i's model. Instead of sending this update directly, `FedLEx` incorporates the global guidance matrix to modulate it:

$$\Delta W_{\text{modulated}_{i,m}} = \Delta W_{i,m} \times G_{\text{global}_m},$$

where $\Delta W_{\text{modulated}_{i,m}}$ is the modulated update for the m-th parameter. Finally, clients send these modulated updates to the server, which then applies federated averaging to update the global model.

Overall, `FedLEx` (Internò et al. 2024a) improves model convergence in non-IID settings through the additional gradient exploration and exchange of local matrices. In the next section, we describe how this FL procedure can be implemented more securely, to prevent a malicious server from inferring sensitive data from gradient and model updates.

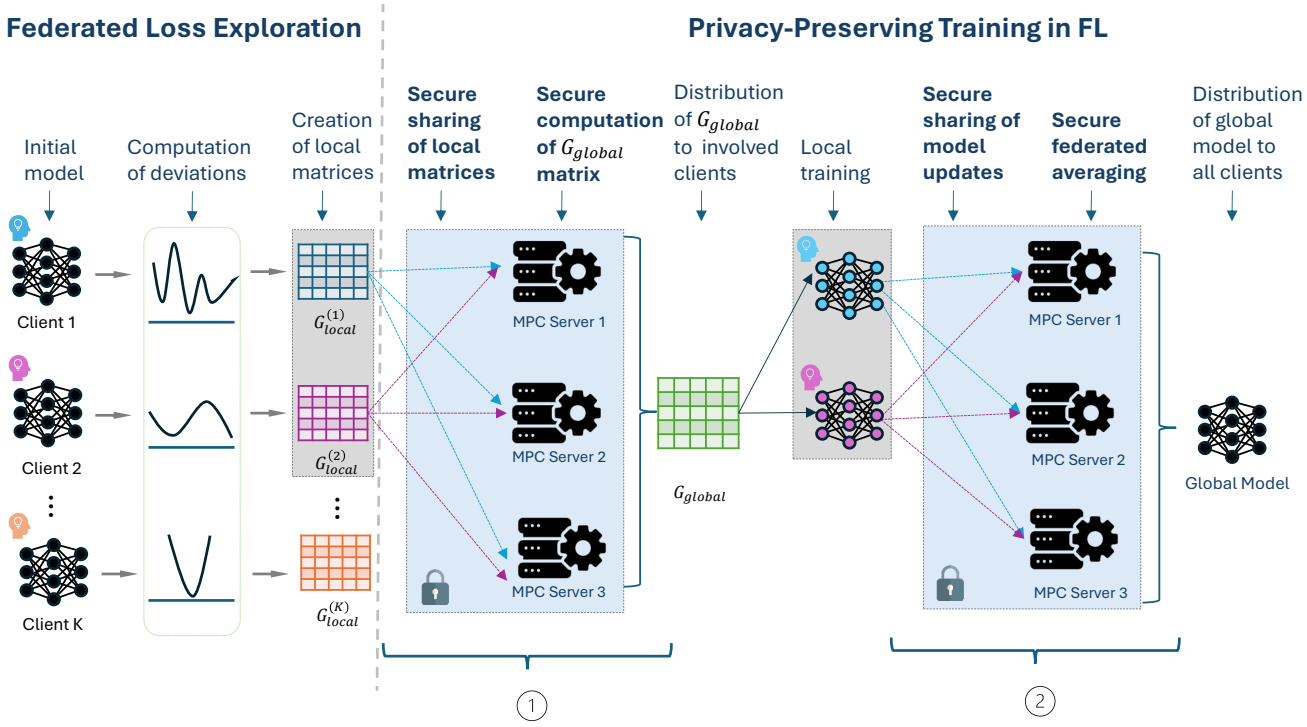

Figure 1: PPFLex: MPC-based privacy-preserving federated learning.

Proposed PPFLex Framework

To address privacy concerns in FedLEx, our PPFLex framework leverages Secure Multi-Party Computation (MPC). To mitigate privacy vulnerabilities associated with single server aggregation in FL, we replace the centralized server in FedLEx with three MPC servers to securely compute G_{global_m} and perform federated averaging in a privacy-preserving manner. Generally, the architecture allows us to implement a setup with two or more servers. The three-server default setting is selected to evaluate various underlying protocols in MP-SPDZ, including honest-majority protocols designed for three parties (Replicated2k, PSReplicated2k), while also supporting dishonest-majority protocols (SPDZ2k, Semi2k). Our approach ensures that the model updates and additional gradient information are shared with MPC servers without revealing privacy-sensitive client data. The proposed enhancements to FedLEx are illustrated in Figure 1.

In our work, we employ *secret sharing–based* MPC protocols. Their core underlying primitive is secret sharing, which enables splitting a secret s among n parties such that any subset of size $t+1$ can reconstruct the secret, while any subset of size at most t learns no information. Linear operations, such as addition and scalar multiplication, can be performed locally on the shares, whereas multiplications of shares rely on preprocessing and interaction among the servers. In our setting, each client decomposes its input into three additive shares, sending one share to each server. The servers jointly compute the aggregated result $\mathbf{G}_{\text{global}}$ and perform federated averaging on secret-shared values, without reconstructing any raw client inputs.

Specifically, PPFLex involves two main MPC operations:

1) Global guidance matrix. At the beginning of the federated learning round, selected participating clients send their gradient deviations (computed locally in the exploration phase) to the three MPC servers. These deviations are secret-shared, and the element-wise minimum and maximum are computed across all local gradient deviation sets, G_{local_i}. As the first MPC computation in PPFLex framework, the MPC servers aggregate them to compute G_{global} for involved clients using Eq. (1) (see ① in Figure 1).

2) Federated Learning. After a local training phase, randomly selected clients send their model updates to the three MPC servers in a secret-shared format. As the second MPC computation in PPFLex framework, the servers perform secure aggregation of updated model parameters of contributing clients to construct a global model (see ② in Figure 1).

Security Assumptions. For both MPC operations, we consider two different security settings. In the semi-honest setting, the servers follow the protocol correctly but attempt to infer sensitive information about the clients' data from their model updates W_i. In the malicious setting, servers can arbitrarily deviate from the protocol by extracting sensitive information about the training dataset from the weights or parameters of the trained model. We evaluate PPFLex under both semi-honest and malicious security assumptions in the Evaluation section.

Our implementation includes four protocols that operate over the arithmetic ring domain of integers modulo a power

Parameter	PPFLex
Dataset, Model	MNIST, 2NN
Layers, hidden layers	2, 200
Num. of epochs in Exploration	150
Batch size, Learning rate	350, 0.0003
Optimizer	Adam,
Loss function	CrossEntropyLoss
Num. clients, total	10
Num. clients, participating	2
Num. numerical params (2NN):	157610
Num. of MPC servers	3
MPC protocols	Semi2k, Replicated SPDZ2k, PsReplicated
Ring size (all protocols)	64 bits (mod 2^{64})
Fixed-point precision	24 fractional bits
Mixed-domain conversions	edaBits

Table 1: Experiment Hyperparameters.

of two (Mod 2^k), where k denotes the bit length of the integers. In a dishonest majority setting, where security is guaranteed even if a majority of the parties are corrupted, SPDZ2k provides malicious security, while Semi2k offers a more efficient variant under semi-honest assumption. In contrast, an honest-majority setting assumes that a majority of parties behave honestly. Under this setting, Replicated2k provides security against semi-honest adversaries, and PSReplicated2k extends it to achieve malicious security(Keller 2020).

Note that while PPFLex prevents server-side leakage through secure MPC computation, it does not defend against malicious clients. Integrating aggregation algorithms robust to malicious clients can be addressed in future work.

Communication Optimization. An alternative approach involves clients sending local model updates G_{local} only once, rather than transmitting them in each round. These updates can be securely stored on the servers in secret-shared format. This allows the servers to reconstruct these local matrices and perform the first MPC computation without repeatedly collecting new inputs from the selected clients.

Implementation

We implement the MPC-based outsourced secure FL aggregation pipeline, `PPFLex`, under MP-SPDZ (Keller 2020). This MPC framework executes MPC programs written in Python-like code under different MPC protocols, supporting more than 40 protocol variants. It allows us to test the solution under different security assumptions and settings. To enable outsourced computation on secret-shared client inputs, we use the `ExternalIO` (MP-SPDZ 2025) interface of MP-SPDZ.

To address the privacy leakages that we identified in `FedLEx`, we implement secure federated computations in MP-SPDZ for both the federated averaging and the guidance matrix computation. Our implementation incorporates two optimizations. First, to minimize MPC communication overhead, the final normalization step for the guidance matrix is performed locally by each client rather than implementing costly non-linear min/max operations under MPC. Second, we use a unified secure aggregation program that handles both the gradient updates and G_{local} matrices and store the results in the filesystem. The source code is publicly available at https://github.com/nergiz-ue/PPFlex.

To connect our `FedLEx` programs to MP-SPDZ, we make targeted adjustments to the FedLEx source by converting the gradient updates and local matrices into fixed-point integers for MPC. Then, we dump necessary MPC inputs to files, and replace plaintext executions with external calls to run MPC clients and computing parties. This generic approach allows us to seamlessly replace plain-text functionality with MPC in the existing FedLEx pipeline, and evaluate the impact on the FL performance.

Evaluation

In this section, we evaluate the `PPFLex` framework. First, we provide evidence of information leakage within guidance information shared by the clients. Second, we measure the impact of the selected MPC functionality on model convergence. Third, we measure the MPC overhead in terms of computation time and traffic. Finally, to assess whether the underlying `FedLEx` optimizer remains suitable for secure deployment in a real-world setting, we conduct a robustness analysis, independent of the MPC integration, under noisy gradient conditions.

For all experiments, we ran the default FedLEx pipeline (as described in the (Internò et al. 2024a) repository) and performed FL on the MNIST dataset in a pathological non-IID setting, with 10 instantiated clients, where all clients participate in the loss exploration phase, and 2 clients are randomly selected in each communication round. The model used is a two-layer network 2NN (McMahan et al. 2016), consisting of 157610 numerical parameters, using $E_{\text{exp}} = 150$, batch size of 350, and a learning rate of 0.0003. The original FedLEx experiments indicate that increasing the number of clients and epochs in the exploration phase can further increase the performance, though there is a saturation point at higher values where additional exploration yields no further gain. Table 1 provides an overview of the different parameters used in the evaluation.

For measuring MPC performance and overhead, we ran computing servers on three *e2-highmem-2* virtual machines (each containing 2 vCPUs with 16GB RAM) on the Google Cloud infrastructure, in a 10GB LAN setting.

An Analysis of Privacy Leakage in FedLEx

The squared deviation values in $\mathbf{G}_{\text{local}}$ reflect how strongly each model parameter adapts to a client's local data distribution in the exploration phase. Since different digit classes activate different neural weights, these deviations form a distinguishable pattern. To empirically demonstrate the privacy vulnerabilities in `FedLEx`'s guidance matrix computation, we conduct a privacy leakage experiment to evaluate whether a malicious server can infer sensitive information about clients' local data distributions from their transmitted

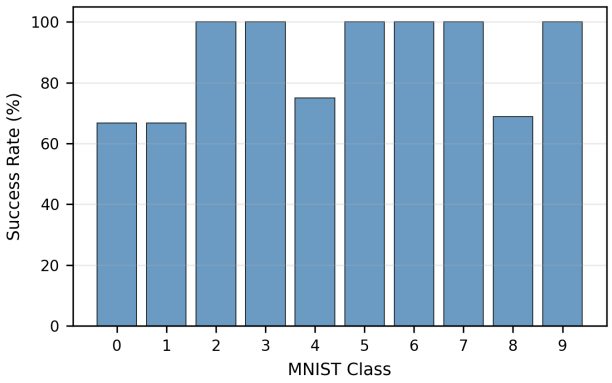

Figure 2: Analysis of Privacy Leakage: MNIST Class Detection Success Rate.

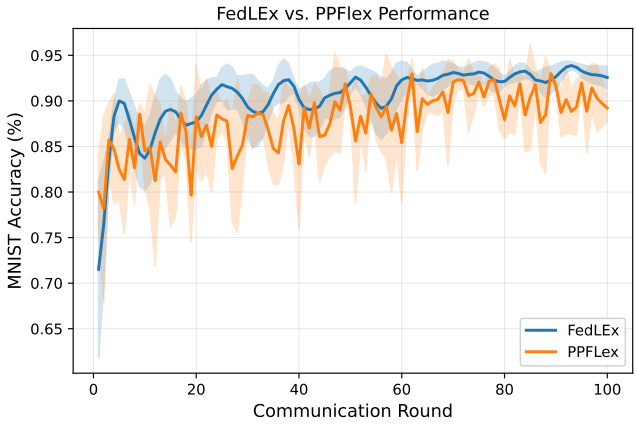

Figure 3: Performance of PPFLex, in comparison to the plain-text FedLEx pipeline.

guidance matrices. We simulate 225 synthetic clients in a pathological non-IID setting, where each client holds data from two out of ten possible MNIST digit classes, covering all 45 class combinations with five clients per combination. Each client trains on 200 samples (100 per class) and generates its G_{local} as a flattened raw deviation vector of 19,850 dimensions, which captures the squared differences between the baseline and scout model parameters. Each deviation vector is paired with the client's corresponding multi-hot encoded class label. To evaluate whether these deviations reveal private information, we use an 80/20 train–test split with a fixed random seed and train a multi-output `RandomForest` classifier to predict which digit classes each client possesses based on its G_{local}.

The experiment results demonstrate measurable privacy exposure. The malicious model correctly infers the data classes of test clients with a 75.6% exact-set-match accuracy. Per-class analysis is shown in Figure 2, with an average 93.5% F1-score, and indicates strong class distinguishability. These findings confirm malicious server's ability to infer the MNIST data classes of the clients in a pathological non-IID setting, and the necessity to apply MPC-based secure aggregation to protect information leakage from the additional gradient information shared by the clients. We believe that even stronger inference attacks based on combining guidance information with model updates can be investigated in future work.

MPC Impact on Model Accuracy

To verify that MPC extensions do not affect the FL performance (as MPC is done on quantized integer values), we run both the original (unsecured) and MPC-enhanced FL pipeline and compare the global model convergence after 100 training rounds. The results are averaged across 5 runs in both settings and presented in Figure 3. `FedLEx` is modular and can be combined with different server-side aggregation strategies, including server momentum-based variants (Hsu, Qi, and Brown 2019), as detailed in the `FedLEx` work. In our comparison (Figure 3), the `FedLEx` baseline uses the momentum-based aggregation (`FedLExAvgM`), whereas `PPFLex` implements plain averaging under MPC without server momentum. Consequently, the MPC implementation achieves slightly lower accuracy and higher per-round variance. Still, we confirm that MPC additions do not affect the overall FL performance and implement the desired functionality correctly.

MPC Overhead

Table 2 shows the overhead of our MPC solution for performing secure federated computations. We use the 2NN model on MNIST to establish a clear performance baseline. We combine the analysis for both FL averaging and guidance matrix computation as their core secure operation is identical: a federated average across client parameters. The final normalization step required for the guidance matrix is handled locally by each client to minimize MPC overhead. In addition, we measure separately the overhead of the online phase only (not including the preprocessing of generating cryptographic randomness such as Beaver triples). We also compare the overhead for four protocols, SPDZ2k, Semi2k, Replicated2k, and PsReplicated2k with semi-honest and malicious settings.

The choice of underlying MPC protocol significantly impacts performance overhead. Across the evaluated protocols, the total execution time for secure computations ranges from 31.1s for the most efficient protocol (Replicated2k) to 2947.6s for the most heavyweight (SPDZ2k) even for this relatively simple model. A similar trend is observed in communication overhead, varying between 27.6 MB (Replicated2k) and 339655.0 MB (SPDZ2k).

This performance gap highlights the trade-off between efficiency and the security guarantees provided by each MPC protocol. SPDZ2k incurs the highest overhead among the protocols due to malicious security under a dishonest majority. Semi2k reduces cost by assuming semi-honest behavior. PsReplicated2k provides malicious security under an honest majority with moderate overhead. Replicated2k is the most efficient due to its semi-honest security model with an honest majority. This experiment highlights the necessity of carefully designed protocols for implementing non-

Table 2: Runtime and communication results for FL Averaging and Guidance Matrix computation across different MPC protocols. The table reports online (o) and total (t) runtime in seconds and communication overhead in MB. All protocols perform computations over an Arithmetic Ring ($\mathbb{Z}_{2^k}$) domain, where k represents the bit length of the integers. Protocols are grouped by security model: Dishonest Majority (DM) or Honest Majority (HM), supporting either Semi-Honest (SH) or Malicious (Mal) adversaries.

MPC Protocol	**# Parties**	**Domain**	**Security**	**Secure Federated Computations (Averaging and Guidance Matrix) via MPC**			
				Time (o)	**Time (t)**	**Comm (o)**	**Comm (t)**
SPDZ2k	3	$\mathbb{Z}_{2^k}$	DM/Mal	3.6	2947.6	10.1	339655.0
Semi2k	3	$\mathbb{Z}_{2^k}$	DM/SH	15.1	124.8	6.7	16957.6
PSReplicated2k	3	$\mathbb{Z}_{2^k}$	HM/Mal	18.5	365.6	5.0	360.6
Replicated2k	3	$\mathbb{Z}_{2^k}$	HM/SH	16.0	31.1	2.5	27.6

linear operations under MPC, and shows that optimization is required to support the computational demands of more complex datasets and deeper network architectures.

Robustness Analysis of FedLEx Convergence Under Noisy Gradients

The robustness analysis presented here is conceptually independent of the MPC implementation. While `PPFLex` ensures private execution on server-side, it does not prevent clients from behaving maliciously, and under MPC, the servers cannot directly observe updates from potentially malicious clients. Therefore, in this experiment, we make a first step to see how `FedLEx` behaves in the presence of noisy data.

To evaluate the robustness to noisy gradients, we introduce `RobFedLEx`, a variant utilizing robust normalization based on the Interquartile Range (IQR). Specifically, `RobFedLEx` replaces the original min-max normalization (Equation (1)) with:

$$G_{\text{global}_m} = \frac{\bar{G}_m - Q_1(\bar{G}_m)}{Q_3(\bar{G}_m) - Q_1(\bar{G}_m)},$$

where Q_1 and Q_3 denote the first and third quartiles, respectively. Experiments were conducted by adding Gaussian noise to client gradients: $\tilde{g}_i = g_i + \alpha\,\epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

The results are shown in Figure 4. On MNIST, RobFedLEx achieves an accuracy of approximately 88%, 80%, and 67% for noise intensities $\alpha = 2, 4, 6$, outperforming FedLEx (82%, 75%, 62%) and FedAvg (80%, 70%, 60%) consistently by 5–10%. CIFAR10 results follow a similar trend, with RobFedLEx improving accuracy by 5–10% over FedAvg across all noise levels. Overall, all methods suffer performance degradation in the presence of noise, but RobFedLEx shows higher accuracy and smoother convergence.

Conclusion

In this paper, we propose `PPFLex`, an MPC-based framework designed to enable secure aggregation specifically in federated loss exploration phase of `FedLEx` framework, supporting m clients and n servers in non-IID settings. The framework provides strong privacy guarantees for gradient information-based optimization, allowing insights to be shared without revealing sensitive client data. Our evaluation results show the trade-offs between efficiency, privacy, and communication overhead across different MPC protocols. These findings emphasize the need for optimized protocols and tailored MPC implementations to improve efficiency without compromising privacy. Additionally, our robustness analysis shows that although `FedLEx` is sensitive to noisy or perturbed client inputs, its robustness can be improved through `RobFedLEx`, indicating a promising direction for integrating robust approaches within secure aggregation in future work. Moreover, we will focus on optimizing computational efficiency, extending scalability to larger models and more clients, and deploying `PPFLex` to a real-world scenario as part of our future research direction.

References

Ahmed, S.; Alshater, M. M.; Ammari, A.; and Hammami, H. 2022. Artificial Intelligence and Machine Learning in Finance: A Bibliometric Review. *Research in International Business and Finance*.

Ben-Itzhak, Y.; Mollering, H.; Pinkas, B.; Schneider, T.; Suresh, A.; Tkachenko, O.; Vargaftik, S.; Weinert, C.; Yalame, H.; and Yanai, A. 2022. ScionFL: Efficient and Robust Secure Quantized Aggregation. *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 490–511.

Bottou, L. 2010. Large-Scale Machine Learning with Stochastic Gradient Descent. In *International Conference on Computational Statistics*.

Bukaty, P. 2019. The California Consumer Privacy Act (CCPA).

Chai, D.; Wang, L.; Yang, L.; Zhang, J.; Chen, K.; and Yang, Q. 2023. A Survey for Federated Learning Evaluations: Goals and Measures. *IEEE Transactions on Knowledge and Data Engineering*, 36: 5007–5024.

Chen, R.; Patel, S.; and Nguyen, L. 2025. Global Aggregation Techniques in Federated Learning: Beyond Weighted Averages. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2345–2353.

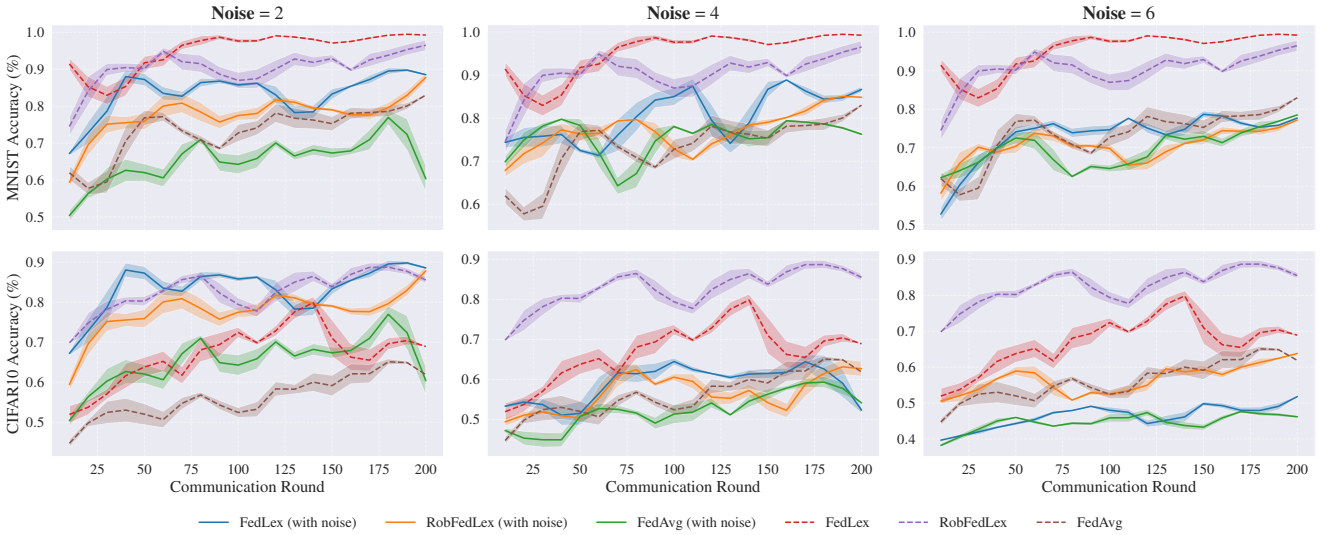

Figure 4: Top row: MNIST accuracy curves (percentage) vs. communication rounds under three noise scaling factors ($\alpha = 2$, 4, and 6). Bottom row: CIFAR10 accuracy curves under the same noise conditions. Three approaches are compared: FedLEx (with noise), RobFedLEx (with noise), and FedAvg (with noise). While all methods suffer performance degradation with increasing noise levels, RobFedLEx shows higher accuracy and smoother convergence, particularly for moderate noise intensities.

Chen, X.; Zhou, Y.; and Wang, J. 2025. Revisiting Global Aggregation in Federated Learning: A Data Distribution Perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2): 567–578.

Chik, W. B. 2013. The Singapore Personal Data Protection Act and an assessment of future trends in data privacy reform. *Comput. Law Secur. Rev.*, 29: 554–575.

Doan, T. V. T.; Messai, M.-L.; Gavin, G.; and Darmont, J. 2023. A survey on implementations of homomorphic encryption schemes. *The Journal of Supercomputing*, 79: 15098–15139.

Fereidooni, H.; Marchal, S.; Miettinen, M.; Mirhoseini, A.; Möllering, H.; Nguyen, T. D.; Rieger, P.; Sadeghi, A.-R.; Schneider, T.; Yalame, H.; and Zeitouni, S. 2021. SAFE-Learn: Secure Aggregation for private FEderated Learning. *2021 IEEE Security and Privacy Workshops (SPW)*, 56–62.

Gehlhar, T.; Marx, F.; Schneider, T.; Suresh, A.; Wehrle, T.; and Yalame, H. 2023. SafeFL: MPC-friendly Framework for Private and Robust Federated Learning. *2023 IEEE Security and Privacy Workshops (SPW)*, 69–76.

Hsu, T.-M. H.; Qi; and Brown, M. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *ArXiv*, abs/1909.06335.

Internò, C.; Olhofer, M.; Jin, Y.; and Hammer, B. 2024a. Federated Loss Exploration for Improved Convergence on Non-IID Data. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Internò, C.; Raponi, E.; van Stein, N.; Bäck, T.; Olhofer, M.; Jin, Y.; and Hammer, B. 2024b. Adaptive Model Pruning in Federated Learning through Loss Exploration. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*.

Ji, Z.; Lipton, Z. C.; and Elkan, C. P. 2014. Differential Privacy and Machine Learning: a Survey and Review. *ArXiv*, abs/1412.7584.

Keller, M. 2020. MP-SPDZ: A Versatile Framework for Multi-Party Computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A. A. A.; Yogamani, S. K.; and P'erez, P. 2020. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23: 4909–4926.

Lee, S.; Park, J.; and Kim, D. 2025. Hierarchical Clustering for Personalized Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1023–1030.

Li, Q.; Zhang, Y.; and Kumar, A. 2025. Federated Transfer Learning for Non-IID Data in Edge Computing. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1): 123–135.

Lin, H.; Chen, X.; and Li, S. 2021. Ensemble Methods for Robust Federated Learning on Non-IID Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 7890–7900.

Lindell, Y. 2020. Secure Multiparty Computation (MPC). *IACR Cryptol. ePrint Arch.*, 2020: 300.

Liu, Y.; Fellow, I. F. R. Y.; Li, X.; Fellow, I. V. C. M. L.; Richard, F.; and Ji, H. 2020. Blockchain and Machine Learning for Communications and Networking Systems. *IEEE Communications Surveys & Tutorials*, 22: 1392–1431.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101.

Lyu, L.; Yu, H.; Ma, X.; Sun, L.; Zhao, J.; Yang, Q.; and Yu, P. S. 2020. Privacy and Robustness in Federated Learning: Attacks and Defenses. *IEEE Transactions on Neural Networks and Learning Systems*, 35: 8726–8746.

Marx, F.; Schneider, T.; Suresh, A.; Wehrle, T.; Weinert, C.; and Yalame, H. 2023. WW-FL: Secure and Private Large-Scale Federated Learning.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics*.

Mothukuri, V.; Parizi, R. M.; Pouriyeh, S.; ping Huang, Y.; Dehghantanha, A.; and Srivastava, G. 2021. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.*, 115: 619–640.

MP-SPDZ. 2025. ExternalIO - MP-SPDZ. Accessed: 14-Feb-2025.

Nikolić, D.; Andrić, D.; and Nikolić, V. 2023. Guided Transfer Learning. ArXiv:2303.16154 [cs].

Qayyum, A.; Qadir, J.; Bilal, M.; and Al-Fuqaha, A. I. 2020. Secure and Robust Machine Learning for Healthcare: A Survey. *IEEE Reviews in Biomedical Engineering*, 14: 156–180.

Rathee, M.; Shen, C.; Wagh, S.; and Popa, R. A. 2023. ELSA: Secure Aggregation for Federated Learning with Malicious Actors. *2023 IEEE Symposium on Security and Privacy (SP)*, 1961–1979.

Ruiz, M.; Garcia, J.; and Silva, A. 2025. Advanced Group Normalization for Accelerated Convergence in Federated Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Sattler, F.; Wiedemann, S.; Müller, K.-R.; and Samek, W. 2019. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5464–5468.

Shao, J.; Li, Z.; Sun, W.; Zhou, T.; Sun, Y.; Liu, L.; Lin, Z.; and Zhang, J. 2023. A Survey of What to Share in Federated Learning: Perspectives on Model Utility, Privacy Leakage, and Communication Efficiency. *ArXiv*, abs/2307.10655.

Shen, L.; Ke, Z.; Shi, J.; Zhang, X.; Sun, Y.; Zhao, J.; Wang, X.; and Zhao, X. 2024. SPEFL: Efficient Security and Privacy-Enhanced Federated Learning Against Poisoning Attacks. *IEEE Internet of Things Journal*, 11: 13437–13451.

Truong, N. B.; Sun, K.; Wang, S.; Guitton, F.; and Guo, Y. 2020. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Comput. Secur.*, 110: 102402.

Wang, L.; Sun, M.; and Zhao, P. 2025. Adaptive Transfer Learning in Federated Settings for Heterogeneous Data. In *Proceedings of the IEEE International Conference on Machine Learning (ICML)*, 456–465.

Wu, Y.; and He, K. 2018. Group Normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Zhu, H.; Xu, J.; Liu, S.; and Jin, Y. 2021. Federated learning on non-IID data: A survey. *Neurocomputing (Amsterdam)*, 465: 371 – 390.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep Leakage from Gradients. In *Neural Information Processing Systems*.