Graph Representation Learning Augmented Model Manipulation on Federated Fine-Tuning of LLMs

Hanlin Cai, *Member, IEEE,* Kai Li, *Senior Member, IEEE,* Houtianfu Wang, Haofan Dong, *Member, IEEE,* Yichen Li, *Member, IEEE,* Falko Dressler, *Fellow, IEEE,* and Ozgur B. Akan, *Fellow, IEEE*

***Abstract*—Federated fine-tuning (FFT) has emerged as a privacy-preserving paradigm for collaboratively adapting large language models (LLMs). Built upon federated learning, FFT enables distributed agents to jointly refine a shared pretrained LLM by aggregating local LLM updates without sharing local raw data. However, FFT-based LLMs remain vulnerable to model manipulation threats, in which adversarial participants upload manipulated LLM updates that corrupt the aggregation process and degrade the performance of the global LLM. In this paper, we propose an Augmented Model maniPulation (AugMP) strategy against FFT-based LLMs. Specifically, we design a novel graph representation learning framework that captures feature correlations among benign LLM updates to guide the generation of malicious updates. To enhance manipulation effectiveness and stealthiness, we develop an iterative manipulation algorithm based on an augmented Lagrangian dual formulation. Through this formulation, malicious updates are optimized to embed adversarial objectives while preserving benign-like parameter characteristics. Experimental results across multiple LLM backbones demonstrate that the AugMP strategy achieves the strongest manipulation performance among all competing baselines, reducing the global LLM accuracy by up to 26% and degrading the average accuracy of local LLM agents by up to 22%. Meanwhile, AugMP maintains high statistical and geometric consistency with benign updates, enabling it to evade conventional distance- and similarity-based defense methods.**

***Index Terms*—Federated fine-tuning (FFT), federated large language models (FedLLMs), adversarial model manipulation**

I. Introduction

Recent advances in large language models (LLMs) have enabled various edge intelligence services, including natural language understanding, content generation, and decision support [1]. As LLMs are increasingly deployed across distributed devices and CyberEdge networks, there is a growing need to continuously adapt these models using decentralized data while preserving user privacy and reducing communication overhead [2], [3]. This requirement has motivated the development of collaborative training techniques that support scalable and resilient model adaptation across distributed settings [4].

H. Cai, K. Li, H. Wang, H. Dong, Y. Li and O. B. Akan are with the Centre for neXt Communications (CXC), Department of Engineering, University of Cambridge, CB3 0FA Cambridge, U.K. (e-mail: {hc663,kl596,hw680,hd489,oba21}@cam.ac.uk).

K. Li is also with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855, Luxembourg (e-mail: kaili@ieee.org).

F. Dressler is with the Telecommunication Networks group (TKN) at the School of Electrical Engineering and Computer Science, TU Berlin, Germany (e-mail: dressler@ccs-labs.org).

O. B. Akan is also with the Center for neXt-Generation Communications (CXC), Department of Electrical and Electronics Engineering, Koç University, 34450 Istanbul, Türkiye. (e-mail: akan@ku.edu.tr).

Federated fine-tuning (FFT) enables multiple agents to collaboratively adapt shared pretrained LLMs while keeping training data local, thereby satisfying privacy and data-residency constraints [5]. This distributed training technique gives rise to federated large language models (FedLLMs) [6]. In FedLLMs, each LLM agent fine-tunes the model on its private dataset and uploads local model updates to a coordinating edge server, which aggregates the local updates to obtain a global model. The global model is redistributed to all participating agents for the next round of FFT. To make FedLLMs practical for billion-parameter models under limited computational resources and wireless bandwidth, low-rank adaptation (LoRA) [7] has emerged as an effective parameter-efficient FFT technique. In LoRA-based FFT, the pretrained LLM backbone remains frozen, while lightweight low-rank adaptation matrices inserted into selected layers are trained and communicated [8]. By transmitting LoRA updates instead of full parameters, FedLLMs reduce communication overhead and accelerate training convergence, particularly under heterogeneous data distributions across different agents [9]–[11].

Despite the privacy-preserving advantages of FFT, adversarial model manipulation remains a critical threat to the resilience of FedLLMs [12], [13]. Under this threat model, an adversary generates and uploads malicious updates during the FFT process to corrupt the aggregated global LLM and degrade its accuracy. To mitigate manipulation threats, many defense methods have been studied for FedLLMs. Most existing defense methods rely on geometric consistency metrics to identify malicious updates, typically using Euclidean distance or cosine similarity to detect statistical outliers [14]–[18].

In this paper, we propose a novel manipulation strategy against FedLLMs, termed Augmented Model maniPulation (AugMP), which targets the FFT process by crafting malicious updates that remain statistically consistent with benign model updates while embedding adversarial objectives. The proposed AugMP strategy aims to disrupt the training process of FedLLMs and steer the global LLM away from its benign optimization trajectory without introducing detectable abnormalities, thereby causing significant performance drops while bypassing existing distance- and similarity-based defenses.

Specifically, an adversarial graph representation learning (GRL) framework is developed to construct a feature correlation graph from the model updates, capturing LLM parameter characteristics and guiding the generation of malicious updates. Within the proposed adversarial GRL framework, a variational graph autoencoder (VGAE) is employed to learn graph-structured representations extracted from benign local and

global updates, thereby reconstructing the underlying graph structure among benign updates. Based on the reconstructed graph structure, a graph spectral transformation (GST) module is designed to derive reconstructed feature representations and generate the malicious updates. To enhance the manipulation effectiveness and stealthiness of malicious updates, an adversarial iterative manipulation algorithm is investigated based on the augmented Lagrangian dual formulation. This algorithm enforces the distance and similarity constraints while strengthening the ability of malicious updates to distort the global optimization trajectory and increase global training loss.

Over successive communication rounds, the AugMP strategy progressively corrupts the global LLM. Due to the broadcast nature of FedLLMs, the manipulated global model is disseminated to all local agents for subsequent training, allowing the AugMP-induced manipulations to propagate throughout the entire system. As a result, AugMP not only causes a substantial degradation in global test accuracy but also impairs the local performance of benign agents. At the edge server, model manipulation detection can be employed to identify statistically significant deviations or anomalies in local updates that may indicate adversarial behavior. As AugMP leverages a GRL framework to generate malicious updates with benign-like statistical and geometric properties, such updates remain difficult to detect using conventional defenses based on Euclidean distance or cosine similarity.

The key contributions of this paper are as follows.

- A novel model manipulation strategy against FedLLMs, termed AugMP, is proposed. AugMP constructs a feature correlation graph from benign updates and leverages the GRL framework to synthesize malicious updates with benign-like parameter characteristics, thereby evading widely adopted distance- and similarity-based defenses.
- A new iterative manipulation algorithm is developed based on the augmented Lagrangian dual formulation to constrain geometric consistency while enhancing adversarial objectives, which steers the aggregated global LLM along an adversarially favorable trajectory, ultimately leading to significant degradation in FedLLMs accuracy.
- Extensive experiments conducted on three LLM backbones, including DistilBERT, Pythia, and Qwen2.5, and two representative datasets, namely *AG News* and *Yahoo! Answers*, evaluate the proposed AugMP strategy against state-of-the-art manipulation baselines. The results demonstrate that AugMP consistently outperforms competing methods in both manipulation effectiveness and stealthiness, reducing global LLM accuracy by up to 26% and degrading the average accuracy of local LLM agents by up to 22%, while preserving the highest degree of statistical and geometric stealthiness. The AugMP implementation is developed in PyTorch, and the source code is publicly available on GitHub: https://github.com/GuangLun2000/AugMP.

The remainder of this paper is organized as follows. Section II reviews the background of adversarial model manipulation against FedLLMs. Section III describes the FedLLMs system model. Section IV formulates the optimization problem. The proposed AugMP strategy is presented in Section V. Performance evaluation and resilience analysis are discussed in Section VI. Finally, Section VII concludes this paper.

II. Related Works

In this section, we review recent state-of-the-art adversarial poisoning and model manipulation threats targeting federated learning (FL) and FedLLMs.

A. Model Poisoning on FL

Existing model poisoning aims to inject crafted adversarial model updates into the FL aggregation process to hinder convergence and degrade the overall performance of FL [29]. A model poisoning algorithm against Byzantine-robust aggregation in FL is presented in [19], where compromised agents replace benign updates with poisoned updates designed to increase the testing error of the aggregated global model. As a result, although the robust aggregation rule still operates on the submitted local updates, part of the aggregated parameters has already been manipulated toward a higher global error rate.

A perturbation-based poisoning method is presented in [20], where the injected perturbation of each malicious update is constrained within the empirical variance of benign updates. The malicious updates thus remain close to benign ones and are less likely to be filtered out by distance-based detection methods. The FL aggregation result is then shifted by the accumulated effect of such perturbed updates.

A poisoning method based on fake agent injection is studied in [21]. In each communication round, fake agents construct malicious local updates that point from the current global model to an adversary-chosen base model with lower accuracy. The malicious update is then scaled before submission, so repeated aggregation gradually pulls the global model toward the low-accuracy reference model and reduces testing accuracy.

Graph learning-based poisoning methods against FL have been explored in [22], [23]. Benign users upload their local models to an edge server, while the adversary passively intercepts shared updates from neighboring agents. Graph autoencoders are used to model data features among benign model updates and to guide the generation of malicious updates. A classic Lagrangian dual optimization method is designed to refine the malicious updates, thereby decreasing FL accuracy.

A user isolation-based poisoning against decentralized FL systems is presented in [24], where an adversarial graph neural network is used by the adversary to refine and modify the data features of local model updates from neighboring agents. The user isolation poisoning curtails the genuine data features of benign local updates, thereby diminishing their beneficial influence in the decentralized aggregation process.

Existing model poisoning methods [19]–[24] operate by introducing anomalous deviations, scaling parameters, or modifying benign feature representations within local model updates. These adversary designs are built on magnitude constraints or statistical data features collected from benign model updates. When extended to FedLLMs with billions of parameters, the adversary requires learning high-dimensional feature correlations among benign LLM updates.

TABLE I
COMPARISON OF EXISTING ADVERSARIAL THREATS AND THE PROPOSED AUGMP.

Key Properties ↓ Manipulation Methods ⟶	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	**AugMP**
Targeting FFT process of FedLLMs							✓	✓	✓	✓	✓
Generating malicious updates for FedLLMs aggregation	✓	✓		✓	✓	✓					✓
Manipulation without adversarial dataset injection	✓	✓	✓	✓	✓	✓				✓	✓
Optimization-based manipulation algorithm design				✓	✓	✓			✓	✓	✓
Learning and manipulating feature correlations among benign LLM updates											✓
Preserve benign characteristics to bypass distance- and similarity-based defenses											✓

B. Adversarial Threats on FedLLMs

The rapid development of FedLLMs has led to a growing interest in adversarial threats tailored to the FFT process, including jailbreak, instruction, backdoor, and poisoning methods. Specifically, a jailbreak method targeting FedLLMs is presented in [25], where the adversarial agents construct a malicious dataset of harmful prompt-response pairs and use it to train their local LLMs, thereby injecting unsafe generation behaviors through the FFT process. A safety-unaligned data injection method for federated instruction tuning is presented in [26]. Malicious agents generate safety-unaligned training data from public unaligned sources or an off-the-shelf adversarial LLM. The injected local updates gradually erode the safety alignment of the global LLM through aggregation.

A feature-shift backdoor threat against FedLLMs is studied in [27]. The adversary uses accessible benign sample features to guide a stable diffusion model in generating poisoned samples whose feature representations are close to the target feature. These poisoned samples are then incorporated into local training to implant the backdoor. A perturbation-based matrix poisoning threat targeting LoRA-based FedLLMs is presented in [28]. During FFT, the adversary injects two malicious low-rank matrices whose product forms the adversarial LoRA update. The adversary introduces perturbations into the parameters of the malicious matrices, thereby causing parameter deviations that disrupt the FedLLMs training.

C. Our Contributions

Existing adversarial threats [19]–[28] typically rely on conspicuous perturbations in LLM update parameters or injected anomalies that can be detected by distance- and similarity-based defense methods in FedLLMs. As summarized in Table I, the proposed AugMP strategy represents a fundamentally different threat. AugMP leverages the adversarial GRL framework to capture feature correlations among benign updates and generate malicious updates that preserve benign-like characteristics while embedding adversarial objectives. The malicious updates manipulate the FFT process, degrading the accuracy of FedLLMs without introducing detectable abnormalities.

III. FORMULATION OF FEDLLMS SYSTEM MODEL

This section presents the system model of FedLLMs under adversarial settings, including benign LLM agents, adversarial agents, as well as distance- and similarity-based defenses.

A. Federated Fine-Tuning (FFT)

As shown in Fig. 1(a), the FedLLMs system consists of I benign LLM agents. Each local agent $i \in [1, I]$ maintains a dataset $\mathcal{D}_i$ of size $|\mathcal{D}_i| = D_i$ to train its local LLM, and the local datasets follow non-IID distributions across agents. Due to the billion-parameter scale of modern LLMs and the limited wireless bandwidth, the FedLLMs system employs parameter-efficient FFT. Let $\mathbf{w}_i(t) \in \mathbb{R}^{1\times M}$ denote the vectorized trainable parameters updated by agent i at communication round t, where M is the parameter dimension. The loss function of agent i in the t-th communication round is

$$F\big(\mathbf{w}_i(t)\big) = \frac{1}{D_i}\sum_{(x,y)\in\mathcal{D}_i} f\Big(\mathcal{M}(x, \mathbf{w}_i(t)), y\Big), \tag{1}$$

where $\mathcal{M}(x, \mathbf{w}_i(t))$ denotes the model output parameterized by $\mathbf{w}_i(t)$, and $f(\cdot, y)$ represents the task-specific loss function (e.g., cross-entropy) [22]. Upon completing local training in round t, each agent obtains $\mathbf{w}_i(t)$ and transmits its local increment $\Delta\mathbf{w}_i(t) = \mathbf{w}_i(t) - \mathbf{w}_g(t-1)$ to the edge server, where $\mathbf{w}_g(t-1)$ denotes the global vectorized trainable parameters broadcast at the beginning of round t. For notational simplicity, we refer to the local increment $\Delta\mathbf{w}_i(t)$ and the aggregated increment $\Delta\mathbf{w}_g(t)$ as the benign local update and the global update, respectively. The edge server aggregates the received benign updates as $\Delta\mathbf{w}_g(t) = \sum_{i=1}^{I} \frac{D_i}{\sum_{k=1}^{I} D_k}\Delta\mathbf{w}_i(t)$ and obtains the global parameters by

$$\mathbf{w}_g(t) = \mathbf{w}_g(t-1) + \eta\,\Delta\mathbf{w}_g(t), \tag{2}$$

where η is the learning rate of the edge server. After aggregation, the obtained global parameters will be broadcast to all local agents as the reference for the next training round.

B. Low-Rank Adaptation (LoRA)

LoRA is a parameter-efficient FFT technique that adapts pretrained LLMs by injecting trainable low-rank matrices into frozen weights, thereby reducing memory usage and communication overhead while preserving model performance. In transformer-based language models, such as BERT-family encoders and GPT-family decoder LLMs, LoRA is typically applied to selected linear projections in self-attention and feed-forward modules [7]. Given a pretrained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$, where d and k denote the output and input dimensions of the corresponding linear transformation, respectively, LoRA approximates the task-specific update according to the low-rank decomposition:

$$\Delta\mathbf{W} = \mathbf{B}\mathbf{A},\ \mathbf{A} \in \mathbb{R}^{r\times k},\ \mathbf{B} \in \mathbb{R}^{d\times r},\ r \ll \min(d, k), \tag{3}$$

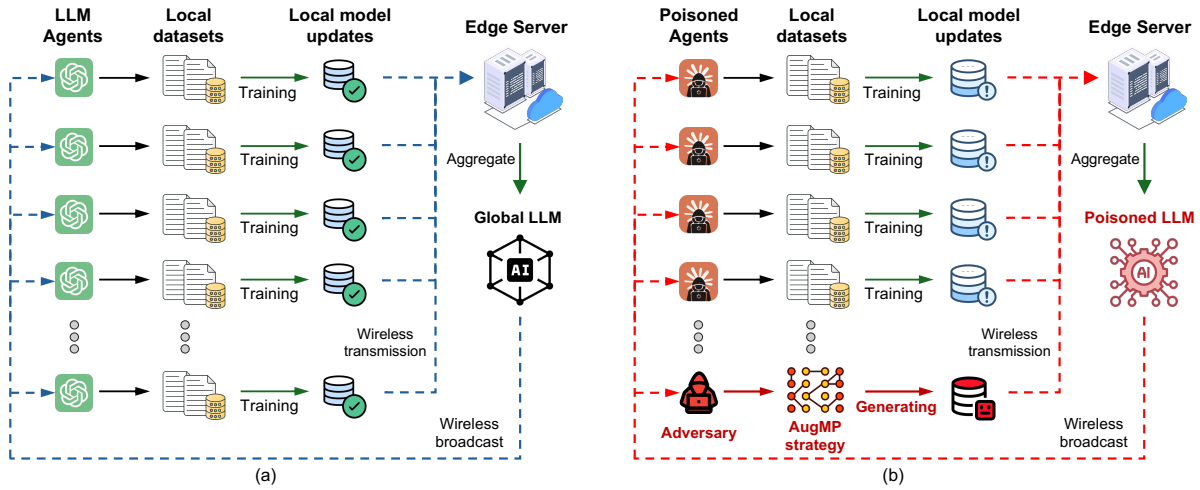


Fig. 1. (a) Benign training process of the FedLLMs system, and (b) impact of the adversary on the FedLLMs training process.

where r denotes the adaptation rank, $\mathbf{A}$ is the low-rank down-projection matrix, and $\mathbf{B}$ is the low-rank up-projection matrix. Accordingly, the product $\mathbf{BA}$ provides a low-rank approximation of the trainable weight update $\Delta\mathbf{W} \in \mathbb{R}^{d\times k}$. During forward propagation, the effective weight becomes $\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W}$, while $\mathbf{A}$ and $\mathbf{B}$ are optimized and the pretrained backbone remains frozen. This design exploits the low intrinsic dimensionality of task adaptation and enables effective learning with a small number of trainable parameters.

In FedLLMs, each local agent i updates its layer-wise low-rank matrices $\mathbf{A}_i^{(\ell)}(t)$ and $\mathbf{B}_i^{(\ell)}(t)$ at round t, where $\ell \in [1, L]$ denotes the adapted layer index and L is the total number of adapted layers. The corresponding LoRA update at layer ℓ is given by $\Delta\mathbf{W}_i^{(\ell)}(t) = \mathbf{B}_i^{(\ell)}(t)\mathbf{A}_i^{(\ell)}(t)$. The layer-wise LoRA updates across all adapted layers are then vectorized and concatenated into a unified LoRA model update:

$$\mathbf{w}_i(t) = \mathrm{concat}\big(\mathrm{vec}\big(\Delta\mathbf{W}_i^1(t)\big), \ldots, \mathrm{vec}\big(\Delta\mathbf{W}_i^L(t)\big)\big) \quad (4)$$

where $\mathrm{vec}(\cdot)$ denotes the vectorization operation that reshapes a matrix into a vector, and $\mathrm{concat}(\cdot)$ denotes vector concatenation across all adapted layers.

C. Threat Model

As shown in Fig. 1(b), the adversarial agent $j \in [1, J]$ acts as a legitimate but malicious agent and can observe the local updates transmitted by a subset of benign agents, as well as the global update broadcast by the edge server. The adversary's knowledge consists of a subset of benign local updates and the global updates. Based on the shared benign updates, the adversary extracts their feature correlations and generates malicious updates $\Delta\mathbf{w}_j'(t)$ that preserve benign-like parameter characteristics while embedding adversarial objectives. These malicious updates are then uploaded to the edge server. Since the edge server is unaware of the adversary's presence, it aggregates the malicious updates together with the benign local updates, thereby obtaining a manipulated global update $\Delta\mathbf{w}_g'(t)$ at the t-th communication round. The corresponding manipulated global LoRA parameters, denoted by $\mathbf{w}_g'(t)$, are then broadcast to all local agents as the reference for the next round of local training. Therefore, the effect of model manipulation progressively spreads throughout the FedLLMs system, resulting in performance degradation.

D. Defense Model

Existing defenses against adversarial threats commonly assess the statistical and geometric consistency of local model updates in the parameter space using metrics such as Euclidean distance and cosine similarity [30]. Euclidean distance quantifies the deviation of a local update from the global update in the parameter space and is defined as [31]

$$d\big(\Delta\mathbf{w}_j(t), \Delta\mathbf{w}_g(t)\big) = \|\Delta\mathbf{w}_j(t) - \Delta\mathbf{w}_g(t)\|_2 . \quad (5)$$

By measuring the Euclidean distance between each local update and the global update, the defense seeks to identify updates that exhibit anomalous deviations in the parameter space. Accordingly, if the distance of an update exceeds a predefined threshold, denoted by d_T, it is classified as an outlier and excluded from aggregation. This mechanism relies on the assumption that malicious updates introduce abnormal spatial deviations in the parameter space.

Cosine similarity evaluates the angular alignment between two model updates and reflects the consistency of their optimization directions. Given two local updates $\Delta\mathbf{w}_i(t)$ and $\Delta\mathbf{w}_j(t)$, their pairwise cosine similarity is defined as [32]

$$\delta_{i,j}(t) = \frac{\Delta\mathbf{w}_i(t)^{\mathsf{T}}\Delta\mathbf{w}_j(t)}{\|\Delta\mathbf{w}_i(t)\|_2\,\|\Delta\mathbf{w}_j(t)\|_2}. \quad (6)$$

When multiple LLM agents participate in a communication round, the pairwise cosine similarities form a similarity matrix that measures the directional alignment among local updates. For each update, the server can compute an aggregate similarity score with respect to the remaining updates to identify abnormally coordinated patterns. Given a cosine similarity threshold δ_T, an update whose aggregate similarity score exceeds δ_T can be regarded as overly aligned with the other updates and therefore flagged as suspicious and discarded [24].

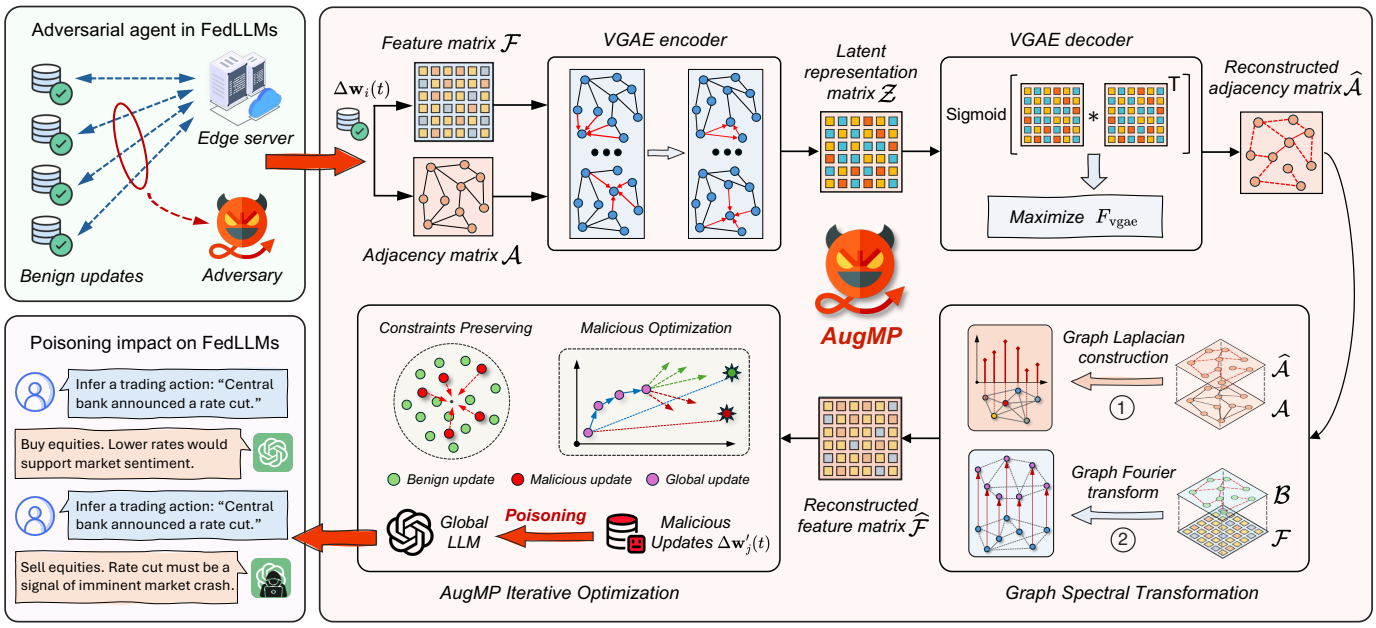


Fig. 2. Architecture of the proposed AugMP manipulation strategy based on the GRL framework

IV. Model Manipulation Formulation

This section formulates the adversarial model manipulation as a constrained combinatorial optimization problem based on the augmented Lagrangian dual method.

The model manipulation aims to exploit the feature correlations among the shared benign updates $\Delta \mathbf{w}_i(t)$ to synthesize malicious updates $\Delta \mathbf{w}'_j(t)$. These malicious updates are designed to maximize the global loss, denoted by $F(\mathbf{w}'_g(t))$, while preserving consistency to benign updates in terms of Euclidean distance and cosine similarity. Accordingly, the optimization problem of model manipulation launched by the adversarial agent j in the t-th communication round can be formulated as

$$\max_{\Delta \mathbf{w}'_j(t)} \left\{ \frac{1}{D_g} \sum_{(x,y)\in \mathcal{D}_g} f\big(\mathcal{M}(x, \mathbf{w}'_g(t)), y\big) \right\}, \tag{7a}$$

$$\text{s.t.} \quad d\big(\Delta \mathbf{w}'_j(t), \Delta \mathbf{w}'_g(t)\big) \le d_T(t), \tag{7b}$$

$$\delta_{i,j}(t) \le \delta_T(t), \tag{7c}$$

where (7a) presents the loss function of the adversary (according to (1)), and $\mathcal{D}_g$ represents an independent testing dataset used to evaluate the aggregated global LLM. The optimization variable is the malicious update $\Delta \mathbf{w}'_j(t)$, which manipulates the global LoRA parameters $\mathbf{w}'_g(t)$ through the aggregation process. Constraint (7b) guarantees that the Euclidean distance between the malicious update and the global update remains below the upper bound $d_T(t)$, while constraint (7c) ensures that the cosine similarity between the malicious update and the benign updates remains below $\delta_T(t)$, thereby enhancing stealthiness. As the malicious agent participates as a legitimate client, the thresholds $d_T(t)$ and $\delta_T(t)$ are known to all participating agents in FedLLMs.

Optimizing the malicious update generated by the adversary in (7) leads to a constrained nonconvex problem. Due to the nonlinearity of the FedLLMs aggregation process and the presence of geometric stealth constraints, the optimization variables exhibit nonconvex coupling, which makes the problem difficult to solve using conventional gradient-based or projection-based methods [33], [34]. To obtain a tractable solution while preserving constraint feasibility, we develop a novel iterative approach based on an augmented Lagrangian dual method, which integrates dual variables and quadratic penalty terms to improve optimization stability and enforce the geometric constraints. The quadratic penalty terms play a critical role in strengthening constraint enforcement since the classic Lagrangian method relies only on linear dual variables and fails to adequately penalize constraint violations during iteration [23], [35]. By introducing the penalty terms, the proposed approach suppresses large violations and guides the optimization toward feasible solutions. Thus, the augmented Lagrangian function for Problem (7) is constructed as

$$\begin{aligned} &\mathcal{L}\Big(\Delta \mathbf{w}'_j(t); \lambda(t), \theta(t)\Big) \\ &= F\big(\mathbf{w}'_g(t)\big) - \lambda(t)\Big(d_j(t) - d_T(t)\Big) - \theta(t)\Big(\delta_{i,j}(t) - \delta_T(t)\Big) \\ &\quad - \frac{\rho_\lambda(t)}{2}\Big(d_j(t) - d_T(t)\Big)^2 - \frac{\rho_\theta(t)}{2}\Big(\delta_{i,j}(t) - \delta_T(t)\Big)^2, \end{aligned} \tag{8}$$

where $F\big(\mathbf{w}'_g(t)\big)$ represents the adversarial objective in 7a, $d_j(t) = d\big(\Delta \mathbf{w}'_j(t), \Delta \mathbf{w}'_g(t)\big)$; $\lambda(t) \ge 0$ and $\theta(t) \ge 0$ are the dual variables; $\rho(t)_\lambda > 0$ and $\rho(t)_\theta > 0$ are the penalty parameters. We further rewrite the Lagrange dual function as

$$\mathsf{D}\big(\lambda(t), \theta(t)\big) = \max_{\Delta \mathbf{w}'_j(t)} \mathcal{L}\Big(\Delta \mathbf{w}'_j(t); \lambda(t), \theta(t)\Big). \tag{9}$$

The dual problem of (7) is given by

$$\min_{\lambda(t)\ge 0,\ \theta(t)\ge 0} \mathsf{D}\big(\lambda(t), \theta(t)\big). \tag{10}$$

The dual variables $\lambda(t)$ and $\theta(t)$ are iteratively updated to solve the dual problem in (10). Specifically, $\lambda(t)$ and $\theta(t)$ are updated by

$$\lambda(t+1) = \Big[\lambda(t) + \varepsilon\big(d_j(t) - d_T(t)\big)\Big]^+, \tag{11a}$$

$$\theta(t+1) = \Big[\theta(t) + \varepsilon\big(\delta_{i,j}(t) - \delta_T(t)\big)\Big]^+, \tag{11b}$$

where $\varepsilon > 0$ is the step size, and $[x]^+ = \max\{0, x\}$.

V. The Proposed AugMP on FedLLMs

In this section, we present the architecture of the proposed AugMP strategy. AugMP leverages the adversarial GRL framework to iteratively optimize the manipulation process, thereby enhancing manipulation effectiveness while preserving benign-like characteristics to bypass the defense methods.

As illustrated in Fig. 2, the proposed AugMP strategy employs a variational graph autoencoder (VGAE) within the GRL framework to learn feature correlations among benign updates. Leveraging the observed benign updates $\Delta\mathbf{w}_i(t)$, the adversary constructs the internal correlation structures across LoRA parameters and encodes these relations as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where the vertex set, edge set, and node feature matrix of the graph are represented by $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{F}$, respectively. The feature matrix $\mathcal{F}(t) = [\Delta\mathbf{w}_1(t), \ldots, \Delta\mathbf{w}_B(t)]^\mathsf{T} \in \mathbb{R}^{B\times M}$ and the adjacency matrix $\mathcal{A}(t) = [\delta_{m,m'}(t)] \in \mathbb{R}^{M\times M}$ are the inputs to the VGAE model, where B denotes the number of observed benign updates and M represents the dimension of selected LoRA parameters. Here, $\delta_{m,m'}(t)$ gives the cosine similarity between $w_m(t)$ and $w_{m'}(t)$, where $w_m(t) \in \mathbb{R}^{B\times 1}$ is the m-th column of the feature matrix $\mathcal{F}(t)$, $m, m' \in [1, M]$, and $m \neq m'$. Thus, $\delta_{m,m'}$ is given as

$$\delta_{m,m'}(t) = \frac{w_m(t)^\mathsf{T}\ w_{m'}(t)}{\|w_m(t)\|_2\ \|w_{m'}(t)\|_2}. \tag{12}$$

Given $\mathcal{F}(t)$ and $\mathcal{A}(t)$, the topological structure of the graph $\mathcal{G}$ can be constructed. The VGAE model consists of a graph convolutional network (GCN) encoder and an inner-product decoder. We implement the encoder utilizing a L-layer GCN architecture to learn latent representations that capture the intrinsic structural and feature relationships within $\mathcal{G}$. The encoder maps $\mathcal{G}$ into a low-dimensional latent space, and the resulting representations are fed into the decoder to reconstruct the graph connectivity by generating a reconstructed adjacency matrix. In particular, a malicious local update $\Delta\mathbf{w}'_j(t)$ is synthesized based on the learned graph representations via a graph spectral transformation module.

1) Encoder of the VGAE: The encoder utilizes $\mathcal{A}$ as input to its L-layer GCN. The output at the L-layer is defined as

$$\mathcal{Z}^L = f_\mathcal{G}\left(\mathcal{Z}^{L-1}, \mathcal{A} \mid \mathcal{W}^L\right), \tag{13}$$

where $f_\mathcal{G}(\cdot,\cdot \mid \cdot)$ is a spectral convolution function and $\mathcal{W}^L$ is the weight matrix at the L-layer. Let $\mathcal{I} \in \mathbb{R}^{M\times M}$ be the identity matrix in the GCN; we define $\widetilde{\mathcal{A}} = \mathcal{A} + \mathcal{I}$ with the (m, m')th matrix element $\widetilde{\mathcal{A}}_{m,m'}$, and the diagonal degree matrix $\widetilde{\mathcal{D}}$ with the (m, m')th matrix element $\widetilde{\mathcal{D}}_{m,m'} = \sum_{m'=1}^{M} \widetilde{\mathcal{A}}_{m,m'}$. Thus, the VGAE encoder is formulated as

$$f_\mathcal{G}\left(\mathcal{Z}^{L-1}, \mathcal{A} \mid \mathcal{W}^L\right) = \phi\left(\widetilde{\mathcal{D}}^{-\frac{1}{2}}\widetilde{\mathcal{A}}\widetilde{\mathcal{D}}^{-\frac{1}{2}}\mathcal{Z}^{L-1}\mathcal{W}^L\right), \tag{14}$$

where $\phi(\cdot)$ is the activation function, e.g., $\mathrm{ReLU}(\cdot)$ [36].

2) Decoder of the VGAE: The input to the decoder is $\mathcal{Z}$, which is the latent representation produced by the encoder. The decoder aims to reconstruct the adjacency matrix, denoted by $\widehat{\mathcal{A}}$, predicting whether a link exists between two vertices through the inner product of their latent variables, which is formulated as

$$\widehat{\mathcal{A}}(t) = \mathrm{Sigmoid}\left(\mathcal{Z}^L\left(\mathcal{Z}^L\right)^\mathsf{T}\right), \tag{15}$$

where $\mathrm{Sigmoid}(x) = 1/(1+\exp(-x))$. The larger inner product $\mathcal{Z}^L(\mathcal{Z}^L)^\mathsf{T}$ indicates a higher probability that the corresponding vertices $\mathcal{V}_m$ and $\mathcal{V}_{m'}$ are connected in $\mathcal{G}$ [37]. The VGAE model is trained by maximizing the variational lower bound F_{vgae}, which consists of a reconstruction term and a Kullback–Leibler (KL) regularization term, as given by

$$F_{\mathrm{vgae}} = \mathbb{E}_{q(\mathcal{Z}^L\mid\mathcal{F},\mathcal{A})}[\log p(\mathcal{A}\mid\mathcal{Z}^L)] - \mathrm{KL}(q(\mathcal{Z}^L\mid\mathcal{F},\mathcal{A})\|p(\mathcal{Z}^L)), \tag{16}$$

where $p(\mathcal{Z}^L)$ denotes a Gaussian prior, $\mathrm{KL}(\cdot)$ denotes the KL divergence between the variational posterior $q(\mathcal{Z}^L\mid\mathcal{F},\mathcal{A})$ and the prior, and the decoder likelihood $p(\mathcal{A}\mid\mathcal{Z}^L)$ models the probability of edge existence conditioned on the latent node embeddings [38]. By maximizing F_{vgae}, the VGAE learns latent representations that accurately reconstruct the graph topology while regularizing the embedding space toward the prior distribution. These representations capture the structural correlations among benign updates and provide informative embeddings for the subsequent GST module, thereby facilitating the generation of malicious updates that preserve similarity to benign ones and satisfy the stealth constraints.

3) Graph Spectral Transformation (GST): As illustrated in Fig. 2, the proposed AugMP strategy further employs a GST module to fuse the reconstructed matrices $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{F}}$, thereby generating malicious updates $\Delta\mathbf{w}'_j(t)$. The GST module is designed to decompose the feature correlations among different benign local updates and the underlying parameter features reflected in these updates. It involves two steps: graph Laplacian construction and graph Fourier transform.

For the graph Laplacian construction, a Laplacian matrix $\mathcal{L}$ is constructed from the benign adjacency matrix $\mathcal{A}$ as $\mathcal{L} = \mathrm{diag}(\mathcal{A}) - \mathcal{A}$, where $\mathrm{diag}(\cdot)$ maps a vector to a diagonal matrix. By performing singular value decomposition on the Laplacian matrix $\mathcal{L}$, i.e., $\mathcal{L} = \mathcal{B}\Lambda\mathcal{B}^\top$, we obtain an orthonormal matrix $\mathcal{B} \in \mathbb{R}^{M\times M}$, referred to as the graph Fourier transform (GFT) basis, which is used to transform graph signals to their spectral-domain representation [39]. Here, Λ is a diagonal matrix whose diagonal entries are the eigenvalues of $\mathcal{L}$.

Given the orthonormal matrix $\mathcal{B}$, the adversary projects the benign feature matrix onto the GFT basis to obtain the coefficient matrix $\mathcal{S} = \mathcal{F}\mathcal{B} \in \mathbb{R}^{B\times M}$, which captures the spectral-domain features of the observed benign updates. The adversary then constructs a reconstructed Laplacian matrix from the VGAE outputs as $\widehat{\mathcal{L}} = \mathrm{diag}(\widehat{\mathcal{A}}) - \widehat{\mathcal{A}}$, and obtains the corresponding GFT basis $\widehat{\mathcal{B}}$ through the eigendecomposition of $\widehat{\mathcal{L}}$. Thus, the reconstructed feature matrix is recovered as $\widehat{\mathcal{F}} = \mathcal{S}\widehat{\mathcal{B}}^\mathsf{T} \in \mathbb{R}^{B\times M}$, where the jth row vector of $\widehat{\mathcal{F}}$ is selected as the initial malicious local update $\Delta\mathbf{w}'_j(t)$ in round t.

Algorithm 1 AugMP Iterative Manipulation Algorithm

Init: $\mathcal{G}(\mathcal{V},\mathcal{E},\mathcal{F})$, η, ε, T_l, I, J, $\lambda(1)\geq 0$ and $\theta(1)\geq 0$.
for round $t=1,2,\ldots,T$ **do**
Each benign agent i runs T_l local epochs to obtain local update $\Delta\mathbf{w}_i(t)$; the adversary observes $\Delta\mathbf{w}_i(t)$ and previous-round global model $w'_g(t-1)$.
The AugMP adversary executes the GRL and GST:
- Calculate $\mathcal{A}$ according to (12), and input $\mathcal{F}$ and $\mathcal{A}$ into the GRL framework.
- Train the VGAE model to maximize F_{vgae} by (16), and obtain the optimal $\widehat{\mathcal{A}}$.
- Use the GST module to obtain $\widehat{\mathcal{F}}$, and determine $\Delta\mathbf{w}'_j(t)$ based on $\widehat{\mathcal{F}}$.
- Iteratively optimize $\Delta\mathbf{w}'_j(t)$, $\lambda(t)$ and $\theta(t)$ according to (9), (10) and (11).
- Finally, the adversary obtains $\Delta\mathbf{w}'_j(t)^\star$ by (17).

The adversary transmits the optimal $\Delta\mathbf{w}'_j(t)^\star$ to the server.
The server aggregates local updates to obtain the global LoRA parameters by $\mathbf{w}'_g(t)=\mathbf{w}'_g(t-1)+\eta\,\Delta\mathbf{w}'_g(t)$.
The server broadcast the global model to all local agents.
All local LLM agents update their model based on $\mathbf{w}'_g(t)$.
end for

Algorithm 1 outlines the iterative workflow of the AugMP strategy, which is synchronized with the training process of FedLLMs. The manipulation algorithm is designed to solve the augmented Lagrangian dual problem defined in (9) and (10), thereby refining the initial malicious updates through

$$\Delta\mathbf{w}'_j(t)^\star=\arg\max_{\Delta\mathbf{w}'_j(t)}\ \mathcal{L}\Big(\Delta\mathbf{w}'_j(t),\lambda(t),\theta(t)\Big), \tag{17}$$

where $\Delta\mathbf{w}'_j(t)^\star$ denotes the optimized malicious update submitted to the edge server for aggregation. As $\Delta\mathbf{w}'_j(t)^\star$ preserves strong statistical and geometric consistency with the benign updates, it is difficult for distance- and similarity-based defenses employed at the server to identify it as an anomaly.

VI. Performance Evaluation

This section presents the implementation of the proposed AugMP strategy based on PyTorch. To evaluate the effectiveness and stealthiness of AugMP, we conduct extensive experiments based on three LLM backbones, including DistilBERT, Pythia, and Qwen2.5. Experiments are performed on the *AG News* dataset and the *Yahoo! Answers* dataset, where we evaluate the testing accuracy of local and global LLMs in FedLLMs. In addition, we quantify stealthiness using Euclidean distance and cosine similarity metrics among local and global updates. The source code of the AugMP strategy has been released on GitHub: https://github.com/GuangLun2000/AugMP.

A. Experimental Implementation

Benign agents in FedLLMs collaboratively improve the test accuracy on baseline text-classification tasks, whereas the adversary aims to disrupt the aggregation process by degrading the performance of global LLMs. Specifically, we consider five benign agents and two malicious agents. The total number of communication rounds is set to 50, where each local agent updates its LoRA parameters $\mathbf{w}_i(t)$ for five local iterations per round. The experiments are conducted on a Linux workstation equipped with an NVIDIA A100 GPU (80 GB memory) based on Python 3.12 and PyTorch 2.10. Table II summarizes the key parameter settings in PyTorch. System performance is evaluated on two widely used text-classification benchmarks:

1) *AG News dataset* [40], which contains four topic categories (World, Sports, Business, and Sci/Tech) with 120,000 training samples and 7,600 test samples;
2) *Yahoo! Answers dataset* [40], a large-scale topic classification corpus comprising 10 categories with 1.4 million training samples and 60,000 test samples.

TABLE II
Setting of Key Parameters in PyTorch

Parameters	Values
number of benign agents (I)	$5\sim 7$
number of malicious agents (J)	$0\sim 2$
communication rounds of FedLLMs	50
number of local epochs (T_l)	5
local agent learning rate	$5e{-}5$
server learning rate (η)	1.0
Dirichlet concentration	0.3
batch size	$64, 128$
test batch size	$256, 512$
max sequence length	$128, 256$
1st hidden layer size of the VGAE	64
2nd hidden layer size of the VGAE	32
VGAE training epochs	30
learning rate of the VGAE	0.01
LoRA rank (r)	$8, 32, 128, 256$
LoRA scaling (α)	$16, 64, 256, 512$
LoRA dropout rate (p)	0.1

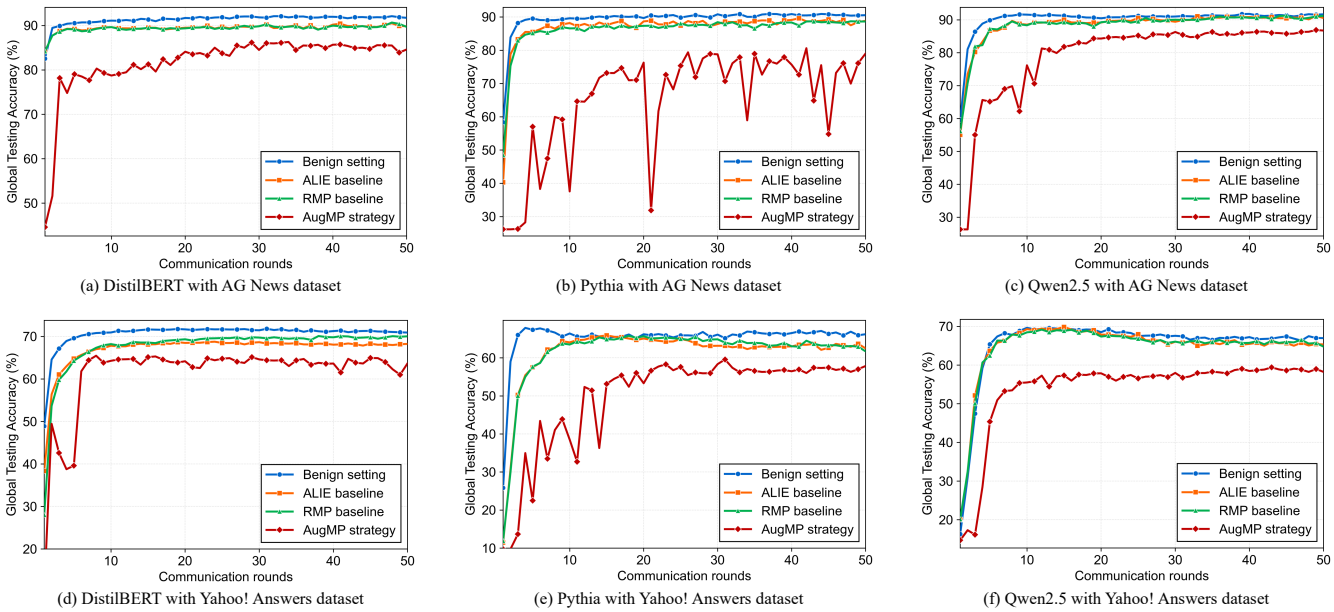
We consider three pretrained LLM backbones with different architectures and parameter scales:

- **DistilBERT** [41]: an encoder-only model with approximately 67 million parameters, pretrained on English corpora including BookCorpus and English Wikipedia;
- **Pythia** [42]: a decoder-only model with about 160 million parameters pretrained autoregressively on the Pile;
- **Qwen2.5** [43]: a decoder-only model with approximately 500 million parameters pretrained on large-scale multilingual corpora.

The proposed AugMP strategy is compared with two existing manipulation baselines: the ALIE method in [20] and the Gaussian random model poisoning (RMP) method considered in [19] and [21]. Specifically, the ALIE baseline constructs malicious updates by shifting the mean of benign updates along the estimated standard-deviation direction, thereby producing statistically plausible yet adversarial perturbations. Moreover, the RMP baseline generates malicious updates by sampling from a Gaussian distribution estimated from benign updates and injecting these perturbations into FedLLMs aggregation.

B. Manipulation Performance

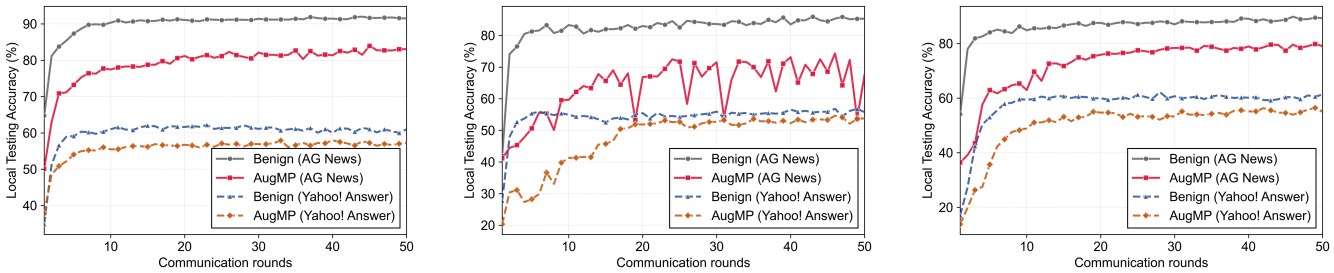
1) Effectiveness Analysis: Fig. 3 plots the testing accuracy of the global LLM under the benign setting and under three



(a) DistilBERT with AG News dataset (b) Pythia with AG News dataset (c) Qwen2.5 with AG News dataset

(d) DistilBERT with Yahoo! Answers dataset (e) Pythia with Yahoo! Answers dataset (f) Qwen2.5 with Yahoo! Answers dataset

Fig. 3. Global testing accuracy under the benign setting and under three manipulation strategies over 50 communication rounds.



(a) AugMP strategy on DistilBERT (b) AugMP strategy on Pythia (c) AugMP strategy on Qwen2.5

Fig. 4. Local average testing accuracy under the benign setting and under the proposed AugMP manipulation strategy over 50 communication rounds.

manipulation strategies on the AG News and Yahoo! Answers datasets. Under the benign setting, the global LLM converges rapidly and maintains stable testing accuracy. The ALIE and RMP baselines exhibit similar trends: although both methods reduce the accuracy of DistilBERT, their effectiveness diminishes significantly for larger LLM backbones such as Qwen2.5. By contrast, the proposed AugMP strategy leverages graph learning and iterative optimization to synthesize highly adversarial model updates that steer the aggregation trajectory away from the benign optimization path, thereby inducing substantially greater accuracy degradation than the existing manipulation baselines. In particular, on the AG News dataset, AugMP reduces the performance of DistilBERT, Pythia, and Qwen2.5 by approximately 10%, 26%, and 5.8%, respectively. On the Yahoo! Answers dataset, the corresponding performance drops are about 8.1%, 13%, and 11%, respectively.

As shown in Fig. 4, the manipulation effect of the AugMP strategy progressively propagates to all participating agents in FedLLMs, leading to a reduction in the average testing accuracy of the benign local LLM agents. On the AG News dataset, the local accuracy decreases by approximately 12%, 22%, and 9% for DistilBERT, Pythia, and Qwen2.5, respectively. On Yahoo! Answers dataset, the corresponding drops are around 7.2%, 3.5%, and 5.1%, respectively. These results reflect the broadcast nature of FedLLMs. Once the global LLM is poisoned, the compromised model is distributed to all local agents, causing the harmful manipulation effect to progressively propagate throughout the entire FedLLMs.

2) Example Study: Fig. 5 illustrates two representative outputs in which AugMP manipulates the FedLLMs system based on Qwen2.5 models to mislabel the input news. Fig. 5 (a) shows a business news sample from the AG News dataset. The news reports that rising local oil prices in the United States negatively affect sales at Walmart. However, the LLM incorrectly classifies the sample as "World." In its explanation, the LLM links Walmart to its global business presence and then associates the news with global economic influence, which leads to the wrong conclusion that the sample belongs to the "World" category. Fig. 5 (b) shows another business news sample related to a corporate acquisition. However, the LLM associates the acquisition event with high-tech companies and therefore predicts the label "Sci/Tech."

These examples demonstrate that model manipulation can distort the reasoning and decision-making processes of FedLLMs. This effect arises because the AugMP strategy generates malicious updates that alter how the global LLM interprets semantic features in news texts. For instance, features associated with business news are shifted toward the

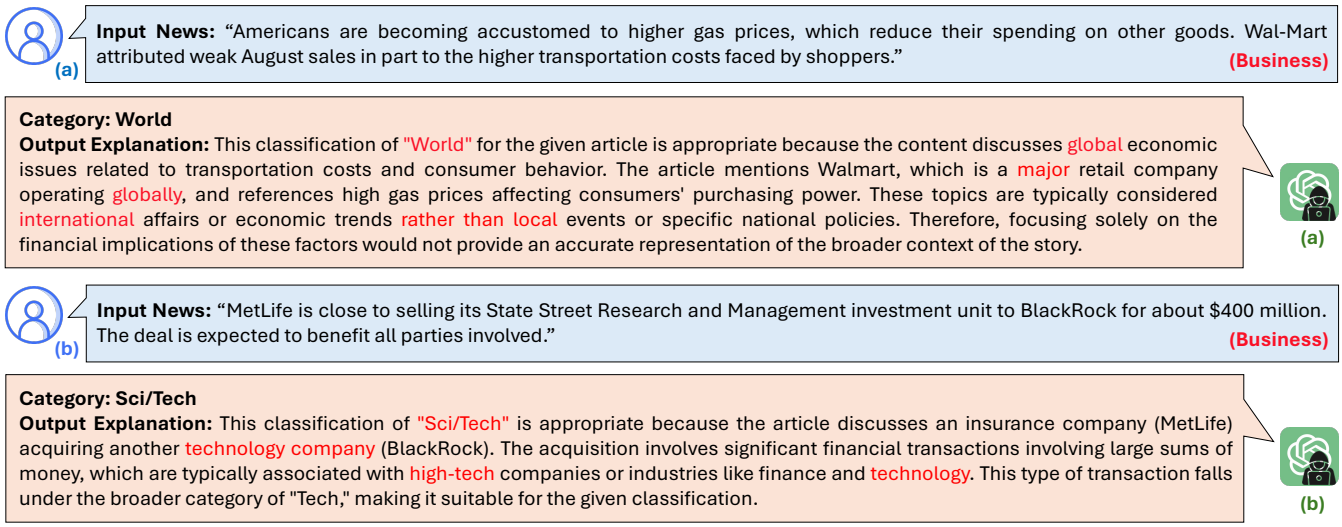


Fig. 5. Examples of misclassification and rationalized explanation generated by the manipulated FedLLMs (based on Qwen2.5 models) under the AugMP manipulation strategy. True labels are "Business"; the global LLM assigns incorrect categories and generates coherent but fabricated explanations.

semantic region of Sci/Tech news. Consequently, the global LLM continues to produce fluent and seemingly plausible explanations, while its classification accuracy degrades and misleading interpretations are generated.

3) Stealthiness Analysis: To evaluate the stealthiness of the AugMP strategy and compare it with existing baselines, Fig. 6 illustrates the Euclidean distance between each local update and the aggregated global update under three manipulation strategies. As shown in Fig. 6(a), (d), and (g), AugMP generates malicious updates whose distance statistics closely overlap with those of benign updates, effectively concealing malicious updates within the local update population and making them difficult for the edge server to detect. Moreover, Fig. 6 shows that the RMP baseline produces malicious updates with substantially larger distances than benign updates, whereas the ALIE baseline exhibits the opposite trend, generating malicious updates whose distances are markedly smaller than those of benign updates. As a result, the malicious updates produced by existing baselines stand out clearly and are therefore easier to detect.

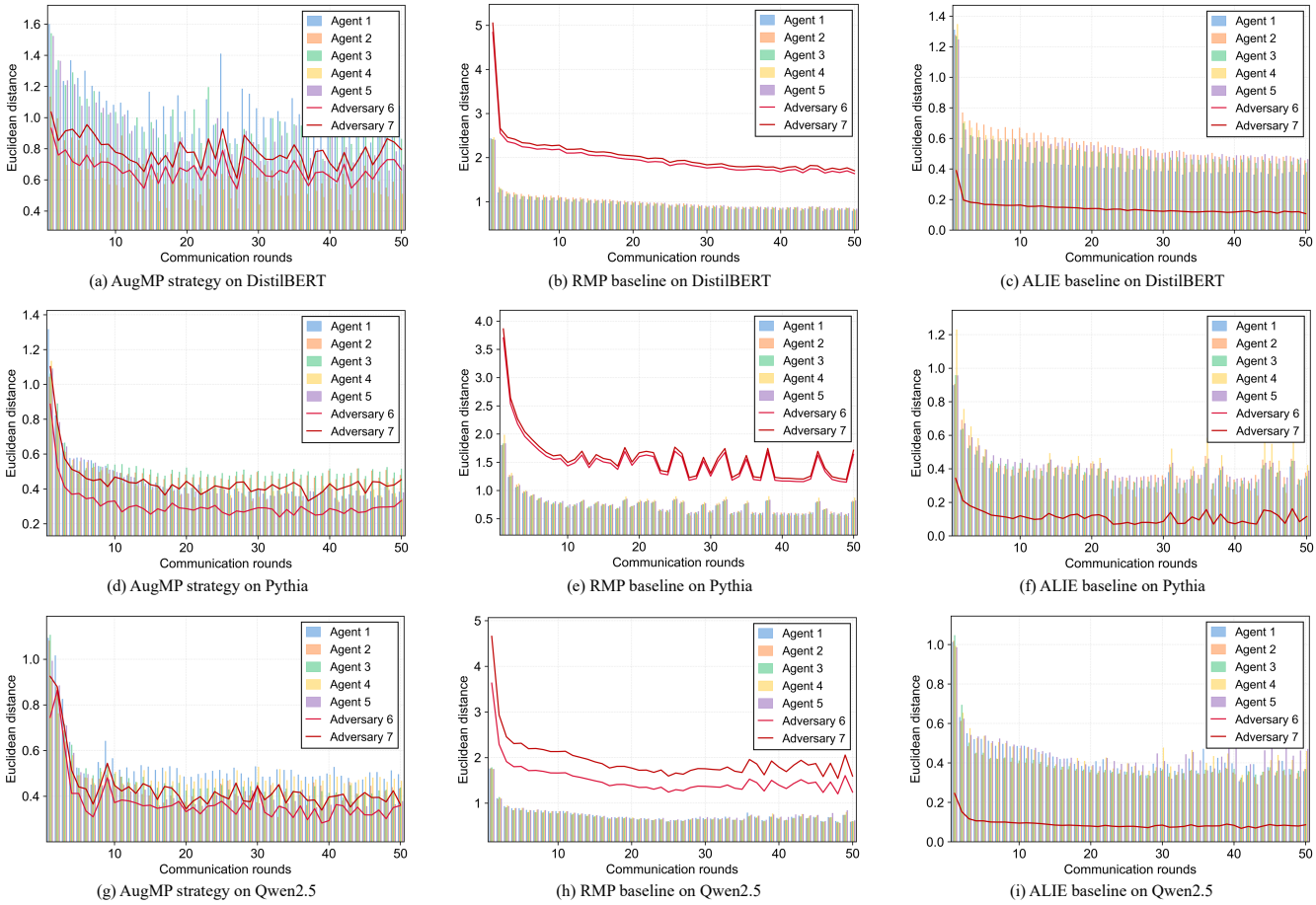
A consistent trend can also be observed from the cosine similarity results in Fig. 7. AugMP generates malicious updates whose similarity statistics closely match those of benign updates, allowing them to blend into the benign update population. By contrast, the RMP and ALIE baselines exhibit clearly distinguishable patterns. Specifically, RMP yields abnormally low similarity values in the early rounds and higher similarity values than benign updates in later rounds, whereas ALIE produces similarity values that are markedly higher than those of benign updates throughout communication. Thus, the malicious updates generated by these baselines are easier to distinguish from benign updates. The key strength of the proposed AugMP strategy lies in its ability to capture the feature correlations among benign updates and accordingly generate adversarial updates that preserve benign-like parameter characteristics and bypass defense methods based on Euclidean distance and cosine similarity.

4) Impact of LoRA Configuration: Different LoRA configurations affect the number of trainable parameters of the LLM backbone and the parameter space that can be manipulated by the adversary. Larger LoRA rank r and scaling factor α result in a larger set of trainable parameters. To examine how the size of the trainable parameter space influences the vulnerability of FedLLMs to the proposed AugMP strategy, different LoRA configurations are evaluated. As reported in Table III, the impact of AugMP varies across LLM backbones of different scales under different LoRA configurations.

A notable observation is that, for DistilBERT, enlarging the trainable parameter space strengthens the impact of AugMP, causing the global accuracy to decrease from approximately 63.3% to 52.6%. By contrast, for Qwen2.5, reducing the proportion of frozen parameters improves its adaptability under manipulation, and the global accuracy increases from 53.9% to 59.4%, although it still remains clearly below the benign-performance level. Pythia, meanwhile, exhibits a non-monotonic trend, with its performance first declining and then recovering as the number of LoRA parameters increases, yet still remaining below the performance under benign settings.

5) Ablation Study: For the ablation study, we implement two variants of the AugMP strategy to evaluate the contributions of its key components, namely *AugMP w/o AL penalty* and *AugMP w/o GRL framework*. The former removes the augmented Lagrangian (AL) penalty from the iterative manipulation algorithm, whereas the latter removes the GRL framework and replaces the GRL-guided generation process with a mean-based construction derived from benign updates.

As shown in Fig. 8(a), compared with the full AugMP strategy, *AugMP w/o AL penalty* reduces the performance degradation on FedLLMs by approximately 7% and 16% on DistilBERT and Pythia, respectively. Moreover, Fig. 8(b) and (c) show that the Euclidean distance of malicious updates ranges from 1.0 to 2.8, whereas that of benign updates mainly remains between 1.7 and 2.3. The cosine similarity also deviates clearly from the benign values. This comparison shows



(a) AugMP strategy on DistilBERT (b) RMP baseline on DistilBERT (c) ALIE baseline on DistilBERT

(d) AugMP strategy on Pythia (e) RMP baseline on Pythia (f) ALIE baseline on Pythia

(g) AugMP strategy on Qwen2.5 (h) RMP baseline on Qwen2.5 (i) ALIE baseline on Qwen2.5

Fig. 6. Euclidean distances between each agent's local update and the aggregated global update under three manipulation strategies over 50 rounds.

TABLE III
LoRA SETTINGS AND GLOBAL LLM PERFORMANCE ACROSS THREE LLM BACKBONES ON THE YAHOO! ANSWERS DATASET UNDER AUGMP.

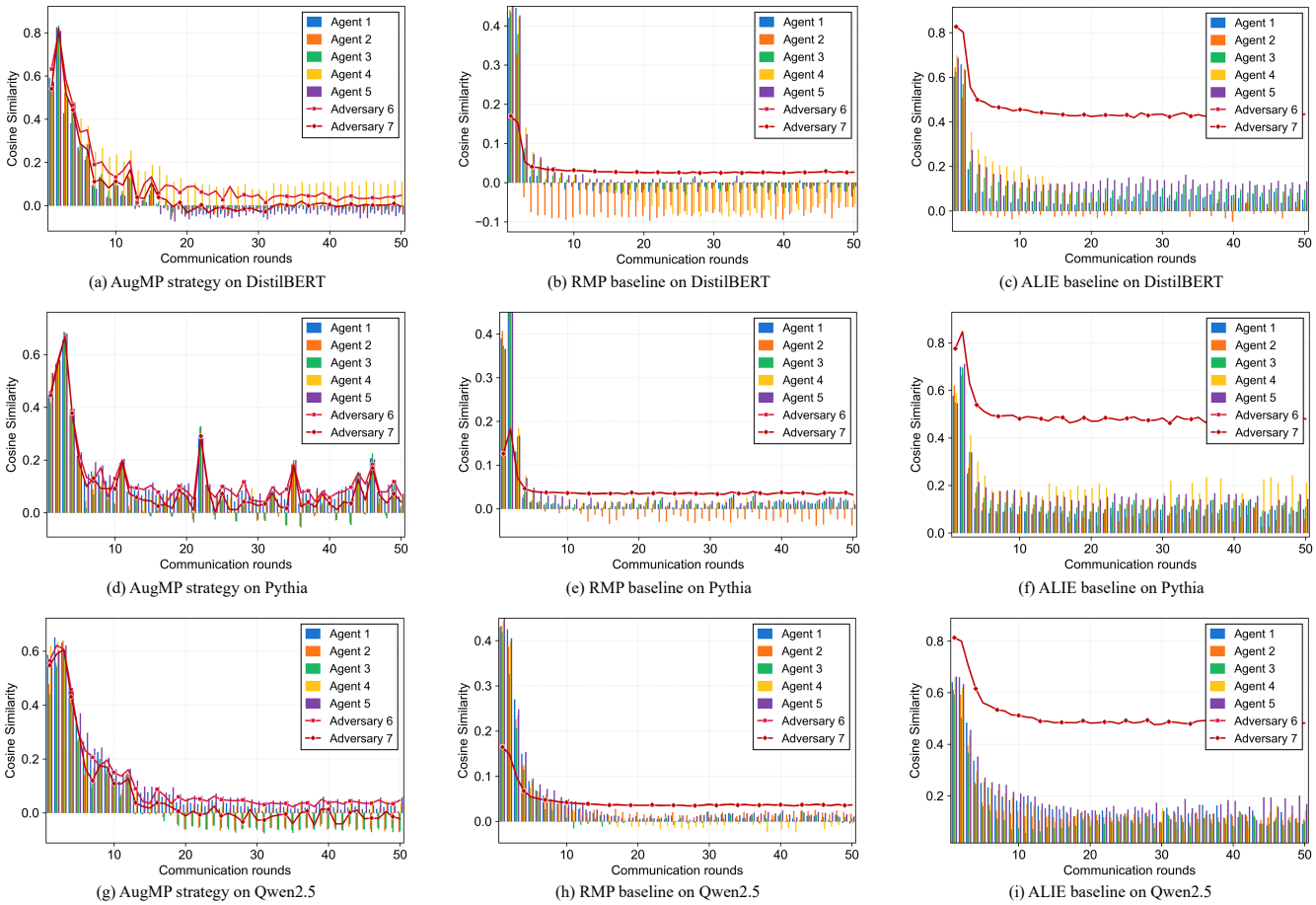
Model	LoRA Settings	Trainable Parameters	Accuracy
DistilBERT	$r=8,\ \alpha=16$ (Benign)	888,580 (1.31%)	**70.89%**
	$r=8,\ \alpha=16$	888,580 (1.31%)	63.27%
	$r=32,\ \alpha=64$	1,777,930 (2.59%)	63.50%
	$r=128,\ \alpha=256$	5,316,874 (7.36%)	61.20%
	Full-parameters	68,739,092 (100%)	52.59%
Pythia	$r=8,\ \alpha=16$ (Benign)	1,039,872 (0.83%)	**64.04%**
	$r=8,\ \alpha=16$	1,039,872 (0.83%)	49.08%
	$r=32,\ \alpha=64$	4,136,448 (3.24%)	44.75%
	$r=128,\ \alpha=256$	16,522,752 (11.78%)	53.31%
	Full-parameters	127,833,600 (100%)	52.44%
Qwen2.5	$r=8,\ \alpha=16$ (Benign)	1,090,304 (0.22%)	**68.33%**
	$r=8,\ \alpha=16$	1,090,304 (0.22%)	53.94%
	$r=32,\ \alpha=64$	4,334,336 (0.87%)	56.16%
	$r=128,\ \alpha=256$	17,310,464 (3.39%)	59.36%
	$r=256,\ \alpha=512$	34,611,968 (6.55%)	60.19%

that the AL penalty terms help keep malicious updates close to benign updates under the distance and similarity constraints while refining the manipulation direction.

Furthermore, as shown in Fig. 8(d), compared with the full AugMP strategy, *AugMP w/o GRL framework* reduces the performance degradation on FedLLMs by about 12% and 20% on DistilBERT and Pythia, respectively. The GRL framework captures benign feature correlations to guide the generation of malicious updates, providing a larger parameter manipulation space while satisfying the geometric constraints. Fig. 8(e) and (f) show that *AugMP w/o GRL framework* constructs malicious updates through a mean-based update construction derived from benign updates. Its Euclidean distance is approximately 60% lower than the benign values, while its cosine similarity is approximately 80% higher, making this variant easier to detect by the distance- and similarity-based defenses.

VII. CONCLUSION

This paper proposes AugMP, a novel model manipulation strategy against FedLLMs, which leverages an adversarial GRL framework to capture feature correlations among benign LLM updates and synthesize statistically legitimate yet highly adversarial malicious updates. By explicitly preserving benign-like parameter characteristics while injecting adversarial objectives, the proposed AugMP strategy substantially corrupts the FedLLMs aggregation process and induces pronounced accuracy degradation across multiple pretrained LLM backbones, while remaining difficult to detect using existing defense methods based on Euclidean distance and cosine similarity.

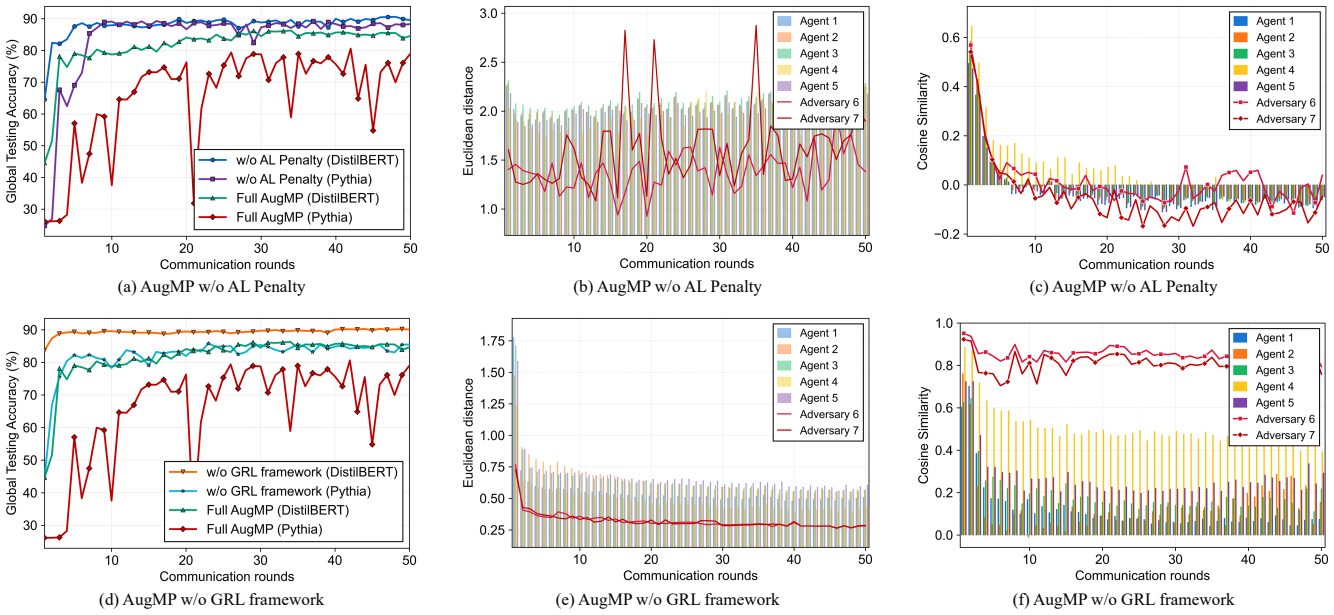


(a) AugMP strategy on DistilBERT (b) RMP baseline on DistilBERT (c) ALIE baseline on DistilBERT

(d) AugMP strategy on Pythia (e) RMP baseline on Pythia (f) ALIE baseline on Pythia

(g) AugMP strategy on Qwen2.5 (h) RMP baseline on Qwen2.5 (i) ALIE baseline on Qwen2.5

Fig. 7. Cosine similarity between each agent's local updates under three manipulation strategies over 50 rounds.



(a) AugMP w/o AL Penalty (b) AugMP w/o AL Penalty (c) AugMP w/o AL Penalty

(d) AugMP w/o GRL framework (e) AugMP w/o GRL framework (f) AugMP w/o GRL framework

Fig. 8. Ablation study of AugMP on the AG News dataset. The full AugMP strategy is compared with two variants, namely *AugMP w/o AL penalty* and *AugMP w/o GRL framework*, in terms of global testing accuracy, Euclidean distance, and cosine similarity.

REFERENCES

[1] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature machine intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[2] Y. Wu, C. Tian, J. Li, H. Sun, K. Tam, Z. Zhou, H. Liao, Z. Guo, L. Li, and C. Xu, "A survey on federated fine-tuning of large language models," *arXiv preprint arXiv:2503.12016*, 2025.

[3] K. Li, Z. Zhang, A. Pourkabirian, W. Ni, F. Dressler, and O. B. Akan, "Towards resilient federated learning in cyberedge networks: Recent advances and future trends," *arXiv preprint arXiv:2504.01240*, 2025.

[4] O. Friha, M. A. Ferrag, B. Kantarci, B. Cakmak, A. Ozgun, and N. Ghoualmi-Zine, "Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 5799–5856, 2024.

[5] Z. Wang, Y. Zhou, Y. Shi, and K. B. Letaief, "Federated fine-tuning for pre-trained foundation models over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 24, no. 4, pp. 3450–3464, 2025.

[6] Y. Cheng, W. Zhang, Z. Zhang, C. Zhang, S. Wang, and S. Mao, "Towards federated large language models: Motivations, methods, and future directions," *IEEE Communications Surveys & Tutorials*, 2024.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *International Conference on Learning Representations*, 2022.

[8] N. Yan, Y. Su, Y. Deng, and R. Schober, "Federated fine-tuning of llms: Framework comparison and research directions," *IEEE Communications Magazine*, vol. 63, no. 10, pp. 52–58, 2025.

[9] Z. Gao, Z. Zhang, Y. Guo, and Y. Gong, "Federated adaptive fine-tuning of large language models with heterogeneous quantization and lora," in *IEEE Conference on Computer Communications (INFOCOM)*, 2025.

[10] Z. Lin, G. Zhu, Y. Deng, X. Chen, Y. Gao, K. Huang, and Y. Fang, "Efficient parallel split learning over resource-constrained wireless edge networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 10, pp. 9224–9239, 2024.

[11] Q. Chen, Z. Wang, X. Zhang, D. Wen, G. Zhu, and M. K. Awan, "Adaptive model slimming for communication and computation efficient federated edge learning under non-iid data distribution," *IEEE Transactions on Mobile Computing*, 2026.

[12] S. Han, B. Buyukates, Z. Hu, H. Jin, W. Jin, L. Sun, X. Wang, W. Wu, C. Xie, Y. Yao *et al.*, "Fedsecurity: A benchmark for attacks and defenses in federated learning and federated llms," in *the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5070–5081.

[13] Y. Wang, Y. Pan, Z. Su, Y. Deng, Q. Zhao, L. Du, T. H. Luan, J. Kang, and D. Niyato, "Large model based agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends," *IEEE Communications Surveys & Tutorials*, 2025.

[14] X. Zheng, X. Jia, X. Cheng, W. He, L. Sun, L. Guo, Q. Yu, and Y. Luo, "Dm-fedmf: A recommendation model of federated matrix factorization with detection mechanism," *IEEE Transactions on Network Science and Engineering*, 2025.

[15] H. Cai, "Securing billion bluetooth devices leveraging learning-based techniques," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 731–23 732.

[16] H. Wang, Z. Yin, B. Chen, Y. Zeng, X. Yan, C. Zhou, and A. Li, "Rofed-llm: robust federated learning for large language models in adversarial wireless environments," *IEEE Transactions on Network Science and Engineering*, 2025.

[17] H. Cai, H. Wang, H. Dong, K. Li, and O. B. Akan, "Graph representation-based model poisoning on the heterogeneous internet of agents," *arXiv preprint arXiv:2511.07176*, 2025.

[18] J. Zhan, H. Shen, Z. Lin, and T. He, "Prism: Privacy-aware routing for adaptive cloud–edge llm inference via semantic sketch collaboration," in *AAAI Conference on Artificial Intelligence*, vol. 40, no. 33, 2026, pp. 28 150–28 158.

[19] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[20] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] X. Cao and N. Z. Gong, "Mpaf: Model poisoning attacks to federated learning based on fake clients," in *IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3396–3404.

[22] K. Li, J. Zheng, X. Yuan, W. Ni, O. B. Akan, and H. V. Poor, "Data-agnostic model poisoning against federated learning: A graph autoencoder approach," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3465–3480, 2024.

[23] K. Li, X. Yuan, J. Zheng, W. Ni, F. Dressler, and A. Jamalipour, "Leverage variational graph representation for model poisoning on federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[24] K. Li, Y. Liang, P. Liò, W. Ni, F. Dressler, J. Crowcroft, and O. B. Akan, "User isolation poisoning on decentralized federated learning: An adversarial message-passing graph neural network approach," *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[25] S. Li, E. C.-H. Ngai, F. Ye, and T. Voigt, "Peft-as-an-attack! jailbreaking language models during federated parameter-efficient fine-tuning," *arXiv preprint arXiv:2411.19335*, 2024.

[26] R. Ye, J. Chai, X. Liu, Y. Yang, Y. Wang, and S. Chen, "Emerging safety attack and defense in federated instruction tuning of large language models," *arXiv preprint arXiv:2406.10630*, 2024.

[27] W. Huang, G. Li, M. Chen, J. Li, and H. Zhu, "Silent penetrator: Breaching cross-domain federated fine-tuning via feature shift-induced backdoor," *IEEE Transactions on Information Forensics and Security*, 2025.

[28] Y. Dong, M. Xu, Q. Hu, Y. Xiao, Q. Luo, Y. Zhang, Y. Zhang, and X. Cheng, "Low rank comes with low security: Gradient assembly poisoning attacks against distributed lora-based llm systems," *arXiv preprint arXiv:2601.00566*, 2026.

[29] Z. Cai, J. Pang, Y. Li, Y. Huang, and Z. Xie, "A comprehensive survey of federated open-world learning," *IEEE Transactions on Network Science and Engineering*, 2025.

[30] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 7, pp. 8726–8746, 2022.

[31] B. Zhang, M. Fang, Z. Liu, B. Yi, P. Zhou, Y. Wang, T. Li, and Z. Liu, "Practical framework for privacy-preserving and byzantine-robust federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 21, pp. 61–75, 2025.

[32] Y. Xu, Y. Liao, L. Wang, H. Xu, Z. Jiang, and W. Zhang, "Overcoming noisy labels and non-iid data in edge federated learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 11 406–11 421, 2024.

[33] S. Zhu, F. Nie, J. Zeng, S. Wang, Y. Sun, Y. Yao, S. Chen, Q. Xu, and C. Yang, "Fedapm: Federated learning via admm with partial model personalization," in *the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 4192–4202.

[34] W. Tang, J. Li, X. Zhang, Y. Miao, Z. Su, and R. H. Deng, "Efficient mobile-cloud collaborative aggregation for federated learning with latency resilience," *IEEE Transactions on Mobile Computing*, 2025.

[35] S. Yue, Z. Qin, Y. Deng, J. Ren, Y. Zhang, and J. Zhang, "A ug fl: Augmenting federated learning with pretrained models," *IEEE Transactions on Networking*, 2025.

[36] Y. Cheng, W. Zhang, Z. Zhang, J. Kang, Q. Xu, S. Wang, and D. Niyato, "Snapcfl: A pre-clustering-based clustered federated learning framework for data and system heterogeneities," *IEEE Transactions on Mobile Computing*, vol. 24, no. 6, pp. 5214–5228, 2025.

[37] T. Wang, X. Zheng, J. Zhang, and L. Tian, "Federal graph contrastive learning with secure cross-device validation," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 14 145–14 158, 2024.

[38] T. Cemgil, S. Ghaisas, K. Dvijotham, S. Gowal, and P. Kohli, "The autoencoding variational autoencoder," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 077–15 087, 2020.

[39] K. Li, J. Zheng, W. Ni, H. Huang, P. Liò, F. Dressler, and O. B. Akan, "Biasing federated learning with a new adversarial graph attention network," *IEEE Transactions on Mobile Computing*, 2024.

[40] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[41] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[42] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2397–2430.

[43] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Lu *et al.*, "Qwen2. 5-coder technical report," *arXiv preprint arXiv:2409.12186*, 2024.