Journal of Cloud Computing

RESEARCH Open Access

GuardChain: multi-stage trust framework for FL-empowered AIGC

Yike Yuan[1], Xinkui Zhao[2*], Shuyi Yu[2], Shengye Pang[3*] and Guobing Zou[3]

Abstract
ChatGPT's growth highlights the importance of Large Language Models (LLMs) in AI innovation, driving a surge in AI-generated content (AIGC). Federated Learning (FL), valued for privacy and cooperation, is drawing interest for using distributed data while ensuring privacy, critical for AI scalability. Yet, incorporating FL into AIGC is challenging, particularly in building trust across LLM training. This paper introduces GuardChain, a novel framework to fortify trustworthiness in every segment of FL-empowered AIGC training processes. We focus on three key designs: (1) Integrating the Nostr protocol into the FL paradigm to facilitate trustworthy cross-data validation. (2) The deployment of a dual-layered (on-chain and off-chain) mechanism for rigorous consistency checks and the identification of adversarial adapters. (3) The enhancement of model aggregation trustworthiness via a dynamic rotating election protocol for the selection of credible aggregators, augmented by a 'Sandwich' smart contract verification methodology. Our empirical studies reveal that GuardChain markedly surpasses existing baseline methods in mitigating diverse adversarial attacks and expediting trust verification processes. Comparative performance experiments reveal that GuardChain efficiently manages various workloads and hostile threats, with blockchain-based verification completed within 1–10 seconds, concurrently improving BLEU and ROUGE scores by 0.14 to 0.17 on the standard public dataset, thereby maintaining its reliability.

Keywords GuardChain, Federated learning, AI-Generated content, Nostr protocol, Dual-layer blockchain

Introduction

ChatGPT's surge in popularity has thrust Large Language Models (LLMs) into the limelight as the powerhouse behind generative AI, which produces a variety of AI-generated content (AIGC), including images, text, and audio [1]. This method harnesses cutting-edge AI technologies such as deep learning, Generative Adversarial Networks (GANs) [2], and Variational Autoencoders (VAEs) [3], blending them to create intricate content with rich detail. The advent of ChatGPT in 2022 stands out as a significant milestone in AIGC's evolution, showcasing its strength in enabling human-computer interaction and improving text generation quality [4]. This progress not only demonstrates AIGC's effectiveness but also indicates a transformative shift in digital content creation driven by AI advancements.

The success of machine learning tasks largely depends on combining data from various sources [5]. However, acquiring high-quality datasets poses significant challenges due to fragmented data ownership and widespread privacy concerns. Federated Learning (FL) emerges as a key solution in this scenario. Unlike traditional centralized training models, FL relies on the iterative processing

*Correspondence:
Xinkui Zhao
zhaoxinkui@zju.edu.cn
Shengye Pang
pangsy@shu.edu.cn
[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]School of Software Technology, Zhejiang University, Hangzhou, China
[3]School of Computer Engineering and Science, Shanghai University, Shanghai, China

of decentralized data, fostering a cooperative learning environment that tackles the complexities of data privacy and ownership in machine learning [6]. This technique not only maintains data confidentiality but also enhances model generalization and robustness, exemplified by FedLLM [7–11]. Yet, incorporating Federated Learning into Artificial Intelligence Generated Content (AIGC) models—resulting in FL-powered AIGC—presents an array of intricate trust issues.

Challenge 1: Within the framework of FL-empowered AIGC, nodes collaborate in a joint training process that begins with each node holding data and continues through ongoing updates and iterations during training, culminating in the aggregation of the final model [12]. At any stage, untrustworthy behavior can affect the model's outcome to different extents. Thus, maintaining trustworthiness throughout the entire FL-empowered AIGC process is a considerable challenge.

Challenge 2: The process of FL-powered AIGC model training involves extensive learning and fine-tuning using vast datasets and parameters. Meanwhile, blockchain technology—known for its trustworthiness—demands significant storage and computational power to manage transaction records and verification on the blockchain. This intense resource usage and potential performance constraints could pose a major obstacle in executing this technological merger [1]. Therefore, finding an effective way to balance performance with trust while maintaining robust security is another crucial issue that needs resolution.

This paper presents GuardChain, a trust assurance framework tailored for Federated Learning (FL) in the creation of an Artificial Intelligence Generated Content (AIGC) model. The primary objective is to ensure the trustworthiness of three critical phases in FL-powered AIGC model training while optimizing time efficiency. To tackle the first challenge during data preparation, we employ multi-party data validators for trust cross-data validation, ensuring data reliability and uniformity. In the adapter update stage, we utilize adapter testers with diverse verification techniques to maintain adapter consistency and integrity, thwarting any malevolent alterations. For the aggregation stage, our novel strategy decentralizes control from a single central node to multiple trusted nodes, greatly reducing potential damage from harmful nodes or activities on training outcomes. Addressing the second challenge involves eschewing traditional on-chain storage methods for gradients and data in favor of storing only adapter hashes along with data and adapter reports. We also introduce a "Sandwich" approach that offloads computation-heavy tasks off-chain while retaining lightweight node elections and result validations on-chain. This method strikes an effective balance between efficiency and trustworthiness.

The primary contributions of this paper are as follows:

- We have created a multi-party cross-data validation mechanism that secures the data foundation's integrity. GuardChain successfully filters out untrustworthy data during data poisoning attack scenarios.
- Our approach includes an adapter verification strategy that integrates both on-chain and off-chain mechanisms, along with a proactive malicious adapter detection method to ensure reliable adapter updates. In simulations of attacks by malicious nodes, GuardChain effectively eliminates harmful adapter updates.
- We employ a "Sandwich" strategy that balances computational efficiency and storage demands while decentralizing control over parameter aggregation to maintain trustworthiness. GuardChain reliably thwarts bad actors attempting to merge compromised models in every epoch of our simulations.
- The security analysis has shown that GuardChain effectively mitigates trust-related attacks at three different stages. We have implemented GuardChain and conducted training on the FL-based AIGC model. Comparative performance experiments reveal that GuardChain efficiently manages various workloads and hostile threats, with blockchain-based verification completed within 1–10 seconds. Additionally, it enhances BLEU and ROUGE scores by 0.14 to 0.17 on a standard public dataset.

The rest of this paper is structured as follows: Section 2 provides a detailed review of the relevant literature. In Sect. 3, we present an illustrative example and outline our trust objectives. Section 4 then describes the GuardChain architecture, explaining each component of the framework. Section 5 reports on experiments conducted to test the effectiveness of our approach. The paper concludes with Sect. 6, which summarizes key results and proposes directions for further research in this area.

Related work

We summarize previous research from three aspects: blockchain-based small model FL, FL-empowered AIGC, and trustworthiness in FL-empowered AIGC.

In the domain of deep learning, small machine learning models possess fewer parameters compared to large-scale models. Before the rise of large language models and AIGC large models, numerous studies had already focused on the trustworthiness issues in small model FL [13–21]. For instance, Biscotti, proposed by Muhammad Shayan et al., aimed to counter harmful attacks and was the first to secure privacy through a safe distributed blockchain ledger [22]. Flock leveraged smart contract technology to build a decentralized FL system, using peer-to-peer review and reward mechanisms to prevent malicious clients [23]. Y Li and others designed a committee mechanism, selecting well-performing nodes as overseers to detect malevolent models [24]. DeepChain [25], using the Corda blockchain smart contracts, incentivized model sharing, ensuring the security and privacy of model updates, demonstrated by its effectiveness with the MNIST dataset in training accuracy and encryption strength of updates. These studies address specific issues in blockchain-powered small model training but have yet to comprehensively consider trustworthiness during the model training process or propose a universal framework.

Recently, the rapid advancement of transformer technologies and their diverse iterations [26–28] has ushered Large Language Models (LLMs) into significant achievements across diverse domains, spanning Computer Vision (CV) to Natural Language Processing (NLP) [29]. The triumph of LLMs is primarily attributed to their pre-training paradigm. However, concurrently, there has been a burgeoning emphasis on addressing security and reliability issues within this domain. For instance, the emergent research avenue of FedLLM [7, 10, 30–34] illuminates potential strategies to alleviate privacy concerns inherent to LLMs. Nevertheless, implementing trustworthiness measures within federated learning settings tailored for AIGC models, which often boast billions of parameters, poses a formidable challenge.

In the field of trustworthiness research for FL-empowered AIGC, Chen and others delved into the security challenges faced by AIGC, forecasting future trustworthiness issues and exploring the potential of combining privacy computing, blockchain, and AIGC [12]. Meanwhile, Li et al. focused on analyzing the opportunities and challenges of integrating blockchain and Web 3.0 with AIGC [35]. Large Language Models (LLMs), representative of AIGC models, currently need help in effectively integrating with blockchain or other Web 3.0 technologies due to their high-performance consumption, resource demands, and storage challenges. Existing research, primarily conducted from a review or survey perspective, highlights these challenges and suggests future directions, yet it needs more specific design and implementation.

Background and motivation

In this section, we present a motivating example to illustrate a specific instance of FL-empowered AIGC. From this example, we will subsequently articulate the trust objectives and the principles of performance balance that our trust framework aims to address.

Motivational example

The increased focus on privacy protection has heightened public interest in privacy-friendly Federated Learning (FL) and fine-tuning of large language models (LLMs), such as FedLLM [7, 10, 36–40]. Figure 1 depicts the framework of the FedLLM ecosystem. In the core process of FedLLM, various edge nodes utilize local data

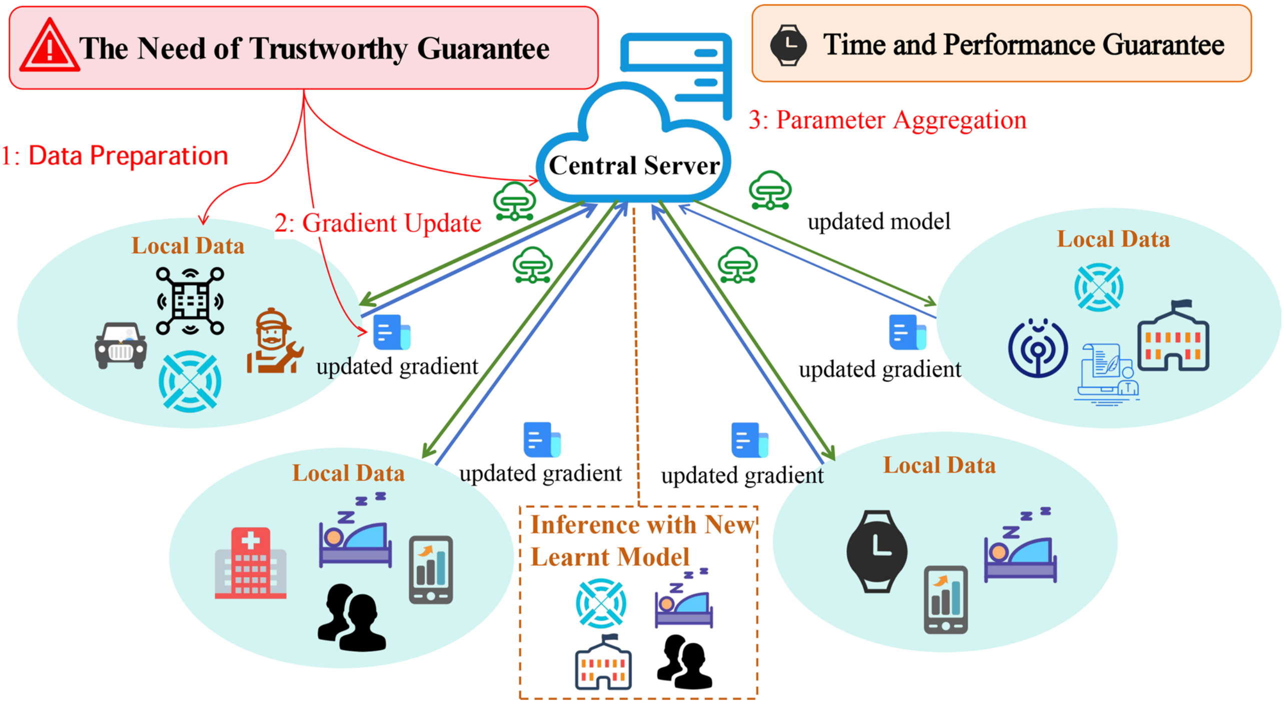


Fig. 1 Trust problems in traditional FL

to train a unified model locally, generating forward gradients or parameter updates. These updates are then transmitted to a central node, where the model aggregation task is performed. Finally, the central node distributes the updated model back to the edge nodes for subsequent training epochs, continuing until the model converges.

As discussed in Sect. 1, FedLLM faces two primary challenges. Firstly, several phases, including data training, edge node updates, and model merging, require trustworthy assurance to ensure the accuracy and reliability of the model training outcomes. Secondly, in AIGC models represented by LLMs, which possess a vast number of parameters and data, determining how to achieve trustworthy assurance in an FL-empowered AIGC scenario with minimal performance and time expenditure is a significant challenge.

Trust goals

Trusted Data Preparation: This phrase prioritizes the reliability of data within an FL context, ensuring it is unbiased and representative. It focuses on sourcing credible data from trustworthy edge nodes, preventing malicious actors' potential exploitation of AIGC models.

Trusted Adapter Update: This phase ensures that adapter updates are non-malicious and come from credible edge nodes. They do not have malicious influences on the final parameter results. The storage of the adapter update files involves ensuring traceability, immutability, and transparency, allowing all participants to verify updates, thus maintaining the integrity and fairness of the model update process.

Trusted Parameter Aggregation: The final phase involves crucial credibility processes in aggregating parameters. In this phase, it is imperative to ensure that the task of parameter aggregation is entrusted to one or more trustworthy nodes, thereby mitigating the risk of malicious activities during the merging process right from the outset. Subsequently, the aggregated results should be verified to guarantee the aggregation outcome. Concurrently, secure storage for each model update cycle is also crucial in maintaining the traceability of FL-empowered AIGC.

Balance among trust guarantee, and performance

One of the AIGC model's most prominent characteristics is its billions of parameters [1]. In the context of FL, when considering integrating trusted technologies such as blockchain to enhance the trustworthiness of the AIGC model, we must take storage and performance into consideration [41]. As a distributed data storage system, blockchain may entail higher costs and slower data retrieval than traditional centralized storage methods [42]. In the realm of FL-empowered AIGC, this can potentially give rise to performance issues, particularly concerning AIGC models.

When applying trusted technologies, a delicate balance must be struck among performance, time efficiency, and trustworthiness. We need to explore novel approaches that can provide a certain level of trust assurance for the AIGC model FL while minimizing the expenditure of time and computational resources. This becomes imperative to effectively address the trusty challenges posed by the scale and complexity inherent in AIGC models.

GuardChain: multi-stage trust framework for FL-empowered AIGC

The FL-empowered AIGC model training framework established based on GuardChain is divided into three trusted stages (trusted data preparation, trusted adapter update, and trusted parameter aggregation) and six layers as shown in Fig. 2. The trusted data preparation stage is primarily concerned with ensuring the reliability and benign nature of the data provided by edge nodes. The "Trusted Adapter Update" focuses on guaranteeing that updates originate from trustworthy nodes and do not maliciously impact the results. Additionally, the traceability of adapter updates is a critical aspect. The trusted parameter aggregation stage ensures that, through competitive processes, the responsibility for the aggregation task is distributed among multiple credible nodes, thereby assuring the trustworthiness of the aggregated outcomes.

There are 6 important roles in GuardChain. When edge nodes play different roles, they will receive different rewards after accurately performing tasks:

- **Data Validator:** Responsible for verification of local edge node data, maintaining data source integrity, and detecting anomalies.
- **Adapter Tester:** Evaluates edge server adapter updates on model replicas, testing performance, and flagging malicious adapters.
- **Adapter Calculator:** Nodes involved in model training, using local data to compute adapters.
- **Aggregate Candidate:** Nodes expressing interest in aggregation tasks after a training epoch.
- **Parameter Aggregator:** Chosen from candidates based on contribution, tasked with model merging and parameter updates. The computation result is validated through smart contracts.
- **Adapter Recorder:** Packages adapter hash transactions for blockchain storage, selected based on computational capacity.

In this section, we adopt the stage-analysis method to solve the trust challenges at each stage in detail.

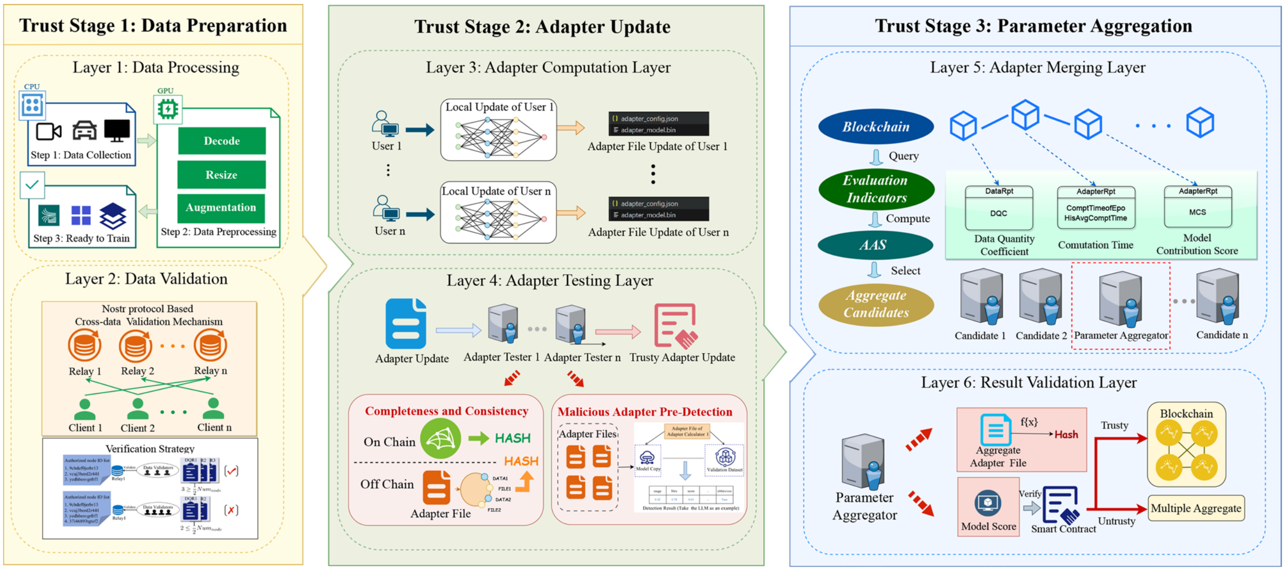


Fig. 2 System overview of GuardChain

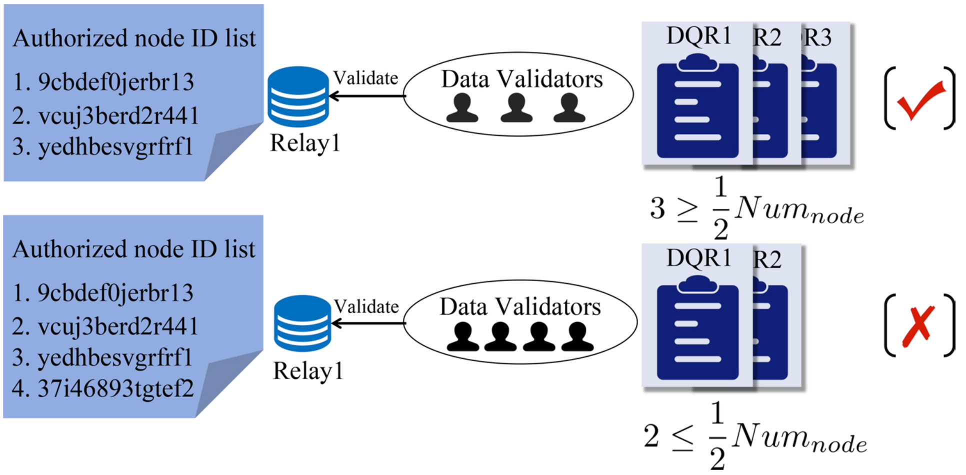


Fig. 3 Cross-validation data mechanism and data validator set configuration in stage 1

Data preparation trust

The data preparation trust stage aims to ensure that the training datasets provided by all nodes are trustworthy, preventing malicious data from participating in the training and influencing the outcomes. The data verification phase is fundamental to ensuring the trustworthiness of large-model training. The trusted data preparation is mainly composed of the data collection and processing layer and the data validation layer:

- **Data Acquisition and Processing Layer:** Sensors and Internet of Things (IoT) devices collect various data types, such as images, sounds, and temperature. These devices transmit raw data to edge servers, where preliminary processing, such as formatting and noise filtering, is performed.
- **Data Validation Layer:** The data validation layer relies on the Nostr protocol, with the database serving as relay nodes and data validators acting as client nodes. Authorized client nodes perform cross-validation and generate multiple data quality certificates stored on the blockchain. Only nodes with sufficient certificate endorsements can upload adapter updates.

Figure 3 shows the cross-validation data mechanism and data validator set configuration. Detailed description is as follows:

To ensure data verification while maintaining data privacy, we propose a multi-party cross-validation data mechanism based on the Nostr protocol access control. Using the Nostr protocol, edge nodes can function as relay and client nodes. As relay nodes, they primarily store datasets trained locally at edge nodes. As client nodes, they can either be the subject of verification, authorizing other edge nodes to verify their data, or act as data validators authorized by other nodes. The data quality report(DQR) generated by the data validator will undergo Ring Signature verification prior to the adapter being uploaded to the blockchain [43] to ensure that a node has sufficient endorsements to be eligible for on-chain submission.

Data validators verify the trustworthiness of data, including the data itself, source inspection to confirm if it originates from a trusted node, and checking for anomalies. After inspection, validators generate a corresponding Data Quality Report, DataRpt = DataID, SignNodeID, SourceNodeID, DataHash, Sign, isValidData, DQC, where DataID, SignNodeID, SourceNodeID represent the Data ID, Data Validator Node ID, and Data Source ID, respectively. DataHash is for verifying data integrity, Sign is the validator's signature on the hash, and isValidData indicates whether the data is verified as valid or non-malicious. DQC represents the data volume coefficient and serves as one of the criteria for subsequent node evaluation.

A particular node must obtain data certificates from at least 1/2 of the nodes to participate in model training, as shown in 3. The rationale for choosing 1/2 of the nodes is based on the following considerations:

Paxos Algorithm: Referencing the Paxos algorithm [44], where a proposal is accepted by the system only if agreed upon by a majority (more than half) of the nodes. For instance, in GuardChain, if there are 10 edge nodes participating in training, each node possesses this characteristic, ensuring that the system can reach consensus even if some nodes fail.

Balancing Performance and Security: Choosing a consensus mechanism involving more than half the nodes balances system performance and security. While a higher consensus threshold could offer stronger security assurances (as in Byzantine fault-tolerant systems, where usually more than 2/3 of the nodes are needed) [45], this could potentially lower system performance and responsiveness.

Preventing Malicious Behavior: In systems containing Byzantine nodes (capable of malicious or unpredictable behavior) [45], a consensus mechanism involving more than half of the nodes is crucial to prevent a minority of malicious nodes from controlling or compromising the entire system.

In an FL environment, this partial verification approach not only ensures data privacy but also provides validation and assurance of data quality.

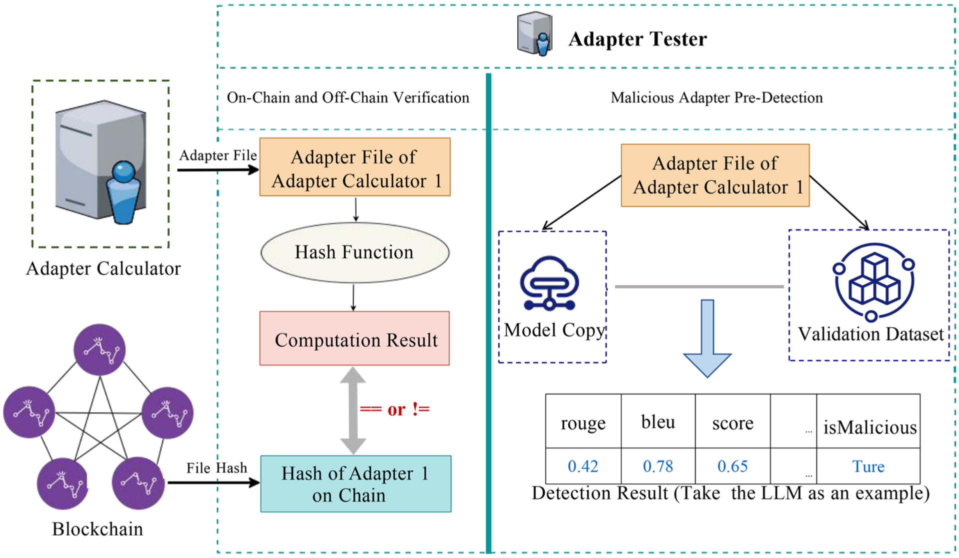
Adapter update trust

During the adapter update stage, our main objective is to ensure that the adapter updates provided by each node are reliable, consistent, and intact, and that these updates do not maliciously impact the final results of the parameters. Ensuring the trustworthiness of adapter updates is a fundamental precondition for successful adapter merging. The adapter update stage includes the adapter computation layer:

- **Adapter Computation Layer:** Edge servers possess significant computational capabilities and execute tasks such as model fine-tuning and adapter calculations. Efficient fine-tuning methods like LoRA are used to generate adapter updates, and the upload of adapter file hashes only occurs when nodes provide sufficient data validation report endorsements.
- **Adapter Testing Layer:** Adapter testers lead the testing of adapter updates, using test sets to check the global model replica for any malicious impacts from the adapters. Nodes under testing send adapter updates to multiple testers who validate the adapter file hash to ensure integrity and detect malicious adapters. The testers ultimately produce adapter detection reports and upload them to the blockchain.

In terms of adapter updates, once the adapter tester obtains the updated adapter file, it is necessary to conduct a dual-level trust verification of the adapter, which includes both integrity checks. Consistency checks and malicious adapter detection, as shown in Fig. 4. Each adapter test report will be individually verified before the subsequent adapter merging process. An adapter is considered trustworthy and can participate in the upcoming adapter merging process only after receiving sufficient adapter test reports from adapter testing nodes.

To ensure the integrity and consistency of adapter updates, we propose a combined on-chain and off-chain verification strategy, which aims to confirm whether the adapter updates come from trustworthy nodes and whether they adhere to the standards of integrity and consistency. The primary task in testing a node's adapter involves checking whether the adapter has been tampered with, ensuring its consistency and integrity. Given the blockchain's storage limitations and performance bottlenecks, we have abandoned the traditional FLChain method of directly storing adapter updates on the blockchain. Instead, we opt to upload the hash values of the adapter files onto the blockchain, a method that better meets the performance requirements of the blockchain. After retrieving the adapter file for adapter updates, the adapter tester performs a local hash computation. It compares this hash with the hash of that node's adapter


Adapter Tester
On-Chain and Off-Chain Verification
Malicious Adapter Pre-Detection
Adapter File
Adapter File of Adapter Calculator 1
Adapter Calculator
Hash Function
Computation Result
== or !=
Blockchain
File Hash
Hash of Adapter 1 on Chain
Adapter File of Adapter Calculator 1
Model Copy
Validation Dataset

rouge	bleu	score	...	isMalicious
0.42	0.78	0.65	...	Ture

Detection Result (Take the LLM as an example)

Fig. 4 Integrity checks, consistency checks and malicious adapter detection in stage 2

update for the current epoch on the blockchain. If they match, the adapter is deemed to have passed the integrity check.

Malicious Adapter Pre-Detection primarily focuses on determining whether the adapter updates are malicious, i.e., whether these updates significantly impact vital metrics such as the model's accuracy and recall rate. To address this issue, after each round of adapter upload to the blockchain, we send testing requests to multiple adapter testers to evaluate these adapters. The main tasks of an adapter tester include reading the updated adapters of a node from the blockchain, applying them to a replica of the global model, performing model performance tests based on a local data test set, and generating an adapter test report. The structure of the adapter test report is defined as AdapterRpt = GRID, AdapterID, SourceNodeID, EndorseNodeID, EndorseNodeSign, DataID, TrainRound, MCS, where SourceNodeID and EndorseNodeID represent the Adapter Source Node ID and Adapter Verification Node ID, respectively. The verification node in the blockchain also acts as the endorser of the adapter. EndorseNodeSign is the endorser's signature, DataID is the dataset ID, TrainRound indicates the current training epoch, and MCS is the contribution score of the model.

Parameter aggregation trust

In FL, the parameter aggregation Trust stage serves as the final safeguard to ensure the credibility of the entire process. It is composed of an adapter merging layer and a result validation layer:

- **Adapter Merging Layer:** When the model merging threshold is reached, smart contracts issue parameter aggregation tasks. Interested edge nodes can respond and are marked as aggregation candidate nodes. The system comprehensively evaluates parameter aggregators based on their composite scores. The parameter aggregator reads trusted adapter updates and various adapter detection reports, aggregating adapters with sufficient detection report endorsements.
- **Result Validation Layer:** The adapter file hash of the aggregated results and model evaluation scores is packaged and uploaded to the blockchain. Smart contracts validate the reasonableness of the results to confirm the effectiveness of the aggregation.

The parameter aggregation trust stage's core objective is to prevent the model merging process from being controlled by any single malicious entity. To achieve this, the power of model merging should be distributed among multiple trustworthy nodes, and the merging results should be thoroughly verified. In the following, we will introduce the strategy for selecting the parameter aggregator and the "Sandwich" task allocation and verification strategy.

Strategy for Selecting the Parameter Aggregator: The Parameter Aggregator, responsible for merging adapter updates from all nodes, plays a pivotal role in the FL process. When selecting a Parameter Aggregator, it is crucial to avoid repeating the selection of the same Aggregate Candidate within a specified number of rounds, as the choice of aggregator significantly impacts the outcome. This selection should consider all factors that affect the results and performance of the model. On the one hand, adapter aggregation is a compute-intensive task, and servers with high computational capabilities can process

Table 1 Notation description in parameter aggregator selection strategy

Notations	Description
i	The ID number of a certain node
HCT	Historical Average Computation Time
CCT	Current Computation Time
CAS	Computational Ability Score
DQC	Data Quantity Coefficient
MCS	Model Contribution Score
AC	Activity Coefficient
parEpo	The number of epochs participated in
sumEpo	The total number of training epochs
CS	Contribution Score
AAS	Aggregator Advantage Score

and integrate adapter updates from different nodes more quickly and effectively, thus accelerating the overall training cycle. On the other hand, model contribution, as the quantitative measure of a node's performance in the network, reflects its historical contributions and reputation. Selecting nodes with high model contribution scores as the adapter aggregator reduces the risk of malicious activities from malicious nodes and motivates all participants to improve their performance and engagement. Furthermore, to prevent any single node from monopolizing the role, it should be stipulated that a parameter aggregator cannot repeat its role in the subsequent k rounds of computation. This strategy ensures the accuracy and reliability of the model while meeting the efficiency requirements of comput-intensive tasks. Table 1 summarizes the key symbols and notations used in the aggregator selection strategy.

The computational capability score primarily comprises a quantitative score for computational ability and a quantitative score for contribution, calculated as follows:

Here, λ is a weight coefficient ranging between 0 and 1, used to balance the influence of Historical Computation Time (HCT) and Current Computation Time (CCT). For example, if $\lambda = 0.5$, then the influence of both historical and current computation times is equal. The formula for the Computational Ability Score (CAS) is given by:

$$\mathrm{CAS} = \frac{\mathrm{DQC}}{\lambda \times \mathrm{HCT} + (1-\lambda) \times \mathrm{CCT}} \tag{1}$$

The Activity Coefficient (AC) is calculated as:

$$\mathrm{AC} = \frac{\mathrm{parEpo}}{\mathrm{sumEpo}} \tag{2}$$

The Contribution Score (CS) is then given by:

$$\mathrm{CS} = (\mathrm{DQC} \times \mathrm{MCS}) \times \mathrm{AC} \tag{3}$$

The Parameter Aggregator Advantage Score (AAS) combines the Contribution Score (CS) and the Computational Power Score (CPS), with a defined weight ratio for both. For instance, the weight of the Contribution Score is α, and the weight of the Computational Ability Score is β, where $\alpha + \beta = 1$.

The AAS is calculated as:

$$\mathrm{AAS} = \alpha \times \mathrm{CS} + \beta \times \mathrm{CAS} \tag{4}$$

"Sandwich" Task Allocation and Verification Strategy: In the trust verification phase of GuardChain, considering the performance bottlenecks of blockchain, we have adopted a "Sandwich" strategy as shown in Fig. 5. This approach entails competitive allocation of compute-intensive tasks on the blockchain, local execution off-chain, and final verification of results on the blockchain.

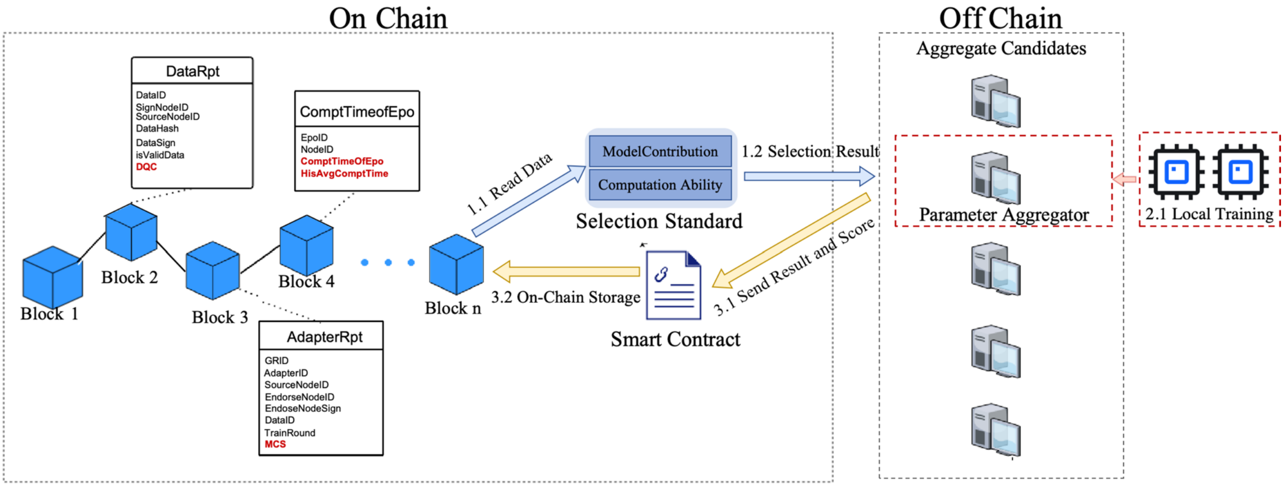


Fig. 5 "Sandwich" task allocation and verification strategy

Specifically, this strategy allocates roles and tasks based on metrics recorded on the blockchain, with edge servers performing compute-intensive tasks locally and using smart contracts on the blockchain to verify the execution results.

After the parameter aggregation is completed, the adapter aggregator uploads the updated adapter hash and the model score of the current epoch to the blockchain for storage. The model's performance may fluctuate in practical applications due to data changes and model adjustments. Hence, the score of each epoch in the entire training process should ideally exhibit a fluctuating upward trend. By understanding the extent of these fluctuations, we can more reasonably determine whether the new aggregation results fall within an acceptable range. If the smart contract verification shows that the aggregation score in the current epoch does not decrease, or even if it decreases, it remains within a reasonable threshold range. We can consider the aggregation result to be trustworthy. The method for calculating this threshold is described as follows:

In the text, x_i represents the model score for the i^{th} round, with a total of n training rounds. We first calculate the mean of these scores.

$$\mathrm{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5}$$

We calculate the standard deviation.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{6}$$

This code snippet will Here, σ represents the standard deviation, which measures the degree of deviation of the scores from their mean value.

Next, we can utilize this standard deviation to establish a threshold. A common practice is to use a multiple of the standard deviation. The choice of the multiplier can vary depending on the requirements of different models. For instance, we might select a threshold of twice the standard deviation.

$$T = 2\sigma \tag{7}$$

Assume that the model score obtained after a new round of aggregation is x_{new}. We compare this new score with the average score $\bar{x}$. If the decrease in the new score is within the threshold T, we consider the result of this aggregation to be within the normal fluctuation range and, therefore, acceptable. That is:

$$|x_{\mathrm{new}} - \bar{x}| \leq T \tag{8}$$

Implementation and experiment

GuardChain, leveraging technologies like blockchain and the Nostr protocol, implements a universal trust assurance system for FL-empowered AIGC training. To thoroughly evaluate the effectiveness of GuardChain in accordance with the proposed trust goals and performance principles, we address the following two research questions (RQs):

RQ1: Can GuardChain maintain the trustworthiness of the training results in the face of various malicious attacks throughout the FL-empowered AIGC model training process?

RQ2: Faced with varying scales of FL and degrees of attacks, can GuardChain demonstrate good scalability and adaptability, completing trustworthy endorsement verification within a shorter time and with reduced performance overhead?

Experimental setting

Prototype and Model We have developed a prototype of GuardChain to cater to stakeholders requiring trustworthy model training. GuardChain employs a novel blockchain architecture, with its state database built on ElasticSearch and the ledger database founded on MySQL. The development was carried out using Go v0.18.4. For model use, we evaluate GuardChain mainly on the popular LLMs. The model we use is—LLaMA2-7B [46]. The LLaMA2-7B model, a variant of the LLaMA (Large Language Model from Meta AI) series, is a potent language model with approximately 7 billion parameters, offering advanced natural language processing capabilities. The model is fine-tuned using LoRA (Low-Rank Adaptation)****, an efficient method for adapting large pre-trained models with minimal additional parameters. We experiment with the NLP datasets—Stanford Alpaca dataset [47]. The Stanford Alpaca dataset is a curated set of instruction-following prompts designed to train language models, particularly the LLaMA model, to better understand and execute task-oriented commands.

FL settings and Hardware As noted in previous FL literature [48–50], our experiments were conducted semi-simulatively on two servers, each equipped with 8 RTX 4090 GPUs. To minimize resource wastage, we simulated various metrics of GuardChain under different numbers of participating edge nodes on each machine. For FL settings, GuardChain and all baselines utilized the same hyperparameters before training as outlined in Table 2, and the default FL aggregation algorithm employed was FedAVG. In the experiment, we chose BLEU and ROUGE scores as the model scores to evaluate the model's accuracy and recall. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual et al.) are

Table 2 Hyperparameter set of model

Parameter	Value
learning rate	10^{-4}
LoRA rank	64
LoRA alpha	128
LoRA dropout	0.05
batch size	2
adapter accumulation steps	8
max sequence length	512

two famous metrics for evaluating language models, particularly in tasks like text summarization (ROUGE) and machine translation (BLEU). Combining these two metrics provides a more comprehensive assessment of a model's performance, focusing on different aspects of language understanding and generation.

Baseline and Experiment Set We established a baseline training framework, configured as a fundamental large-model federated learning setup without additional trust assurance mechanisms. This baseline operates in an environment and utilizes a dataset identical to that employed in GuardChain, but is executed under conditions devoid of trust safeguards. Our experimental design is bifurcated into two distinct segments. The first segment is the security analysis experiment, primarily focused on how GuardChain mitigates various attacks to ensure model trustworthiness. The second segment is the accuracy and verification efficiency analysis experiment, concentrating on how GuardChain effectively balances performance with model accuracy.

Security analysis (RQ1)

In this experimental section, we simulate various attacks, including data poisoning, adapter tampering, and malicious participant attacks. We compare the model scores under these attack scenarios between the baseline trainer and GuardChain. This analysis is aimed at evaluating the effectiveness of the GuardChain trust framework in mitigating and defending against these attacks. The experiments are conducted in three modules:

Data Poisoning Attacks: Data poisoning attacks typically occur during the data collection phase, where attackers deliberately manipulate or inject erroneous data to deviate the model's training outcome from its intended path. This can be executed by introducing samples with incorrect labels or, more subtly, by altering the data distribution to impact the model's generalization capability. We simulate this malicious influence on the model by injecting malevolent data into one of the edge nodes in both our baseline trainer and GuardChain.

Malicious Node Attacks: In a distributed or federated learning environment, multiple participants collaborate to train a model, each computing their adapter on their data and sending it to a server for aggregation. Malicious node attackers may attempt to compromise the overall training of the model by modifying adapters or uploading malicious adapter files to skew the training results or degrade model performance. We simulate attacker behavior by tampering with the generated adapter files on one of the edge nodes.

Model Poisoning Attacks: In a federated learning environment, multiple edge servers' parameters must be aggregated. A cloud or central server is traditionally chosen to perform this merging task. If an attacker gains control over the central server or node responsible for parameter aggregation, they could deliberately aggregate these parameters incorrectly, leading to the model's inability to learn correctly or deviating from its intended learning path. We simulate a scenario where a node acts as a malicious parameter aggregator, deliberately merging the model maliciously during the parameter merging process of a particular epoch.

Figures 6, 7, and 8 distinctly demonstrate the impacts of data poisoning attacks, malicious node attacks, and model poisoning attacks on performance metrics, including BLEU scores, ROUGE-1 F1 scores, ROUGE-2 F1 scores, and ROUGE-L F1 scores. During the initial training phase, precisely the first epoch, the baseline trainer and the GuardChain model exhibited comparable

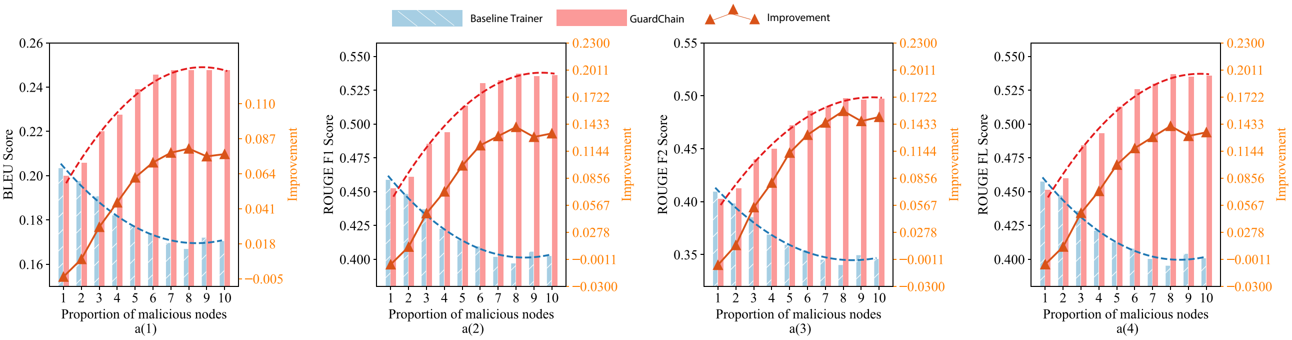


Fig. 6 The model performance of the baseline trainer and GuardChain under a data poisoning attack with 10 nodes. The model evaluation metrics for a(1), a(2), a(3), and a(4) are BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively

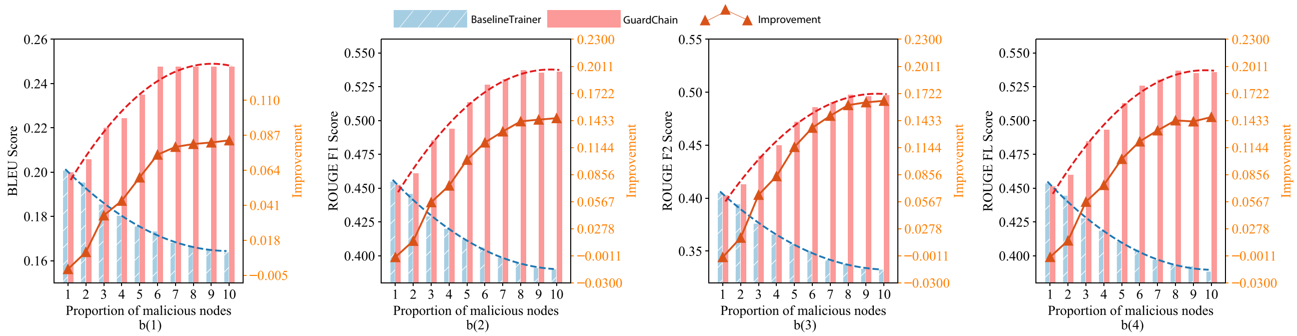


Fig. 7 The model performance of the baseline trainer and GuardChain under malicious node attacks with 10 nodes. The model evaluation metrics for b(1), b(2), b(3), and b(4) are BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively

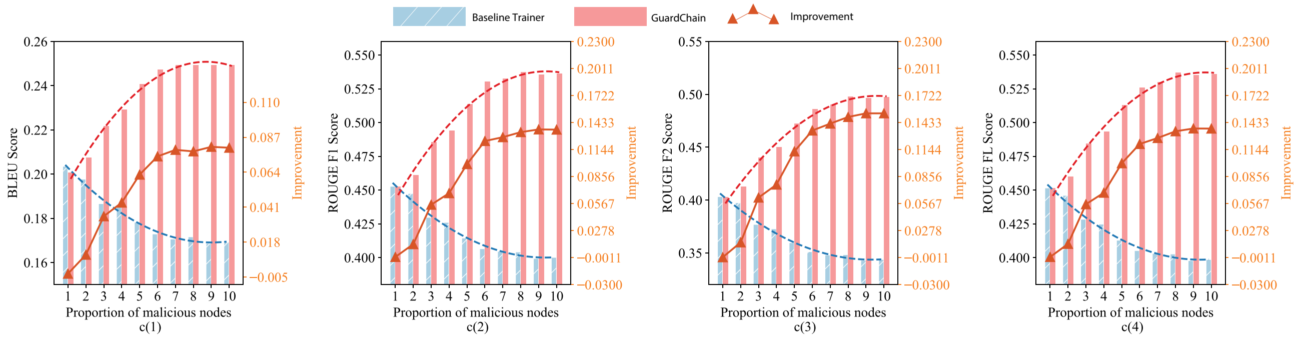


Fig. 8 The model performance of the baseline trainer and GuardChain under model poisoning attacks. The model evaluation metrics for c(1), c(2), c(3), and c(4) are BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively

performances under these three types of attacks, with only marginal differences in scores.

However, as the number of epochs increased, the trend graphs distinctly illustrated a divergence in performance between the two: the baseline trainer, lacking robust security measures, showed a downward trend in overall scores. In contrast, benefiting from the trusted protection mechanisms of GuardChain, there was a noticeable upward trend in its scores. Finally, at the 10th epoch, the score difference was around 0.14. With the persistence of malicious attacks, the performance disparity between the two training approaches becomes increasingly pronounced. This result will be more apparent when applying good model calculations.

Accuracy and verification efficiency analysis (RQ2)

In evaluating blockchain-based trust-enabled federated learning frameworks, it is crucial to consider the balance between performance and model accuracy. To achieve this objective, we devised two pivotal experiments. These involve comparing the accuracy of GuardChain with that of a baseline trainer with multiple malicious nodes, which contributed malicious adapter updates during the first epoch, and assessing the time efficiency of completing adapter trust verification within the GuardChain framework. This dual-experimental approach comprehensively evaluates the framework's ability to maintain a balance between high model accuracy and efficiency in complex real-world data environments, as detailed in the accompanying Table 3.

Table 3 Independent variable settings of accuracy and verification efficiency analysis experiment

Experiment Number	Sum Node Number	Proportion of malicious nodes
1	5, 10, 15	40%
2	10	0, 10%, 20%, 30%, 40%

Experiment 2.1 We maintain the total number of nodes with a constant percentage of malicious nodes as the independent variable, aiming to observe the efficiency and accuracy of the framework under different workload conditions. This design enables us to assess the framework's performance when processing FL-empowered AIGC of varying scales.

As depicted in Fig. 9, a dual-colored bar graph represents the model scores for Baseline Trainer and GuardChain. In contrast, the line graph highlights the score enhancements achieved by GuardChain over the Baseline Trainer. Observing the scenarios with 5, 10, and 15 nodes in federated learning, GuardChain's BLEU scores improved between 0.0001 and 0.0006. For 5-node scenarios, the ROUGE-1, ROUGE-2, and ROUGE-L F1 scores

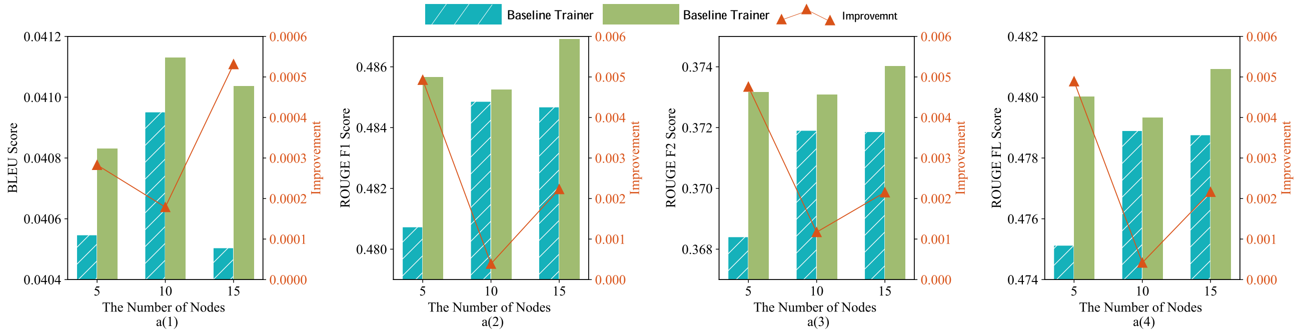


Fig. 9 With the total count of participating nodes varied sequentially among 5, 10, and 15. The histogram in our findings depicts the performance scores of both baseline trainer and GuardChain models under the condition of malicious nodes contributing to adapter updates. Complementarily, the accompanying line chart delineates the disparity in scores between the two models, effectively capturing the differential, namely $Score_{GuardChain} - Score_{Baseline}$. The model evaluation metrics for a(1), a(2), a(3), and a(4) are BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively

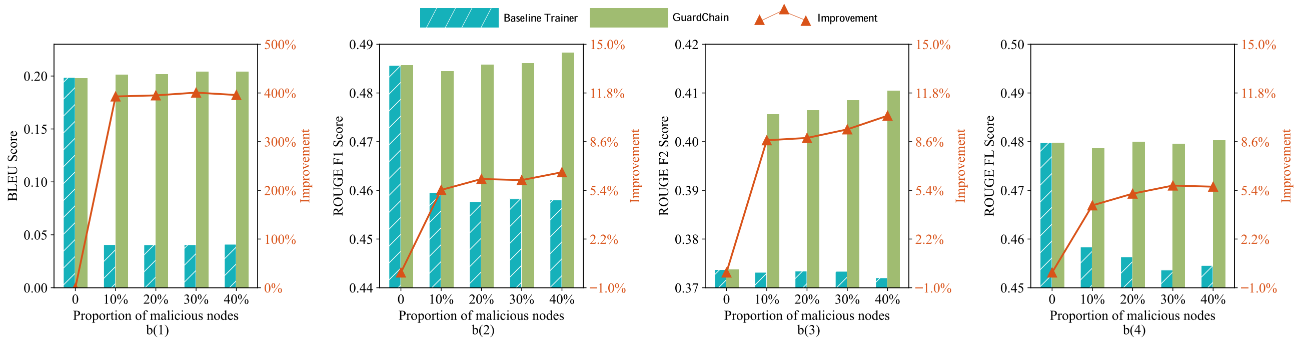


Fig. 10 With a fixed number of ten nodes participating in federated learning (FL), we simulate real-world scenarios of varying scales of malicious attacks by altering the proportion of malicious nodes. Starting from 0%, this proportion increments in steps of 10%, reaching up to 40%. The bar graph in our study illustrates the disparity in model performance between baseline trainer and GuardChain under these conditions. Additionally, a line graph is employed to represent the variation in accuracy between the two models. The model evaluation metrics for b(1), b(2), b(3), and b(4) are BLEU, ROUGE-1, ROUGE-2, and ROUGE-L, respectively

saw an increase of approximately 0.005, with 10-node scenarios exhibiting a rise of about 0.001 and 15-node scenarios around 0.002. GuardChain also displayed incremental improvements in Precision and Recall for ROUGE scores. Notably, these enhancements were modest due to our experimental setup involving fine-tuning pre-trained models. However, for models highly susceptible to malicious scenarios or those without prior training, GuardChain is anticipated to deliver more remarkable and impactful results.

Regarding the endorsement time for adapter quality reports on GuardChain, as shown in Fig. 10, on-chain verification for 5 nodes could be completed in around 5 seconds, about 50 seconds for 10 nodes, and approximately 100 seconds for 15 nodes. In configurations with fewer nodes, the on-chain verification time per node could reach around 1 second. This duration is negligible compared to the days or even months of training time typical for AIGC models, thus being considered acceptable.

Experiment 2.2 With a constant total number of nodes, we vary the percentage of malicious nodes as our independent variable. This allows us to evaluate the framework's robustness and accuracy when dealing with inputs of varying data quality. This experimental setup reflects the fluctuating quality of real-world data, aiding our understanding of how the framework copes with imperfections in data.

In Fig. 11, we present the effectiveness of GuardChain in managing the participation of malicious nodes during the FL-empowered AIGC training process, particularly when the number of participating nodes is fixed at ten, and the proportion of malicious nodes escalates from 0% to 40%. Notably, with a 0% malicious node proportion, the training results of Baseline Trainer and GuardChain are nearly identical, demonstrating the model's fundamental performance. However, as the proportion of malicious nodes increases to 10%, there is a precipitous decline in the model score of the Baseline Trainer, with the BLEU score dropping by approximately 0.2 and the ROUGE F1 score decreasing by around 0.3. This significant reduction indicates the detrimental impact of malicious nodes on model accuracy. Throughout the presence of malicious nodes, a considerable difference in

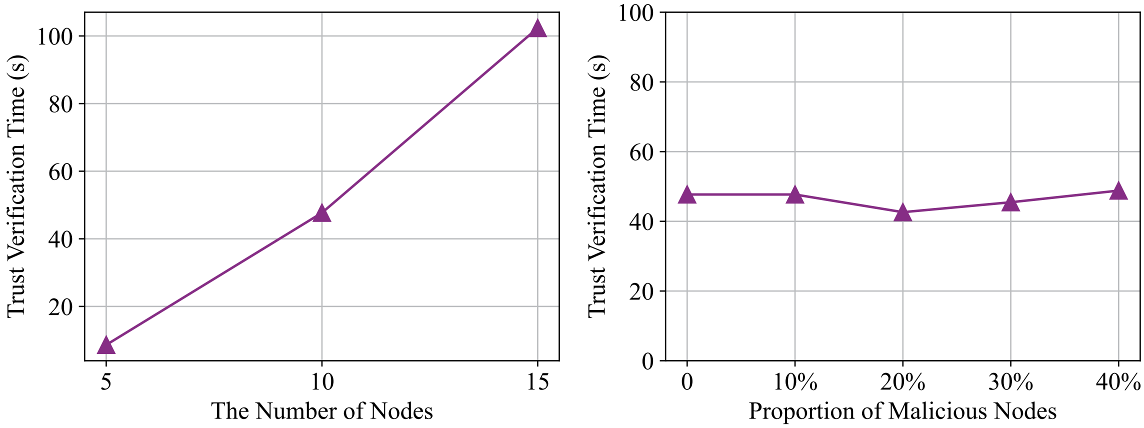


Fig. 11 The left-hand graph in our presentation depicts the time required by GuardChain for trustworthy endorsement of adapter verification reports as the number of nodes varies. The right-hand graph, on the other hand, illustrates the time taken by GuardChain to reliably endorse adapter verification reports in response to changes in the percentage of malicious nodes. The values presented in this experiment represent the average results obtained after ten iterations

model scores between Baseline Trainer and GuardChain persists.

Moreover, as the proportion of malicious nodes increases, the line graph distinctly illustrates the widening gap in model scores between GuardChain and Baseline Trainer. This trend is especially evident in the ROUGE-F2 scores, which exhibit a noticeable upward trajectory, signifying that the more significant the proportion of malicious nodes, the more substantial their impact on model accuracy. Therefore, it is foreseeable that with many nodes participating in FL-empowered AIGC, there is an increased reliance on GuardChain to provide trust assurance for model training, ensuring the integrity and accuracy of the model are not compromised by malicious interventions.

In scenarios where the proportion of malicious nodes increases while the number of nodes remains fixed at ten, the endorsement time for adapter quality reports in GuardChain consistently stabilizes between 40 and 60 seconds. Consequently, each node requires approximately 5 seconds for endorsement. This efficiency is crucial, especially in the context of model training, where the presence of malicious nodes can fluctuate. GuardChain demonstrates its capability to complete verifications swiftly, even amidst such variability in the network's trust environment.

Conclusion and future work

In this study, we first introduced the concept of end-to-end trust assurance to the FL scenario of AIGC models and proposed and implemented the GuardChain system. Using a phased analysis approach, we ensured the trustworthiness of the three crucial phases: data preparation, adapter updates, and parameter aggregation. We have successfully realized the GuardChain system and conducted extensive experiments on popular large language models and publicly available real datasets. The evaluation results have validated the effectiveness of GuardChain. Furthermore, the GuardChain system is open-source, and we welcome contributions from the research community.

Given the resource-intensive nature of training AIGC models and the inherent characteristics of blockchain, there are limitations in the extent to which the aggregation results of parameters can be verified on the blockchain. This constraint leads to a scenario where only a partial, fuzzy verification of parameter aggregation results is feasible rather than a complete on-chain process replication. Consequently, this does not guarantee the absolute accuracy of the outcomes. In future research, we aim to investigate comprehensive methods for validating the computational results of AIGC model aggregation nodes within this framework.

Acknowledgements
This work was supported in part by the National Science Foundation of China under Grant (62472375), and in part by the Major Program of National Natural Science Foundation of Zhejiang (LD24F020014, LD25F020002), and in part by the Zhejiang Pioneer (Jianbing) Project (2024C01032), and in part by the Ningbo Yongjiang Talent Programme (2023A-198-G).

Author contributions
The individual contributions of the authors are as follows: Yike Yuan conceived the study, designed the overall GuardChain architecture, and drafted the initial manuscript. Xinkui Zhao and Shuyi Yu developed the prototype system, implemented the simulation environment, and conducted all experiments. Shengye Pang designed the security analysis experiments, analyzed the experimental data, and prepared Figures 1-8. Guobing Zou contributed to the theoretical foundation, designed the "Sandwich" strategy and trust verification algorithms, and all authors reviewed the manuscript.

Funding
This work was supported in part by the National Science Foundation of China under Grant (62472375), and in part by the Major Program of National Natural Science Foundation of Zhejiang (LD24F020014, LD25F020002), and in part by the Zhejiang Pioneer (Jianbing) Project (2024C01032), and in part by the Ningbo Yongjiang Talent Programme (2023A-198-G).

Data availability

No datasets were generated or analysed during the current study.

Declarations

Competing interests

The authors declare no competing interests.

Received: 18 October 2025 / Accepted: 6 January 2026
Published online: 28 January 2026

References

1. Cao Y, Li S, Liu Y, Yan Z, Dai Y, Yu PS, et al (2023) A comprehensive survey of ai-generated content (aigc): a history of generative ai from gan to chatgpt. arXiv preprint arXiv:230304226
2. Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA (2018) Generative adversarial networks: an overview. IEEE Signal Process Mag 35(1):53–65
3. Xia W, Zhang Y, Yang Y, Xue JH, Zhou B, Yang MH (2022) Gan inversion: a survey. IEEE Trans Pattern Anal Mach Intel 45(3):3121–3138
4. Guo D, Chen H, Wu R, Wang Y (2023) AIGC challenges and opportunities related to public safety: a case study of ChatGPT. J Saf Sci Resilience 4(4):329–339
5. L'heureux A, Grolinger K, Elyamany HF, Capretz MA (2017) Machine learning with big data: challenges and approaches. IEEE Access 5:7776–7797
6. Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R et al (2019) A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM workshop on artificial intelligence and security, pp 1–11
7. Cai D, Wang S, Wu Y, Lin FX, Xu M (2023) Federated few-shot learning for mobile NLP. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, pp 1–17
8. Pang S, Zhao X, Luo J, Zheng B, Yin J, Zheng X (2023) Incentive-driven pricing game for multi-edge service providers towards optimal profits. In 2023 IEEE International Conference on Web Services (ICWS), pp 350–359
9. Yao M, Qi S, Tian Z, Li Q, Han Y, Li H et al (2025) Quantifying bytes: understanding practical value of data assets in federated learning. Tsinghua Sci Technol 30(1):135–147. https://doi.org/10.26599/TST.2024.9010034
10. Zhang J, Vahidian S, Kuo M, Li C, Zhang R, Wang G, et al (2023) Towards building the federated GPT: federated instruction tuning. arXiv preprint arXiv:230505644
11. Zhao H, Du W, Li F, Li P, Liu G (2023) FedPrompt: communication-efficient and privacy-preserving prompt tuning in federated learning. In ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE 1–5
12. Chen C, Wu Z, Lai Y, Ou W, Liao T, Zheng Z (2023) Challenges and remedies to privacy and security in AIGC: exploring the potential of privacy computing, blockchain, and beyond. arXiv preprint arXiv:230600419
13. Shan Z, Chen X, Zhang Y, He Y, Wang D (2025) Exploration and practice of constructing trusted public IT systems using blockchain-based service network. Tsinghua Sci Technol 30(1):124–134. https://doi.org/10.26599/TST.2023.9010159
14. Zhang K, Tsai PW, Tian J, Zhao W, Cai X, Gao L et al (2024) Towards privacy in decentralized IoT: a blockchain-based dual response DP mechanism. Big Data Min Analytics 7(3):699–717. https://doi.org/10.26599/BDMA.2024.9020023
15. Kumar R, Khan AA, Kumar J, Golilarz NA, Zhang S, Ting Y et al (2021) Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. IEEE Sensors J 21(14):16301–16314
16. Wang F, Chen C, Liu W, Fan T, Liao X, Tan Y et al (2024) CE-RCFR: robust counterfactual regression for consensus-enabled treatment effect estimation. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp 3013–3023
17. Rehman MH, Salah K, Damiani E, Svetinovic D (2020) Towards blockchain-based reputation-aware federated learning. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE p 183–188
18. Cui L, Su X, Ming Z, Chen Z, Yang S, Zhou Y et al (2020) CREAT: blockchain-assisted compression algorithm of federated learning for content caching in edge computing. IEEE Internet Things J 9(16):14151–14161
19. Zhang W, Lu Q, Yu Q, Li Z, Liu Y, Lo SK et al (2020) Blockchain-based federated learning for device failure detection in industrial IoT. IEEE Internet Things J 8(7):5926–5937
20. Sharma PK, Park JH, Cho K (2020) Blockchain and federated learning-based distributed computing defence framework for sustainable society. Sustain Cities Soc 59:102220
21. Wang F, Chen C, Liu W, Qi L, Zhang X, Tan Y et al (2025) Cluster-enhanced dual discrete collaborative filtering for efficient recommendation. IEEE Trans Knowl Data Eng 1–13. https://doi.org/10.1109/TKDE.2025.3624659
22. Pang S, Zhao X, Yu S, Chen J, Deng S, Yin J (2025) TrustPay: a dual-layer blockchain-based framework for trusted service transaction. IEEE Trans Serv Comput 18(2):1068–1080. https://doi.org/10.1109/TSC.2025.3534619
23. Dong N, Sun J, Wang Z, Zhang S, Zheng S (2022) Flock: defending malicious behaviors in federated learning with Blockchain. arXiv preprint arXiv:221104344
24. Li Y, Chen C, Liu N, Huang H, Zheng Z, Yan Q (2020) A blockchain-based decentralized federated learning framework with committee consensus. IEEE Network 35(1):234–241
25. Weng J, Weng J, Zhang J, Li M, Zhang Y, Luo W (2019) Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Trans Dependable Secure Comput 18(5):2438–2455
26. Wang F, Qi L, Liu W, Yu B, Chen J, Xu Y (2025) Inter- and intra- similarity preserved counterfactual incentive effect estimation for recommendation systems. ACM Trans Inf Syst. https://doi.org/10.1145/3722104
27. Liu W, Zhou P, Zhao Z, Wang Z, Deng H, Ju Q (2020) Fastbert: a self-distilling bert with adaptive inference time. ACL
28. Jdmwc K, Toutanova LK (2019) Bert: pre-training of deep bidirectional transformers for language understanding. In Proceedings of naacL-HLT, vol 1. pp 2
29. Jin M, Wen Q, Liang Y, Zhang C, Xue S, Wang X, et al (2023) Large models for time series and spatio-temporal data: a survey and outlook. arXiv preprint arXiv:231010196
30. Chen C, Feng X, Zhou J, Yin J, Zheng X (2023) Federated large language model: a position paper. arXiv preprint arXiv:230708925
31. Li C, Gu B, Zhao Z, Qu Y, Xin G, Huo J et al (2025) Federated transfer learning for on-device LLMs efficient fine tuning optimization. Big Data Min Analytics 8(2):430–446. https://doi.org/10.26599/BDMA.2024.9020068
32. Xu M, Wu Y, Cai D, Li X, Wang S (2023) Federated fine-tuning of billion-sized language models across mobile devices. arXiv preprint arXiv:230813894
33. Chen J, Wang F, Pang S, Chen M, Xi M, Zhao T et al (2025) A privacy policy text compliance reasoning framework with large language models for healthcare services. Tsinghua Sci Technol 30(4):1831–1845. https://doi.org/10.26599/TST.2024.9010089
34. Yu J, Yao H, Ouyang K, Cao X, Zhang L (2025) BPS-FL: Blockchain-based privacy-preserving and secure federated learning. Big Data Min Analytics 8(1):189–213. https://doi.org/10.26599/BDMA.2024.9020053
35. Wenzheng L (2023) The characteristics, relationships and challenges of metaverse, Web 3.0 and AIGC. In 2023 IEEE 13th International Conference on Electronics Information and Emergency Communication (ICEIEC), IEEE p 32–37
36. Liu C, Guo S, Dang F, Qiu X, Shao S (2024) Large-scale model meets federated learning: a hierarchical hybrid distributed training mechanism for intelligent intersection large-scale model. Big Data Min Analytics 7(4):1031–1049. https://doi.org/10.26599/BDMA.2024.9020029
37. Kuang W, Qian B, Li Z, Chen D, Gao D, Pan X, et al (2023) Federatedscope-llm: a comprehensive package for fine-tuning large language models in federated learning. arXiv preprint arXiv:230900363
38. Chen J, Wang F, Pang S, Tan S, Chen M, Zhao T et al (2024) UniGM: unifying multiple pre-trained graph models via adaptive knowledge aggregation. In Proceedings of the 32nd ACM International Conference on Multimedia. MM'24, Association for Computing Machinery, New York, NY, USA, 8556–8565. Available from: https://doi.org/10.1145/3664647.3681018
39. Che T, Liu J, Zhou Y, Ren J, Zhou J, Sheng VS et al (2023) Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In: Confon empirical methods in natural language Processing(EMNLP). Association for Computational Linguistics, Singapore
40. Han S, Buyukates B, Hu Z, Jin H, Jin W, Sun L, et al (2023) FedMLSecurity: a benchmark for attacks and defenses in federated learning and LLMs. arXiv preprint arXiv:230604959
41. Xu M, Du H, Niyato D, Kang J, Xiong Z, Mao S et al (2024) Unleashing the power of edge-cloud generative ai in mobile networks: a survey of aigc services. IEEE Commun Surv & Tutorials

42. Zarrin J, Wen Phang H, Babu Saheer L, Zarrin B (2021) Blockchain for decentralization of internet: prospects, trends, and challenges. Cluster Comput 24(4):2841–2866
43. Rivest RL, Shamir A, Tauman Y (2001) How to leak a secret. In Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold, Springer, Coast, Australia, 552–565. December 9–13, 2001 Proceedings 7
44. Lamport L (2019) The part-time parliament. Concurrency: The Works Leslie Lamport 277–317
45. Lamport L, Shostak R, Pease M (2019) The Byzantine generals problem. Concurrency: The Works Leslie Lamport 203–226
46. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al (2023) Llama: open and efficient foundation language models. arXiv preprint arXiv:230213971
47. Taori R, Gulrajani I, Zhang T, Dubois Y, Li X, Guestrin C et al Stanford Alpaca: an instruction-following LLaMA model. GitHub. https://github.com/tatsu-lab/stanford_alpaca
48. Li A, Sun J, Li P, Pu Y, Li H, Chen Y (2021) Hermes: an efficient federated learning framework for heterogeneous mobile clients. In Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, pp 420–437
49. Li C, Zeng X, Zhang M, Cao Z (2022) PyramidFL: a fine-grained client selection framework for efficient federated learning. In Proceedings of the 28th Annual International Conference on Mobile Computing and Networking, pp 158–171
50. Lin BY, He C, Zeng Z, Wang H, Huang Y, Dupuy C, et al (2021) Fednlp: benchmarking federated learning methods for natural language processing tasks. arXiv preprint arXiv:210408815

Publisher's Note