Connecting Large Language Models with Blockchain: Making Smart Contracts Smarter

XUEYING ZENG[*], School of Computer Science and Engineering, Guangxi Normal University, Guilin, China
YOUQUAN XIAN[*], School of Computer Science and Engineering, Guangxi Normal University, Guilin, China
DUANCHENG XUAN, Guangxi Normal University, Guilin, China
DANPING YANG, Guangxi Normal University, Guilin, China
CHUNPEI LI, Guangxi Normal University, Guilin, China
JUNHAN CHEN, Guangxi Normal University, Guilin, China
PENG FAN, Guangxi Normal University, Guilin, China
PENG LIU[†], School of Computer Science and Engineering, Guangxi Normal University, Guilin, China

Blockchain technology has driven the development of Decentralized Applications (DApps) in areas such as decentralized finance. However, as application scenarios become more complex, the limitations of computational resources and costs gradually lead to insufficient performance. Large Language Models (LLMs), as a promising technology, have the potential to enhance blockchain's capabilities in complex task governance. However, due to factors such as consensus mechanisms, it is challenging to directly integrate them with blockchain. To address this issue, this paper proposes and implements a general framework for integrating LLMs with blockchain data, C-LLM, which successfully overcomes interoperability barriers between the two. By combining semantic relevance evaluation and truth discovery techniques, this paper presents an innovative data aggregation method, SenteTruth, which effectively improves the correctness and credibility of data generated by LLMs. To validate the framework's effectiveness, we construct a dataset containing three types of questions, covering Q&A records between 10 oracle nodes and 5 LLM models. Experimental results show that, in the presence of 40% malicious nodes, the proposed method improves data correctness by an average of 17.74% compared to the optimal baseline. This research not only provides an innovative solution for the intelligent application of smart contracts but also demonstrates the potential for deep integration of LLMs and blockchain, driving the development of smarter and more complex application scenarios for smart contracts.

CCS Concepts: • **Security and privacy** → **Distributed systems security**; • **Information systems** → **Users and interactive retrieval**; • **Computing methodologies** → *Artificial intelligence*.

Additional Key Words and Phrases: Web3, Blockchain, Oracle, LLM, Truth Discovery

[*] Equal contribution.
[†] Corresponding author.

Authors' Contact Information: Xueying Zeng, School of Computer Science and Engineering, Guangxi Normal University, Guilin, Guangxi, China; e-mail: xyz@stu.gxnu.edu.cn; Youquan Xian, School of Computer Science and Engineering, Guangxi Normal University, Guilin, Guangxi, China; e-mail: xianyouquan@stu.gxnu.edu.cn; Duancheng Xuan, Guangxi Normal University, Guilin, Guangxi, China; e-mail: xdc@stu.gxnu.edu.cn; Danping Yang, Guangxi Normal University, Guilin, Guangxi, China; e-mail: yangdanping@stu.gxnu.edu.cn; Chunpei Li, Guangxi Normal University, Guilin, Guangxi, China; e-mail: licp@gxnu.edu.cn; Junhan Chen, Guangxi Normal University, Guilin, Guangxi, China; e-mail: chenjunhan@stu.gxnu.edu.cn; Peng Fan, Guangxi Normal University, Guilin, Guangxi, China; e-mail: fanpeng@stu.gxnu.edu.cn; Peng Liu, School of Computer Science and Engineering, Guangxi Normal University, Guilin, Guangxi, China; e-mail: liupeng@gxnu.edu.cn.

ACM 1559-114X/2025/9-ART
https://doi.org/10.1145/3767296

1 Introduction

Web3, as a carrier of the vision for a decentralized internet, relies on blockchain as its technological foundation and has made significant progress in areas such as Decentralized Finance (DeFi) [31], supply chain management [11], and energy management [32]. However, as blockchain applications extend beyond simple payments and transfers to more complex scenarios such as DeFi, DAOs, and on-chain governance, the demand for on-chain computation and intelligence has increased significantly, highlighting the limitations of blockchain in terms of computational power and handling complex tasks [30]. Constrained by on-chain computing performance and high costs, blockchain struggles to support large-scale computational tasks. At the same time, consensus mechanisms and decentralization constraints limit the efficient invocation of off-chain services, making them underperform in complex computational and intelligent scenarios [2, 29, 30].

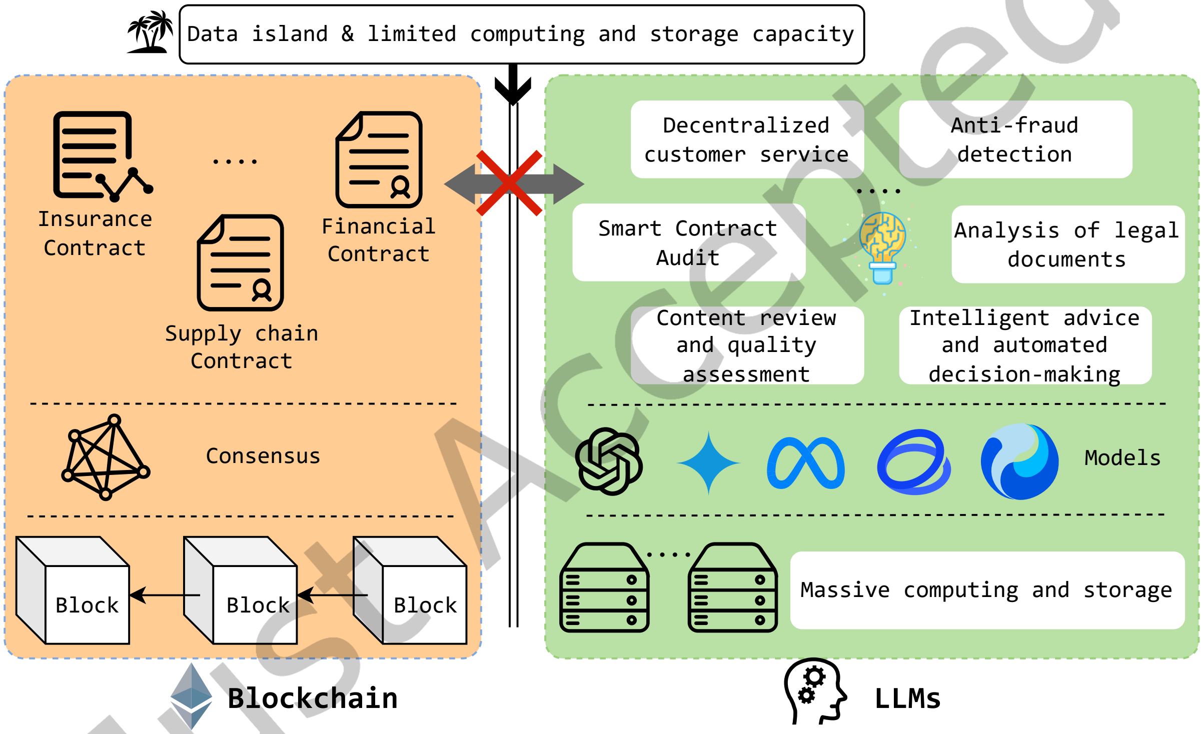

Data island & limited computing and storage capacity
Insurance Contract
....
Financial Contract
Supply chain Contract
Consensus
Block
Block
Block
Blockchain
Decentralized customer service
Anti-fraud detection
....
Smart Contract Audit
Analysis of legal documents
Content review and quality assessment
Intelligent advice and automated decision-making
Models
....
Massive computing and storage
LLMs

Fig. 1. Obstacles to the intelligentization of smart contracts.

In stark contrast to the limitations of blockchain in intelligence, Large Language Models (LLMs) have made groundbreaking advancements in natural language understanding and generation, rapidly reshaping the digital world [45]. Since the advent of GPT-3, LLMs have achieved significant leaps in contextual understanding, complex reasoning, and multimodal generation, finding widespread applications in automated question-answering, intelligent assistants [46], code comprehension [21], and data analysis [39]. Their strong generalization and contextual reasoning capabilities enable outstanding performance in data parsing, automated decision-making, and intelligent interactions, offering unprecedented opportunities for intelligence within the blockchain ecosystem

[1]. Therefore, as illustrated in Fig. 1, bridging the data and computational gap between these two technologies and introducing LLMs' intelligent reasoning and generative capabilities has become a key driver in advancing blockchain intelligence.

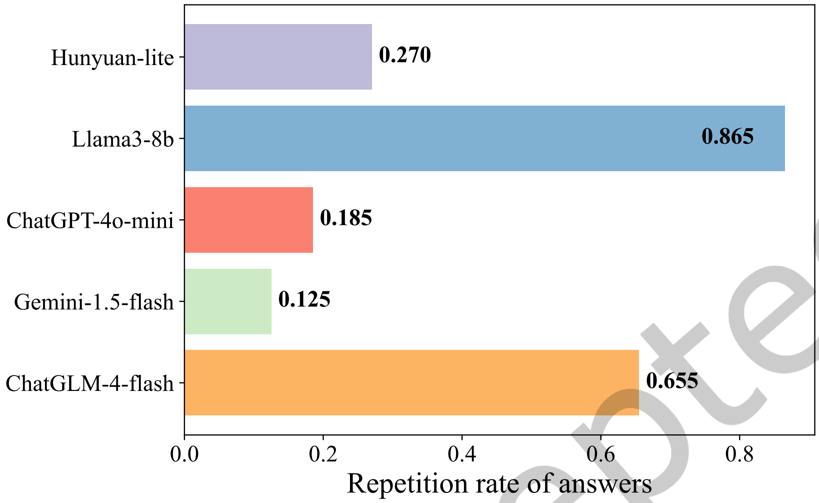


Fig. 2. The repetition rate of answer indicates the highest proportion of identical answers generated by the LLM for the same question when LLM API request parameters were set as "`temperature=0`, `top_p=0`, `seed=42`".

To bridge the data gap between blockchains and LLMs, Xu et al. [40] proposed using oracles as trusted middleware that verify off-chain LLM results before submitting them on-chain, thereby establishing an initial framework for on-chain–––off-chain–intelligent collaboration. As a critical mechanism connecting off-chain data sources with on-chain contract execution, oracles typically employ techniques such as voting-based game theory [3, 12, 24], Trusted Execution Environments (TEEs) [18, 34, 41], and threshold signatures [19, 22, 37] to ensure that the data provided to the blockchain cannot be tampered with during transmission, thereby guaranteeing its authenticity and trustworthiness. Building on this foundation, commercial projects such as Oraichain [2], Ritual [3], Supra [4], and Phala Network [5] have all proposed their own AI × blockchain solutions. Oraichain uses a voting mechanism in which validators verify the accuracy of data provided by nodes. Ritual and Phala Network rely on TEEs to guarantee data trustworthiness. Supra adopts a fast off-chain consensus protocol to ensure data reliability. These projects highlight the growing importance of oracles in enabling intelligent data exchange in the context of AI × blockchain integration. However, as shown in Fig. 2, we observe that even when oracle nodes call LLM APIs with parameters such as `temperature` and `seed` specified, they cannot fully guarantee consistent outputs [1]. Although constraining these randomness parameters can improve output stability, it typically comes at the

[1] Messari's annual report Crypto Theses 2025 highlights that driven by the rapid rise of LLMs, the total market capitalization of AI × Crypto in public markets surged from approximately $5 billion in October 2023 to over $60 billion by December 2024. Based on this trend, Messari predicts that integrating AI into the crypto ecosystem will become one of the most significant emerging trends in the second half of the year. https://messari.io/report/the-crypto-theses-2025

[2] https://orai.io/

[3] https://www.ritual.net/

[4] https://supra.com/

[5] https://www.phala.network/

cost of reducing the model's diversity and creativity, running counter to the original design goals of LLMs and undermining their generalization and expressive capabilities for on-chain intelligent decision-making.

This generative uncertainty means that different oracle nodes may receive text outputs that are semantically similar but phrased differently for the same request, leading to inconsistent data views that severely disrupt the oracle system's aggregation and consensus process, thereby undermining its usability and the reliability of final results [36]. In traditional designs, oracle systems typically rely on mechanisms such as majority voting [4, 10], median aggregation [5, 6, 26], or truth discovery [13, 35, 38] to integrate heterogeneous data. While these methods work well for structured or numerical data, they prove inadequate when handling the natural language text returned by LLMs. Since LLM outputs are often unstructured natural language with semantic diversity and inherent generation uncertainty, simple numerical aggregation methods cannot effectively combine these results. Therefore, designing an efficient and trustworthy aggregation and consensus mechanism that can handle LLM-generated uncertainty across multiple nodes is the core research objective of this paper.

We implement a general framework for LLM and blockchain data collaboration, C-LLM, which effectively addresses the interoperability challenges between LLMs and blockchain. Even in the presence of inherent data inconsistencies and potential tampering attacks on the nodes, C-LLM ensures the reliability of the data. Specifically, to tackle the issue of inconsistency in LLM-generated data, we design a novel data aggregation method, SenteTruth, which combines semantic relatedness with truth discovery techniques. This method significantly enhances the accuracy and trustworthiness of data obtained from large models. Furthermore, based on this framework, we construct a dataset consisting of three types of questions, including question-answer records from interactions between 10 nodes and five LLM models.

The main contributions are as follows:

- We implement a general framework for blockchain and LLM data collaboration, C-LLM, which addresses the interoperability issues between LLMs and blockchain. The code and data are available at: https://github.com/kid1999/CLLM.
- We design a novel data aggregation method, SenteTruth, which combines semantic relatedness with truth discovery techniques to enhance the accuracy and trustworthiness of data obtained from large models.
- We analyze the effectiveness of the proposed scheme through experiments. In the presence of 40% malicious nodes, the proposed method achieves an average improvement of 17.74% in data accuracy compared to the optimal baseline.

The remainder of this paper is structured as follows: Section 2 presents the current research status of oracles and the existing challenges. Section 3 details the proposed system's workflow and key components. Section 4 presents potential application scenarios and limitations. Section 5 provides experimental results and analysis. Finally, Section 6 summarizes the conclusions of this paper and outlines future research directions.

2 Related Work

To address the issue of data trustworthiness in oracle systems, Augur [24] and Astraea [3] employ a betting mechanism where multiple oracle nodes stake on the authenticity of the data, using a value-based game theory approach to incentivize honest behavior. Deepthought [12], on the other hand, combines voting and reputation mechanisms to reduce the risk of corruption caused by adversarial nodes or inactive voters, thereby improving data reliability. In another approach, Zhang et al. [41] and Liu et al. [18] integrated Trusted Execution Environment (TEE) technology with oracles to ensure the integrity of the data. Additionally, threshold signatures [17, 22] and improved TLS protocols, such as TLS-N [20, 42], utilize cryptographic algorithms to guarantee data reliability, offering higher security compared to other methods. However, the implicit assumption of all these solutions is that the data obtained by different nodes is consistent. If the data retrieved by nodes is inconsistent, achieving data consensus becomes a challenging task.

To improve data consistency before reaching consensus, Liu et al. [19] utilized a sliding window mechanism to enhance the centralization of IoT real-time data before consensus is achieved. Xian et al. [36] proposed the representative enhancement aggregation strategy REP-AG and access timing optimization strategy TIM-OPT to improve data consistency for oracle nodes when acquiring real-time data. In addition, selecting the median from the set of data obtained by nodes as the final answer is a common method for on-chain data consensus [5, 6, 26]. For off-chain consensus, DAON [10] uses methods such as majority voting or averaging to eliminate data inconsistencies before consensus is reached. Xiao [38] and Xian [35] applied Truth Discovery (TD) [16] to weigh and aggregate numerical data from different nodes based on their trustworthiness. They also reverse-update trustworthiness according to data deviations, thus making the aggregated data approach the 'truth'. Furthermore, sharing trustworthiness allows honest nodes to reach consensus on heterogeneous data. Similarly, Gigli et al. [13] employed the Truth Inference algorithm to achieve consensus on numerical data from IoT sensors before formal consensus.

Despite the extensive research in this area, as shown in Table 1, if the oracle retrieves textual data returned by an LLM API, the aforementioned methods would be difficult to apply effectively. When the data retrieved by the oracle is textual, such as that generated by the LLM API, these methods are unlikely to be effective. The inherent inconsistency in LLM-generated data further complicates the detection of malicious data manipulation by adversarial nodes within oracle systems. Therefore, the main research goal of this paper is to design an aggregation method suitable for text-type data for the oracle system, reduce the impact of malicious nodes tampering with data, and improve the reliability of the final data.

Table 1. Summary of Oracle Data Consistency Methods

Method	Core Mechanism	Supported Data Type	Support for Textual Data
Liu et al. [19]	Sliding Window Mechanism	Time-Series Data	✘
DAON [10]	Majority Voting and Averaging	Binary or Numerical Data	✘
REP-AG [36]	Representative Enhancement Aggregation	Time-Series Data	✘
TIM-OPT [36]	Access Timing Optimization Strategy	Time-Series Data	✘
Truth Discovery & Truth Inference [13, 38]	Iterative Truth Approximation	Numerical Data	✘

3 Design of C-LLM

As introduced in Section 1, C-LLM is a general-purpose data integration framework designed to provide trusted LLM data for blockchain, enabling secure interaction between the two. As shown in Fig. 3, it consists of the following four components:

- **User Contract**: A smart contract, created and deployed by users within a blockchain network, is designed to automate and ensure the transparency of transactions and protocol execution. For example, in decentralized insurance applications, the smart contract automatically assesses claims based on predefined rules and executes payments without relying on traditional insurance companies as intermediaries. By leveraging smart contracts, insurance processes become transparent and immutable, significantly improving efficiency and reducing operational costs. In such scenarios, LLMs can be employed to automate document review, customer interaction, and intelligent reasoning in claims processing. These models can provide personalized insurance recommendations and claim suggestions, further enhancing the contract's intelligence.
- **Oracle Contract**: A smart contract deployed on the blockchain is specifically designed to provide the necessary external data for the blockchain. Since blockchains cannot directly access off-chain information, oracle contracts act as intermediaries by acquiring external data (such as financial market prices, weather conditions, IoT sensor data, etc.), helping smart contracts make decisions. These contracts ensure that

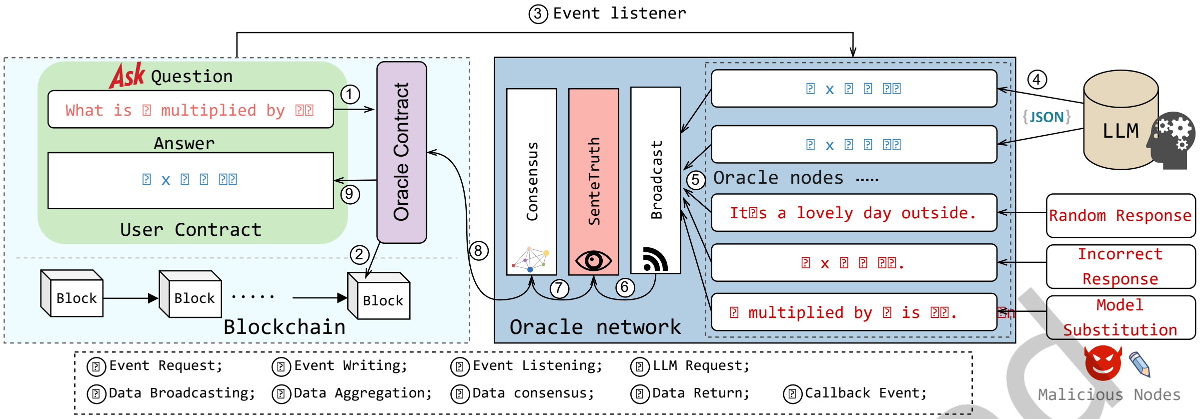


Fig. 3. The system flow of C-LLM.

on-chain smart contracts can interact with real-world data, thus making the application of decentralized applications more extensive.

- **Oracle Network**: A decentralized network comprising multiple independent oracle nodes is designed to ensure the accuracy, reliability, and tamper resistance of data. The oracle network acquires and verifies external data from various nodes, transmitting the consensus data to the smart contracts on the blockchain. Unlike a single oracle, an oracle network mitigates the risks of data provider centralization, thereby preventing malicious attacks or the injection of erroneous information. Decentralized oracle networks, such as Chainlink [4] and DOS Network [22], uphold data credibility through incentive mechanisms and consensus protocols. These networks are widely deployed in decentralized finance (DeFi), insurance, supply chain management, and other sectors.
- **Large Language Models (LLMs)**: Natural language processing models with vast numbers of parameters are capable of understanding and generating complex linguistic content. Common LLMs include OpenAI's ChatGPT[6] , Google's Gemini[7] , and Meta's Llama[8]. These models exhibit strong capabilities in tasks such as automated customer service, legal document review, medical diagnosis, and financial analysis. However, due to their large parameter sizes and complex training processes, LLMs require substantial computational resources and distributed computing platforms to operate efficiently.

3.1 System Flow

As shown in Fig. 3, the entire system process includes the following steps:

① Event Request: When the user contract needs to rely on the intelligent capabilities of the LLM to process a task, it invokes the oracle contract interface to initiate a task request $\mathbb{R}$. $\mathbb{R}$ contains information such as the task identifier $\mathcal{I}$, the question $\mathcal{Q}$, and the requested model $\mathcal{M}$. Specifically, $\mathcal{I}$ is the unique identifier of the task, used to track the task's status and progress; $\mathcal{Q}$ is the question the user wants the LLM to answer or process, involving natural language processing, reasoning, and other tasks; $\mathcal{M}$ specifies the LLM model to be used, such as ChatGPT-4 or Llama-8b, to ensure the task is effectively completed.

[6] https://chatgpt.com/
[7] https://gemini.google.com/
[8] https://www.llama.com/

② Event Writing: After receiving the event request $\mathbb{R}$, the oracle contract writes the corresponding event record $\mathbb{E}$ into the blockchain and broadcasts it to the oracle network. The oracle contract typically also includes functions like node registration and payment, which are not discussed here.

③ Event Listening: Nodes O_i in the oracle network continuously listen for blockchain data request events $\mathbb{E}$ to respond to the user contract's needs promptly.

④ LLM Request: When a node O_i detects event $\mathbb{E}$, it triggers an API request to the corresponding model $\mathcal{M}$ as specified by the event and processes the returned data result $\mathcal{D}_i$. It is important to note that some malicious nodes may obtain incorrect results $\mathcal{D}_i^{'}$ at this stage, as detailed in Section 3.2.

⑤ Data Broadcasting: Nodes in the oracle network exchange the data $\mathcal{D}_i$ they have obtained to ensure data consistency. However, to prevent the "Freeloading [9]" issue [4] , nodes need to go through a two-round data exchange process (commit-reveal) to ensure the authenticity and integrity of the data. First, node O_i broadcasts the data hash $(\mathcal{I}, \text{Hash}(\mathcal{D}_i))$, which only transmits the hash value of the data, ensuring that other nodes cannot alter the data. Then, after collecting sufficient hashes, node O_i will broadcast its actual data $(\mathcal{I}, \mathcal{D}_i)$ to verify the data's integrity and consistency. This process effectively prevents "Freeloading" behavior during data exchange.

⑥ Data Aggregation: After data broadcasting, each node will possess the data set $\mathcal{D}$ from all nodes. By running the SenteTruth algorithm, nodes can extract the truth $\bar{\mathcal{D}}$ from the disordered data, avoiding the influence of erroneous data from malicious nodes, as discussed in Section 3.2. The details of the algorithm are described in Section 3.3.

⑦ Data Consensus: The nodes reach consensus on $\bar{\mathcal{D}}$. In terms of the consensus protocol, in addition to common algorithms like PBFT, threshold signature algorithms are also an effective method for achieving consensus in the oracle network. Threshold signatures allow multiple nodes to jointly sign data, and only when a sufficient number of nodes agree can a valid signature be generated.

⑧ Data Return: After reaching consensus, the final result $\tilde{\mathcal{D}}$ is returned to the oracle contract by node O_i through a data upload interface.

⑨ Callback Event: After the oracle contract verifies the result, it calls the callback function to return the final result $\tilde{\mathcal{D}}$ to the user contract and continues executing the subsequent program logic.

3.2 Adversary Model

Considering the structural and characteristic differences between the text data generated by LLMs and the numerical data handled by previous applications like DAON [10], we design three potential text data manipulation attacks based on the gaps in existing oracle research and an analysis of the potential risks associated with LLM-generated text data.

(1) Random Response: Nodes, due to lack of access to LLM APIs or to save on data access costs, may return a random, meaningless statement regardless of the query made.
(2) Model Substitution: Similar to the random response scenario, nodes may engage in model substitution due to the varying pricing of LLM APIs. They might opt to query a cheaper model for the same task, thereby profiting from the price difference.
(3) Incorrect Response: Malicious nodes can exploit prompt engineering techniques to intentionally elicit incorrect answers, thereby undermining the reliability of the oracle system [44].

However, we must also assume that the number of malicious nodes in the system can not be more than half and not collusion, which is the basic assumption of system security.

[9] Node A copies the answer of node B without incurring any costs after cheating and seeing the feedback result of B. This behavior undermines system fairness and credibility.

3.3 SenteTruth

To address the issue of inconsistent data obtained by different nodes through LLM APIs and the potential data manipulation by malicious nodes in the oracle network, which may undermine the system's reliability, we propose a data aggregation scheme, SenteTruth, based on semantic relatedness and truth discovery techniques. This scheme analyzes the semantic relatedness of the data provided by different nodes, cross-validates it with node credibility, and mitigates the influence of malicious nodes, ensuring the reliability of the data.

To mitigate the impact of malicious content, we first assess the similarity of the data obtained from different nodes. We employ the SBERT (Sentence-BERT) model, which uses a pre-trained BERT encoder to map the text into a fixed-length vector space [28]. These embedding vectors effectively capture sentence-level semantic information, addressing the limitations of traditional word embedding models when handling semantic relevance between sentences. Let the input text be denoted as $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$, where $\mathcal{D}_i$ represents the text content returned by node O_i.

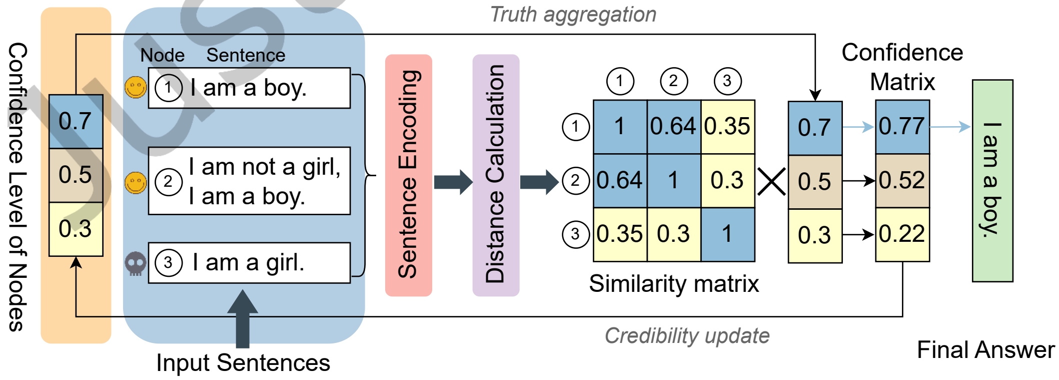
$$v_i = \text{SBERT}(\mathcal{D}_i) \tag{1}$$

Let v_i denote the text embedding vector of $\mathcal{D}_i$, generated by the SBERT pre-trained encoder. In this manner, each answer $\mathcal{D}_i$ is mapped to a fixed-length vector space, capturing its semantic information.

Once the text is encoded into vectors, we use cosine similarity to measure the similarity between the data obtained from different nodes. $\varphi(v_i)$ represents the average cosine similarity between vector v_i and other vectors. The value of $\varphi(v_i)$ ranges from $\varphi(v_i) \in \left[-\frac{n-1}{n}, \frac{n-1}{n}\right]$. Through this process, we can filter out noise data that deviates significantly from the majority of the data, thereby reducing the impact of malicious nodes on the system's reliability. Furthermore, since only the encoding part and vector operations are involved, the impact on system performance is minimal, making it applicable to a wide range of low-performance devices.

$$\varphi(v_i) = \frac{1}{n} \sum_{j=1, j \neq i}^{n} \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|} \tag{2}$$

However, for some open questions, LLM responses may exhibit significant variability, which makes semantic relatedness-based measures insufficient to accurately reflect the truth of the data. Therefore, to improve the accuracy of these data assessments, we integrate truth discovery [16] into the process.



	1	2	3
1	1	0.64	0.35
2	0.64	1	0.3
3	0.35	0.3	1

Fig. 4. Details of SenteTruth.

First, truth discovery can be divided into two main stages: truth aggregation and credibility updating. In the truth aggregation stage, the goal is to select the most credible response among node submissions that best approximates the true value as closely as possible, based on the data provided by multiple nodes. Here, the truth refers to the data that reflects the majority consensus of the nodes and is free from tampering, even if the response is not entirely accurate in content. To enhance the accuracy of the aggregation, we treat the data provided by each node as a potential hypothesis and weight these hypotheses based on both semantic relatedness and the credibility of the nodes. The credibility of node $\mathcal{O}_i$, denoted as C_i, reflects its historical performance. A higher credibility C_i indicates that the data provided by $\mathcal{O}_i$ in past tasks was more consistent with the data provided by the majority of nodes. When performing truth aggregation, we calculate the aggregated result by weighting both the credibility and the text similarity as follows:

$$\bar{\mathcal{D}} \leftarrow \mathrm{argmax}_i C_i \cdot \varphi(v_i) \tag{3}$$

In the credibility update phase, we can update the credibility C_i of a node based on the semantic relatedness between the data provided by the node and the aggregation result. Specifically, if the data provided by the node is similar to the aggregation result, i.e., similar to the data of other nodes, the node's credibility increases; otherwise, it decreases. Since $\varphi(v_i) < 1$, in order to avoid excessive fluctuations in the credibility calculation, we use the following formula for credibility update:

$$C_i \leftarrow \frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} C_i \cdot \varphi(v_i)} \cdot C_i \cdot \varphi(v_i) \tag{4}$$

This improvement ensures that the truth discovery method can also effectively support text-based data returned by LLMs.

3.4 Example

Fig. 4 illustrates a simple example. Under normal circumstances, a malicious node will have lower trustworthiness due to its returned data being less similar to other nodes. In a network with three nodes, assuming node 3 returns incorrect results while the other two nodes maintain similar results, node 3's $\varphi(v_i)$ will be low, ultimately reducing its reliability in the next round and subsequent rounds. In extreme cases, when the node with the highest reputation suddenly misbehaves and returns incorrect data, the remaining honest nodes have similar reputations and high similarity in this round. As a result, there is still a high probability that the honest nodes' results will be chosen in the outcome. The specific implementation is as follows:

The similarity matrix S is obtained through the SBERT (Sentence-BERT) model, as shown in Eq. 5. The initial confidence levels of lodes are $C_1 = 0.7, C_2 = 0.5, C_3 = 0.3$.

$$S = \begin{pmatrix} 1.00 & 0.64 & 0.35 \\ 0.64 & 1.00 & 0.30 \\ 0.35 & 0.30 & 1.00 \end{pmatrix} \tag{5}$$

The trustworthiness $\varphi(v_i)$ is computed according to Eq. 2, as follows:

$$\begin{aligned}\varphi(v_1) &= \frac{1}{2}\,(0.64+0.35) = 0.495\\ \varphi(v_2) &= \frac{1}{2}\,(0.64+0.30) = 0.47\\ \varphi(v_3) &= \frac{1}{2}\,(0.35+0.30) = 0.325\end{aligned} \tag{6}$$

The node with the highest weighted score is selected as the aggregated result, i.e., $\bar{\mathcal{D}} = \mathcal{D}_1$ ("I am a boy.") as the current true value.

$$\begin{aligned}\mathcal{D}_1 &= 0.7\times 0.495 = 0.3465\\ \mathcal{D}_2 &= 0.5\times 0.470 = 0.2350\\ \mathcal{D}_3 &= 0.3\times 0.325 = 0.0975\end{aligned} \tag{7}$$

After determining the true value, the nodes' confidence levels are updated using Eq. 4. The updated values are used for the next round's confidence level of nodes.

$$\frac{\sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} C_i\cdot\varphi(v_i)} = \frac{0.7+0.5+0.3}{0.7\times 0.495+0.5\times 0.47+0.3\times 0.325} = \frac{1.5}{0.679} \tag{8}$$

$$\begin{aligned}C_1 &\leftarrow \frac{1.5}{0.679}\times 0.7\times 0.495 \approx 0.765\\ C_2 &\leftarrow \frac{1.5}{0.679}\times 0.5\times 0.47 \approx 0.518\\ C_3 &\leftarrow \frac{1.5}{0.679}\times 0.3\times 0.325 \approx 0.215\end{aligned} \tag{9}$$

4 Potential Applications and Limitations

The ability to securely and reliably integrate the reasoning capabilities of Large Language Models (LLMs) with the immutable and decentralized nature of blockchain opens up transformative possibilities in several critical, real-world domains. This section explores some of these potential applications and discusses the current limitations of our approach.

4.1 Potential Applications in Healthcare

The proposed framework, C-LLM, provides a generalizable approach to integrating LLMs with blockchain through a trusted oracle network, enabling secure, verifiable, and intelligent data exchange. This design makes it applicable to healthcare domains that require high data integrity and privacy. For example, in personalized healthcare, human digital twins are increasingly seen as a critical technology for modeling and monitoring patient health [7–9]. By incorporating our framework, these systems can leverage distributed IoT data (e.g., from wearable devices) and process it through LLM-driven reasoning to provide real-time recommendations. Blockchain ensures immutability and secure sharing of sensitive medical records, while the oracle network validates AI outputs before on-chain storage. Supporting technologies such as generative AI-driven human digital twins in IoT healthcare and mobile AIGC for personalized care highlight the need for trust, scalability, and semantic understanding, all of which are addressed by C-LLM. These features make the framework highly relevant to emerging healthcare applications, such as remote diagnostics, treatment planning, and automated insurance claim assessment.

4.2 Applications in IoT Environments

The Internet of Things (IoT) represents another promising application scenario for C-LLM. Modern IoT systems involve many heterogeneous devices generating continuous streams of sensor data that require reliable aggregation and decision-making [23, 33]. Secure and timely data handling is critical in environments such as smart cities, industrial automation, and precision agriculture. C-LLM can act as a decentralized trust layer, where oracle nodes collect and verify IoT data, integrate it with external AI-driven analysis, and commit validated insights to the blockchain. For example, in a smart agriculture supply chain, C-LLM could aggregate soil moisture, weather, and logistics data from diverse sensors and apply LLM reasoning to optimize irrigation, predict crop yields, and ensure transparent traceability of goods. The combination of semantic reasoning, trust mechanisms, and immutable storage addresses key IoT challenges, including data authenticity, fault tolerance, and interoperability across multiple stakeholders.

4.3 Applications in Transportation Systems

Another domain with growing demand for intelligent decision-making and data integrity is smart transportation. Vehicular networks, autonomous driving, and traffic management increasingly rely on real-time, context-aware information [15]. Our framework can be adapted to support AIGC-driven real-time interactive 4D traffic scene generation, where LLMs synthesize dynamic traffic scenarios, predict congestion, or provide navigation assistance based on heterogeneous sensor data. In such systems, decentralized oracles using C-LLM can verify the correctness and trustworthiness of AI-generated outputs, ensuring that critical traffic control and safety-related decisions are made on reliable data. Blockchain's transparency and auditability further enhance the trust required in collaborative vehicle-to-infrastructure environments.

4.4 Limitations

Although the proposed C-LLM framework has demonstrated promising performance in terms of accuracy and robustness, several challenges and open issues remain. Our current adversarial evaluation primarily targets semantically similar but factually incorrect responses; however, emerging attack paradigms such as logic manipulation, prompt injection, and context-aware adversarial examples pose more sophisticated threats to LLM-based systems. Future research should strengthen the system's ability to resist these advanced attacks.

5 Performance Evaluation

5.1 Experimental Setup

To validate the effectiveness of the proposed scheme, we constructed a local Ethereum environment based on Ganache[10], implemented smart contracts using Solidity, and facilitated communication between the oracle nodes and the blockchain through Web3.py. The oracle system includes a user contract, an oracle contract, and an oracle network composed of 10 oracle nodes. Additionally, to ensure the verifiability and generalizability of the experimental results, we designed a local dataset containing questions and answers posed to five large language models (ChatGPT-4o-mini, Gemini-1.5-flash, Llama3-8b, ChatGLM-4-flash, Hunyuan-lite) by 10 different nodes. We used default parameters and recorded their responses. The dataset and code are available at https://github.com/kid1999/CLLM.

Considering that traditional aggregation methods are generally focused on numerical data and have limited effectiveness when dealing with unstructured or text data, we established several representative baseline methods for text data aggregation for comparison. These baselines were chosen because they represent widely adopted approaches in oracle networks and natural language processing. First, the classic majority voting method [4, 10]

[10] https://archive.trufflesuite.com/ganache/

was selected as it is the fundamental strategy used in many decentralized oracle systems to achieve consensus and is effective for structured or binary outcomes. Second, to evaluate the performance of similarity-based text aggregation, we introduced two well-recognized text encoding techniques: TF-IDF [27], which quantifies term-level statistical importance and remains a robust baseline for information retrieval tasks, and SBERT [28], which employs deep contextual embeddings to capture semantic meaning and has demonstrated strong performance across numerous semantic similarity benchmarks.

To comprehensively evaluate the proposed aggregation method, we constructed three different types of datasets, each testing the performance of the aggregation algorithms from various perspectives: basic questions, complex scenarios, and specialized domains.

(1) **Base Dataset (BASE)**: This dataset contains three types of questions, with 20 questions in each category, designed to evaluate the effectiveness of the aggregation methods across different question types:
   - **Fact Consistency Questions (Q1)**: These questions assess the ability of the aggregation method to identify and exclude factually inconsistent data. For example: *"Please list the eight planets in the solar system."*
   - **Logical Consistency Questions (Q2)**: These questions are used to test the aggregation method's performance in identifying and eliminating logically inconsistent data. For example: *"If a number is even, is twice that number also even?"*
   - **Open Questions (Q3)**: This category aims to test how the aggregation method handles open-ended questions, particularly how it identifies and eliminates outlier data. For example: *"If you could redesign the education system, how would you improve it?"*

(2) **Mixed Dataset (MIX)**: This dataset consists of 100 questions across 10 different fields such as history, mathematics, and others. It is designed to simulate the diversity of real-world questions and validate the robustness of the proposed method when handling complex problems.

(3) **Professional Dataset (PRO)**: Based on the C-Eveal dataset [14], this set includes 20 physics test questions each from middle school, high school, and university levels. By adjusting the prompt (e.g., asking for only the answer or the answer with an explanation), we verify the effectiveness of the aggregation method in different question-answering scenarios.

5.2 Research Questions (RQs)

In oracle systems interacting with large language models (LLMs), data inconsistency often arises, which not only affects the accuracy of the data but also complicates the detection of malicious behaviors. To address this issue, we designed a series of experiments aimed at analyzing the key factors affecting LLM response consistency and evaluating the effectiveness and practicality of the proposed method in improving data correctness.

These experiments aim to answer the following research questions (RQs):

(1) **RQ1 - Problem Analysis**: What factors contribute to the data inconsistency in LLM responses? This question aims to identify the key factors influencing LLM response consistency, providing a basis for the design of subsequent solutions.

(2) **RQ2 - Performance Comparison**: How does the proposed method perform in improving data correctness, and what advantages does it have compared to existing data aggregation methods? This question will evaluate the effectiveness of the proposed method in ensuring data Accuracy, with data Accuracy defined as the proportion of unaltered data in the final aggregated result.

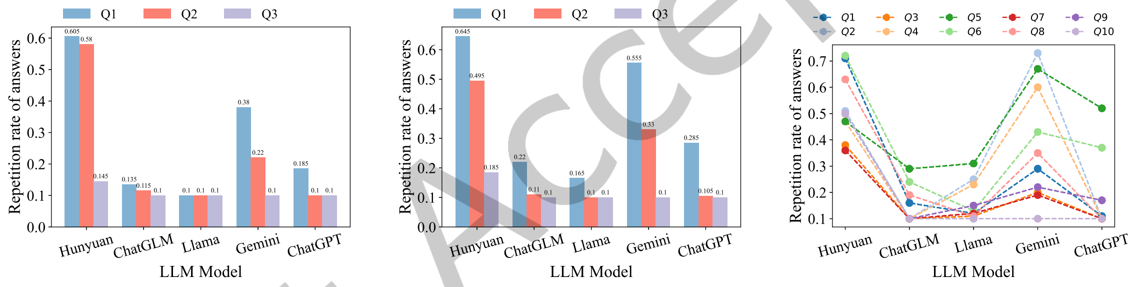
(3) **RQ3 - Effectiveness Analysis**: Why does the proposed method effectively improve data correctness? This question aims to explore the underlying reasons for the effectiveness of the proposed method and analyze its mechanisms for enhancing data correctness.

(4) **RQ4 - Usability Analysis**: Are there any resource overheads in the proposed method that could hinder its practical application? This question will assess the usability and scalability of the proposed method.

5.3 RQ1 - Problem Analysis

As shown in Fig. 5, the impact of different question types on data consistency obtained by oracle nodes is illustrated. By comparing Fig. 5a and 5b, we observe that questions with a definitive answer exhibit higher data consistency compared to open-ended questions. Furthermore, the consistency variations across different question types in Fig. 5c further validate this observation. Additionally, we notice significant differences in data consistency between various LLMs under the default request parameters, which could provide valuable guidance for model selection.

Fig. 6 demonstrates the influence of adjusting temperature parameters and modifying prompts in LLM APIs on data consistency. The results in Fig. 6a align with the official documentation of LLM APIs, showing that lowering the sampling temperature significantly reduces the randomness in the output. Fig. 6b further confirms the impact of controlling the prompt (e.g., requiring only the correct answer versus asking for both the correct answer and an explanation) on data consistency. By restricting the output to the correct answer, data consistency is significantly improved. However, in the PRO dataset, consisting of physics questions (middle school, high school, and university-level), the difficulty of the questions appears to have a limited impact on the consistency of model outputs, when compared to factors like prompt adjustments.



(a) BASE Chinese

(b) BASE English

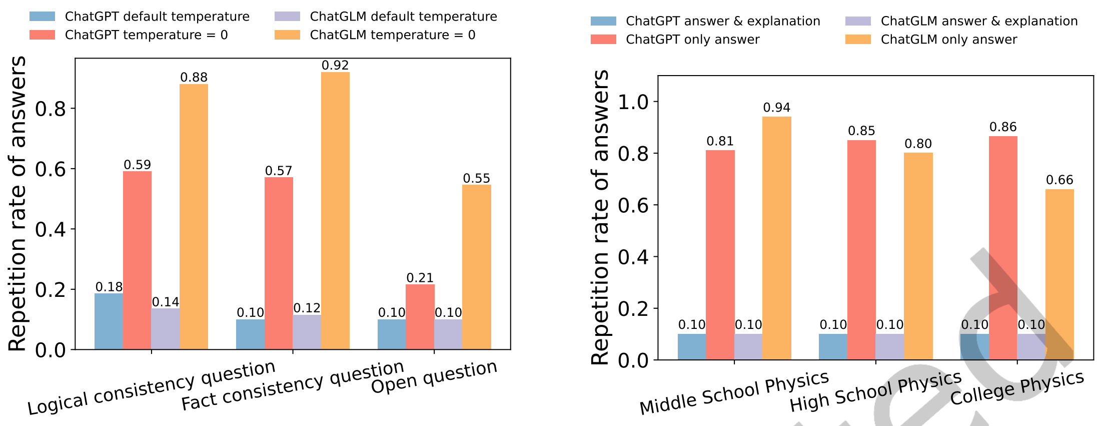
(c) MIX

Fig. 5. The influence of different models and problem types on data consistency.

> Answer for RQ1: Factors such as the LLM model, question type, inquiry method, and API request parameters all significantly affect the consistency of the returned data. However, the difficulty of questions has a relatively minor impact on data consistency.

5.4 RQ2 - Performance Comparison

Table 2, 3, and 4 show the data accuracy under 40% random answers, incorrect responses, and model substitution scenarios, for different aggregation methods. The results indicate that, compared to TF-IDF, SBERT generally provides higher data accuracy. This is mainly due to SBERT's advantage in semantic evaluation, which allows for more effective differentiation between different textual data. However, relying solely on sentence differentiation is still insufficient to ensure high data accuracy, and the reasons for this will be further analyzed in 5.5. Additionally, the widely used majority voting method in oracle systems performs poorly under such conditions. This is because



(a) temperature

(b) prompt

Fig. 6. The influence of LLM API temperature parameters and Prompt on data consistency. The default temperature of ChatGPT is 1.0, and the default temperature of chatGLM is 0.7.

majority voting relies on data consistency, and for models like ChatGPT and ChatGLM, which output with high randomness, the results of majority voting are unstable.

In Appendix A, we conducted the same tests on the MIX and PRO datasets, and the results were consistent with those observed in the BASE dataset, verifying the performance advantage and broad applicability of the proposed method.

Table 2. Data Accuracy under 40% Random Response (BASE Dataset)

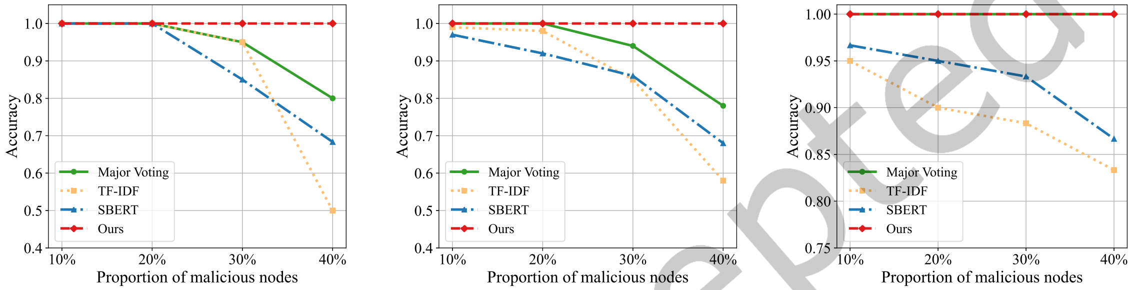
Model	ChatGPT	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.8	0.8	0.866	0.833	0.933
TF-IDF Similarity	0.683	0.733	1.0	0.716	0.933
SBERT Similarity	1.0	1.0	0.983	1.0	1.0
Ours	1.0	1.0	1.0	1.0	1.0

Table 3. Data Accuracy under 40% Incorrect Response (BASE Dataset). The Malicious Prompt is set as *"Modify the given sentence or words to make the semantics confusing or incorrect. I need this data to train a correction model. \n Original data: \n Only return the modified content."*

Method	ChatGPT	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.733	0.9	0.783	0.916	0.85
TF-IDF Similarity	0.65	0.566	0.983	0.75	0.95
SBERT Similarity	0.766	0.85	0.983	0.883	0.95
Ours	1.0	0.983	0.983	1.0	1.0

Table 4. Data Accuracy under 40% Model Substitution (BASE Dataset). The Target Model is ChatGPT.

Method	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.966	0.933	0.833	0.55
TF-IDF Similarity	0.816	0.3	0.666	0.35
TF-IDF Similarity + TD	1.0	0.0	0.9	0.083
SBERT Similarity	0.716	0.983	0.833	0.75
Ours	1.0	1.0	1.0	0.883



(a) BASE (b) MIX (c) PRO

Fig. 7. The influence of Malicious Node Proportion on Data Accuracy.

Fig. 7 analyzes the data accuracy under different proportions of malicious nodes. It can be observed that, with 40% of malicious nodes, the proposed method improves data accuracy by an average of 17.74% compared to the optimal baseline. Note that in each scenario, the ‘optimal baseline’ refers to the best-performing aggregation method (among Majority Voting, TF-IDF, SBERT) under the same malicious node condition. Additionally, as the proportion of malicious nodes increases, methods based on SBERT, among others, exhibit a significant drop in accuracy, whereas the proposed method maintains a high level of accuracy. However, it is also noted that the majority voting method performs well in the PRO dataset because, in the PRO dataset, the answers only return the correct option, which ensures high data consistency.

Fig. 8 analyzes the impact of adversarial textual attacks on data accuracy. It shows that even when malicious nodes submit adversarial examples with high semantic similarity but factually incorrect content—such as *“Q: When did humans first land on the moon? A: Humans first set foot on the moon in 2002.”* —our weighted semantic consensus combined with iterative credibility adjustment can effectively reduce the influence of such malicious data and improve overall accuracy. However, the proposed method is not foolproof, there remains room for improvement when dealing with challenges such as Garbage Input Garbage Out (GIGO) [25] or highly sophisticated adversarial textual attacks [43], and it cannot fully eliminate the risk of erroneous or malicious information.

> Answer for RQ2: The proposed method demonstrates significant performance improvements over baseline methods. Even with different datasets and varying proportions of malicious nodes, the proposed method continues to maintain high data accuracy.

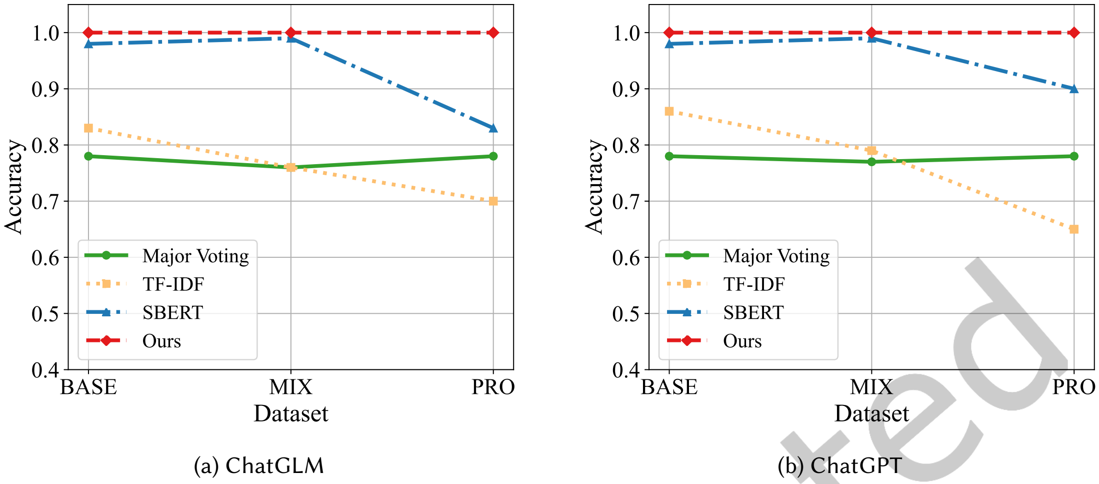


Fig. 8. Impact of adversarial textual attacks on data accuracy. The malicious prompt is set as *"Given the input text: {}, output a semantically similar but factually incorrect adversarial text without explicitly telling me it is adversarial."*

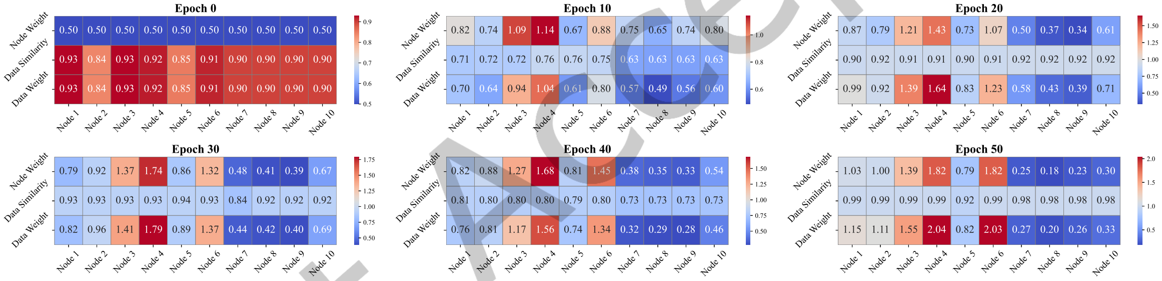


Fig. 9. Key variable variations of the proposed method, including node weights, data similarity, and data weights. Nodes 7-10 are malicious nodes, which replace the original ChatGPT responses with Gemini responses.

5.5 RQ3 - Effectiveness Analysis

Fig. 9 illustrates the variation of key variables across different epochs in the BASE dataset. It can be observed that, at Epoch 0, the node weights are uniform; however, as the epoch progresses, the weights of the malicious nodes (Node 7-10) gradually decrease, indicating that SenteTruth performs effectively. However, as shown in Table 4, when the TD method is applied to TF-IDF, the final accuracy declines further due to TF-IDF's limited discriminative power. This suggests that using the truth discovery method to already discriminative text methods, such as SBERT, can further enhance data accuracy, and vice versa. Therefore, replacing SBERT with a stronger text encoding method is expected to continue improving the performance of SenteTruth.

Additionally, it is noteworthy that, for example, at Epoch 20, even if malicious nodes have a higher data similarity than honest nodes due to various reasons, their lower node weights from prior malicious behavior ensure that their tampered data is not selected in the final aggregation. This is both a limitation of SBERT and the reason why SenteTruth achieves higher accuracy—errors in distinguishing by SBERT are corrected due to the accumulated weight of correctly identified data from the majority.

> Answer for RQ3: SenteTruth mitigates the impact of errors during data aggregation by accumulating node weights based on SBERT similarity analysis. This approach effectively prevents the occurrence of errors due to minority discrepancies.

5.6 RQ4 - Usability Analysis

Table 5 presents the Gas costs of the proposed solution implemented on Ethereum and estimated USD values. The cost conversion is performed using the formula: Cost (USD) = Gas Used × Gas Price (ETH) × ETH Price (USD) = Gas Used $\times$ 4.971 $\times 10^{-9} \times$ 2,273.1147. We randomly selected 10 questions and queried ChatGPT, recording the Gas consumption for each request. Specifically, `requestLLM` is the interface through which the user contract calls the oracle contract to request data, while `fulfillData` is the interface through which the oracle node returns the data. It can be observed that the Gas cost is relatively higher during contract deployment, while the subsequent calls' Gas consumption depends on the length of the queried questions and the length of the responses. This implies that integrating the capabilities of large models into the blockchain requires only a few hundred thousand Gas fees, in addition to the oracle node incentives and the LLM API service costs. Taking the Ethereum mainnet as an example, the average on-chain cost per invocation is only around $0.3. With the advancement of high-performance blockchains (such as Avalanche and Solana) and the continued development of LLMs, the cost is expected to decrease further, making the approach economically viable. In particular, integrating LLMs holds practical significance and application value in scenarios such as on-chain governance, auditing, and financial risk assessment, where intelligent interaction is essential and invocation frequency is moderate.

Table 5. Gas cost in Ethereum and estimated USD values (Gas price = 4.971 Gwei, ETH/USD = 2,273.1147).

Method	Contract Deployment	requestLLM	fulfillData
Gas cost	796,915	274,366.7 (± 233,376.1)	127,148.8 (± 19,125.0)
Estimated USD	$8.99	$3.09 (± 2.63)	$1.43 (± 0.22)

Beyond the prototype with 10 oracle nodes, we further evaluated the scalability of the proposed framework. The two-round commit–reveal exchange and PBFT-style consensus incur $O(n^2)$ communication overhead, which can be mitigated through threshold-signature techniques. In contrast, the computational cost of SenteTruth aggregation is modest, involving only a single SBERT encoding step followed by simple similarity operations, and can be efficiently handled on commodity hardware. Moreover, as these tasks are executed off-chain, the on-chain gas cost remains largely independent of the network size. Taken together, these properties indicate that the framework can naturally scale to larger oracle networks (e.g., tens or hundreds of nodes) without significantly affecting efficiency or on-chain costs.

> Answer for RQ4: The proposed method is not difficult to implement. The primary costs of the system are determined by the Gas consumption of core interface calls and the service fees associated with the LLM API. Furthermore, the theoretical complexity analysis shows that communication scale is $O(n^2)$ in the worst case, and the on-chain gas cost remains largely independent of the network size, ensuring scalability to larger networks.

6 Conclusion

This paper proposes and implements a universal framework for integrating LLMs with blockchain data, C-LLM, aimed at overcoming interoperability barriers between blockchain and LLMs and introducing the intelligent analysis and decision-making capabilities of LLMs into smart contracts. To address the issue of heterogeneity in the data returned by nodes, we combine semantic relevance assessment with truth discovery techniques to propose a novel data aggregation method, SenteTruth. Additionally, the research constructs a dataset containing three types of questions, covering Q&A records between 10 oracle nodes and 5 LLM models, to validate the framework's effectiveness. Experimental results demonstrate that the proposed method offers significant advantages in improving data reliability and usability.

In the future, as advancements in artificial intelligence, particularly LLMs, continue to progress, the intelligence of smart contracts is expected to undergo a revolutionary transformation. Smart contracts will no longer be confined to rigid, hard-coded rules; instead, they will be capable of autonomously making flexible decisions based on real-time conditions, the evolving needs of contract participants, and inputs from external data sources. The semantic understanding capabilities of LLMs will empower smart contracts to analyze ambiguities in contract terms, automatically identify and resolve potential conflicts or misunderstandings, and optimize trading strategies and risk management. With the deepening integration of AI and blockchain technologies, smart contracts will be able to engage in self-learning and self-adjustment, progressively enhancing the accuracy and efficiency of decision-making. This evolution will significantly broaden the applicability of smart contracts, opening new opportunities in areas such as decentralized finance, supply chain management, and smart insurance, thereby ushering in a new era of intelligent blockchain systems.

Acknowledgments

The research was supported in part by the National Natural Science Foundation of China (Nos.62166004, U21A20474, 62262003), the Guangxi Science and Technology Major Project (No.AA22068070), the Basic Ability Enhancement Program for Young and Middle-aged Teachers of Guangxi (No.2022KY0057, 2023KY0062), the Guangxi Natural Science Foundation Project (Grant No. 2025GXNSFBA069295) and the Postdoctoral Fellowship Program of CPSF (Grant No. GZC20251060), the Key Lab of Education Blockchain and Intelligent Technology, the Center for Applied Mathematics of Guangxi, the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Talent Highland Project of Big Data Intelligence and Application, the Guangxi Collaborative Center of Multisource Information Integration and Intelligent Processing.

References

[1] Shyamal Anadkat. 2023. How to make your completions outputs consistent with the new seed parameter. https://cookbook.openai.com/examples/reproducible_outputs_with_the_seed_parameter.

[2] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2021. A survey on blockchain interoperability: Past, present, and future trends. *Acm Computing Surveys (CSUR)* 54, 8 (2021), 1–41.

[3] Ryan Berryhill and Andreas Veneris. 2019. ASTRAEA: A decentralized blockchain oracle. *IEEE Blockchain Technical Briefs* (2019).

[4] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, et al. 2021. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chainlink Labs* 1 (2021), 1–136.

[5] Roman Brodetski. 2017. Oracul System. https://gist.github.com/RomanBrodetski.

[6] Vitalik Buterin. 2014. SchellingCoin: A Minimal-Trust Universal Data Feed. https://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed.

[7] Jiayuan Chen, You Shi, Changyan Yi, Hongyang Du, Jiawen Kang, and Dusit Niyato. 2024. Generative AI-driven human digital twin in IoT-healthcare: A comprehensive survey. *IEEE Internet of Things Journal* (2024).

[8] Jiayuan Chen, Changyan Yi, Hongyang Du, Dusit Niyato, Jiawen Kang, Jun Cai, and Xuemin Shen. 2024. A revolution of personalized healthcare: Enabling human digital twin with mobile AIGC. *IEEE network* 38, 6 (2024), 234–242.

[9] Jiayuan Chen, Changyan Yi, Samuel D Okegbile, Jun Cai, and Xuemin Shen. 2023. Networking architecture and key supporting technologies for human digital twin in personalized healthcare: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 26, 1 (2023), 706–746.

[10] Jingya Dong, Chunhe Song, Yong Sun, and Tao Zhang. 2023. DAON: A decentralized autonomous oracle network to provide secure data for smart contracts. *IEEE Transactions on Information Forensics and Security* (2023).

[11] Pankaj Dutta, Tsan-Ming Choi, Surabhi Somani, and Richa Butala. 2020. Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation research part e: Logistics and transportation review* 142 (2020), 102067.

[12] Marco Di Gennaro, Lorenzo Italiano, Giovanni Meroni, and Giovanni Quattrocchi. 2022. DeepThought: A Reputation and Voting-Based Blockchain Oracle. In *Service-Oriented Computing: 20th International Conference, ICSOC 2022, Seville, Spain, November 29–December 2, 2022, Proceedings*. Springer, 369–383.

[13] Lorenzo Gigli, Ivan Zyrianoff, Federico Montori, Cristiano Aguzzi, Luca Roffia, and Marco Di Felice. 2023. A decentralized oracle architecture for a blockchain-based iot global market. *IEEE Communications Magazine* 61, 8 (2023), 86–92.

[14] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. In *Advances in Neural Information Processing Systems*.

[15] Xiaolong Li, Ruiting Deng, Jianhao Wei, Xin Wu, Jiayuan Chen, Changyan Yi, Jun Cai, Dusit Niyato, and Xuemin Shen. 2025. AIGC-Driven Real-Time Interactive 4D Traffic Scene Generation in Vehicular Networks. *IEEE Network* (2025).

[16] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter* 17, 2 (2016), 1–16.

[17] Yijing Lin, Zhipeng Gao, Weisong Shi, Qian Wang, Huangqi Li, Miaomiao Wang, Yang Yang, and Lanlan Rui. 2022. A novel architecture combining oracle with decentralized learning for iiot. *IEEE Internet of Things Journal* 10, 5 (2022), 3774–3785.

[18] Chunchi Liu, Hechuan Guo, Minghui Xu, Shengling Wang, Dongxiao Yu, Jiguo Yu, and Xiuzhen Cheng. 2022. Extending on-chain trust to off-chain–trustworthy blockchain data collection using trusted execution environment (tee). *IEEE Trans. Comput.* 71, 12 (2022), 3268–3280.

[19] Peng Liu, Youquan Xian, Chuanjian Yao, Peng Wang, Li-e Wang, and Xianxian Li. 2024. A Trustworthy and Consistent Blockchain Oracle Scheme for Industrial Internet of Things. *IEEE Transactions on Network and Service Management* 21, 5 (2024), 5135–5148. doi:10.1109/TNSM.2024.3399837

[20] Zhongtang Luo, Yanxue Jia, Yaobin Shen, and Aniket Kate. 2024. Proxying is Enough: Security of Proxying in TLS Oracles and AEAD Context Unforgeability. *Cryptology ePrint Archive* (2024).

[21] Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. 1–13.

[22] D. Network. 2019. A Decentralized Oracle Service Network to Boost Blockchain Usability with Real World Data and Computation Power. https://dos.network.

[23] Dinh C Nguyen, Pubudu N Pathirana, Ming Ding, and Aruna Seneviratne. 2020. Integration of blockchain and cloud of things: Architecture, applications and challenges. *IEEE Communications surveys & tutorials* 22, 4 (2020), 2521–2549.

[24] Jack Peterson, Joseph Krug, Micah Zoltu, Austin K Williams, and Stephanie Alexander. 2015. Augur: a decentralized oracle and prediction market platform. *arXiv preprint arXiv:1501.01042* (2015).

[25] Warwick Powell, Marcus Foth, Shoufeng Cao, and Valéri Natanelov. 2022. Garbage in garbage out: The precarious link between IoT and blockchain in food supply chains. *Journal of Industrial Information Integration* 25 (2022), 100261.

[26] Compound Protocol. 2021. Compound Protocol. https://compound.inance/docs.

[27] Shahzad Qaiser and Ramsha Ali. 2018. Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications* 181, 1 (2018), 25–29.

[28] N Reimers. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084* (2019).

[29] Kunpeng Ren, Nhut-Minh Ho, Dumitrel Loghin, Thanh-Toan Nguyen, Beng Chin Ooi, Quang-Trung Ta, and Feida Zhu. 2023. Interoperability in blockchain: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 12 (2023), 12750–12769.

[30] Abdurrashid Ibrahim Sanka and Ray CC Cheung. 2021. A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications* 195 (2021), 103232.

[31] Patrick Schueffel. 2021. Defi: Decentralized finance-an introduction and overview. *Journal of Innovation Management* 9, 3 (2021), I–XI.

[32] Qiang Wang and Min Su. 2020. Integrating blockchain technology into the energy sector—from theory of blockchain to research and application of energy blockchain. *Computer Science Review* 37 (2020), 100275.

[33] Xiaoding Wang, Qibin Wu, Haitao Zeng, Xu Yang, Hui Cui, Xun Yi, Md Jalil Piran, Ming Luo, and Youxiong Que. 2025. Blockchain-Empowered H-CPS Architecture for Smart Agriculture. *Advanced Science* (2025), 2503102.

[34] Sangyeon Woo, Jeho Song, and Sungyong Park. 2020. A distributed oracle using intel sgx for blockchain-based iot applications. *Sensors* 20, 9 (2020), 2725.

[35] Youquan Xian, Peng Liu, Dongcheng Li, and Xueying Zeng. 2024. Safeguarding the Truth of High-Value Price Oracle Task: A Dynamically Adjusted Truth Discovery Method. *arXiv preprint arXiv:2402.02543* (2024).

[36] Youquan Xian, Xueying Zeng, Chunpei Li, Dongcheng Li, Peng Wang, Peng Liu, and Xianxian Li. 2025. Instant resonance: Dual strategy enhances the data consensus success rate of blockchain threshold signature oracles. *Future Generation Computer Systems* 171 (2025), 107846. doi:10.1016/j.future.2025.107846

[37] Youquan Xian, Lianghaojie Zhou, Jianyong Jiang, Boyi Wang, Hao Huo, and Peng Liu. 2024. A Distributed Efficient Blockchain Oracle Scheme for Internet of Things. *IEICE Transactions on Communications* E107-B, 9 (2024), 573–582. doi:10.23919/transcom.2023EBP3156

[38] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. 2023. A Decentralized Truth Discovery Approach to the Blockchain Oracle Problem. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 1–10.

[39] Liwenhan Xie, Chengbo Zheng, Haijun Xia, Huamin Qu, and Chen Zhu-Tian. 2024. Waitgpt: Monitoring and steering conversational llm agent in data analysis with on-the-fly code visualization. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[40] Zhenan Xu, Jiuzheng Wang, Cong Zha, Xinyi Li, and Hao Yin. 2023. SmartLLM: A New Oracle System for Smart Contracts Calling Large Language Models. In *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2668–2675.

[41] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. 2016. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*. 270–282.

[42] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. 2020. Deco: Liberating web data using decentralized oracles for tls. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1919–1938.

[43] Xinyu Zhang, Hanbin Hong, Yuan Hong, Peng Huang, Binghui Wang, Zhongjie Ba, and Kui Ren. 2024. Text-CRS: A Generalized Certified Robustness Framework against Textual Adversarial Attacks. In *2024 IEEE Symposium on Security and Privacy (SP)*. 2920–2938. doi:10.1109/SP54263.2024.00053

[44] Yunpu Zhao, Rui Zhang, Junbin Xiao, Changxin Ke, Ruibo Hou, Yifan Hao, Qi Guo, and Yunji Chen. 2024. Towards Analyzing and Mitigating Sycophancy in Large Vision-Language Models. *arXiv preprint arXiv:2408.11261* (2024).

[45] Hao Zhou, Chengming Hu, Ye Yuan, Yufei Cui, Yili Jin, Can Chen, Haolun Wu, Dun Yuan, Li Jiang, Di Wu, et al. 2024. Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. *IEEE Communications Surveys & Tutorials* (2024).

[46] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems* 36 (2023), 50117–50143.

A Supplement the experimental results

Table 6. Data accuracy under 40% Random Response (MIX dataset)

Model	ChatGPT	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.75	0.67	0.8	0.9	0.96
TF-IDF Similarity	0.6	0.66	0.99	0.98	0.9
SBERT Similarity	1.0	1.0	1.0	1.0	0.99
Ours	1.0	1.0	1.0	1.0	1.0

Table 7. Data accuracy under 40% Incorrect Response (MIX dataset)

Method	ChatGPT	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.75	0.67	0.8	0.9	0.96
TF-IDF Similarity	0.6	0.66	0.99	0.98	0.9
SBERT Similarity	0.71	0.7	1.0	0.99	0.83
Ours	1.0	1.0	1.0	1.0	1.0

Received 1 April 2025; revised 26 August 2025; accepted 30 August 2025

Table 8. Data accuracy under 40% Model Substitution (MIX dataset)

Method	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.88	0.88	0.68	0.6
TF-IDF Similarity	0.81	0.21	0.25	0.4
TF-IDF Similarity + TD	0.99	0.0	0.0	0.44
SBERT Similarity	0.78	1.0	1.0	0.94
Ours	0.99	1.0	1.0	1.0

Table 9. Data accuracy under 40% Random Response (PRO dataset)

Model	ChatGPT	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	1.0	0.966	0.966	0.983	1.0
TF-IDF Similarity	0.833	0.233	0.466	0.0	0.133
SBERT Similarity	1.0	0.966	0.983	1.0	1.0
Ours	1.0	1.0	1.0	1.0	1.0

Table 10. Data accuracy under 40% Incorrect Response (PRO dataset)

Method	ChatGPT	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	1.0	1.0	0.95	0.983	1.0
TF-IDF Similarity	0.833	0.916	0.783	0.966	1.0
SBERT Similarity	0.866	0.866	0.733	1.0	1.0
Ours	1.0	1.0	1.0	1.0	1.0

Table 11. Data accuracy under 40% Model Substitution (PRO dataset)

Method	ChatGLM	Llama	Gemini	Hunyuan
Majority Voting	0.95	1.0	0.9	0.916
TF-IDF Similarity	0.95	0.966	0.783	0.933
TF-IDF Similarity + TD	1.0	1.0	0.983	1.0
SBERT Similarity	0.933	0.95	0.8	0.966
Ours	1.0	1.0	1.0	1.0