

# When MoE Meets Blockchain: A Trustworthy Distributed Framework of Large Models

Weihao Zhu, Long Shi, Kang Wei, Zhen Mei, Zhe Wang, Jiaheng Wang, and Jun Li

**Abstract**—As an enabling architecture of Large Models (LMs), Mixture of Experts (MoE) has become prevalent thanks to its sparsely-gated mechanism, which lowers computational overhead while maintaining learning performance comparable to dense LMs. The essence of MoE lies in utilizing a group of neural networks (called experts) with each specializing in different types of tasks, along with a trainable gating network that selectively activates a subset of these experts to handle specific tasks. Traditional cloud-based MoE encounters challenges such as prolonged response latency, high bandwidth consumption, and data privacy leakage. To address these issues, researchers have proposed to deploy MoE over distributed edge networks. However, a key concern of distributed MoE frameworks is the lack of trust in data interactions among distributed experts without the surveillance of any trusted authority, and thereby prone to potential attacks such as data manipulation. In response to the security issues of traditional distributed MoE, we propose a blockchain-aided trustworthy MoE (B-MoE) framework that consists of three layers: the edge layer, the blockchain layer, and the storage layer. In this framework, the edge layer employs the activated experts downloaded from the storage layer to process the learning tasks, while the blockchain layer functions as a decentralized trustworthy network to trace, verify, and record the computational results of the experts from the edge layer. The experimental results demonstrate that B-MoE is more robust to data manipulation attacks than traditional distributed MoE during both the training and inference processes.

**Index Terms**—Large Models, trustworthy edge, Mixture of Experts, blockchain.

## I. INTRODUCTION

RECENTLY, Mixture of Experts (MoE) [1] has gained extensive popularity as a commonly used technique in mainstream Large Models (LMs) such as Mistral and DeepSeek-MoE. With reference to classical MoE frameworks like Gshard [2] and FastMoE [3], MoE consists of a group of separate neural networks that are divided into a gating network and several experts according to their functions. The essence of MoE is that, the gating network first calculates the weight of

Weihao Zhu, Long Shi, and Zhen Mei are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: zhuwh0813@njust.edu.cn; slong1007@gmail.com; meizhen@njust.edu.cn). Kang Wei is with the School of Cyber Science and Engineering, Southeast University, Nanjing, 211189, China. He is also with the Engineering Research Center of Blockchain Application, Supervision and Management (Southeast University), Ministry of Education. (e-mail: kang.wei@seu.edu.cn). Zhe Wang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China (email: zwang@njust.edu.cn). Jiaheng Wang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 211102, China, and also with the Purple Mountain Laboratories, Nanjing 211111, China (e-mail: jhwang@seu.edu.cn). Jun Li is with the National Mobile Communications Research Laboratory, Southeast University, China (e-mail: jun.li@seu.edu.cn).

each expert based on the input task and sparsely activates the experts with Top- $K$  weights. Then, the task is processed by the activated experts, and the obtained results are aggregated with weights to generate the final output of MoE. In the context of MoE, LMs can augment model parameter scales without introducing excessive computational overhead, while gaining comparable learning performance to dense LMs.

It is noted that the mainstream MoE systems are commonly implemented in the cloud servers equipped with sufficient computing resources. However, these cloud-based MoE frameworks are challenged by several limitations, e.g., prolonged response latency, high bandwidth consumption, potential privacy leakage, and high deployment costs. To address these issues, recent works of [4], [5] have proposed to deploy MoE over the distributed edge networks (i.e., edge-based MoE) towards real-time performance, privacy protection, and economic effectiveness [6]. Specifically, compared with the cloud-based MoE, edge-based MoE reduces the response latency by shortening the physical distance to data sources, and alleviates the bandwidth consumption through local pre-processing (e.g., feature extraction or lightweight analysis), which only requires exchanging key information across edges. Moreover, edge-based MoE mitigates the risks of privacy leakage by enabling local or near-source data processing, thereby diminishing the necessity of transmitting sensitive information. Additionally, edge-based MoE reduces the economic costs due to lower purchase or rental costs of edge computing devices.

Within an edge-based MoE framework, the experts are distributed across different edge computing devices, wherein data interaction occurs among the distributed experts to facilitate the training and inference processes [7]. In this distributed framework, [8] studies the issue of workload imbalance among different experts, and proposes to replicate busy experts across multiple devices to balance the computational load. Later, [9] investigates traffic congestion in distributed MoE caused by all-to-all communications, and designs an expert migration algorithm to optimize the overall latency. Despite facilitating the deployment of edge-based MoE, a key concern of this traditional distributed MoE framework is the lack of trust in data interactions among distributed experts without the surveillance of any trusted authority, and thereby prone to potential attacks such as data manipulation (e.g., adversarial perturbation and model poisoning) [10]. Specifically, malicious edges can manipulate either the computational results of the experts or the model parameters of the gating network and experts to degrade the learning performance of MoE. However, through a thorough literature review, we find that there has been little research on the security issue of untrustworthy data

interaction in distributed MoE networks.

Motivated by this non-trivial and critical issue, we develop a blockchain-aided trustworthy MoE (B-MoE) framework to enhance the robustness of distributed MoE against data manipulation attacks. The contributions of this article are summarized as follows:

- We investigate the security issues of untrustworthy data interaction of traditional distributed MoE. Based on the experimental results, we find that the traditional framework suffers severe performance degradation under data manipulation attacks during both the training and inference processes. Specifically, during the training process, the gating network can reduce the frequency of activating the manipulated experts, however, this not only comes at the cost of inefficient utilization of expert resources (i.e., workload imbalance), but also degrades the learning performance of MoE. During the inference process, the gating network cannot identify malicious edges, which significantly degrades the inference performance.
- We propose a B-MoE framework to promote the security of data interaction among distributed edges against data manipulation attacks. For each learning task, under the coordination of the blockchain layer, each edge in the edge layer downloads all the activated experts from the storage layer to execute training or inference tasks and uploads the computational results of each expert to the blockchain layer. The blockchain layer verifies and globally agrees on the trustworthy computational results, and records the data interaction behaviors on the chain in a decentralized manner. Thus, B-MoE can effectively detect malicious edges thanks to tamper proofing, global verification, and decentralized record of blockchain.
- We conduct experiments on Fashion-MNIST and CIFAR-10 to evaluate the training and inference performance of B-MoE under data manipulation attacks. The experimental results show that under data manipulation attacks, compared with traditional distributed MoE, B-MoE can achieve at least 45% and 44% test accuracy improvement during the training and inference processes, respectively. This corroborates that B-MoE is more resilient to data manipulation attacks than traditional distributed MoE.

The remainder of this article is organized as follows. Section II introduces the preliminaries of MoE, blockchain, and smart contracts. Section III investigates the security issues of traditional distributed MoE and explains the challenge and motivation of this article. Section IV presents the framework and workflow of B-MoE. Section V presents the experimental results. Section VI lists future research directions of B-MoE. Finally, Section VII concludes this article.

## II. PRELIMINARIES

In this section, we brief the preliminaries of MoE, blockchain, and smart contracts.

### A. MoE

The concept of MoE was initially proposed more than three decades ago [11], and it has resurfaced in interest with the rapid

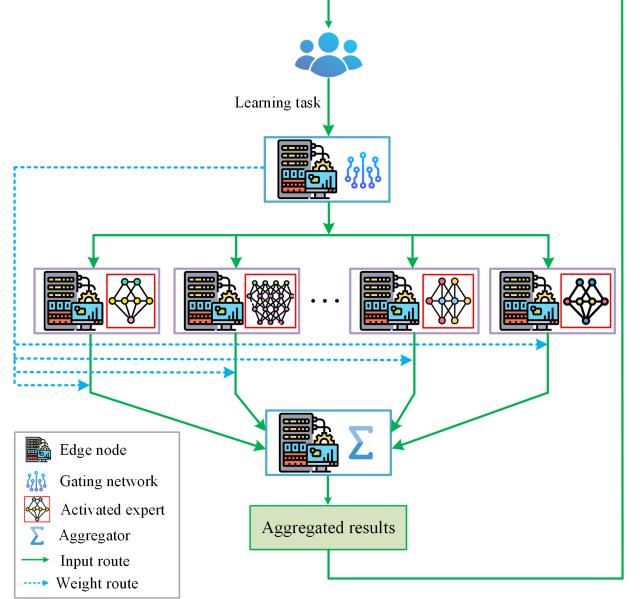


Fig. 1: The architecture of edge-based sparsely-gated MoE.

development of LMs in recent years. An MoE layer consists of a gating network and a set of experts. For each training or inference task (e.g., processing a batch of data samples), the gating network first evaluates weights for each expert on every data sample. A higher weight assigned to a particular expert indicates greater specialization. Subsequently, the experts process the tasks and generate the computational results. Finally, the aggregator aggregates the results to produce the final output of MoE for each data sample.

To reduce the significant computational burden of expanding the scale of MoE, [12] develops a sparsely-gated MoE framework. Fig. 1 shows the framework of distributed sparsely-gated MoE in the edge network. In this framework, the gating network and experts are distributed across a group of edges, leveraging the computing power of these edges to execute training or inference. Considering a total of  $N$  experts, MoE only activates  $K$  ( $K \leq N$ ) experts with the top- $K$  highest weights to process each data. This sparsely-gated approach has been widely adopted in LMs due to its advantage of augmenting model parameter scales without imposing excessive computational overhead while maintaining the learning performance comparable to dense LMs.

### B. Blockchain and Smart Contracts

Blockchain is a distributed database technology known for its key characteristics of decentralization, high security, and strong transparency [13]. Initially introduced as the underlying technology for Bitcoin, blockchain has now found wide-ranging applications across various domains. The core concept of blockchain involves storing data in a series of continuously growing “blocks” linked together using cryptographic techniques to form an immutable “chain”. Each block contains transaction data from a certain time period, and each new block is validated by multiple nodes in the network before being added to the end of the chain. The main features of blockchain include:

(1) Decentralization: Blockchain operates without reliance on a single centralized authority. Instead, data is maintained and managed collectively by distributed nodes in a decentralized manner.

(2) Security: Blockchain ensures data security and integrity through cryptographic techniques. Each block contains the hash of the previous block, making any tampering on the chain detectable and rejectable by honest nodes.

(3) Transparency: On-chain transaction information is publicized by all the blockchain nodes. This transparency enhances trustworthiness and helps prevent fraud and misconduct.

Smart contracts are computer protocols designed to automatically execute and verify the contract terms that are defined by computer code. Once the triggering conditions are met, the contracts can automatically execute without human intervention. On top of blockchain, the process of smart contracts execution is transparent, immutable, and trustworthy, ensuring the security and reliability of the contract without the need of any centralized operation.

### III. CHALLENGES AND MOTIVATIONS

In the edge network, traditional distributed MoE frameworks face the security issues of untrustworthy data interaction among the distributed gating network and experts. For example, potential adversaries can control edges to manipulate the computational results or model parameters of the gating network and experts. However, it is notable that MoE sparsely activates the experts with the Top- $K$  weights for each task, raising a question of whether the gating network (assuming it is employed by the honest edge) itself can detect and deactivate manipulated experts with the aim of preserving the learning performance.

Motivated by this open issue, we conduct an experiment on sparsely-gated MoE to analyze the learning performance under data manipulation attacks. Fig. 2 shows the activation ratio (i.e., the ratio of the number of samples processed by the expert over the total number of samples) of each expert under data manipulation attacks during both the training and inference processes. We set the number of experts as  $N = 10$  employed by  $M = 10$  edges and the number of activated experts for each sample as  $K = 3$ , respectively. Moreover, we let each edge employ a distinct expert, i.e., edge  $i$  employs expert  $i$ ,  $\forall i \in \mathcal{N} = \mathcal{M} = \{1, 2, \dots, 10\}$ , where  $\mathcal{N}$  and  $\mathcal{M}$  denote the index set of experts and edges, respectively. In this figure, we let experts 7-9 be employed by malicious edges.

**Impact on the training process:** From Fig. 2, it is observed that under data manipulation attacks, the activation ratios of experts 7, 8, and 9 are significantly dropped during the training process compared with that before. This indicates that the activation frequencies of the experts employed by malicious edges are significantly reduced. Thus, the gating network is able to detect manipulated computational results of the experts during the training process. This is due to the fact that the gating network can update itself during the backward propagation of the training process. Therefore, to minimize the MoE loss function, the gating network reduces the probability of activating experts deployed by malicious

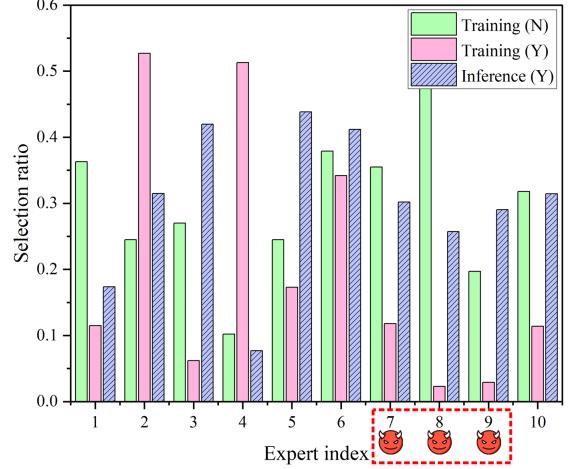


Fig. 2: The activation ratio of each expert of traditional distributed MoE. ‘Y’ denotes the MoE is attacked by data manipulation, while ‘N’ denotes not.

edge nodes. However, this incurs two critical problems: (1) inefficient expert resource utilization (i.e., extreme workload imbalance), which exacerbates expert overfitting and impairs the generalization capability of MoE [12]; and (2) learning performance degradation, which is shown in Fig. 4 (a).

**Impact on the inference process:** To evaluate the activation ratios of the experts under data manipulation attacks during the inference process, we first train the MoE in a trustworthy environment and then deploy the trained MoE in a network with malicious edges. Fig. 2 shows that under data manipulation attacks, the activation ratios of experts 7, 8, and 9 during the inference process are significantly higher than that during the training process. This indicates that a large number of samples have been processed by the experts employed by malicious edges. This is due to the fact that the gating network cannot detect the manipulated results of the experts since it is no longer updated during the inference process. As a result, the MoE’s inference performance suffers severe degradation.

Generally, the traditional distributed MoE framework is vulnerable to data manipulation attacks, and how to address such security issue remains challenging. A major challenge lies in the fact that, in traditional distributed MoE, the edge cannot verify the computational results from its peers due to the divergence of experts across different edges. This disadvantage is particularly significant during the inference phase. To address this issue, a redundancy mechanism can be employed for distributed MoE, which is a common approach in the Internet of Things and edge computing domains for fault tolerance [14]. The core of the mechanism lies that, the edges are allowed to employ the same experts to process the same learning task. In this context, the honest edges can identify any manipulated results from malicious edges by comparing all the results published by the edges and accepting the most consistent results. Inspired by this approach, we develop a blockchain-assisted distributed MoE framework to enhance the security of data interaction of MoE. Within this framework, blockchain acts as a decentralized trustworthy infrastructure to trace, verify, and record the trustworthy computational results.

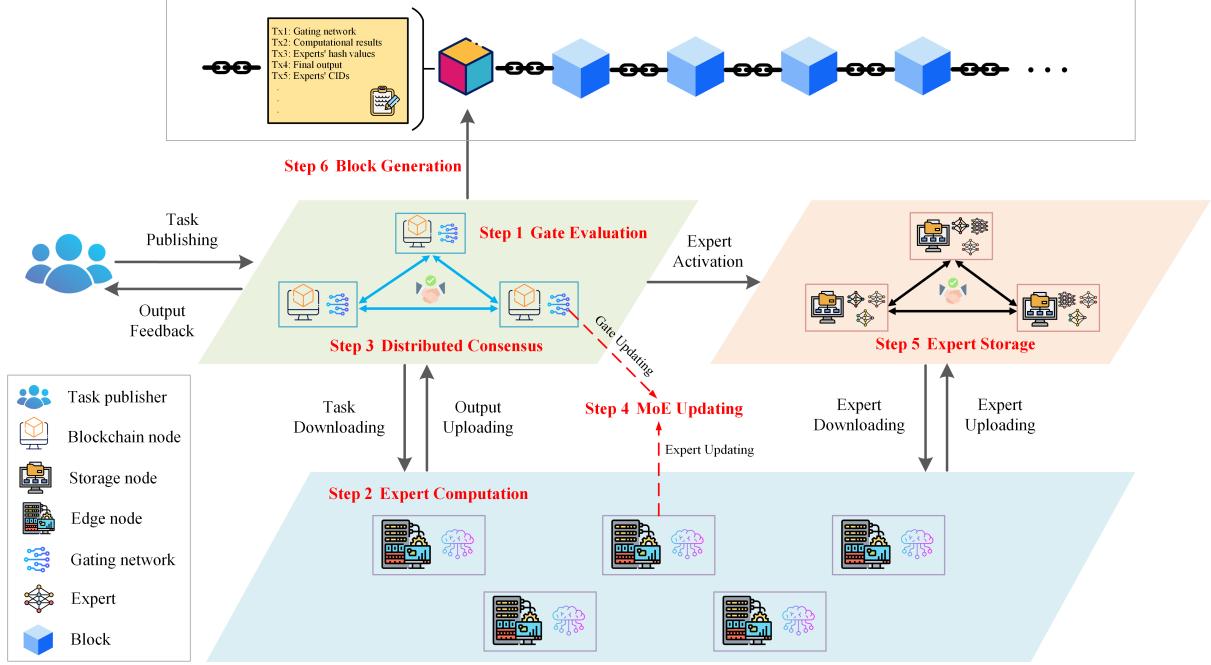


Fig. 3: The framework and workflow of B-MoE.

#### IV. SYSTEM MODEL

In this section, we present the workflow of the proposed B-MoE framework.

##### A. Framework and Workflow of B-MoE

As illustrated in Fig. 3, the B-MoE framework consists of a task publisher and three layers: the edge layer, the blockchain layer, and the storage layer.

(1) Task publisher: The task publisher publishes its training or inference tasks on the blockchain. For privacy-preserving purposes, raw data of these learning tasks can be desensitized.

(2) The edge layer: The edge layer consists of an edge network with a group of edges that have sufficient computing resources to deploy the experts for the learning task computation. Each edge downloads the experts from the storage layer for training or inference.

(3) The blockchain layer: The blockchain layer consists of a blockchain network that collects, verifies, and reaches a global agreement on the trustworthy computational results of the experts from the edge layer. Moreover, the gating network is stored on the chain to evaluate the weights of experts.

(4) The storage layer: The storage layer consists of a decentralized storage network with a group of storage nodes to supply decentralized storage services, such as the existing technique of Interplanetary File System (IPFS). The storage layer is employed to store all the experts and B-MoE can query or download any expert according to its sole content identifier (CID) generated by the storage layer, which is stored on the blockchain.

With the above exposition, let us illustrate the workflow of B-MoE in Fig. 3 as follows:

**Step 1 (Gate Evaluation):** The blockchain network receives the learning task from the task publisher. Based on the task, the on-chain gating network evaluates the weights of the

experts. After that, the blockchain network records the task and activates the experts through the top- $K$  sparse activation.

**Step 2 (Expert Computation):** The edges download the recorded task from the blockchain network and the activated experts from the storage layer, respectively. It is stressed that each edge is required to download all the activated experts to facilitate the redundancy mechanism. After that, the edges utilize the activated experts to process the learning task and upload the computational results to the blockchain network.

**Step 3 (Distributed Consensus):** The blockchain nodes collect the computational results of the experts uploaded by the edge network and propagate these data as blockchain transactions among the entire blockchain network. Subsequently, the blockchain nodes sequentially verify the validity of the results from each activated expert by comparing that of the same expert across different edges, and globally agree on the most consistent results as the trustworthy ones. Finally, each blockchain node aggregates the trustworthy results of the activated experts to generate the final outputs of MoE.

**Step 4 (MoE Updating):** The edges download the value of the loss function generated by the blockchain network according to the aggregated outputs of MoE. After that, the edges update the activated experts and the blockchain nodes update the gating network via the gradient descent algorithm. Finally, the edges upload the updated experts' hash values to the blockchain network and the updated experts to the storage network, respectively.

**Step 5 (Expert Storage):** The blockchain network globally verifies the validity of the updated experts' hash values and agrees on the trustworthy ones. After that, the storage network stores the updated experts corresponding to the trustworthy hash values and generates the CIDs for them.

**Step 6: (Block Generation)** Each blockchain node packages the data such as the trustworthy computational results, the CIDs of the updated experts (if B-MoE is during the training

process), the final output of MoE, and the gating network into a block and competes to generate a new block under the distributed consensus. The task publisher can download the final output of MoE from the blockchain network and then publish a new learning task.

The aforementioned workflow involves both the training and inference processes of B-MoE. Notably, the training process encompasses both forward propagation and backward propagation, whereas the inference process only requires forward propagation. In this context, **Steps 4-5** are skipped if B-MoE operates in the inference process.

Within the workflow of B-MoE, the blockchain layer is compatible with existing consensus protocols such as Proof of Work (PoW) and Practical Byzantine Fault Tolerance (PBFT). In this article, we tentatively adopt PoW, the prominent consensus of mainstream blockchain networks, for B-MoE. Moreover, the data interaction across different layers can be automatically triggered by smart contracts, e.g., the task downloading and result uploading between the edge and blockchain layers, the expert downloading and uploading between the edge and storage layers, and the CID generation of the experts within the storage layer.

### B. Security Analysis of B-MoE

From Section IV-A, B-MoE decouples the deployment of the gating network and the experts of MoE, wherein the gating network is deployed over the blockchain network, and the experts are deployed over the edges. In this context, data manipulation attacks may be conducted by malicious blockchain nodes and malicious edges. Let us present the security analysis of B-MoE with particular focus on on-chain and off-chain scenarios.

**Scenario (1):** In the blockchain layer, MoE is vulnerable to data manipulation from malicious blockchain nodes, wherein malicious blockchain nodes can tamper the intermediate data of MoE computation, e.g., the model parameters and computational results of the gating network within the blockchain layer, and the computation results of experts uploaded from the edge layer. With the aid of distributed consensus, only intermediate data verified by the majority of blockchain nodes can be accepted and recorded on the chain. Since we adopt PoW as the on-chain consensus of B-MoE, the threshold of on-chain malicious ratio (i.e., the proportion of computing power controlled by malicious collusion over the total computing power) in the blockchain layer is 50% according to the prior work of [15]. Specifically, if the on-chain malicious ratio exceeds this threshold, malicious collusion possesses sufficient computing power to dominate the block generation process and bias the entire blockchain network towards the untrustworthy results of MoE. However, such an attack costs extremely high computational power under PoW and is therefore unlikely to occur in practice [15].

**Scenario (2):** In the edge layer, malicious edges can manipulate the computational results and model parameters of experts. To address this issue, the redundancy mechanism of B-MoE stipulates that each edge downloads all the activated experts for computation. With the aid of the distributed consensus,

the blockchain network can filter out the untrustworthy results (e.g., the computation results and hashes of the experts) from the malicious edges and accept the majority-consistent results as trustworthy. The key of this solution is that, the trustworthy results of the experts from honest edges remain identical, whereas those from malicious edges vary diversely. This enables the blockchain network to distinguish between malicious and honest edges and achieve a global consensus on the trustworthy results. However, if the off-chain malicious ratio (i.e., the proportion of the number of malicious colluding edges in the total number of edges in the edge layer) surpasses 50%, the blockchain is misled by the manipulated results of the experts, which degrades the learning performance of MoE.

### C. Scalability Analysis of B-MoE

In reality, MoE can scale up by deploying more experts and enlarging the model size to process more complex tasks and achieve higher learning performance. In this context, let us present the scalability challenges and tentative solutions for large-scale B-MoE from the perspectives of storage, computation, and communication overheads as follows.

**(1) Storage Overhead:** As the number of experts and the model size enlarge, B-MoE faces higher storage demands. To address this issue, a tentative solution is to scale up the storage layer by employing more storage nodes to accommodate the increasing demand of storage load.

**(2) Computation Overhead:** From the prior works of [1]–[3], a prominent advantage of MoE is its ability to gain comparable learning performance to dense models by activating only a small subset of experts. Therefore, if the number of activated experts (i.e., the value of  $K$ ) remains fixed, the computation overhead on the edge layer for both training and inference processes remains stable, although the total number of experts goes up.

**(3) Communication Overhead:** As the scale of MoE expands, communication overhead between the edge layer and the storage layer increases, since the edge layer needs to download (upload) more experts from (to) the storage layer. To address this issue, a tentative solution is to apply model compression techniques (e.g., sparsification or quantization) to shrink the transmission size of experts (see Section VI-B). Meanwhile, if the number of activated experts  $K$  is fixed, the communication overhead between the blockchain layer and the storage layer remains stable. Moreover, communication overhead caused by uploading additional expert hash values from the edge layer to the blockchain layer is negligible, as each hash is represented by only a small number of bits.

## V. EXPERIMENTAL RESULTS

In this section, we conduct experiments to evaluate the performance of B-MoE under data manipulation attacks from the malicious edges.

### A. Experiment Setting

**(1) Hardware environment:** One Intel-13900KF CPU (3.0GHz and 64GB RAM) and two RTX-4090 GPUs (24GB RAM).

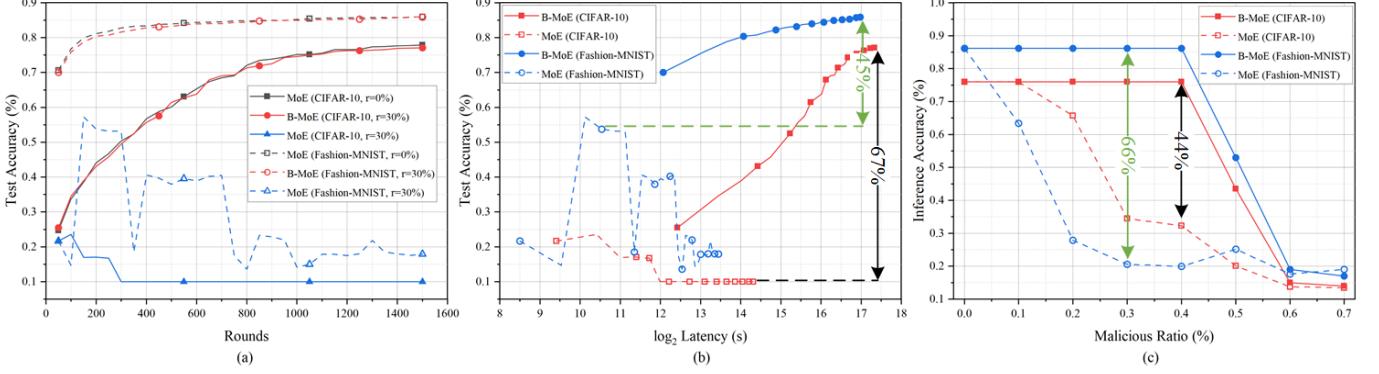


Fig. 4: Performance comparison of B-MoE and traditional distributed MoE: (a) Training performance, (b) Latency, and (c) Inference performance.

(2) Software environment: Pytorch platform with Python 3.9 version.

(3) Datasets: We evaluate the performance of B-MoE on Fashion-MNIST and CIFAR-10, respectively. The Fashion-MNIST contains 60,000 training samples and 10,000 test samples, each of which is a  $28 \times 28$  grayscale image of clothes. The CIFAR-10 dataset contains 50,000 training samples and 10,000 test samples, each of which is a  $32 \times 32$  RGB color image with labels including aircraft, cars, birds, cats, deer, dogs, frogs, horses, boats, and trucks.

(4) Experiment parameters: We consider  $N = 10$  experts and  $M = 10$  edges in the experiments. For traditional distributed MoE, we let each edge employ a distinct expert. In the training process, we set the learning rate as 0.01 and 0.1 for Fashion-MNIST and CIFAR-10, respectively. We train both B-MoE and traditional distributed MoE over  $T = 1500$  rounds, wherein the task publisher publishes 1000 samples (i.e., learning task) for processing in each round.

(5) Models: We use the linear model as the gating network, and use two types of neural network models as the experts for different datasets.

- Multi-layer Perceptron (MLP): For Fashion-MNIST, we let all the experts be MLPs. Each MLP consists of two fully connected layers, wherein the number of neurons of hidden layer is 256 and the activation function is ReLu.
- Convolutional Neural Network (CNN): For CIFAR-10, we let all the experts be CNNs. Each CNN consists of three convolutional layers and two fully connected layers.

To simulate data manipulation attacks, we let malicious edges inject random Gaussian noise into the employed experts in each round, and we also set each malicious edge to launch an attack with a probability of 0.2.

### B. Performance Analysis of B-MoE

In this part, we compare the training and inference performance of B-MoE and traditional distributed MoE under data manipulation attacks. We let the number of activated experts for processing each sample be  $K = 3$ . For B-MoE, we consider that malicious edges collude with each other to publish the same manipulated results of the experts to threaten the security of blockchain consensus.

Fig. 4 (a) compares the test accuracy of B-MoE and traditional distributed MoE during the training process under data manipulation attacks, where  $r$  denotes the malicious ratio. It is observed that B-MoE consistently outperforms traditional distributed MoE. Moreover, B-MoE gains almost the same accuracy as traditional distributed MoE without data manipulation attacks. These observations demonstrate B-MoE's robustness against data manipulation attacks during the training process.

Fig. 4 (b) compares the latency of B-MoE and traditional distributed MoE under manipulation attacks in the training process, where the experiment settings follow those of Fig. 4 (a). It is observed that B-MoE gains significant test accuracy improvement (e.g., 45% on Fashion-MNIST and 67% on CIFAR-10) over traditional distributed MoE at the cost of higher latency.

Fig. 4 (c) compares the inference accuracy of both well-trained B-MoE and traditional distributed MoE under different malicious ratios. It is observed that B-MoE gains higher inference accuracy than traditional distributed MoE (e.g., 66% on Fashion-MNIST and 44% on CIFAR-10) when the malicious ratio is below 50%. This observation demonstrates B-MoE's enhanced resilience against data manipulation attacks than traditional distributed MoE during the inference process. However, it is also observed that B-MoE suffers significant performance degradation when the malicious ratio surpasses 50%. This is due to the fact that, as the number of malicious edges surpasses that of honest edges, the blockchain agrees on the manipulated results of the experts, thereby degrading the inference performance of MoE.

## VI. FUTURE DIRECTIONS

### A. Resource Optimization

Deploying B-MoE over the edge networks demands sufficient computing and bandwidth resources. However, the computing resources and communication conditions of edge devices are typically limited, heterogeneous, and dynamic. In this context, the design of resource optimization strategies for B-MoE is an urgent problem. Inspired by the previous work on edge computing [5], a tentative solution is to use Lyapunov optimization and reinforcement learning to maximize the long-term system throughput (or minimize the long-term overall

latency) by dynamically allocating computational loads and communication bandwidth among the edges.

### B. Latency Investigation

B-MoE improves the security performance at the cost of high latency compared with traditional distributed MoE, which is mainly caused by frequent off-chain data interaction and on-chain consensus. In this context, it is of paramount importance to improve the overall efficiency of B-MoE while maintaining its robustness to malicious attacks. One tentative solution is to use model compression techniques (e.g., quantization and sparsification) to reduce the communication overhead of transmitting model parameters of experts. Another tentative solution is to design a lightweight consensus mechanism to balance the trade-off between network security and consensus efficiency. For example, inspired by the previous work [13], we can design a reputation-aided hybrid consensus based on PoW and Proof of Stake, wherein the reputation serves as the stake to dynamically adjust the consensus difficulty for each blockchain node. Specifically, the difficulty is inversely proportional to the reputation, which incentivizes the nodes with higher reputation to generate blocks with higher probabilities.

### C. Workload Balance

Workload imbalance in MoE refers to the situation wherein a small group of experts (or their located edges) processes most of the tasks at the cost of resource wastage, reduced efficiency, and imbalanced training updates. However, B-MoE also faces a critical problem of workload imbalance. In this context, the optimization of workload balance for B-MoE remains open. In light of the trade-off between the learning performance and the degree of workload imbalance of B-MoE, a tentative solution is to design control mechanisms to let experts with lighter workloads actively participate in processing the tasks. Additionally, considering resource-limited edges, dynamic expert scheduling methods can be optimized to balance the workloads of B-MoE.

### D. Incentive Mechanism

As previously discussed in Section IV-B, the learning performance of B-MoE is significantly degraded if the malicious ratio exceeds the security threshold. Although the computational costs of such data manipulation attacks are extremely high in practice, there still exist potential security risks. To alleviate these risks, a tentative solution is to incentivize more edges towards honest behaviors by designing efficient incentive mechanisms for B-MoE. However, there are significant challenges in designing such incentive mechanisms. For example, the optimization of resource allocation among selfish edges is generally formulated as a multi-agent stochastic game due to information asymmetry and network dynamics. In this context, how to guarantee the incentive compatibility and budget balance remains open and challenging.

## VII. CONCLUSION

In this article, we have investigated the security issues of untrustworthy data interaction in traditional edge-based distributed MoE frameworks, and found that the traditional framework suffers severe learning performance degradation under data manipulation attacks. To cope with this problem, we have proposed a novel B-MoE framework that reconciles the blockchain, edge, and decentralized storage networks to enhance the robustness of distributed MoE against data manipulation. The core of this proposal lies in that, the blockchain network bridges the edge network and the storage network to trace, verify, and record trustworthy computational results of the experts. The experimental results have demonstrated that compared with traditional distributed MoE, B-MoE exhibits better robustness against data manipulation attacks and can tolerate up to 50% malicious ratio. Finally, we have provided potential research trends of B-MoE from the perspectives of resource optimization, latency investigation, workload balance, and incentive mechanism.

## REFERENCES

- [1] D. Yu, L. Shen, H. Hao, W. Gong, H. Wu, J. Bian, L. Dai, and H. Xiong, “Moesys: A distributed and efficient mixture-of-experts training and inference system for internet services,” *IEEE Transactions on Services Computing*, vol. 17, no. 5, pp. 2626–2639, 2024.
- [2] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, “GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding,” in *Proc. ICLR*, 2021.
- [3] J. He, J. Qiu, A. Zeng, Z. Yang, J. Zhai, and J. Tang, “Fastmoe: A fast mixture-of-expert training system,” *arXiv preprint arXiv:2103.13262*, 2021.
- [4] N. Xue, Y. Sun, Z. Chen, M. Tao, X. Xu, L. Qian, S. Cui, and P. Zhang, “Wdmoe: Wireless distributed large language models with mixture of experts,” in *Proc. IEEE GLOBECOM*, 2024, pp. 2707–2712.
- [5] J. Wang, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, and K. B. Letaief, “Toward scalable generative ai via mixture of experts in mobile edge networks,” *IEEE Wireless Communications*, vol. 32, no. 1, pp. 142–149, 2025.
- [6] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, “Toward an intelligent edge: Wireless communication meets machine learning,” *IEEE Communications Magazine*, vol. 58, no. 1, pp. 19–25, 2020.
- [7] M. Ryabinin and A. Gusev, “Towards crowdsourced training of large neural networks using decentralized mixture-of-experts,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 3659–3672.
- [8] J. He, J. Zhai, T. Antunes, H. Wang, F. Luo, S. Shi, and Q. Li, “Fastermoe: modeling and optimizing training of large-scale dynamic pre-trained models,” in *Proc. PPoPP*, 2022, pp. 120–134.
- [9] J. Liu, J. H. Wang, and Y. Jiang, “Janus: A unified distributed training framework for sparse mixture-of-experts models,” in *Proc. SIGCOMM*, 2023, pp. 486–498.
- [10] Z. Wang, Q. Wang, G. Yu, and S. Chen, “TDML—A trustworthy distributed machine learning framework,” *Future Generation Computer Systems*, vol. 174, p. 107951, 2026.
- [11] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts,” *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.
- [12] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” in *Proc. ICLR*, 2017.
- [13] L. Shi, T. Wang, Z. Xiong, Z. Wang, Y. Liu, and J. Li, “Blockchain-aided decentralized trust management of edge computing: Toward reliable off-chain and on-chain trust,” *IEEE Network*, vol. 38, no. 5, pp. 182–188, 2024.
- [14] M. Mudassar, Y. Zhai, and L. Lejian, “Adaptive fault-tolerant strategy for latency-aware iot application executing in edge computing environment,” *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 250–13 262, 2022.
- [15] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the security and performance of proof of work blockchains,” in *Proc. ACM CCS*, 2016, pp. 3–16.