RLChain: A DRL Approach for Blockchain Performance Optimization Towards IIoT

Min An , Xuan Zhang , *Member, IEEE*, Jishu Wang , *Member, IEEE*, Qiyuan Fan, Chen Gao, Linyu Li , Cuizhen Lu, Nan Li, and Yingchen Liu

***Abstract*—With the development of communication technology and Internet of Things, Industrial Internet of Things (IIoT) is proposed in the automation industry for complex scenarios. Blockchain is applied in IIoT to solve data security and privacy issues related to centralized data storage and processing. However, there are inevitably performance issues with throughput constraints when blockchain manages large amounts of device data. This paper proposes a blockchain-supported performance optimization framework for IIoT systems using deep reinforcement learning (DRL) methods. We model the blockchain performance optimization problem as a Markov decision process that optimizes the blockchain's throughput by dynamically adjusting the block size and interval through DRL while satisfying security constraints. We use the double deep Q-network (DDQN) to deal with the dynamic and complexity of optimization problems due to the heterogeneity of equipment and diversified requirements. We also alleviate the overestimation problem caused by DQN. Meanwhile, we study the impact of the number of network layers and different activation units on the performance optimization method in DDQN. Finally, we prove that our work is feasible and effective through the case study based on actual IIoT scenario datasets. Experimental results demonstrate that our proposed scheme enhances blockchain performance in IIoT systems. The detailed qualitative comparison with related work demonstrates the superiority and innovation of our work and proves that it improves the shortcomings of existing work.**

***Index Terms*—Blockchain, deep reinforcement learning (DRL), Industrial Internet of Things (IIoT), performance optimization, scalability, latency.**

This work was supported by Xingdian Talent Support Program Industrial Innovation Talent Project; Science Foundation of Young and Middle-aged Academic and Technical Leaders of Yunnan under Grant No. 202205AC160040; Science Foundation of Yunnan Jinzhi Expert Workstation under Grant No. 202205AF150006; Major Project of Yunnan Natural Science Foundation under Grant No. 202302AE09002003; Special funding projects for the national guidance of provincial science and technology development under Grant No. 202407AB110010; Knowledge-driven Smart Energy Science and Technology Innovation Team of Yunnan Provincial Department of Education; Open Foundation of Yunnan Key Laboratory of Software Engineering under Grant No. 2023SE101; Research and Innovation Fund for Recommended Postgraduates without Examination of Yunnan University under Grant TM-23236809. *(Corresponding authors: Xuan Zhang.)*

Min An and Qiyuan Fan are with the School of Software, Yunnan University, Kunming 650091, China (e-mail: kirinmin@hotmail.com; 804957581@qq.com).

Xuan Zhang is with the Yunnan Key Laboratory of Software Engineering; the School of Software, Yunnan University, Kunming 650091, China (e-mail: zhxuan@ynu.edu.cn).

Jishu Wang and Chen Gao are with the School of Information Science and Engineering, Yunnan University, Kunming 650091, China (e-mail: cswangjishu@hotmail.com; 929165733@qq.com).

Linyu Li is with the Key Laboratory of High Confidence Software Technologies, Ministry of Education; the School of Computer Science, Peking University, Beijing 100871, China. (e-mail: xltx_youxiang@qq.com;).

Cuizhen Lu is with the College of Electronics and Information Engineering, Sichuan University, Chengdu, Sichuan, China (email: 1041508033@qq.com).

Nan Li is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, Liaoning, China (email: 1782348177@qq.com).

Yingchen Liu is with the College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, Hunan, China (email: liuyingchen@nudt.edu.cn).

I. Introduction

BLOCKCHAIN is a new distributed computing and storage paradigm that stores transaction information using a time-stamped chain structure [1], [2]. As an emerging technology, it solves the problems of trust building and privacy protection in complex production environments with multi-organization participation at a low cost based on the results of multiple technology research. It has been widely used in industry [3], smart cities [4], electronic transactions [5], intelligent transportation [6], energy/public utilities [7], healthcare [8], and other fields, and has become a research hotspot in recent years. It can be seen that the value and development potential of blockchain technology is that it is widely used in multiple fields.

With the emergence of the Internet of Everything concept, Internet of Things (IoT) technology has become a major research focus, attracting significant attention from academia and industry. The global number of IoT devices is expected to increase from 8.74 billion in 2020 to over 25.4 billion by 2030 [9]. As the number of IoT devices grows, the amount of data generated is increasing exponentially. Among these technologies, the Industrial Internet of Things (IIoT), a key application of IoT in the industrial sector, plays a crucial role in driving industrial development.

The development of network communication technology and IoT technology has led to an increasing number of IIoT devices, and there are various security attacks in the IoT, which requires a large amount of data to be stored, shared and calculated securely. This raises the issue of data security and processing efficiency for IIoT applications. The rapid growth of the IIoT requires a secure and reliable infrastructure to store and share massive amounts of data. Fortunately, blockchain technology's ability to build trust in untrusted environments [10] makes it promising to solve problems mentioned above in the IIoT. As an open, decentralized, distributed, and immutable ledger [11], blockchain has been widely used in the IIoT to ensure data security and privacy [12]–[14]. Its decentralized characteristics solve the security risks and high latency problems caused by the centralized storage and data processing in the traditional IIoT [15].

However, while blockchain improves the reliability and security of the IIoT, the growth in the number of devices makes

the performance bottleneck of blockchain systems increasingly apparent. Blockchain performance has become a key issue for blockchain as a common platform for different services and applications. Although blockchain has been applied to various distributed application scenarios such as IIoT, smart grid, Internet of Vehicles (IoV), medical health, and content distribution networks [16]–[21], most applications expect high transaction throughput performance and low latency transaction confirmation. Therefore, it is necessary and vital to develop an intelligent, self-organizing optimization scheme to optimize the performance of the blockchain.

Blockchain essence as a distributed ledger mainly used to record a variety of transaction data or other data after the generalization of transaction records; its latency and throughput are the keys to optimizing the performance of the blockchain. For example, in a networked vehicle scenario, where fast vehicle decisions are needed to reduce traffic jams or accidents [22], blockchain is required to guarantee low latency and high throughput. In a distributed power trade scenario, blockchain requires low latency to quickly complete power transactions without the high throughput needed in the IoV.

There have been studies that typically use off-chain, on-chain, traditional deep learning methods, and DRL methods to optimize blockchain performance. Among them, deep learning is a new technology. Still, it is difficult for traditional deep learning models to obtain dynamic optimal solutions to optimize the performance of blockchain systems in specific scenarios. Reinforcement learning (RL) is used to solve dynamic selection and control problems by maximizing agents' long-term rewards by interacting with the environment [23]. The traditional model-based policy iteration and value iteration methods are limited to some extent by the size of the state space and action space, so they can not complete the RL task well in the IIoT scenario. DRL combines RL with deep neural networks and becomes the primary method to solve complex dynamic programming problems and dynamic optimization problems of multiple state spaces [24].

In DRL, rapid developments in recent years have provided powerful tools and methods for solving complex problems. DRL algorithms have achieved remarkable results in dealing with highly dynamic and uncertain environments. DQN, which has attracted much attention as a combination of the strengths of deep learning and RL, offers agents the possibility to learn excellent strategies in complex tasks. However, while DQN is outstanding at tackling complex, challenging problems, its performance is still affected by the network structure. Our research examines the impact of different network layers and activation units in Double Deep Q-Network (DDQN) on blockchain performance optimization. The motivation for this focus is that by deeply understanding the role of network structural parameters, we can better adapt the model to the needs of a specific task, thereby improving its performance and generalization.

Although the current DRL-based blockchain performance optimization schemes [25]–[33] all adopt the DRL method to optimize the blockchain performance, the following three problems exist:

1) The overestimation of the Q-value that exists in the DRL method is not taken into account.
2) No research has been done on the impact of the network structure of DNNs in the DRL on performance optimization.
3) The actual scenario dataset is not used.

In a nutshell, the contributions of this paper are summarized as follows.

- We propose a DDQN-based method for dynamically selecting block sizes and adjusting block intervals to optimize blockchain throughput. DDQN fully inherits all the advantages of DQN and alleviates the overestimation problem.
- We study the influence of the number of network layers and different activation units in DDQN on the performance optimization method. Previous studies only used DRL methods to optimize the performance of blockchain systems. Still, they did not consider the impact of the network structure in the selected method on performance optimization.
- The experimental results show that our scheme has effective convergence and can improve the blockchain performance. Most previous studies did not consider the verification of blockchain performance optimization in real-world application scenarios. We combine with actual data in IIoT scenarios to verify the effectiveness of our method.

The rest of this article is structured as follows. Section II introduces the research background of blockchain performance optimization and discusses the related work. Section III provides a general presentation of the proposed approach, limiting technical formalisms. We introduce the proposed system model in Section IV. Section V discusses the expression and solution of the joint optimization problem with the DDQN method. The performance of the proposed method is evaluated through simulations in Section VI, including a case study with two real scenarios. Section VII concludes this paper and looks forward to future work.

II. Related Work

This section will briefly review the work related to IIoT systems powered by blockchain technology. Then, we summary and analyze the work related to blockchain performance optimization. In addition, some work related to DRL-based blockchain IIoT systems is discussed to provide some necessary background and introduce the motivation for our work.

A. Blockchain-Enabled IIoT

Blockchain-based Industrial Internet of Things (IIoT) has a broad research prospect, as it leverages blockchain technology to streamline the circulation and control of data within IIoT systems, thereby promoting the flow of data elements and value conversion. There are many IIoT devices supported by blockchain technology, and the normal operation of these devices will affect the whole system. In recent years, with the emergence of blockchain, the application of combining blockchain with the IoT has received extensive attention and research.

Blockchain's features of preventing tampering and decentralized consensus mechanisms can solve the security problems in IIoT systems. Shen et al. [34] proposed a blockchain-assisted cross-domain IIoT secure device authentication mechanism using consortium blockchain to establish trust between different domains. Z. Li et al. [35] implemented an energy trading system based on consortium blockchain, but for power-constrained IoT devices, there is an overload problem for power-constrained IoT devices. Misra et al. [36] proposed integrating heterogeneous IoT edge devices into blockchain nodes to extend distributed security features to resource-constrained IoT. To ensure that sensors upload data securely and reliably, Volker et al. [37] proposed a lightweight communication protocol based on blockchain. They combined it with distributed features to enhance the security of wireless sensor systems. Liang et al. [38] study focused on secure data transmission techniques for blockchain to achieve reliable transactions based on the power blockchain sharing model. Zhang et al. [39] proposed an authentication system architecture based on blockchain networks to solve the problem of fast authentication and collaborative sharing among IIoT networks and showed through experiments that blockchain can realize trust and cooperation among multiple subjects in IIoT scenarios.

In summary, blockchain-enabled IIoT can enhance system utility by improving overall system security and data reliability. Blockchain can drive a highly reliable and transparent smart IIoT with secure, verifiable data.

B. Blockchain Performance Optimization

In scenarios where IIoT resources are limited, compute and storage resources are limited per node, so there are performance bottlenecks when applying blockchain to IIoT. In addition, due to the block size and block time limitations, the transactions that can be processed by the blockchain in a certain period of time are fixed, which does not fit most transaction scenarios. At present, the performance optimization research of blockchain mainly focuses on redesigning the blockchain, storage optimization, sharding, and deep learning methods to achieve performance enhancement or improvement.

To address the challenges of blockchain systems in IIoT, J. Huang et al. [19] redesigned the blockchain structure. They proposed a blockchain with a directed acyclic graph structure based on the IIoT credit consensus mechanism for power-limited IoT devices, reducing consensus power consumption and improving system throughput.

Although blockchain performs well in privacy protection, storing a large amount of data on blockchain reduces the storage efficiency and exacerbates the blockchain bloat problem. To address the privacy issue in data sharing, Lu et al. [40] designed a secure data-sharing architecture with distributed multiple parties authorized by a blockchain using federated learning to store learned models in the blockchain to improve storage efficiency and protect data privacy. Abdella et al. [7] implemented a Peer-to-Peer (P2P) energy trading system based on the blockchain through the Istanbul byzantine fault-tolerant consensus algorithm with better scalability, success rate, and security of the blockchain-based P2P energy trading system.

The use of state channel techniques to optimize the performance of blockchain is more prevalent, and the work [41] proposes a state channel network optimization architecture to improve the scalability of blockchain. In addition, there is also a study [42] that reduces the number of necessary blockchain interactions to improve blockchain scalability in blockchain-based decentralized power transactions by using state channel techniques. The paper [43] proposed a cross-chain data migration model based on the blockchain island problem to optimize blockchain performance regarding blockchain data sharing.

Segmentation technology reduces node load by giving different parts of the blockchain to various subsets of nodes for processing to improve blockchain throughput and optimize blockchain scalability. Huang et al. [44] propose a segmented reorganization scheme based on the new data structure, which enhances the efficiency of segmented reorganization while guaranteeing the security of the segmented blockchain. Elastico [45] divides the transactions into different slices, and each slice is verified in parallel by a different collection of nodes. However, the sharding technique is limited by improving throughput while needing to ensure blockchain security and has difficulties in applying it to IIoT scenarios with large-scale characteristics.

Deep learning approaches should not be ignored, Wang et al. [46] used temporal convolutional networks to predict blockchain transaction arrival rates and proposed a blockchain performance optimization framework to train blockchain performance prediction models. Although predicting blockchain parameters through deep learning regression to optimize blockchain performance can effectively complement existing research results, such methods are limited by the quantity and quality of data, leading to additional data collection and processing costs.

Based on the above research, we find that blockchain parameters, network latency, read/write efficiency, consensus speed, interaction mechanism, security, etc., all impact the blockchain system's performance. In this paper, we optimize blockchain performance in combination with IIoT scenarios to promote a better practical application of blockchain.

C. Blockchain-Enabled IIoT With DRL

RL belongs to a type of machine learning, different from supervised and unsupervised learning, in which an intelligent body continuously interacts with its environment (i.e., takes actions) and thus receives rewards to constantly optimize its action strategy in anticipation of maximizing its long-term payoff (sum of rewards). DRL models the action value function in RL with the help of deep learning methods to achieve the optimal optimization strategy and maximize the long-term reward. DRL can solve large-dimensional and complex problems by combining it with deep neural networks. Therefore, DRL can be used to optimize blockchain performance in IIoT systems.

Liu et al. [25] used a DRL approach in the IIoT scenario to optimize blockchain scalability as much as possible without affecting other performance metrics of the blockchain.

However, not all IIoT systems need to obtain high throughput while ensuring other metrics. Liu et al. [27] dynamically select block producers and adjust block parameters to maximize the transaction throughput of the blockchain through DRL based on the IoV scenario.

Yang et al. [30] proposed an IIoT framework for blockchain and Mobile Edge Computing (MEC) integration using DQN to optimize device energy efficiency and system computation overhead. Gao et al. [28] investigated the task scheduling problem for IoV and proposed four DRL algorithms to solve the task scheduling problem by skillfully combining transactional scheduling with block assembling. Ning et al. [47] solved the task scheduling problem through a multi-objective optimization problem to minimize the delay of the ITS system and maximize data security and user utility.

Feng et al. [31] proposed a blockchain-enabled MEC framework to optimize the computation rate of the MEC system and throughput of the blockchain system through a multi-objective function. Abegaz et al. [48], to solve the resource trading problem in multi-drone-assisted IIoT, proposed an intelligent resource trading framework integrating multi-intelligent DRL, blockchain and Stackelberg gaming to manage a dynamic resource trading environment.

Zhang and Yu et al. [49] proposed a Trust-based DRL-driven framework for sharded blockchain in IoT, addressing scalability and security. The framework employs deep reinforcement learning to optimize node allocation and counter collusion attacks, enhancing throughput while safeguarding network security. Zhang and Lin et al. [50] proposed PBRL-TChain, a permissioned blockchain optimized for time-critical IoT applications. They used deep reinforcement learning to implement priority ordering, fast retransmission, and dynamic adjustment methods that significantly reduce latency and enhance reliability.

None of the above studies have considered the over-estimation problem of DQN. High dynamics and large dimensionality characterize blockchain-based IIoT systems supported by blockchain, and relying only on DQN methods may result in sub-optimal performance. Therefore, this paper uses a DDQN framework to address the blockchain performance optimization problem and mitigate the over-estimation problem in DQN. Table I summarises the characteristics of the work in the related work.

TABLE I
THE SUMMARISES OF THE RELATED WORKS

Works	IIoT	Blockchain	Performance Optimization	DRL
[34]	✓	✓	✗	✗
[35]	✓	✓	✗	✗
[36]	✓	✓	✗	✗
[37]	✓	✓	✗	✗
[38]	✓	✓	✗	✗
[39]	✓	✓	✗	✗
[19]	✓	✓	✓	✗
[40]	✓	✓	✓	✗
[7]	✓	✓	✓	✗
[41]	✗	✓	✓	✗
[42]	✗	✓	✓	✗
[43]	✗	✓	✓	✗
[44]	✗	✓	✓	✗
[45]	✗	✓	✓	✗
[46]	✗	✓	✓	✗
[25]	✓	✓	✓	✓
[27]	✓	✓	✓	✓
[30]	✓	✓	✓	✓
[28]	✓	✓	✗	✓
[47]	✓	✓	✗	✓
[31]	✓	✓	✓	✗
[48]	✓	✓	✗	✗
[49]	✓	✓	✓	✓
[50]	✓	✓	✓	✓

III. APPROACH

A. Approach Overview

Before discussing the model and algorithms in detail, let us first consider two specific application scenarios to illustrate a simple yet highly significant example. The application of blockchain in IIoT is often significantly influenced by the business factors of different IIoT scenarios. However, the performance settings of traditional blockchain systems are typically static and cannot flexibly respond to these scenario changes. This limitation makes it challenging to ensure processing efficiency in highly dynamic and complex IIoT environments. From a practical perspective, it is crucial for the performance of blockchain systems integrated into IIoT to be dynamically adjustable according to the specific demands of different scenarios.

In intelligent transportation, traffic sensors, cameras, and other devices collect real-time road traffic information and vehicle behaviour data. During peak hours, the rate of data generation increases dramatically, requiring blockchain systems to process many transactions quickly. If the blockchain is unable to adjust performance parameters based on traffic conditions dynamically, it can lead to delays in data processing, resulting in traffic congestion.

In a smart grid, power usage data and generation information need to be continuously recorded and analyzed to achieve dynamic load balancing of the grid. As weather and electricity consumption patterns change, transaction volumes fluctuate significantly, especially during peak transaction periods. The blockchain system must be able to dynamically adjust the block size and interval based on the current transaction load to ensure timely power dispatch and system stability.

Therefore, we are interested in the key question: "In a given state, what is the optimal block size and interval to process <IIoT data transaction>efficiently?" We use the DRL method to address this, which continuously observes the system state and performs corresponding actions to obtain a reward value. By maximizing this reward, the DRL model provides the best

blockchain performance parameters from possible answers, such as {block size=6MB, block interval=8s}. Through this approach, we can significantly enhance the performance of blockchain systems in IIoT applications, enabling them to effectively adapt to varying transaction loads across different scenarios, thereby ensuring processing efficiency and system security.

B. A Running Example

To better illustrate the workflow and interaction of the different components in the proposed blockchain-based IIoT system, we will walk through an example involving a smart grid application within an IIoT environment. Imagine a smart grid system where multiple households and industrial facilities generate and consume electricity. Each household and facility has smart meters that monitor electricity usage and generation. These smart meters regularly collect data on electricity consumption, generation (from renewable sources like solar panels), and any excess energy sent back to the grid.

1) **Data Collection (IIoT Facilities and Users Layer):** Each smart meter in the system collects data on electricity usage and generation in real-time. For instance, Household A generates 10 kWh of solar energy and consumes 8 kWh, resulting in an excess of 2 kWh. The smart meter stores this information and periodically prepares to upload it to the data aggregator.
2) **Data Aggregation (Data Aggregation Layer):** The smart meters from multiple households and industrial facilities send their data to the local data aggregator. For example, the aggregator for a specific neighbourhood receives data from all smart meters in that area, including the 2 kWh excess energy data from Household A. The aggregator processes and packages the data to reduce the volume and enhance security before sending it to the blockchain system. This reduces the communication load on the blockchain network and ensures only relevant, aggregated data is transmitted.
3) **Data Transmission and Verification (Blockchain System Layer):** The packaged data from the aggregator is sent to the blockchain system for verification and recording. Using the PBFT consensus protocol, the blockchain randomly selects some consensus nodes to validate the transaction. The selected node verifies the integrity of the data (e.g., confirming that the 2 kWh surplus reported by Household A is accurate) and, if valid, includes it in a new block. Once verified by other nodes, the block is added to the blockchain, ensuring an immutable transaction record.
4) **Optimization (DRL Agent Layer):** The DRL agent monitors the system's performance, such as the time it takes to verify and add a block to the blockchain and its overall security. If the agent detects that the system is experiencing delays (e.g., due to a surge in transactions), it dynamically adjusts parameters like block size or interval to optimize performance. For instance, it might reduce the block interval during peak hours to ensure faster processing of transactions. Over time, the DRL agent learns the optimal settings for different conditions, ensuring that the blockchain system remains efficient and secure even as transaction volumes fluctuate.

C. Markov Decision Processes

In short, Markov Decision Processes (MDPs) are a cyclic process in which an agent interacts with the environment by continuously taking actions to change its state, obtain rewards, and interact with the environment. MDP can be expressed as $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathrm{Ps}, \mathrm{a}, \mathrm{R}, \gamma >$, where S and A represent a limited set of states and actions respectively, $\mathrm{Ps}, \mathrm{a}(s'|s,a)$ represents the probability of taking action a in state s to transfer to a new state s', $\mathrm{R}(s,a)$ represents the immediate reward after the agent takes action a, $\gamma \in \ (0,1]$ represents the discount factor, which is a constant. The solution process of MDP is the process of finding the optimal strategy to maximize future returns. The solution process can be divided into two steps: prediction and action. The prediction step evaluates the corresponding state value function and state-action value function through the given strategy, and the action step selects the optimal action corresponding to the current state based on the value function.

D. Deep Reinforcement Learning

The core concept of RL lies in learning to learn, in other words, mapping situations to behaviours to maximize reward signals. DRL combines deep learning with RL and is an effective method for achieving optimal optimization strategies and maximizing long-term rewards.

Given an action in MDP, there is an action-value function based on the state and the expected payoff of the action.DRL is an iterative value-based approach that generally evaluates actions in different states utilizing an action–state value function, as shown below.

$$\begin{aligned} Q^{\pi}(s,a) &= \mathbb{E}_{\tau\sim\pi}[R(\tau) \mid S_0 = s, A_0 = a] \\ &= \mathbb{E}_{A_t\sim\pi(\cdot|S_t)}\left[\sum_{t=0}^{\infty}\gamma^t R(S_t,A_t) \mid S_0 = s, A_0 = a\right] \end{aligned} \tag{1}$$

where $\mathbb{E}_{A_t\sim\pi(\cdot|S_t)}[*]$ denotes the mathematical expectation, and γ is a discount factor that weighs short-term rewards against future rewards. $R(S_t, A_t)$ is the short-term reward for time period t under strategy π. The agent will iterate the action-state values based on (1), where α is the learning rate, and $\max_{a'} Q(s',a')$ is the maximum action-state value in the new state.

$$Q^{\pi}(s,a) = Q(s,a) + \alpha\left[r(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{2}$$

IV. System Model

In this section, we will introduce the architectural model of the proposed blockchain-based IIoT system.

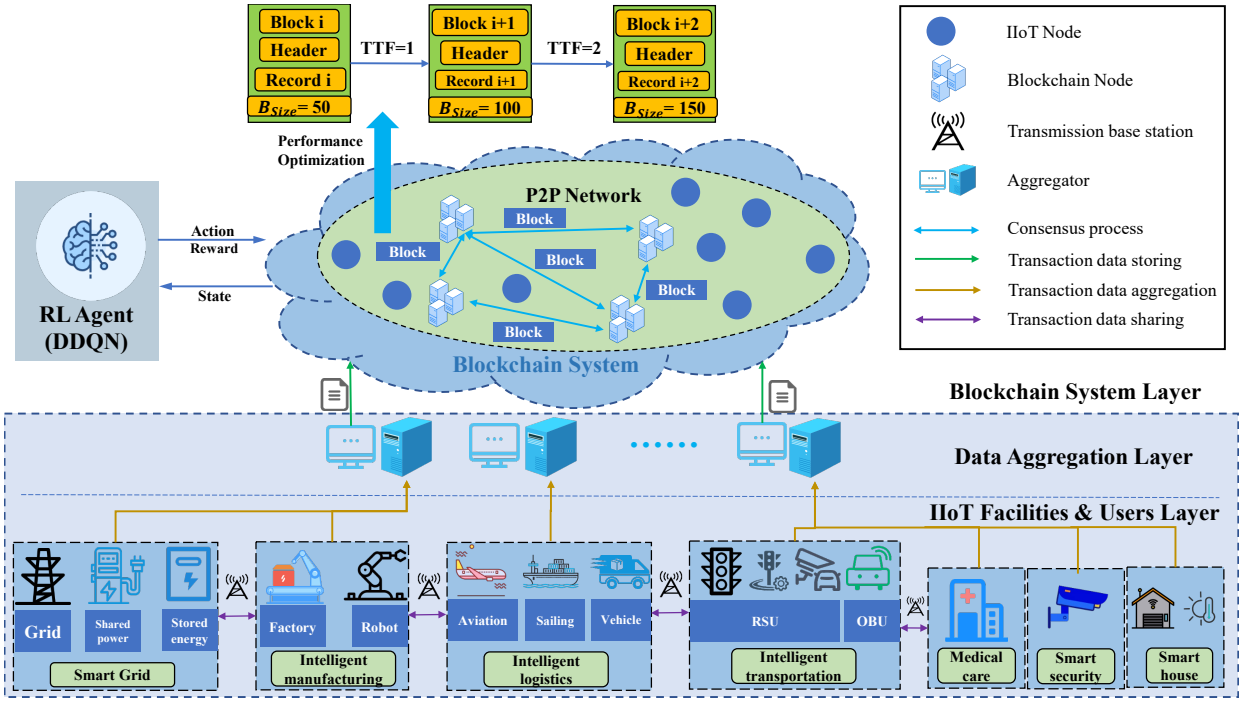


Fig. 1. Structure of the proposed system model.

A. System Overview

As shown in Fig. 1, the system model consists of four parts: 1) the IIoT device and user layer; 2) data aggregation layer; 3) blockchain system layer; and 4) DRL agent. Our proposed four-part model is described as follows. Fig. 1 illustrates the complete architecture of the proposed blockchain-based IIoT system, detailing the interactions between the various layers. The figure emphasizes the flow of data from IIoT devices to the blockchain system, highlighting the role of data aggregators in managing communication and ensuring efficient data processing. Additionally, it depicts the role of the DRL agent in optimizing system performance through real-time adjustments based on incoming data and current system states. The role of each component in Fig. 1 is described in detail below.

B. IIoT Facilities and Users Layer

This layer is mainly composed of IIoT devices and users; according to the different application scenarios in the IIoT, this layer is distributed with many IIoT devices, such as smart grid equipment, intelligent transportation equipment, smart medical equipment, and so on. These smart device terminals collect and store data in various formats, such as pictures, videos, and documents, through photoelectric devices, sensor devices, etc., and complete data sharing through the IIoT network. We assume that these different categories of devices are subsystems, and these devices regularly upload data to the aggregator. Then, the aggregator completes the data aggregation and packaging upload to the blockchain system for data consensus and storage.

C. Data Aggregation Layer

Each IIoT subsystem has its corresponding data aggregator, and each data aggregator receives various IIoT data from the local subsystem. Data aggregators can reduce the transmission consumption of direct communication between terminal devices and the blockchain system and alleviate the access pressure and potential security risks caused by the direct interaction of massive terminal devices with the blockchain system. This decoupled architectural pattern reduces the operating load of IIoT devices with low power consumption and limited computing resources and improve the stability of blockchain-enabled IIoT systems.

Let the subsystem set be $\mathcal{C} = 1, 2, \ldots, S$, then the set of data aggregators can be expressed as $\mathcal{C}^* = \mathcal{C} = 1, 2, \ldots, S$. Data aggregators collect IIoT data from corresponding subsystems. The load and computing power of each data aggregator can be denoted by: $E(t) = (e_1(t), e_2(t), \ldots, e_s(t))$ and $C(t) = (c_1(t), c_2(t), \ldots, c_s(t))$. After data processing or encapsulation, the aggregator transmits the unverified data to the blockchain system.

D. Blockchain System Layer

The blockchain system layer mainly completes the task of block generation and block consensus. In a blockchain-supported IIoT system, this paper assumes that there are n consensus nodes in the blockchain system, expressed as $N = \{1, 2, \ldots, n\}$. The block size is B_{Size} (MB), and the block interval is B_{Interval} (s). The consensus node is responsible for verifying the data submitted by the aggregator and then records the verified blocks on the blockchain. For IIoT devices with limited computing resources, consensus protocols such as PoW and PoS have poor compatibility with application

scenarios. They cannot provide large system throughput to meet the needs of IIoT systems. Therefore, the blockchain system layer adopts PBFT, widely adopted by Hyperledger, EOS, etc., as a consensus protocol.

Initially, the aggregator forwards unverified transaction data and verification requests to the blockchain system. Upon receiving the data, the blockchain system randomly assigns a node as the primary validation node to complete the validation of transaction block data and information. Specifically, the primary node verifies the block's signature and Message Authentication Code (MAC). If the information is successfully verified, the block is considered valid. Subsequently, the signatures and MACs of each transaction within the block are further verified.

E. RL Agent(DDQN)

The primary objective of DRL agents is to find the optimal mapping between states and actions that leads to maximum rewards. The agent utilizes available state samples and computes estimated rewards achievable by taking action from the state. The DRL agent used in this article is DDQN. DDQN optimizes system performance by dynamically adjusting block size and interval parameters. The agent continuously learns the optimal state-to-action mapping, thereby maximizing long-term rewards and ensuring that the blockchain operates efficiently, even under varying load conditions.

V. Problem Definition and Solution

A. Performance Analysis Indicators for Blockchain

Blockchain systems are decentralized distributed ledgers, and developers will always encounter Mundellian Trilemma when building blockchains. The blockchain in the application process faces scalability, security, and decentralization and can only sacrifice one party to meet the other two parties. In combination with the above theorem, we mainly consider the following attributes when analyzing the performance of the blockchain.

1) Scalability: Scalability is vital in blockchain applications. A well-scalable blockchain platform can efficiently handle the high volume and fast transactions generated by different users. Transaction per second (TPS) refers to system throughput and is one of the most important indicators to measure the performance of a blockchain system. The higher the TPS, to a certain extent, the more stable the performance of the blockchain system. If the TPS is too low, it can easily cause network congestion.

Generally, the TPS of a blockchain system is calculated as follows:

$$TPS = \frac{Number_{transactions}}{Time_{response}} \tag{3}$$

where $Number_{transactions}$ represents the number of transactions, and $Time_{response}$ represents the response time to process the transaction.

Three main parameters affect system throughput in a blockchain system: block size, block spacing, and transaction size. Where the block size represents the maximum number of transactions that can be stored in the block, the block interval represents the time required for new blocks to be generated; that is, the block time and the transaction size represent the size of each transaction stored in the block.

Thus, the transaction throughput of a blockchain can be denoted by:

$$\Psi(t) = \frac{\lfloor B_{\text{Size}}/T_{\text{Size}} \rfloor}{B_{\text{Interval}}} \tag{4}$$

where T_{Size} is the average transaction size, and B_{Size} and B_{Interval} are the block size and the block interval (the average time required to produce a new block) at time slot t,respectively.

Through the above formula, we find that the throughput of the blockchain system can be improved by increasing the block size or reducing the block time and transaction size, thereby optimizing system stability and availability.

2) Latency: In deep learning, latency refers to a model's time to process a single data unit. In blockchain, latency is the time it takes for the network to verify and execute transactions to store the information on the blockchain. This article represents blockchain latency using finalization latency (LTF/TTF). LTF measures the minimum time required to write irreversible transactions in the blockchain. Typically, transaction processing consists of two phases: block generation and block verification. Therefore, the LTF of a transaction consists of the block generation time and the block verification time.

Let $T_c(t)$ represent the time cost of the consensus process. We divide the consensus process into two parts: message propagation and message verification. Message verification consists of signature verification, message authentication codes (MACs) generation, and MAC verification. Then, the delay of the consensus process in time slot t is denoted by:

$$T_c(t) = T_T(t) + T_\nu(t) \tag{5}$$

where $T_T(t)$ and $T_\nu(t)$ represent the message propagation time and the verification time in time slot t,respectively.

Therefore, TTF is represented by

$$T_{\text{latency}}(t) = T_c(t) + T^I(t), \quad T^I(t) = B_{\text{Interval}} \tag{6}$$

where $T_c(t)$ and $T^I(t)$ are the time spent in the consensus process and the time spent in producing a block, respectively.

IIoT systems often want low latency, so a single block should be written to the blockchain over multiple consecutive block intervals. At the same time, to meet the blockchain's finality, the delay should be smaller than the block generation cycle. Therefore, TTF needs to satisfy the following constraints:

$$T_{\text{latency}}(t) \le \xi \cdot T^I(t), \quad \xi > 1 \tag{7}$$

where ξ represents the number of block intervals required for the final confirmation of the transaction.

B. Modeling Optimization Problem using MDP

To maximize blockchain performance in IIoT scenarios, joint decision-making on representative parameter adjustments that affect blockchain performance is required. We use DRL to solve the joint optimization problem. Therefore, the joint

optimization problem needs to be expressed as an MDP consisting of system state, system action, and reward function. The specific expression is shown in the Fig. 2.

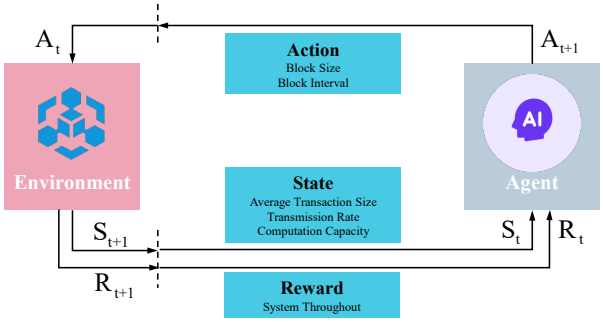


Fig. 2. MDP formulation of joint optimization problem.

1) State Space: The arrival of new transactions in the blockchain system follows the Poisson point process. The Poisson point process is a continuous-time random process whose state may change at any time. The agent learns experience and updates decisions by observing the state at each time slot.

This article expresses the system's state at each discrete time element t ($t = 1, 2, 3, \ldots, T$) as $S^{(t)}$. $S^{(t)}$ is defined as the set of average transaction size, aggregator transmission rate, aggregator computing power, and transaction arrival rate.

$$S^{(t)} = [T_{\text{Size}}, \mu, \sigma, T_R]^{(t)} \tag{8}$$

where T_{Size} represents the average transaction size, μ represents the transmission rate between the aggregator and the blockchain system, σ represents the computing power of the aggregator, and T_R represents the transaction arrival rate.

2) Action Space: Through the previous definition of blockchain performance analysis, we can find that block size and block interval are important performance parameters of the blockchain system. Therefore, we define the action space at decision time t as

$$\mathcal{A}^{(t)} = \left[B_{\text{Size}}, T^I\right]^{(t)} \tag{9}$$

According to the limits fractional method, the block size B_{Size} and block interval T^I respectively decisions are respectively given by

$$B_{\text{Size}}(t) \in [1, 2, \ldots, \dot{B}_{\text{Size}}] \tag{10}$$

$$T^I(t) \in [0.5, 1, \ldots, \dot{T}^I] \tag{11}$$

where $\dot{B}_{\text{Size}}$ and $\dot{T}^I$ are the block size limit and the maximum block interval, respectively.

3) Reward Function: Optimization goals are achieved by maximizing long-term rewards. We define the reward function as maximizing transaction throughput while ensuring the finality and security of the blockchain system.

The consensus protocol mainly ensures the security of the blockchain system in the IIoT scenario. Among many consensus protocols, the consensus protocol that can better support a more significant number of power-limited IIoT devices is PBFT. The PBFT consensus mechanism only requires some participants to be honest nodes to ensure the system's security under all network conditions [51]. To ensure the security of the blockchain system, for a single chain using the PBFT protocol, the number of malicious nodes f and the total number of verifiers v must satisfy

$$3f + 1 \leq N_v \tag{12}$$

Therefore, the number of malicious nodes f should satisfy the following constraints.

$$f \leq N_m, \quad N_m = \left\lfloor \frac{N-1}{3} \right\rfloor \tag{13}$$

where N_m denotes the maximum number of malicious participants that the system can tolerate.

When the system security and delay constraints are met, the reward function can be denoted by

$$\begin{aligned} &\max_{A} \quad Q(S, A) \\ &\text{subject to:} \\ &\quad (C1): T_{\text{latency}}(t) \leq \xi \cdot T_I(t), \quad \xi > 1 \\ &\quad (C2): f \leq N_{\text{m}}, \quad N_{\text{m}} = \left\lfloor \frac{N-1}{3} \right\rfloor \end{aligned} \tag{14}$$

where $Q(S, A)$ is the long-term reward of the system.

The instant reward $R(t)$ is defined as follows

$$R(t) = \begin{cases} \Psi(t), & \text{if C1, C2 are satisfied} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

In summary, long-term incentives can be denoted by

$$Q(S, A) = \left[\sum_{t=0}^{\infty} \rho^t R(t) \left(S^{(t)}, \mathcal{A}^{(t)}\right)\right] \tag{16}$$

The above equation weighs immediate and future rewards by a discount factor $\rho \in (0, 1]$.

C. Solving Optimization Problem via DDQN

Adding deep neural networks to DRL improves the ability to handle complex, large-dimensional problems compared to previous RL methods. Previous traditional methods have had difficulty solving the uncertainty problems existing in the complex environment of IIoT, and DRL can better deal with uncertainty and nonlinear problems, providing a new solution to solve the performance optimization problem of blockchain in IIoT.

IIoT systems integrating blockchain and data aggregators have high-dynamic and large-dimensional characteristics and suffer from the joint optimization objective problem. Therefore, the state space represented is large and complex in dimension, and it is impossible to explore each state and obtain the action-state values under each policy. Fortunately, DRL based on deep neural networks can obtain low-dimensional features from large-dimensional data by adjusting the network parameters [52], and at the same time, use the approximate action-state function to get the action-state values, which allows the agent to output the approximate Q-values of all the possible actions after inputting the state features obtained from observation into the DNN.

Traditional DQN has instability in the training process, which may lead to poor or failed training results. DDQN completely inherits the advantages of DQN, and at the same time, it can solve the problem of over-estimation of Q-value in the training process and improve training efficiency and stability. Therefore, we use the DDQN method [53] to optimize long-term rewards. The learning process uses two sets of weights θ and θ^-. The former is used for online learning, with a small batch of data randomly sampled from the experience replay queue D as input. The latter is periodically updated to the weights of corresponding terms that change more frequently, where 160 steps are used as the update period. This detailed process is shown in Algorithm 1.

Algorithm 1 DDQN-Based Performance Optimization Algorithm for Blockchain-Enabled IIoT

Initialization:
Initialize the system state $S^{(t)}$;
Initialize the action network of $Q(S, A, \theta)$ with weights and biases θ;
Initialize the target network of $Q^-(S, A, \theta^-)$ with weights and biases θ^-;
Initialize replay memory M with the P;
Input maximum training episode $\dot{E}$, maximum training step $\dot{S}$;
Initialize the greedy coefficient ϵ.
DDQN Learning Process:
for each training episode $= 1, \ldots, \dot{E}$ **do**
  for each training step $= 1, \ldots, \dot{S}$ **do**
    Select a random probability δ;
    if $\delta < \epsilon$ **then**
      Select a random action
    else
      $A(t) = \arg\max_A Q(S(t), A(t), \theta)$
    end if
    Decrease ϵ;
    Execute action $A(t)$ to adjust parameters, and observe reward $R(t)$ and proceed to next state $S(t+1)$;
    Store the experience $[S(t), A(t), R(t), S(t+1)]$ into the replay memory M;
    Random Sample a mini-batch M^- of state transition $[S(i), A(i), R(i), S(i+1)]$ from the replay memory M;
    Calculate the target Q^- value from the target Q network: $y(t)^{\text{DDQN}} = r_t + \rho Q(S_{t+1}, \text{argmax}_A Q(S_{t+1}, A_t; \theta); \theta^-)$.
    Update target network by performing the gradient descent of loss function for every G step: $L(\theta) = \left(y(t)^{\text{DDQN}} - Q(S_t, A_t; \theta)\right)^2$.
  end for
end for

The training process of DDQN is shown in Fig. 3, where the DDQN action network is Multi-Layer Perceptron. The MLP consists of an input layer, a hidden layer, and an output layer; the different layers are fully connected. In implementing blockchain performance optimization with the DQN algorithm, deep neural networks are needed to extract information from the system state volume to fit the RL gains. Through hidden layers, MLP can obtain state information from inputs with long-term and short-term impacts on the outputs.

The DDQN training process consists of two main parts: a deep neural network and an online deep Q-network. The former approximates the action value function to evaluate the action, and the latter dynamically updates the network parameters and accomplishes the policy selection. The DDQN lets the agent interact with the environment, continuously learn the policy selection from experience, and finally obtain the optimal policy based on the value function. Specifically, at each decision moment, the agent obtains system state information through the simulation environment and then selects and executes an action consisting of block size and interval. Finally, the agent receives a short-time reward and updates the system environment to the next state. That is, the environment gives an initial state s_t, and then the action network outputs an action a_t based on the state of the environment and feeds it back to the environment, which returns the corresponding short-term reward r_t and the next state s_{t+1} based on the action. The agent repeats this process, interacting with the environment continuously, and ends at the termination state.

To learn the best action for a particular state, some random exploration actions are first taken. All the experiences generated by exploration, including states, actions taken, rewards obtained and new states, are stored in the experience replay cache. As shown in Fig. 3, the DDQN inputs a certain amount of randomly sampled data from the experience cache into the action network through the small batch sampling of exploration strategy pairs from the simulation environment, which avoids the correlation of experience due to the temporal order and is conducive to the enhancement of the agent's exploration capability. The action network generates the approximate value $Q(S, A, \theta)$, minimizes the DDQN loss through the loss function, continuously adjusts the weights and bias values in the action network, and updates the parameters of the target network at every G-steps. The temporary preservation of the network parameters through the target network reduces the correlation between the approximate Q value of the action network and the target Q value, making the training process more stable. Finally, the action network selects the action values to be fed back to the environment. The Q-network is trained by minimizing the loss function given by

$$L(\theta) = \left(y(t)^{\text{DDQN}} - Q(S_t, A_t; \theta)\right)^2. \tag{17}$$

The target for use by DDQN is

$$y(t)^{\text{DDQN}} = r_t + \rho Q(S_{t+1}, \text{argmax}_A Q(S_{t+1}, A_t; \theta); \theta^-). \tag{18}$$

Differentiating the loss function with respect to the weights we arrive at the following gradient

$$\Delta_\theta L(\theta) = 2 \cdot \left(y(t)^{\text{DDQN}} - Q(S_t, A_t; \theta)\right) \cdot \left(\Delta_\theta y(t)^{\text{DDQN}} - \Delta_\theta Q(S_t, A_t; \theta)\right) \tag{19}$$

We optimize the loss function by stochastic gradient descent (SGD).

The specific details of the action network are shown in Fig. 4. In DDQN, we use an action network and a target network

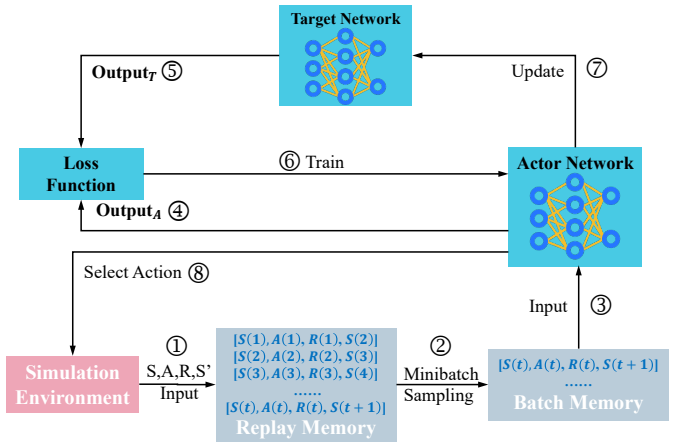


Fig. 3. DDQN training architecture diagram.

with the same structure. Both networks consist of an input layer, two densely connected hidden layers, and an output layer, where each hidden layer contains 32 neurons. To scale the output of the hidden neuron layers before computing the activation function and to reduce undesirable effects of the training process, such as gradient vanishing and gradient explosion, we add a batch normalization layer after each hidden layer. The output of the action network is an approximation of the Q-value of each available action.

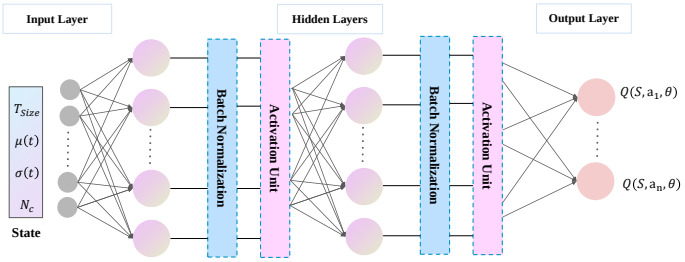


Fig. 4. Action network architecture.

VI. Experiment Results and Analysis

In this section, we first introduce the simulation environment and parameters. The simulation experiments are conducted using the proposed method, and the results are analyzed and discussed. Additionally, the qualitative analysis through some key features and related work is summarized as shown in Table IV.

To evaluate the validity of the proposed approach, we consider the following RL-based and non-DRL schemes.

1) *Our Work*: Using the method proposed in this paper.
2) *Q-learning-based Scheme* [54]: The problem is optimized using Q-learning methods.
3) *DQN-based Scheme* [55]: Optimize the proposed problem by DQN.
4) *Our Schemes Without Block Size Adjustment*: The size of the generated blocks is fixed.
5) *Our Schemes Without Adjusting Block Interval*: The interval for generating blocks is unchanged.
6) *Static Scheme*: No adjustments or selections are adopted.

A. Simulation Parameters

We use a workstation with hardware configuration of Intel Core i7-10700 2.90GHz CPU and NVIDIA GeForce RTX 2060 GPU to complete the experiments. The IDE for the experiments is Pycharm, the programming language version is Python 3.8, and the deep neural network included in the DRL-based framework is implemented using TensorFlow 2.4.1. Python and the framework are widely used for deploying experimental environments and deep learning algorithms. We consider the experimental scenario with a blockchain system with 20 nodes and 6 aggregators. The parameter settings used in the simulation are summarized in Table II.

TABLE II
Simulation Parameters Settings

Symbols	Parameters	Value
n	The number of nodes	20
C	Number of subsystems and aggregators	6
T_{Size}	Average transaction size	200B
$\dot{B}_{\text{Size}}$	Maximum block size	10MB
$\dot{T}^I$	Maximum block interval	10s
c_s	Aggregator computational power	1-20GHz
S_{cn}	Data transfer rate between the aggregator and the blockchain system layer	10-100Mbps
α	The computing cost for verifying signatures	2 MHz
β	The computing cost for generating/verifying MACs	1 MHz
ξ	The number of block intervals required for a new block to be verified	6
ϵ	The greedy coefficient	0.8
P	Replay memory size	3600
M^-	Sample mini-batch	128
$\dot{E}$	Maximum training episode	6000
$\dot{S}$	Maximum training step	160

Meanwhile, we also study the impact of the number of layers of the action network and the activation function on the reward and running time of the proposed model in deep Q-networks. Table III lists the relevant parameter settings.

TABLE III
Model Parameters Setting

Parameter	Setting
No.layers	{2,4,6,8}
Activation	{ReLU, tanh, Mix(ReLU & tanh)}
Learning rate	0.0001
Optimizer	Adam

B. Analysis of Convergence

We provide a portion of the experimental result plots to analyze the convergence performance of the proposed and baseline schemes. Fig. 5 illustrates the convergence of the proposed scheme at 10,000 episodes with different learning rates. When the learning rate is a moderate $1e^{-4}$, the convergence rate is moderate while ensuring that better strategies are

learned and better rewards are obtained. Therefore, we choose $1e^{-4}$ as the fixed learning rate in the simulation experiments.

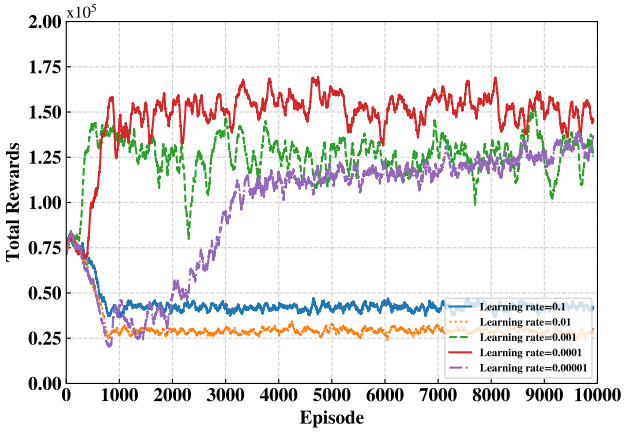


Fig. 5. Convergence performance of different learning rates.

Fig. 6 shows the convergence performance of our scheme compared to different baseline schemes. Compared to the other two DRL-based baselines, our proposed scheme achieves higher total rewards while maintaining a more stable convergence performance. Q-learning-based schemes are ineffective in exploring and evaluating policies in IIoT scenarios, as the limited and incomplete nature of exploration prevents the model from learning better policies. DQN and DDQN use deep neural networks to estimate Q-values, enabling them to handle large-dimensional state spaces. However, the DQN-based scheme requires more time in the early stages of exploration and suffers from the problem of overestimation. DDQN, on the other hand, uses a target network to evaluate the Q-value of the next state when selecting an action, minimizing the overestimation issue. Therefore, DDQN outperforms DQN in addressing the overestimation problem and improving stability, and it surpasses Q-learning in handling large-dimensional problems and enabling effective exploration.

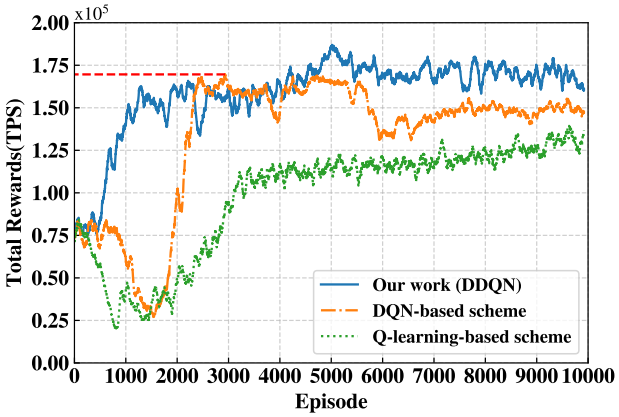


Fig. 6. Convergence performance of different schemes.

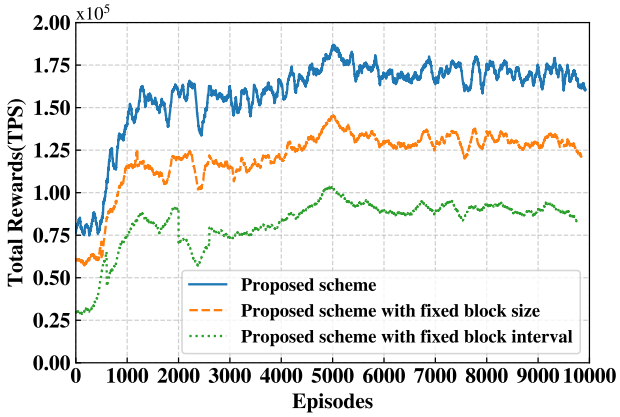


Fig. 7. Total rewards under different schemes.

Fig. 7 illustrates the convergence performance of our proposed DDQN-based performance optimization method under different scenarios. As shown in the figure, the total reward value is low at the beginning of the learning process. However, as the number of episodes increases, the reward value rises and stabilizes after approximately 3,000 episodes, indicating that our proposed scheme has good convergence performance. Additionally, we observe that the proposed scheme achieves higher throughput than the other two schemes, demonstrating its advantage. Both the fixed block size scheme and the fixed block interval scheme have some impact on system performance.

C. Performance Analysis

We compare the proposed scheme with the baselines of limiting block size, limiting block spacing, and the static scheme. Fig. 8 and Fig. 9 illustrate the system throughput of the proposed scheme compared to the baseline approaches under different parameters.

Fig. 8 shows the effect of different block size parameters on throughput. We find that as the block size increases, the throughput of all three schemes except the fixed block size scheme gradually increases. However, the throughput does not keep rising as the block size increases because the block spacing limits the maximum number of transactions in each block.

As shown in Fig. 9, the transaction throughput of all schemes increases as the block interval increases. Compared to the other three baseline schemes, our scheme achieves the highest throughput, followed by the fixed block size scheme, the fixed block spacing scheme, and the static scheme. The fixed block size scheme outperforms the fixed block interval scheme in terms of throughput; this is because, with fixed block size, the system can adjust the block interval to increase transaction throughput, whereas the fixed block interval scheme cannot generate more blocks within a short period. However, it can still increase transaction throughput by adjusting the block size. The static scheme performs the worst among the four, validating the effectiveness of our proposed DDQN-based scheme. Meanwhile, changing the block interval

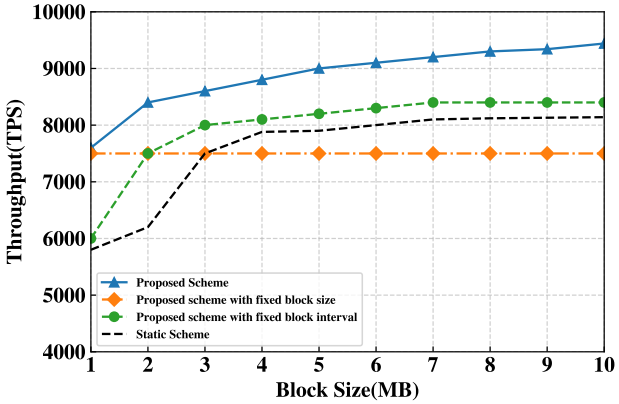


Fig. 8. Throughput with different block size.

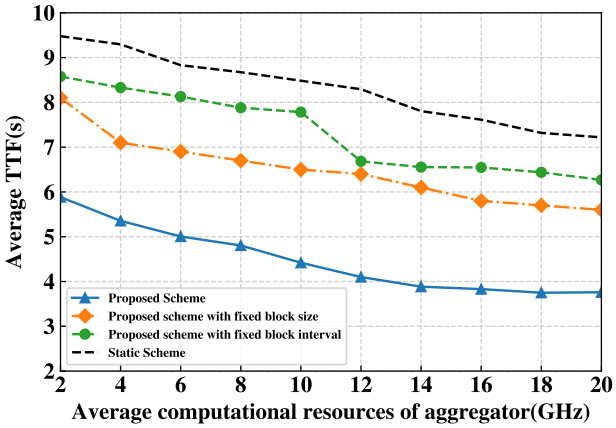


Fig. 10. The latency versus average computational resources of aggregator.

to optimize blockchain performance is more effective than adjusting the block size.

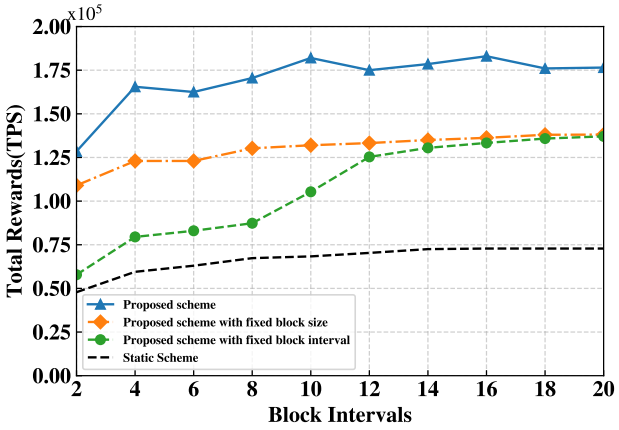


Fig. 9. Total reward with different block interval.

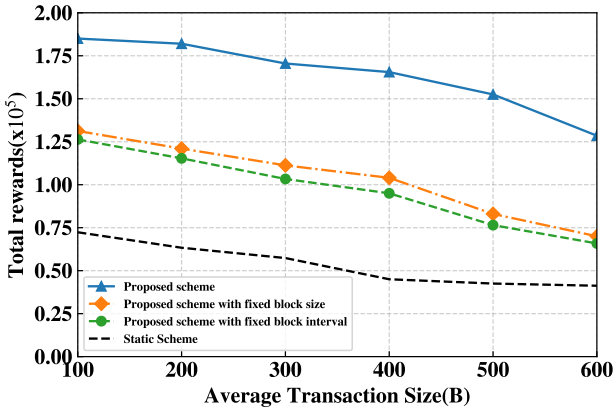


Fig. 11. Total reward versus average transaction size.

In Fig. 10, we investigate the impact of the average computational resources of the aggregator on the system latency. As with the general common sense results, the system latency of all four schemes shows a decreasing trend as the computational resources of the data aggregator increase. An interesting observation is that the fixed-block-size scheme has lower latency than the fixed-block-spacing scheme, and a plausible explanation is that the fixed-block-size scheme reduces the system waiting time by adjusting the block spacing. Our proposed scheme obtains a lower average latency than the other three baseline schemes, again validating the effectiveness of our proposed DDQN-based approach to optimize blockchain performance.

The results of the experiments with different average transaction sizes are shown in Fig. 11. From the figure, we can observe that the rewards of all the schemes decrease as the average transaction size increases. This is because as the average transaction size increases, the number of transactions that can be recorded within a block keeps decreasing. Moreover, we can find that as the average transaction size varies, our proposed scheme gets the highest reward, followed by the fixed block size scheme and the fixed block spacing scheme, with the static scheme performing the worst.

D. Network Structure Analysis

Fig. 12 shows the convergence of the proposed scheme over 2000 episodes with different numbers of layers and activation units. We selected tanh, ReLU, and a mixture of both as nonlinear activation units, combined with varying numbers of layers, to explore optimal configurations.

A comprehensive analysis concluded that the model's convergence gradually accelerates as the number of hidden layers increases. However, more layers do not always lead to better performance. Specifically, when the number of layers reaches 8, we observe that the fluctuation in total rewards becomes more pronounced compared to other configurations. The optimal convergence is achieved with six layers. Regarding activation units, different nonlinear activation functions perform differently depending on the layer configuration.

Fig. 13 shows the scheme's average reward for 2000 episodes of different layers and activation units. In the three activation unit modes, the average reward increased with the number of layers, but decreases when the number of layers reaches 8. When the activation unit selects Mix mode, the average reward value is the best performance of the three.

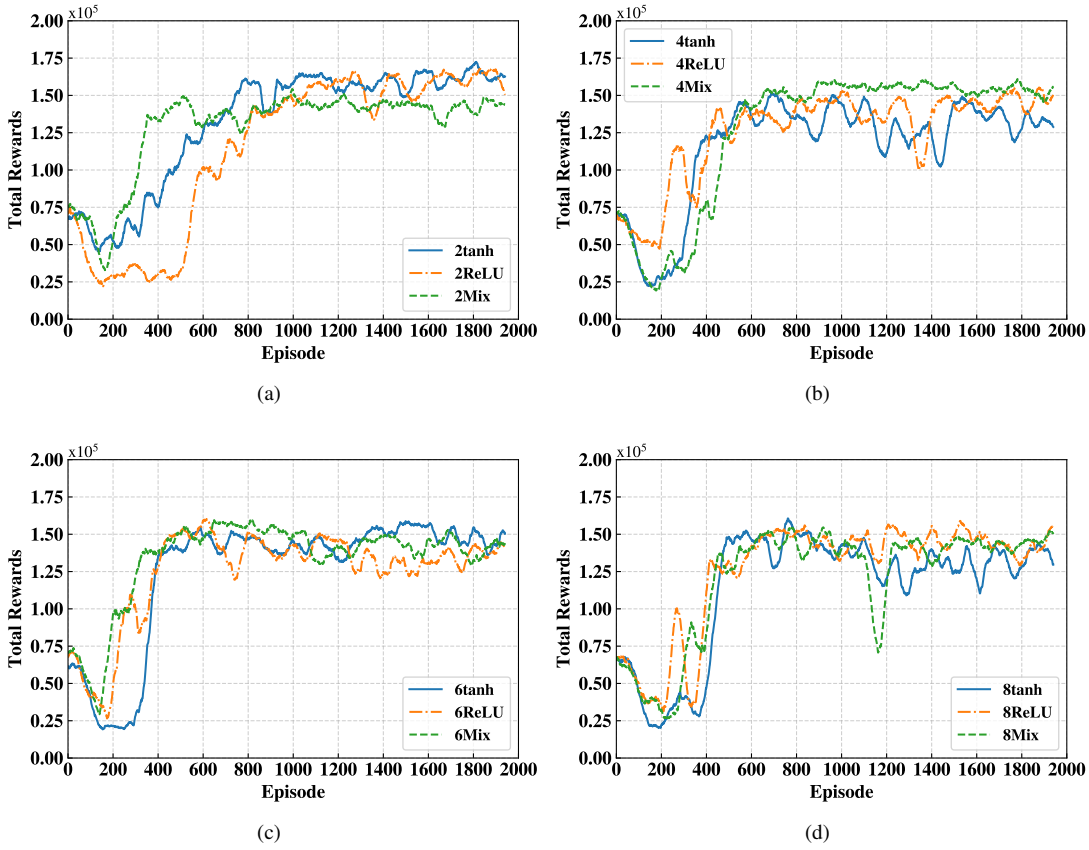


Fig. 12. Impact of activation units and layers on total rewards.

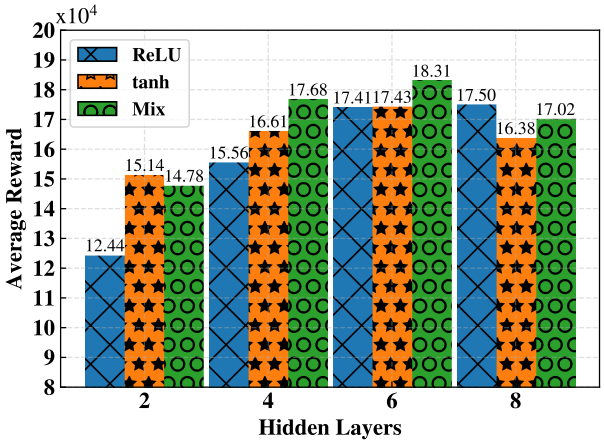


Fig. 13. Impact of activation units and layers on average rewards.

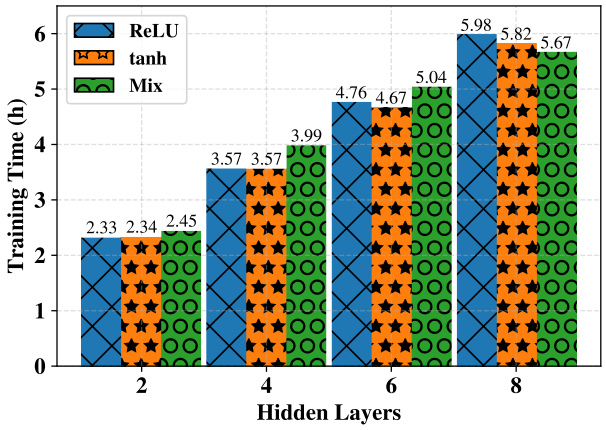


Fig. 14. The time complexity for training. The training time increases with additional layers. The type of activation units shows little impact on the complexity of model training.

Fig. 14 gives the time complexity analysis of the proposed scheme at 2000 episodes with the different number of layers and activation units. From the study, it can be concluded that the training time increases with the number of layers. The type of activation unit has little effect on the complexity of model construction. However, there is still a difference in the time spent by different activation units, and overall, the Mix model takes more time than the other two models because the Mix model combines the computational complexity of ReLU and tanh. An interesting observation is that when the number of layers is small, the ReLU mode requires less training time than tanh and Mix due to its lower computational complexity, as it avoids exponential operations. However, as the number of layers increases, the training time for ReLU also increases, eventually surpassing that of the Mix mode.

To fully demonstrate the innovation and effectiveness of RLChain, we compare it with other related works. Table IV

provides a qualitative comparison between existing blockchain performance optimization efforts and our proposed scheme. The table highlights the strengths and limitations of each work based on several key features.

Through the comparison, we find that our scheme and most of the existing work employ Deep Reinforcement Learning (DRL) methods (C_1), which are crucial for solving complex decision-making problems in blockchain systems. There are three pieces of work that consider scalability, latency, and security together in performance optimization, but most of the research, comprising nine studies, focuses on only one or two of these factors. Scalability (C_2) remains a significant challenge in blockchain systems. Our scheme, like [25] and [26], effectively addresses the scalability issue. This feature is critical to ensure that blockchain networks do not experience performance degradation as the number of transactions and participants increases. Latency and completion time (C_3) are important metrics for evaluating the responsiveness of a blockchain system. Our scheme with [25] and [32], among other works, focuses on reducing latency, which is critical for real-time applications. While much of the existing work prioritizes security, there is also work such as [31] and [56], where security is either not fully addressed or remains unclear. Our scheme ensures security during blockchain operation. One of the significant advantages of our scheme is the ability to support aggregator integration (C_5) and perform deep neural network (DNN) parameter analysis (C_6), which most of the existing work fails to address. These capabilities are essential to optimize the performance of DRL models and ensure efficient processing in large-scale blockchain networks. The comparison with related work effectively demonstrates the innovation, effectiveness, and scalability of RLChain.

TABLE IV
THE COMPARISON OF THE EXISTING WORKS AND OUR SCHEME

Works	C_1	C_2	C_3	C_4	C_5	C_6
[25], 2019	✓	✓	✓	✓	✗	✗
[26], 2020	✓	✓	N/A	✓	✗	✗
[31], 2020	✓	✓	✓	N/A	✗	✗
[32], 2021	✓	✓	✓	✓	✗	✗
[47], 2022	✓	✗	✓	✓	✗	✗
[56], 2022	✓	✗	✓	N/A	✗	✗
[22], 2023	✗	✓	✓	✓	✗	✗
[28], 2023	✓	✗	✓	N/A	✗	✗
[33], 2023	✓	✗	✗	✗	✗	✗
[46], 2024	✗	✓	✓	N/A	✗	✗
[49], 2024	✓	✓	✗	✓	✗	✗
[50], 2024	✓	✓	✓	✗	✗	✗
Ours	✓	✓	✓	✓	✓	✓

Note:C_1: **DRL-Based**. C_2: **Scalability**. C_3: **Latency/TTF**. C_4: **Security**. C_5: **Aggregator**. C_6: **DNN Network Parameter Analysis**.
Note:✓ is "**Yes**" (related to this content), ✗ is "**No**" (not related to this content), N/A is "**Unknown**" can not be identified).

E. Case Study

In recent years, energy transactions between smart IoT devices (e.g., smart meters, new energy vehicles, etc.) and the energy internet have been facilitated due to the development of IIoT. The number of participants in energy transactions has been increasing, and the forms of transactions have been diversified [57]. Energy trading is an important part of the IIoT, and blockchain, as a distributed public ledger technology, has been widely used in designing new energy trading schemes [58]. Blockchain-based energy trading applications encourage market members to trade energy with each other without a third party, leading to the emergence of energy prosumers (producers and consumers), which is of great significance for maintaining grid stability.

However, as the number of blockchain nodes participating in the market increases, it will affect the stability and reliability of the system, which can lead to a decrease in transaction performance and poor system scalability. Therefore, we consider a blockchain-based peer-to-peer electricity trading scenario to evaluate the effectiveness of our proposed approach. We use data from a real P2P power transaction dataset [59] in Western Australia as transaction information to evaluate the efficiency of the proposed scheme to optimize the performance of blockchain in an energy IoT scenario. We conduct simulation experiments on the power transaction dataset using the transaction arrival rate to measure the number of transactions.

The experimental results are shown in Fig. 15. We compare and analyze the performance of blockchain transaction processing based on electricity transaction data from January 2023 for both the conventional static mode and our proposed dynamic adjustment approach. Our proposed scheme adjusts the blockchain block size and generation time with the actual transaction arrival rate change to obtain the throughput matching the transaction arrival rate, which is impossible with the static scheme. The experimental results show that our proposed approach can effectively optimize the performance of the blockchain to meet the demands of power trading.

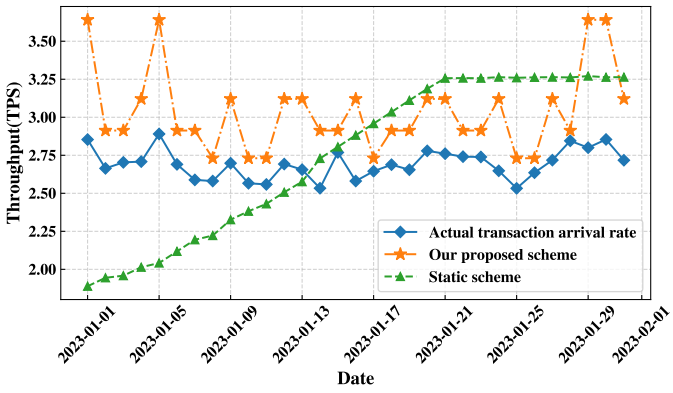


Fig. 15. The power trading case study.

IoV is also one of the primary use cases of IIoT [60]. To further evaluate the effectiveness of our proposed approach in different IIoT scenarios, we conduct simulation experiments

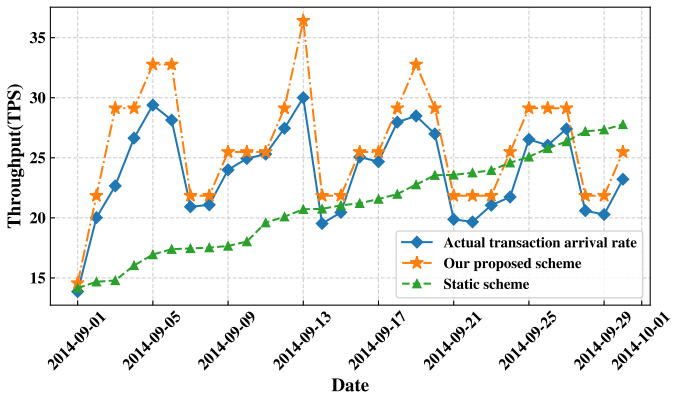


Fig. 16. The IoV case study.

using a real dataset from an IoV scenario to assess the effectiveness of blockchain performance optimization. Specifically, we simulate the IoV scenario using a dataset containing more than 4.5 million Uber pickup records in New York City from April to September 2014 [61].

In the experiment, we used daily data from September 2014 as transaction requests. The experimental results are shown in Fig. 16, demonstrating that our proposed method can effectively optimize blockchain performance in the IoV scenario. Our proposed scheme can adjust throughput according to the actual transaction arrival rate, whereas the static scheme cannot adjust dynamically. A clear drawback of the static scheme is that it continues to increase throughput even as the actual transaction arrival rate decreases.

F. Take-away Key Points

1) ***Adaptability Advantage from Dynamic Adjustment***: Through extensive experiments, we verify that the proposed DDQN framework significantly improves blockchain throughput and reduces latency in IIoT scenarios. The results show that dynamically adjusting block size and interval can better adapt to the changing transaction load in IIoT environments and ensure that the blockchain system can operate efficiently under various load conditions. Throughput improves as block size and interval increase, but a block size too large is limited by block interval and cannot sustainably improve system performance.
2) ***Effective Application of Deep Reinforcement Learning***: By adopting the DDQN approach, we successfully mitigate the problem of overestimating the Q-value in the traditional DQN model. Experimental validation shows that this improvement improves the stability and reliability of the optimization. Compared to DQN and Q-learning, DDQN performs better in dealing with the Q-value overestimation problem, improving model stability, coping with large dimensionality problems, and exploring efficiency.
3) ***Impact of Network Structure on Model Performance***: A reasonable configuration of the DNN structure can significantly enhance the performance of the DDQN model. Increasing the number of hidden layers can accelerate the model's convergence speed, but too many layers may cause increased fluctuations in the reward value. A mixed activation function yields the best average reward performance but also increases the training complexity.
4) ***Applicable to Multiple IIoT Application Scenarios***: Through the experimental validation of datasets in real-world power transactions and IoV scenarios, our scheme can dynamically adjust the block size and generation time based on the actual transaction arrival rate. This adjustment enables the system to achieve throughput that aligns with the transaction load, demonstrating its broad applicability across multiple IIoT applications. This dynamic adjustment capability significantly outperforms traditional static models. A notable drawback of the static scheme is that the throughput continues to increase even when the actual transaction arrival rate decreases, leading to a mismatch with the actual demand.

VII. Conclusions and Future Work

Currently, blockchain integration in the IIoT faces performance optimization issues. In this paper, we address the performance optimization challenges of blockchain in IIoT systems, where high dynamics and large-scale data introduce complexity. We propose a framework that improves blockchain throughput while maintaining system security. By modelling the joint optimization problem as an MDP, we apply a DRL approach and use DDQN to mitigate overestimation issues, optimizing long-term rewards. We also explore the impact of network layers and activation functions on performance. Case studies using real-world power trading and IoV data demonstrate that our scheme outperforms benchmark methods in convergence and effectiveness.

Nevertheless, certain challenges remain unresolved. One key challenge is ensuring that the performance of blockchain systems in IIoT environments remains stable across various conditions and use cases. For future work, we will focus on performance-tuning recommendations for blockchain in IIoT systems, considering the need for stability and minimal parameter adjustments. We plan to use deep learning for monitoring performance trends and time-series prediction to anticipate future performance needs. Additionally, we aim to address the challenge of limited performance data by integrating large language models for data generation, improving the accuracy and generalization of our predictions.

References

[1] Y. Yuan, F.-Y. Wang *et al.*, "Blockchain: The state of the art and future trends," *Acta Autom. Sin.*, vol. 42, no. 4, pp. 481–494, 2016.
[2] M. An, Q. Fan, H. Yu, B. An, N. Wu, H. Zhao, X. Wan, J. Li, R. Wang, J. Zhen, Q. Zou, and B. Zhao, "Blockchain technology research and application: A literature review and future trends," *Journal of Data Science and Intelligent Systems*, Oct. 2023.
[3] A. P. Kalapaaking, I. Khalil, M. S. Rahman, M. Atiquzzaman, X. Yi, and M. Almashor, "Blockchain-Based Federated Learning With Secure Aggregation in Trusted Execution Environment for Internet-of-Things," *IEEE Trans. Ind. Inf.*, vol. 19, no. 2, pp. 1703–1714, Feb. 2023.
[4] Y. Bai, Q. Hu, S.-H. Seo, K. Kang, and J. J. Lee, "Public participation consortium blockchain for smart city governance," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2094–2108, 2022.

[5] N. Satoshi, "Bitcoin: A peer-to-peer electronic cash system," *Bitcoin: A Peer-to-peer Electronic Cash System*, 2008.

[6] J. Wang, R. Zhu, T. Li, F. Gao, Q. Wang, and Q. Xiao, "ETC-Oriented efficient and secure blockchain: Credit-based mechanism and evidence framework for vehicle management," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11 324–11 337, 2021.

[7] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, "An Architecture and Performance Evaluation of Blockchain-Based Peer-to-Peer Energy Trading," *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3364–3378, Jul. 2021.

[8] F. Firouzi, S. Jiang, K. Chakrabarty, B. J. Farahani, M. Daneshmand, J. Song, and K. Mankodiya, "Fusion of IoT, AI, edge-fog-cloud, and blockchain: Challenges, solutions, and a case study in healthcare and medicine," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 3686–3705, 2023.

[9] "IoT connected devices worldwide 2019-2030 | Statista," https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[10] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[11] P. Marc, "Blockchain technology: Principles and applications," *Edward Elgar*, 2016.

[12] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, 2018.

[13] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, 2018.

[14] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2017, pp. 618–623.

[15] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 5, pp. 840–852, 2018.

[16] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial internet of things: A dueling deep {Q} -Learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, 2019.

[17] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Inf.*, vol. 13, no. 6, pp. 3154–3164, 2017.

[18] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, 2019.

[19] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards Secure Industrial IoT: Blockchain System With Credit-Based Consensus Mechanism," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.

[20] W. Wang, D. Niyato, P. Wang, and A. Leshem, "Decentralized caching for content delivery based on blockchain: A game theoretic perspective," in *2018 IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, May 20-24, 2018*. IEEE, 2018, pp. 1–6.

[21] L. Xue, H. Huang, F. Xiao, and W. Wang, "A Cross-Domain Authentication Scheme Based on Cooperative Blockchains Functioning With Revocation for Medical Consortiums," *IEEE Trans. Netw. Serv. Manage.*, vol. 19, no. 3, pp. 2409–2420, Sep. 2022.

[22] J. Wang, C. Zhu, C. Miao, R. Zhu, X. Zhang, Y. Tang, H. Huang, and C. Gao, "BPR: Blockchain-Enabled Efficient and Secure Parking Reservation Framework With Block Size Dynamic Adjustment Method," *IEEE Trans. Intell. Transport. Syst.*, vol. 24, no. 3, pp. 3555–3570, Mar. 2023.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[24] H. Yao, S. Ma, J. Wang, P. Zhang, C. Jiang, and S. Guo, "A Continuous-Decision Virtual Network Embedding Scheme Relying on Reinforcement Learning," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 2, pp. 864–875, Jun. 2020.

[25] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Performance Optimization for Blockchain-Enabled Industrial Internet of Things (IIoT) Systems: A Deep Reinforcement Learning Approach," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3559–3570, Jun. 2019.

[26] J. Luo, Q. Chen, F. R. Yu, and L. Tang, "Blockchain-Enabled Software-Defined Industrial Internet of Things With Deep Reinforcement Learning," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5466–5480, Jun. 2020.

[27] M. Liu, Y. Teng, F. R. Yu, V. C. M. Leung, and M. Song, "Deep reinforcement learning based performance optimization in blockchain-enabled internet of vehicle," in *2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019*. IEEE, 2019, pp. 1–6.

[28] Y. Gao, P. Si, K. Jin, T. Sun, and W. Wu, "Performance Comparison of Different Deep Reinforcement Learning Algorithms for Task Scheduling Problem in Blockchain-Enabled Internet of Vehicles," *IEEE Trans. Veh. Technol.*, pp. 1–15, 2023.

[29] Z. Yang, R. Yang, F. R. Yu, M. Li, Y. Zhang, and Y. Teng, "Sharded Blockchain for Collaborative Computing in the Internet of Things: Combined of Dynamic Clustering and Deep Reinforcement Learning Approach," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16 494–16 509, Sep. 2022.

[30] L. Yang, M. Li, P. Si, R. Yang, E. Sun, and Y. Zhang, "Energy-Efficient Resource Allocation for Blockchain-Enabled Industrial Internet of Things With Deep Reinforcement Learning," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2318–2329, Feb. 2021.

[31] J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6214–6228, 2020.

[32] H. Lin, S. Garg, J. Hu, G. Kaddoum, M. Peng, and M. S. Hossain, "Blockchain and deep reinforcement learning empowered spatial crowdsourcing in software-defined internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3755–3764, 2021.

[33] N. K. Akrasi-Mensah, A. S. Agbemenu, H. Nunoo-Mensah, E. T. Tchao, A.-R. Ahmed, E. Keelson, A. Sikora, D. Welte, and J. J. Kponyo, "Adaptive Storage Optimization Scheme for Blockchain-IIoT Applications Using Deep Reinforcement Learning," *IEEE Access*, vol. 11, pp. 1372–1385, 2023.

[34] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, and M. Guizani, "Blockchain-assisted secure device authentication for cross-domain industrial IoT," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 942–954, 2020.

[35] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things," *IEEE Trans. Ind. Inf.*, pp. 1–1, 2017.

[36] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the edge: Performance of resource-constrained IoT networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 174–183, 2020.

[37] V. Skwarek, "Blockchains as security-enabler for industrial IoT-applications," *Asia Pacific Journal of Innovation and Entrepreneurship*, vol. 11, no. 3, pp. 301–311, 2017.

[38] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, and K.-C. Li, "A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things," *IEEE Trans. Ind. Inf.*, vol. 15, no. 6, pp. 3582–3592, 2019.

[39] P. Zhang, P. Yang, N. Kumar, C.-H. Hsu, S. Wu, and F. Zhou, "RRV-BC: Random reputation voting mechanism and blockchain assisted access authentication for industrial internet of things," *IEEE Trans. Ind. Inf.*, 2023.

[40] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Inf.*, vol. 16, no. 6, pp. 4177–4186, 2019.

[41] A. Ajorlou and A. Abbasfar, "An optimized structure of state channel network to improve scalability of blockchain algorithms," in *2020 17th International ISC Conference on Information Security and Cryptology (ISCISC)*. IEEE, 2020, pp. 73–76.

[42] S. Dziembowski, S. Faust, and K. Hostáková, "General state channel networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 949–966.

[43] Z. Gao, H. Li, K. Xiao, and Q. Wang, "Cross-chain oracle based data migration mechanism in heterogeneous blockchains," in *40th IEEE International Conference on Distributed Computing Systems, ICDCS 2020, Singapore, November 29 - December 1, 2020*. IEEE, 2020, pp. 1263–1268.

[44] H. Huang, Y. Zhao, and Z. Zheng, "tMPT: Reconfiguration across Blockchain Shards via Trimmed Merkle Patricia Trie," in *2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS)*. Orlando, FL, USA: IEEE, Jun. 2023, pp. 1–10.

[45] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of*

the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 17–30.

[46] J. Wang, Y. Wang, X. Zhang, Z. Jin, C. Zhu, L. Li, R. Zhu, and S. Lv, "LearningChain: A Highly Scalable and Applicable Learning-Based Blockchain Performance Optimization Framework," *IEEE Trans. Netw. Serv. Manage.*, pp. 1–1, 2024.

[47] Z. Ning, S. Sun, X. Wang, L. Guo, S. Guo, X. Hu, B. Hu, and R. Y. K. Kwok, "Blockchain-Enabled Intelligent Transportation Systems: A Distributed Crowdsensing Framework," *IEEE Trans. Mob. Comput.*, vol. 21, no. 12, pp. 4201–4217, Dec. 2022.

[48] M. S. Abegaz, H. N. Abishu, Y. H. Yacob, T. A. Ayall, A. Erbad, and M. Guizani, "Blockchain-Based Resource Trading in Multi-UAV-Assisted Industrial IoT Networks: A Multi-Agent DRL Approach," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 1, pp. 166–181, 2023.

[49] Z. Zhang, G. Yu, C. Sun, X. Wang, Y. Wang, M. Zhang, W. Ni, R. P. Liu, A. Reeves, and N. Georgalas, "Tbdd: A new trust-based, drl-driven framework for blockchain sharding in iot," *Computer Networks*, vol. 244, p. 110343, 2024.

[50] Y. Zhang, J. Lin, Z. Lu, Q. Duan, and S.-C. Huang, "Pbrl-tchain: A performance-enhanced permissioned blockchain for time-critical applications based on reinforcement learning," *Future Generation Computer Systems*, vol. 154, pp. 301–313, 2024.

[51] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.

[52] A. Zhu, S. Guo, M. Ma, H. Feng, B. Liu, X. Su, M. Guo, and Q. Jiang, "Computation offloading for workflow in mobile edge computing based on deep Q-learning," in *28th Wireless and Optical Communications Conference, WOCC 2019, Beijing, China, May 9-10, 2019*. IEEE, 2019, pp. 1–5.

[53] H. van Hasselt, "Double Q-learning," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a Meeting Held 6-9 December 2010, Vancouver, British Columbia, Canada*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 2613–2621.

[54] B. J. A. Kröse, "Learning from delayed rewards," *Robotics Auton. Syst.*, vol. 15, no. 4, pp. 233–235, 1995.

[55] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nat.*, vol. 518, no. 7540, pp. 529–533, 2015.

[56] C. Qiu, H. Yao, C. Jiang, S. Guo, and F. Xu, "Cloud Computing Assisted Blockchain-Enabled Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 247–257, Jan. 2022.

[57] Z. Guan, X. Lu, N. Wang, J. Wu, X. Du, and M. Guizani, "Towards secure and efficient energy trading in IIoT-enabled energy internet: A blockchain approach," *Future Gener. Comput. Syst.*, vol. 110, pp. 686–695, 2020.

[58] R. Huo, S. Zeng, Z. Wang, J. Shang, W. Chen, T. Huang, S. Wang, F. R. Yu, and Y. Liu, "A Comprehensive Survey on Blockchain in Industrial Internet of Things: Motivations, Research Progresses, and Future Challenges," *IEEE Commun. Surv. Tutorials*, vol. 24, no. 1, pp. 88–122, 2022.

[59] "Short-Term Energy Market (STEM) Bids and Offers Data," https://data.wa.aemo.com.au/#stem-bids-and-offers.

[60] S. Yeasmin and A. Baig, "Permissioned Blockchain-based Security for IIoT," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. Vancouver, BC, Canada: IEEE, Sep. 2020, pp. 1–7.

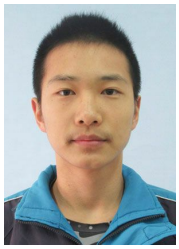
[61] "Uber Pickups in New York City," https://www.kaggle.com/datasets/fivethirtyeight/uber-pickups-in-new-york-city.

Min An received the B.E. degree in computer science and technology from Northwest Minzu University, Lanzhou, China, in 2022. He is currently pursuing the M.S. degree with the School of Software, Yunnan University, Kunming, China.

His current research interests include reinforcement learning, time series forecasting, blockchain technology and federated learning. He has authored peer-reviewed papers in *Future Generation Computer Systems*.

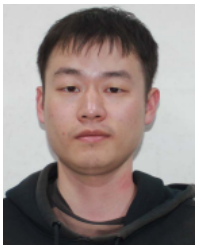
Xuan Zhang (Member, IEEE) received the B.S. and M.S. degrees in computer science from Yunnan University, Yunnan. She received the Ph.D. degree in system analysis and integration from Yunnan University, Yunnan. She is a professor with the School of Software, Yunnan University, Yunnan, China. She is author of 3 books and more than 120 papers. She has been principal investigator for more than 30 national, provincial, and private grants and contracts. She is the core scientist of Yunnan Key Laboratory of Software Engineering and leader of Yunnan Software Engineering Academic Team. Her research interests include knowledge engineering, natural language processing and blockchain.

Jishu Wang (Member, IEEE) He is currently working toward the Ph.D. degree with the School of Information Science and Engineering, Yunnan University, Kunming, China. He has authored peer-reviewed papers in IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. His current research interests include blockchain, intelligent transportation, and deep learning.



Qiyuan Fan received his Bachelor's degree in Wuhan University of Technology of School of Computer Science, China in 2022.He is currently studying for a master's degree in the School of Software, Yunnan University, Kunming, China.

His current research interests include tensor decomposition, reinforcement learning, big data, and network security. He has authored peer-reviewed papers in *Future Generation Computer Systems* and *Information Sciences*.



Chen Gao (Student Member, IEEE) received the M.E. degree in software engineering from Yunnan University, Kunming, China, where he is currently pursuing the Ph.D. degree with the School of Information Science and Engineering. He published three articles in the field of information extraction.

His research interests include natural language processing and knowledge graph, especially information extraction.

Linyu Li (Student Member, IEEE) received the M.E. degree in software engineering from Yunnan University, Kunming, China, in 2024. He is currently pursuing the Ph.D. degree with the School of Computer Science, Peking University, Beijing, China.

His current research interests include using deep learning to solve knowledge graph-related problems and requirements engineering. He has authored peer-reviewed papers in *Information Sciences* and *Knowledge-Based Systems*.

Cuizhen Lu received the B.E. degree in electronic information engineering from Yunnan University, Kunming, China, in 2022. She is currently pursuing the M.S. degree with the College of Electronics and Information Engineering, Sichuan University, Chengdu, China.

Her current research interests include deep learning and image processing.

Nan Li received the B.E. degree in computer science and technology from Northwest Minzu University, Lanzhou, China, in 2024. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, Northeastern University, Shenyang, China.

Her current research interests include deep learning and IoT security.

Yingchen Liu received the B.E. degree in computer science and technology from Northwest Minzu University, Lanzhou, China, in 2023. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, National University of Defense Technology, Changsha, China.

His current research interests include high-performance computing and randomized numerical algorithms.