Article | Received 5 September 2024; Accepted 1 November 2024; Published 28 November 2024
https://doi.org/10.55092/blockchain20240008

TriGuard: mitigating bribery in DPoS-based blockchain governance for Web 3.0

Jingyu Liu[1,†], Xingchen Sun[1,†], Yunfeng Xia[1], Yifei Yang[1], Wenyuan Yang[2] and Chao Li[1,*]

[1]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China
[2]School of Cyber Science and Technology, Sun Yat-sen University, Guangzhou, China

† These two authors contributed equally.

* Correspondence author; E-mail: li.chao@bjtu.edu.cn.

Abstract: The evolution of Web 3.0 has brought decentralized governance to the forefront, with Delegated Proof-of-Stake (DPoS) mechanisms playing a pivotal role. However, bribery and collusion pose significant risks to the integrity of DPoS systems, undermining the decentralization that is fundamental to Web 3.0's vision. This paper presents TriGuard, an enhanced governance mechanism designed to curb bribery and promote fair participation within DPoS frameworks. TriGuard integrates a tripartite evolutionary game model with incentive mechanisms tailored for voting participation, bribery reporting, and supervisory actions. Through extensive simulations and theoretical analysis, we demonstrate that TriGuard effectively increases voter engagement, strengthens supervisory oversight, and diminishes the influence of malicious actors. The proposed mechanism reduces centralization risks and enhances security, creating a more decentralized governance framework for Web 3.0 ecosystems.

Keywords: blockchain governance; decentralization; evolutionary game theory; incentive mechanism

1. Introduction

Web 3.0 [1], often heralded as the next generation of the internet, represents a profound shift from centralized platforms to decentralized, user-centric ecosystems. Unlike Web 2.0, which is dominated by a few powerful corporations that control data and online interactions, Web 3.0 seeks to restore control to individuals by leveraging blockchain technology. At its core, Web 3.0 is built on the principles of decentralization, data sovereignty, and peer-to-peer interactions. This new iteration of the web envisions a landscape where users are not merely participants but co-owners of the platforms they use, fostering more open, transparent, and censorship-resistant environments. Innovations such as decentralized finance (DeFi) [2], decentralized autonomous organizations (DAOs) [3], and non-fungible tokens (NFTs) [4] are already demonstrating the potential of Web 3.0 to transform industries by removing intermediaries and allowing for a more equitable distribution of power and resources [5]. However, as this decentralized vision gains momentum, the need for robust and secure governance mechanisms becomes increasingly crucial. Ensuring that the foundational principles of decentralization are upheld requires careful consideration of how governance is implemented across various blockchain networks [6].

Among the consensus mechanisms utilized in blockchain platforms, Delegated Proof-of-Stake (DPoS) [7] has emerged as a popular model due to its ability to balance decentralization with operational efficiency. DPoS allows token holders to elect a set of representatives—often referred to as "super representatives"l—who are responsible for managing the network and

making critical governance decisions. This approach is favored for its scalability and lower energy consumption compared to Proof-of-Work (PoW) [8] systems, making it a practical choice for high-throughput blockchains like TRON [9] and EOS [10]. However, DPoS [11] is not without significant challenges. The concentration of decision-making power in a limited number of representatives can lead to governance centralization, which directly contradicts the decentralized ethos of Web 3.0. One of the most critical issues in DPoS governance is the risk of bribery, where representatives or external actors offer financial rewards or incentives to voters in exchange for their votes. Such practices can distort governance outcomes, allowing a small group of well-resourced actors to dominate the network, ultimately compromising the fairness, security, and integrity that decentralized systems are intended to guarantee.

Bribery [12] in DPoS-based blockchain governance [13] is a multifaceted problem that poses a significant threat to the sustainability of decentralized ecosystems. When voting is driven by financial incentives rather than the genuine interest in maintaining a decentralized network, governance centralization can occur, undermining the legitimacy of decision-making and increasing vulnerability to collusion and corruption. In the context of Web 3.0, where decentralization is a core principle, allowing governance centralization to take root can have long-term consequences that erode trust and stifle innovation. Without effective measures to curb bribery and other forms of manipulation, the decentralized promise of Web 3.0 may remain unfulfilled. Addressing this pressing issue requires innovative governance models that realign incentives, discourage malicious behavior, and promote transparency, fairness, and inclusivity in the decision-making process.

To address these challenges, this paper introduces TriGuard, an enhanced governance mechanism specifically designed to mitigate bribery within DPoS-based blockchain governance. TriGuard is built upon a game-theoretical foundation and incorporates incentive mechanisms aimed at promoting honest participation, encouraging the reporting of bribery attempts, and enhancing oversight by independent parties. By embedding these incentives into the governance framework, TriGuard creates a more resilient and decentralized system capable of effectively countering bribery and maintaining network integrity. To validate this approach, we construct a tripartite evolutionary game model that simulates the interactions among various participants in the governance process. This model enables us to analyze how different incentive structures influence behavior and governance outcomes, providing insights into the effectiveness of TriGuard in real-world scenarios.

In addition to theoretical modeling, we implement and simulate TriGuard in a real-world setting to further assess its impact on governance dynamics. The results show that TriGuard reduces bribery risk and promotes decentralization by encouraging wider participation in governance. Through a detailed analysis of equilibrium points, we validate the rationality and effectiveness of TriGuard, providing a solid foundation for future improvements in DPoS-based blockchain governance. Our findings suggest that aligning incentives with decentralized principles can significantly enhance the security and fairness of blockchain networks, thereby contributing to the development of a truly decentralized Web 3.0 ecosystem.

Contributions. The key contributions of this paper are summarized as follows:

- We propose TriGuard, an enhanced governance mechanism specifically designed to address the bribery issue in DPoS systems. TriGuard encourages voter participation, promotes accurate reporting of bribery attempts, and enhances supervisory efforts, effectively strengthening decentralized governance.
- We construct and analyze a tripartite evolutionary game model involving candidates, voters, and supervisors. This model captures the strategic interactions among these participants, demonstrating how TriGuard guides the system toward a more secure and decentralized equilibrium.
- We validate the effectiveness of TriGuard under various conditions with a combination of theoretical analysis and simulation experiments. Our simulations demonstrate that TriGuard

boosts voter engagement and reduces bribery, enhancing the security of governance.

The rest of this paper is organized as follows: Section 2. introduces the background. Section 3. details the theoretical framework and incentive mechanism design, introducing our tripartite game model and its equilibrium analysis. Section 4. presents the experimental setup and simulation results, demonstrating the effectiveness of TriGuard under various conditions. Section 5. discusses related work on incentive mechanisms and governance. Finally, Section 6. concludes the paper with a summary of key contributions and directions for future research.

2. Background

2.1. Game theory

Game theory, introduced in 1973 [14], provides a mathematical framework for analyzing interactions and decision-making among rational agents, encompassing both conflict and cooperation. It has found applications across various domains, including economics [15], finance [16], and computer science [17].

Classical Game Theory and Nash Equilibrium. Classical game theory examines how rational players make decisions in competitive and cooperative scenarios, aiming to predict and analyze behavior in interactive environments. A central concept in this field is the Nash equilibrium, where each player's strategy is optimal, given the strategies of others. At a Nash equilibrium, no player can improve their outcome by unilateral ly changing their strategy, assuming others' strategies remain constant.

Evolutionary Game Theory and Stable Strategies. Evolutionary game theory extends classical game theory by incorporating ideas from evolutionary biology to study how strategies evolve over time in dynamic, incomplete-information environments. This branch focuses on the concept of "stable strategies," which persist through repeated interactions and natural selection. A stable strategy is resistant to invasion by alternative strategies, and if widely adopted within a population, no new strategy can outcompete it. Evolutionary game theory is particularly useful for explaining behaviors in social dynamics, animal behavior, and economic decision-making. To address the bribery scenario in DPoS-based blockchain governance, evolutionary game theory offers valuable insights. This theoretical framework models the strategic interactions between token holders and delegates over time, analyzing how bribery behaviors evolve. By simulating strategies like honest voting, bribery acceptance, and resistance, evolutionary dynamics reveal conditions under which bribery becomes dominant or is suppressed. This approach helps identify stable equilibria and the impact of governance mechanisms, such as transparency and accountability measures, on reducing bribery. Integrating evolutionary game theory thus provides a deeper understanding of bribery dynamics and helps design robust governance to promote fairness in DPoS systems.

2.2. DPoS-based blockchain governance

DPoS is a consensus algorithm that significantly influences blockchain governance by integrating a structured, on-chain decision-making process. Unlike traditional PoW systems like Bitcoin and Ethereum, where governance typically occurs off-chain through informal discussions among developers and stakeholders, DPoS embeds governance directly into the blockchain. In a DPoS system, token holders(or voters) vote for a select number of delegates who are entrusted with validating transactions and managing the network. This representative model allows for swift and efficient decision-making, as delegates can quickly implement protocol changes and resolve disputes. The accountability mechanism ensures that delegates act in the network's best interests, as they can be voted out if they fail to meet the community's expectations. However, DPoS-based blockchain governance faces challenges, such as the potential concentration of voting power among a few large token holders, leading to central-

ization risks. Additionally, the election process can be prone to manipulation and collusion, potentially undermining the democratic principles that DPoS aims to promote. To address these issues, DPoS-based blockchains must incorporate robust mechanisms ensuring transparency, fairness, and broad participation in the governance process. Despite these challenges, DPoS offers a promising framework for achieving efficient consensus and responsive governance in blockchain technology.

The DPoS consensus mechanism structures blockchain governance into three distinct phases: staking, voting, and governing [13].

Staking Phase: In the initial phase, participants intending to vote must stake their tokens, which grants them voting power proportional to the amount and duration of their stake. This phase is critical as it aligns governance participation with a financial commitment, ensuring that those who vote have a genuine vested interest in the network. Different blockchain platforms implement staking with varying nuances. For instance, TRON uses TRX tokens for staking, while Steem employs Steem tokens. The relationship between stake and voting power differs across platforms, influencing the dynamics of governance.

Voting Phase: Once tokens are staked, participants enter the voting phase, where they elect representatives or delegates tasked with governing the blockchain. Voters cast their votes, and the candidates with the most support are selected to form the governing committee. This process occurs in rounds, with each round updating the committee based on the latest voting results. The voting phase encapsulates the community's collective decision-making power, balancing the interests of diverse stakeholders within the ecosystem.

Governing Phase: In the final phase, the elected committee members engage in the active governance of the blockchain. They deliberate and make decisions on proposals that shape the blockchain's development and operational direction. Typically, proposals require approval from multiple committee members before implementation, ensuring that decision-making is decentralized and minimizing the risk of concentrated control.

As blockchain ecosystems have matured, the token-based incentive mechanisms that once spurred growth and engagement have also revealed potential drawbacks. Particularly in DPoS systems, these mechanisms can inadvertently contribute to governance centralization and collusion. To address these challenges, this paper introduces TriGuard, a novel incentive-driven mechanism designed to mitigate bribery and enhance the integrity of DPoS-based blockchain governance.

3. TriGuard

This section introduces TriGuard, an enhanced governance mechanism for DPoS-based blockchain systems. TriGuard is designed to foster broader participation, enhance security, and uphold the principles of decentralized governance by incorporating an improved voting scheme supported by a tripartite evolutionary game model. This model includes three key stakeholders: candidates, voters, and spontaneous supervisors, each playing a distinct role with specific strategies. The following subsections will detail the game-theoretical foundation of TriGuard, the design of its incentive mechanisms, and an analysis of the system's stability.

3.1. Evolutionary game theory in DPoS-based blockchain governance

In the context of Web 3.0 blockchain governance, stakeholders such as candidates and voters engage in decision-making strategies that influence the overall governance dynamics. These participants continuously adjust their strategies based on consensus rules, learning from outcomes until they converge toward a stable equilibrium. Evolutionary game theory, which models interactions among groups rather than individual players, is particularly well-suited for capturing the collective behaviors observed in decentralized systems. Leveraging this approach, we analyze governance dynamics under the DPoS consensus mechanism to explore

the strategic decisions that stakeholders make.

Before introducing the incentive mechanisms within TriGuard to address governance centralization and collusion, we first conduct a dynamic analysis of the existing DPoS governance process using evolutionary game theory. This analysis lays the groundwork for understanding the strategic environment in which TriGuard operates. Based on our model of DPoS governance, we establish the following assumptions:

(1) **Participants and Strategy Spaces:** The participants in this game model are the candidate group C and the voter group V, each operating with bounded rationality. The strategy space for the candidate group C is $S_C = \{S_{C1}, S_{C2}\} = \{\text{bribe}, \text{not bribe}\}$, while the strategy space for the voting group V is $S_V = \{S_{V1}, S_{V2}\} = \{\text{accept}, \text{not accept}\}$.

(2) **Payoff Structure for Candidates:** If the candidate group C opts for the "not bribe" strategy, they can secure block production and governance power through legitimate voting processes, yielding a profit of Q_1 from block generation. Conversely, if the candidate group C chooses to bribe and the voting group V accepts, the candidate must bear an additional bribe cost ω, but can gain a collusion profit q_1.

(3) **Payoff Structure for Voters:** When the candidate group C does not engage in bribery, the voting group V earns a consistent profit Q_2 regardless of whether they choose "accept" or "not accept." However, if the candidate group C opts for bribery and the voting group V chooses "accept," the voters gain an additional collusion profit q_2 alongside their regular profit. If the voters choose "not accept," they still secure the standard profit Q_2.

(4) **Probability Distributions:** The probability that the candidate group C chooses the "bribe" strategy is x $(0 < x < 1)$, while the probability they choose "not bribe" is $1 - x$. Similarly, the probability that the voter group V chooses the "accept" strategy is y $(0 < y < 1)$, and the probability they choose "not accept" is $1 - y$.

Based on these assumptions, we construct the evolutionary game payoff matrix shown in Table 1.

Table 1. Evolutionary game payoff matrix before setting incentive mechanisms.

Gaming Strategy		Voter	
		Accept (y)	Not Accept $(1-y)$
Candidate	Bribe (x)	$(Q_1 - \omega + q_1, Q_2 + q_2)$	(Q_1, Q_2)
	Not Bribe $(1-x)$	(Q_1, Q_2)	(Q_1, Q_2)

Given the above assumptions, the expected payoff for candidates choosing the "bribe" strategy is $U_c^1 = y(Q_1 - \omega + q_1) + (1-y)Q_1$, while the expected payoff for candidates choosing "not bribe" is $U_c^2 = yQ_1 + (1-y)Q_1$. Thus, the average payoff for the candidate group is $\overline{U_C} = xU_c^1 + (1-x)U_c^2$. Similarly, the expected payoff for voters choosing "accept" is $U_V^1 = x(Q_2 + q_2) + (1-x)Q_2$, and the expected payoff for voters choosing "not accept" is $U_V^2 = xQ_2 + (1-x)Q_2$. Therefore, the average payoff for the voter group is $\overline{U_V} = yU_V^1 + (1-y)U_V^2$. From this analysis, it is evident that when the bribe cost ω is less than the collusion profit q_1, both candidates and voters are likely to adopt the evolutionary strategy of (bribe, accept).

3.2. Incentive mechanisms and enhanced governance

The analysis in the previous section underscores that without effective incentive mechanisms, governance centralization and collusion are likely to persist, as candidates and voters can easily adopt improper strategies in elections. To counter these issues, we introduce an enhanced incentive-driven voting scheme specifically designed to mitigate the risks inherent in DPoS blockchain governance. At the core of this scheme lies a reporting mechanism that rewards voters for exposing unethical behaviors, such as bribery, while simultaneously penalizing dishonesty. Additionally, a new participant role—the spontaneous supervisor—is introduced. This role encourages community members to voluntarily engage in network supervision, fostering a more transparent and secure governance process. By integrating a reward and

punishment mechanism, the proposed scheme effectively deters bribery while promoting active and fair participation, thereby contributing to the sustainable development of DPoS-based blockchain systems.

The assumptions underpinning the evolutionary game model for DPoS blockchain governance with the proposed incentive mechanisms are as follows:

(1) **Participants and Strategy Spaces:** The game participants include the candidate group C, the voter group V, and the spontaneous supervisor group A, all of whom exhibit bounded rationality. The strategy space for the candidate group C is $S_C = \{S_{C1}, S_{C2}\}$ = {Bribe, Not Bribe}; the strategy space for the voter group V is $S_V = \{S_{V1}, S_{V2}\}$ = {Accept, Report}; and the strategy space for the spontaneous supervisor group A is $S_A = \{S_{A1}, S_{A2}\}$ = {Supervise, Not Supervise}.

(2) **Payoff Structure:** If the candidate group C chooses "Not Bribe," they receive a reward Q_1 for completing the block generation task, while the voter group V earns a standard return of Q_2 for participating in the voting process. Conversely, if the candidate group C chooses "Bribe" and the voter group V accepts, and if the spontaneous supervisor group A chooses "Not Supervise," the candidate group C incurs a bribe cost ω but gains a collusion profit q_1, and the voter group V additionally earns a collusion profit q_2.

(3) **Reporting and Supervision Dynamics:** When the candidate group C does not engage in bribery, if the voter group V opts to "Report" while the spontaneous supervisor group A does not supervise, the voters may incur a penalty m for falsely reporting. However, if both the voter group V reports and the spontaneous supervisor group A supervises, penalties are avoided, and all honest participants—candidates, supervisors, and voters—receive rewards n_1, n_2 and n_3 respectively. Importantly, the spontaneous supervisor incurs a supervision cost α, which accounts for resource expenditures such as bandwidth.

(4) **Incentives and Penalties for Bribery:** When the candidate group C bribes, and the voter group V reports while the spontaneous supervisor group A supervises, both the voter and supervisor groups receive a return. On the other hand, if voters accept the bribe and supervisors choose to supervise, the supervisor gains a return R_A^1, while candidates and voters face penalties p_1 and p_2 respectively, where $p_1 + p_2 = p$. If the voter group reports and the supervisor does not supervise, the voter group gains R_V^1, and the candidate incurs a penalty p. If both voters report and supervisors supervise, the voter group gains R_A^2, and the supervisor group gains R_V^2 while the candidate faces a penalty p.

(5) **Probability Distribution of Strategies:** The probability that the candidate group C chooses the "Bribe" strategy is x $(0 < x < 1)$, meaning the probability of choosing "Not Bribe" is $1 - x$. Similarly, the probability that the voter group V opts for the "Accept" strategy is y $(0 < y < 1)$, while the probability of choosing "Report" is $1 - y$. Lastly, the probability that the spontaneous supervisor group A chooses the "Supervise" strategy is z, with the probability of "Not Supervise" being $1 - z$.

Based on these assumptions, the evolutionary game payoff matrix with incentive mechanisms is presented in Table 2.

3.3. Evolutionary game analysis

3.3.1. Replicator dynamics equations

In this subsection, we derive the replicator dynamics equations that describe the evolutionary stability of strategies for the candidate group, voter group, and spontaneous supervisor group. These equations model how the proportions of different strategies evolve over time based on the payoffs associated with each strategy.

Table 2. Evolutionary game payoff matrix after setting incentive mechanisms.

Strategy Combination			Strategy Benefits		
			Candidate	Voter	Spontaneous Supervisor
Bribe (x)	Accept (y)	Supervise (z)	$Q_1-\omega-p_1$	Q_2-p_2	$R_4^1-\alpha$
		Not Supervise $(1-z)$	$Q_1-\omega+q_1$	Q_2+q_2	0
	Report$(1-y)$	Supervise (z)	$Q_1-\omega-p$	$Q_2+R_V^2$	$R_4^2-\alpha$
		Not Supervise $(1-z)$	$Q_1-\omega-p$	$Q_2+R_V^1$	0
Not Bribe $(1-x)$	Accept (y)	Supervise (z)	Q_1	Q_2	$-\alpha$
		Not Supervise $(1-z)$	Q_1	Q_2	0
	Report$(1-y)$	Supervise (z)	Q_1+n_1	Q_2+n_3	$n_2-\alpha$
		Not Supervise $(1-z)$	Q_1	Q_2-m	0

$$\begin{aligned} U_c^{1\prime} &= yz(Q_1-\omega-p_1)+y(1-z)(Q_1-\omega+q_1)+(1-y)z(Q_1-\omega-p) \\ &\quad +(1-y)(1-z)(Q_1-\omega-p) \\ &= y(p+q_1)-yz(p_1+q_1)+Q_1-\omega-p \\ U_c^{2\prime} &= yzQ_1+y(1-z)Q_1+(1-y)z(Q_1+n_1)+(1-y)(1-z)Q_1 \\ &= Q_1+zn_1-yzn_1 \\ \overline{U_c{}'} &= xU_c^1+(1-x)U_c^{2\prime} \end{aligned} \tag{1}$$

The replicator dynamics equation for the candidate group, representing the rate of change of candidates adopting the "Bribe" strategy over time, is given by:

$$\begin{aligned} F(x) = \frac{dx}{dt} &= x(U_c^{1\prime}-\overline{U_c'}) \\ &= x(1-x)(U_c^{1\prime}-U_c^{2\prime}) \\ &= x(1-x)[y(p+q_1)-zn_1-yz(p_1+q_1-n_1)-\omega-p] \end{aligned} \tag{2}$$

The expected returns for the voter group choosing the “Accept” strategy, the expected returns for voters choosing “Report,” and the average returns for the voter group’s strategy choices are:

$$\begin{aligned} U_V^{1\prime} &= x(1-z)q_2-xzp_2+Q_2 \\ U_V^{2\prime} &= (x-1)(1-z)m+(1-x)n_3+xz(R_V^2-R_V^1)+xR_V^1+Q_2 \\ \overline{U_V{}'} &= yU_V^{1\prime}+(1-y)U_V^{2\prime} \end{aligned} \tag{3}$$

The replicator dynamics equation for the voter group choosing the “Accept” strategy is:

$$\begin{aligned} F(y) = \frac{dy}{dt} &= y(U_V^{1\prime}-\overline{U_V'}) \\ &= y(1-y)(U_V^{1\prime}-U_V^{2\prime}) \\ &= y(1-y)\big[x(1-z)q_2-xzp_2-(x-1)(1-z)m \\ &\quad -(1-x)zn_3-xz(R_V^2-R_V^1)-xR_V^1\big] \end{aligned} \tag{4}$$

The expected payoffs for the group of spontaneous supervisors choosing to “Supervise” and the expected payoffs for supervisors choosing “Not Supervise,” along with the average payoffs for the supervisor group’s strategic choices, are:

$$\begin{aligned} U_A^{1\prime} &= xy(R_A^1-\alpha)+x(1-y)(R_A^2-\alpha)+(1-x)y(-\alpha)+(1-x)(1-y)(n_2-\alpha) \\ &= xyR_A^1+x(1-y)R_A^2+(1-x)(1-y)n_2-\alpha \\ U_A^{2\prime} &= 0 \\ \overline{U_A{}'} &= zU_A^{1\prime}+(1-z)U_A^{2\prime} \end{aligned} \tag{5}$$

The replicator dynamics equation for the spontaneous supervisor group choosing the "Supervise" strategy is:

$$\begin{aligned} F(z) = \frac{dz}{dt} &= z(U_A^{1\prime}-\overline{U_A'}) \\ &= z(1-z)(xyR_A^1+x(1-y)R_A^2+(1-x)(1-y)n_2-\alpha) \end{aligned} \tag{6}$$

These replicator dynamics equations provide a foundation for analyzing the stability of different strategies in the blockchain governance model, allowing us to identify conditions under which participants may adopt cooperative or collusive behaviors.

3.3.2. Replicator dynamics and evolutionary stability analysis

Analysis of Candidate Group Bribery Strategies. The first-order derivative of the equation for the replication dynamics of the candidate population is given by:

$$\frac{d[F(x)]}{dx} = (1-2x)[y(p+q_1)-zn_1-yz(p_1+q_1-n_1)-\omega-p] \tag{7}$$

Let $G(z)=(p+q_1)y-(p_1y+q_1y-n_1y+n_1)z-\omega-p$, According to the stability theorem of differential equations, the probability of the candidate group choosing a bribery strategy being in a stable state must satisfy: $F(x)=0$ and $d[F(x)]/dx=0$. Since $\partial G(z)/\partial z<0$, $G(z)$ is a decreasing function. Therefore, when $z=[(p+q_1)y-\omega-p]/[(p_1+q_1-n_1)y+n_1]=y^*$, $G(y)=0$, and at this point $d[F(x)]/dx\equiv 0$, the candidate group cannot determine a stable strategy. When $y<y^*$, $G(y)>0$, $x=1$ is the Evolutionary Stable Strategy (ESS) for the candidate group. Conversely, when $y>y^*$, $G(y)<0$, $x=0$ is the ESS for the candidate group.

Analysis of Voter Group Acceptance Strategies. The first-order derivative of the equation for the dynamics of voter group replication is given by:

$$\frac{d[F(y)]}{dy} = (1-2y)[x(1-z)q_2-xzp_2-(x-1)(1-z)m-(1-x)zn_3-xz(R_V^2-R_V^1)-xR_V^1] \tag{8}$$

Let $H(z)=-(xq_2+xp_2-xR_V^2+xR_V^1+m-xm+n_3-xn_3)z+(q_2-m-R_V^1)x+m$. For the probability of the voter group choosing the acceptance strategy to be in a stable state, it must satisfy $F(y)=0$ and $d[F(y)]/dy=0$. Since $H(z)$ is a decreasing function, when $z=[(q_2-m-R_V^1)x+m]/(xq_2+xp_2-xR_V^2+xR_V^1+m-xm+n_3-xn_3)=z^*$, $H(z)=0$, and at this point, $d[F(y)]/dy\equiv 0$, the voter group cannot determine a stable strategy. When $z<z^*$, $H(z)>0$, $y=1$ is the ESS for the voter group. Conversely, when $z>z^*$, $H(z)<0$, $y=0$ is the ESS for the voter group.

Analysis of Supervisory Strategies of Spontaneous Supervisors. The first-order derivative of the equation for the dynamics of spontaneous supervisor population replication is given by:

$$\frac{d[F(z)]}{dz} = (1-2z)[xyR_A^1+x(1-y)R_A^2+(1-x)(1-y)n_2-\alpha] \tag{9}$$

Let $L(x)=(R_A^2-n_2+yR_A^1-yR_A^2+yn_2)x-n_2y+n_2-\alpha$. For the probability of spontaneous supervisor groups choosing the supervisory strategy to be in a stable state, it must

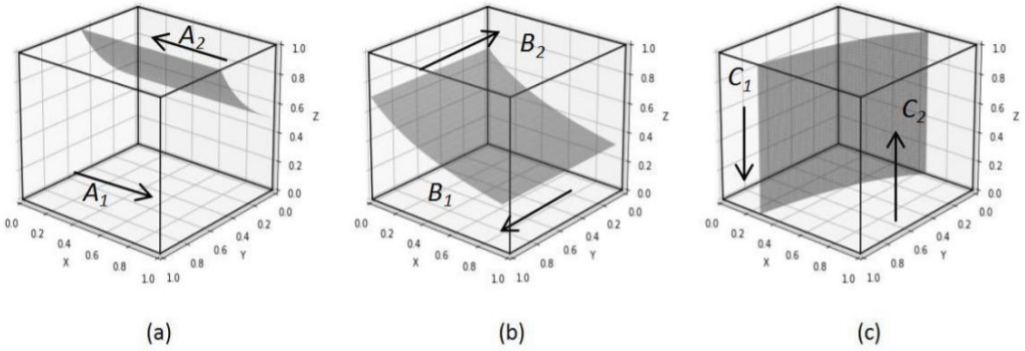

Figure 1. Evolution process of third-party agent strategies. Each subplot represents different strategic phases or scenarios, with axes labeled X, Y, and Z, typically representing different variables or dimensions relevant to the strategies. The shaded areas represent regions of influence or stability where agents tend to converge or stabilize.

satisfy $F(z) = 0$ and $d[F(z)]/dz = 0$. Since $L(x)$ is an increasing function, when $x = (\alpha - n_2 + n_2 y)/(R_A^2 - n_2 + yR_A^1 - yR_A^2 + n_2 y) = x^*$, $L(x) = 0$, and at this point, $d[F(z)]/dz \equiv 0$, the supervisor group cannot determine a stable strategy. When $x < x^*$, $L(x) < 0$, $z = 0$ is the ESS for the supervisor group. Conversely, when $x > x^{**}$, $L(x) > 0$, $z = 1$ is the ESS for the supervisor group.

The strategy evolution phase diagram for the three parties is shown in Figure 1.

3.3.3. Stability analysis of the evolutionary game model for DPoS blockchain governance

Given $F(x) = 0$, $F(y) = 0$, $F(z) = 0$, the system equilibrium points can be obtained as: $E_1(0,0,0)$, $E_2(0,0,1)$, $E_3(0,1,0)$, $E_4(0,1,1)$, $E_5(1,0,0)$, $E_6(1,0,1)$, $E_7(1,1,0)$, $E_8(1,1,1)$, $E_9(x_1,y_1,z_1)$, $E_{10}(x_2,y_2,z_2)$, $E_{11}(x_3,y_3,z_3)$, $E_{12}(x_4,y_4,z_4)$, $E_{13}(x_5,y_5,z_5)$, $E_{14}(x_6,y_6,z_6)$, where:

$$
\begin{aligned}
&x_1 = 1, y_1 = \frac{\alpha - R_A^2 + n_2}{R_A^1 - R_A^2 + n_2}, z_1 = \frac{q_2 - R_V^1}{q_2 + p_2 - R_V^2} \\
&x_2 = \frac{\alpha}{R_A^1}, y_2 = 1, z_2 = \frac{q_1 - \omega}{p + q_1} \\
&x_3 = \frac{n_3}{R_V^2 - 2R_V^1 + n_3}, y_3 = \frac{\omega + p - n_1}{p - p_1 + n_1}, z_3 = 1 \\
&x_4 = 0, y_4 = \frac{n_2 - \alpha}{n_2}, z_4 = \frac{m}{m + n_3} \\
&x_5 = \frac{\alpha - n_2}{R_A^2 - n_2}, y_5 = 0, z_5 = \frac{-\omega - p}{n_1} \\
&x_6 = \frac{m}{R_V^1 + m - q_2}, y_6 = \frac{\omega + p}{p + q_1}, z_6 = 0
\end{aligned}
$$

Since x, y, and z are in the range $[0,1]$, the equilibrium point E_{13} is meaningless; the equilibrium points E_9 to E_{12} and E_{14} are meaningful under certain conditions.

The Jacobian matrix of the three-way evolutionary game system is:

$$J = \begin{bmatrix} \frac{\partial F(x)}{\partial x} & \frac{\partial F(x)}{\partial y} & \frac{\partial F(x)}{\partial z} \\ \frac{\partial F(y)}{\partial x} & \frac{\partial F(y)}{\partial y} & \frac{\partial F(y)}{\partial z} \\ \frac{\partial F(z)}{\partial x} & \frac{\partial F(z)}{\partial y} & \frac{\partial F(z)}{\partial z} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}$$

$$J_{11} = (1-2x)(yp + yq_1 - zn_1 - yz(p_1 + q_1 - n_1) - \omega - p)$$
$$J_{12} = x(1-x)(p + q_1 - zp_1 - zq_1 - zn_1)$$
$$J_{13} = x(1-x)(-n_1 - yp_1 + q_1y - n_1y)$$
$$J_{21} = y(1-y)(R_V^1 z - R_V^1 - R_V^2 z + mz + n_3 z - m - p_2 z - q_2 z + q_2)$$
$$J_{22} = (1-2y)(R_V^1 xz - R_V^1 - R_V^2 xz + mxz + n_3 xz - mx - n_3 z - mz + m - p_2 xz - q_2 xz + q_2 x)$$
$$J_{23} = y(1-y)(R_V^1 x - R_V^2 x + mx + n_3 x - m - n_3 - p_2 x - q_2 x)$$
$$J_{31} = z(1-z)(R_A^1 y - R_A^2 y + n_2 y - n_2 + R_A^2 x)$$
$$J_{32} = z(1-z)(R_A^1 x - R_A^2 x + n_2 x - n_2)$$
$$J_{33} = (1-2z)(R_A^1 xy - R_A^2 xy + n_2 xy - n_2 x - n_2 y + R_A^2 x - n_2 - \alpha)$$

According to Lyapunov's first method, if all the eigenvalues of the Jacobian matrix have negative real parts, the equilibrium point is asymptotically stable; conversely, if at least one eigenvalue of the Jacobian matrix has a positive real part, the equilibrium point is unstable; furthermore, if all the eigenvalues of the Jacobian matrix, except for those with zero real parts, have negative real parts, the equilibrium point is in a critical state, and its stability cannot be determined solely by the signs of the eigenvalues. Therefore, when analyzing the stability of each equilibrium point, the above three cases should be comprehensively considered.

Table 3 shows the eigenvalues of each point, their signs, and stability calculated from the Jacobian matrix. According to Lyapunov's first method, possible asymptotically stable equilibrium points are identified.

Table 3. Stability analysis of equilibrium points.

Equilibrium Point	Eigenvalue $(\delta_1, \delta_2, \delta_3)$	Sign	Stability	Condition
(0, 0, 0)	$-p-\omega, m, -\alpha+n_2$	(-, +, *)	unstable	/
(0, 0, 1)	$-n_1-p-\omega, -n_3, \alpha-n_2$	(-, +, *)	unsure	①
(0, 1, 0)	$q_1-\omega, -m, -\alpha$	(*, -, -)	unsure	②
(0, 1, 1)	$-p_1-\omega, n_3, \alpha$	(-, -, +)	unstable	/
(1, 0, 0)	$p+\omega, -R_V^1+q_2, R_A^2-\alpha$	(+, *, *)	unstable	/
(1, 0, 1)	$n_1+p+\omega, -R_V^2-p_2, \alpha-R_A^2$	(+, -, *)	unstable	/
(1, 1, 0)	$-q_1+\omega, R_V^2-q_2, R_A^1-\alpha$	(*, *, *)	unsure	③
(1, 1, 1)	$p_1+\omega, R_V^2+p_2, \alpha-R_A^1$	(+, +, *)	unstable	/
(x_1, y_1, z_1)	$a_1, b_1 = -c_1$	(*, -, +)	unstable	/
(x_2, y_2, z_2)	$a_2, b_2 = -c_2$	(*, -, +)	unstable	/
(x_3, y_3, z_3)	$a_3, b_3 = -c_3$	(*, -, +)	unstable	/
(x_4, y_4, z_4)	$a_4, b_4 = -c_4$	(*, -, +)	unstable	/
(x_6, y_6, z_6)	$a_6, b_6 = -c_6$	(*, -, +)	unstable	/

Note: * indicates an uncertain symbol. The specific values of equilibrium points E_9 to E_{14} will be provided in the appendix.

Under ideal conditions, we hope that the evolutionary game can achieve stability under pure strategy conditions. Therefore, we first conduct an in-depth analysis of the pure strategy set, and then separately discuss the remaining mixed strategy set and equilibrium points in a critical state, to ensure a comprehensive and rigorous assessment of the game's stability.

Based on the eigenvalues and their signs, we analyze three equilibrium points: $E_2(0,0,1)$, $E_3(0,1,0)$, and $E_7(1,1,0)$.

Lemma 1. When $q_1 > \omega$, $q_2 > R_V^2$, and $\alpha > R_A^1 > n_2$, the replicator dynamic system has a unique stable point $E_7(1,1,0)$.

Proof 1. According to Table 3, condition ③ is satisfied, but conditions ① and ② are not. Hence, $E_7(1,1,0)$ is an asymptotically stable point under the current conditions, while $E_2(0,0,1)$ and $E_3(0,1,0)$ are unstable points.

Lemma 1 indicates that when the collusion benefits obtained by the candidates exceed their bribery costs, the collusion benefits obtained by the voters exceed the rewards for reporting, and the cost for spontaneous supervisors exceeds the supervisory benefits, the strategy combination will stabilize at (Bribe, Accept, No Supervision) depending on the initial strategy choices. At this time, the supervisory effectiveness of spontaneous supervisors is low and the collusion willingness of candidates and colluders is strong, posing a severe threat to the blockchain's system security. To avoid the emergence of this strategy combination, sufficient reward amounts should be set to play the role of the incentive mechanism.

Lemma 2. When $q_1 < \omega$ and $\alpha > n_2$, the replicator dynamic system has a unique stable point $E_3(0,1,0)$.

Proof 2. According to Table 3, condition ② is satisfied, but conditions ① and ③ are not. Hence, $E_3(0,1,0)$ is an asymptotically stable point under the current conditions, while $E_2(0,0,1)$ and $E_7(1,1,0)$ are unstable points.

Lemma 2 shows that when the bribery costs borne by the voters exceed their collusion benefits, the strategy combination evolution will stabilize at (No Bribe, Accept, No Supervision). In this state, there is no bribery behavior in the system. The bribery investment and collusion benefits of voters in reality are dynamically changing, and the relationship between them lacks clear correlation. In other words, the relationship between bribery investment and collusion benefits is not externally controlled. Therefore, achieving the condition where bribery costs exceed collusion benefits is something we cannot artificially control for a blockchain system. Thus, we believe this particular stable point lacks practical application value.

Lemma 3. When $q_1 > \omega$, $R_V^2 > q_2$, $R_A^1 > \alpha$, and $n_2 > \alpha$, the replicator dynamic system has a unique stable point $E_2(0,0,1)$.

Proof 3. According to Table 3, condition ① is satisfied, but conditions ② and ③ are not. Hence, $E_2(0,0,1)$ is an asymptotically stable point under the current conditions, while $E_7(1,1,0)$ and $E_3(0,1,0)$ are unstable points.

Lemma 3 indicates that when the bribery costs borne by the voters exceed their collusion benefits, the rewards for reporting chosen by the voters exceed their collusion benefits, and the rewards for supervision chosen by the spontaneous supervisors exceed their costs, the strategy combination evolution will stabilize at (No Bribe, Report, Supervise). In this state, there is no bribery behavior in the system, and voters always choose the reporting strategy while spontaneous supervisors always choose the supervision strategy.

Through the above three Lemmas, we can conclude that by appropriately setting parameters, the evolutionary game can be guided to achieve a stable state that is beneficial for blockchain governance. This finding is significant for designing effective incentive mechanisms and improving the governance quality of blockchain systems.

3.4. Incentive mechanism design

In the previous section, we constructed a tripartite game model for DPoS-based blockchain governance that incorporates incentive mechanisms and discussed its equilibrium points. This analysis demonstrated the feasibility of optimizing the DPoS-based blockchain governance

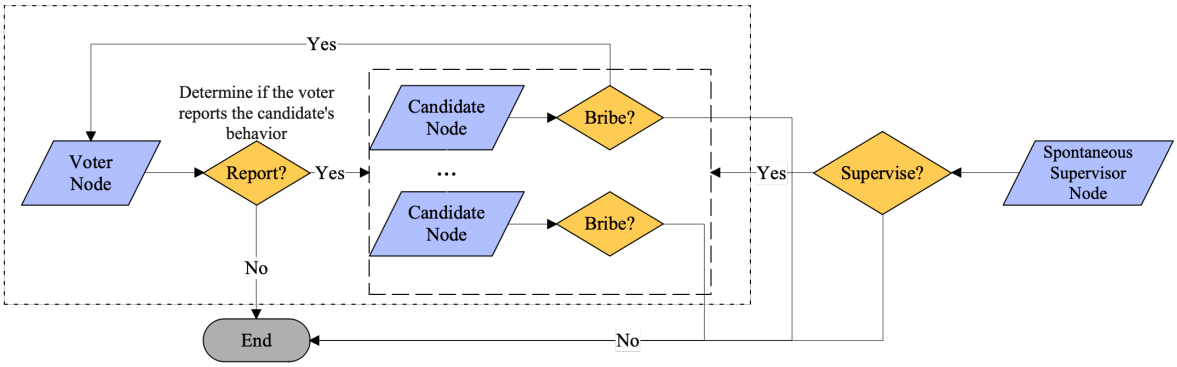

Figure 2. The interaction between the three parties in the game.

mechanism through carefully designed incentive structures. By precisely adjusting the numerical parameters in the improved scheme, the evolutionary game process of blockchain governance can be effectively guided towards a secure and desirable equilibrium state. These parameter relationships essentially reflect the logic and rules of reward distribution within the governance mechanism.

Building on this foundation, this section elaborates on the specific incentive mechanism designed for DPoS-based blockchain systems. This mechanism distributes voting rewards based on the number of voting nodes in the voter group and introduces a reporting mechanism and spontaneous supervisory nodes. When bribery is reported by voting nodes or supervised by spontaneous supervisory nodes, the amount used for bribery and the fines imposed on the candidate are allocated as additional rewards to both the voting nodes and the spontaneous supervisory nodes. These rewards are distributed according to specific rules and weight ratios. The design aims to encourage greater voter participation in blockchain governance and, by incorporating spontaneous supervisory nodes, to foster broader indirect participation in the governance process. This approach effectively counters the risks of centralization and enhances the overall security of the DPoS-based blockchain governance process.

3.4.1. Incentive mechanism design phases

Based on the assumptions outlined in Section 3.2., our study delves into the interaction dynamics among the three parties involved in blockchain voting with the introduction of a governance mechanism. Figure 2 illustrates the interactions among the candidate nodes, voting nodes, and spontaneous supervisory nodes in the tripartite game. In each voting round, candidate nodes must decide whether to engage in bribery, voting nodes must decide whether to report such bribery, and spontaneous supervisory nodes must determine whether to fulfill their supervisory role. The strategic choices made by these parties are independent and do not directly influence one another.

Building on this interaction framework, the incentive mechanism design consists of the following phases:

Phase 1: Voting Decision Phase. In this phase, nodes determine whether to participate in the voting process by staking tokens. Once a node chooses to participate, it enters the tripartite game interaction process. Importantly, voting rewards are not distributed immediately during this phase. Instead, the number of participating nodes is tallied after each election round, and an effective reward distribution period is determined based on this tally. This design encourages active participation in the voting process while mitigating risks associated with immediate reward distribution, such as manipulation.

Phase 2: Reward Qualification Confirmation Phase. Once the reward distribution period determined in Phase 1 ends, this phase verifies the validity of reporting and supervisory

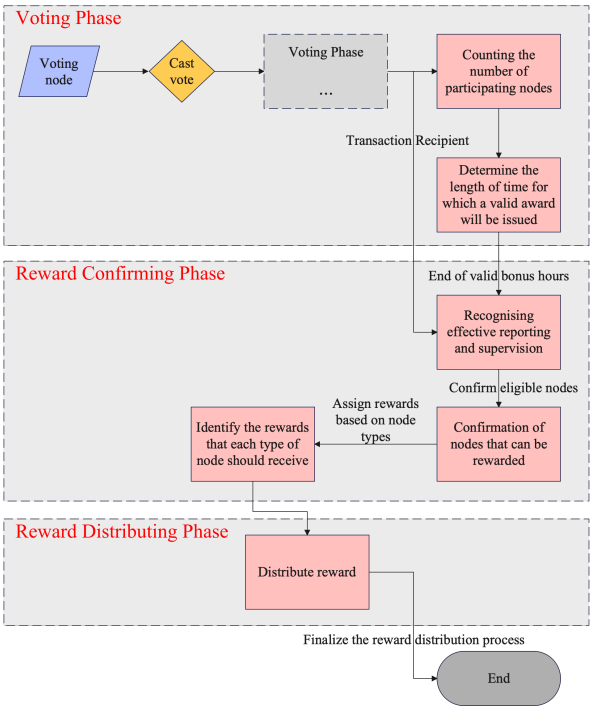

Figure 3. Incentive mechanism process.

actions based on transaction data from the voting decision phase. It identifies which candidate nodes, voting nodes, and spontaneous supervisory nodes qualify for rewards in the respective voting round and calculates the corresponding reward amounts. The primary objective of this phase is to ensure fairness, transparency, and positive behavioral incentives in the reward distribution process.

Phase 3: Reward Distribution Execution Phase. Once the reward qualifications and amounts for each node are confirmed, this phase executes the reward distribution, disbursing the calculated reward amounts to the respective nodes. The reward distribution must ensure timeliness and accuracy to maintain the system's credibility and the participants' motivation. The process of the three phases of the incentive mechanism is illustrated in Figure 3.

3.4.2. Reward functions

Voting Reward Function: For each node that participates in voting and whose voting behavior is compliant and valid, the system will issue rewards based on the node's performance in the last ten voting rounds and the actual participation of voting nodes in the current round. The reward function for voting nodes is specifically expressed as follows:

$$reward_{voter} = \begin{cases} \frac{Q_2}{N_v} + p_2 * \sum_{i=1}^{10} W_i * S(V_i) / \sum_{i=1}^{10} W_i, \ if \ N_v \leq \frac{2L}{3} \\ \frac{Q_2}{N_v}, \ if \ N_v > \frac{2L}{3} \end{cases} \tag{10}$$

where N_v represents the number of valid nodes participating in the voting, P_2 is the reward baseline set by the system, W_i is the weight assigned to each of the last ten voting rounds, and $S(V)$ indicates whether the voting node participated in the voting in that round. In other words, the more actively a voting node participates, the more voting rewards it can obtain.

Reporting Reward Function: The reward function for reporter nodes is specifically expressed as follows:

$$reward_{reporter} = a * (\omega + p) * R_{V_i} / \sum_{j=1}^{n} R_{V_j} \tag{11}$$

where a represents the reward coefficient set by the system for reporting behavior, R_{V_i} indicates the number of valid reports made by the voting node (V_i) in the current round. The more valid reports a voting node makes, the more rewards it can obtain.

Supervisory Reward Function: The reward function for spontaneous supervisory nodes is specifically expressed as follows:

$$reward_{monitor} = \gamma * (\omega + p) * R_{A_i} / \sum_{j=1}^{n} R_{A_j} \tag{12}$$

where γ represents the reward coefficient set by the system for supervisory behavior, R_{A_i} indicates the number of valid supervisory actions performed by the spontaneous supervisory node (A_i) in the current round. Similarly, the more valid supervisory actions a spontaneous supervisory node performs, the more rewards it can obtain.

3.4.3. Algorithm implementation

Next, the algorithms implementing these reward functions are introduced.

The Voting Reward Calculation Algorithm 1 determines rewards for voters based on their recent voting history. It first checks if a node's current vote is valid. If so, it looks at how many voters and token owners are participating. When fewer than two-thirds of token owners vote, the reward is higher and considers the node's voting history over the last ten rounds. If more people vote, the reward is simply shared equally among all voters. After calculating the reward, the algorithm updates how likely the node is to vote next time and records its current voting performance. This approach encourages consistent voting, especially when overall participation is low.

The Reporting Reward Calculation Algorithm 2 determines rewards for participants who

Algorithm 1 Voting Reward Calculation Algorithm

Input: $H, n,$ recent_participation$(H) = \{S(V_1), \ldots, S(V_{10})\}$
Output: Reward_voter(H, n)
if $V_n(H) > 0$ **then**
    // If node H is in this round and the vote is valid, calculate its reward
    $N_v =$ count(voter(n))
    $L =$ count(token_owner)
    if $N_v < 2L/3$ **then**
        for i=1,...,10i = 1, ..., 10 **do**
            weight+ = weight(i) $* S(V_1)$
            sum+ = weight(i)
        end for
        Reward_voter(H, n) = $\frac{Q_2}{N_v} + q^2 * \varepsilon$
    end if
    if $N_v > 2L/3$ **then**
        Reward_voter(H, n) = $\frac{Q_2}{N_v}$
    end if
else
    Reward_voter(H, n) = 0
end if
willing_to_vote(H) = willing_to_vote(H) + [Reward_voter$(H,n) \times Q_2]/N_v \times 5\%$ // Willingness to next vote
for i = 2, ..., 10 **do**
    // Updated performance in the last ten rounds of voting
    $S(V_i) = S(V_{i+1})$
end for
$S(V_i) = V_i(H)$

Algorithm 2 Reporting Reward Calculation Algorithm

Input: H,n,report(H)=R1,...,Rn
Output: Reward_reporter(H, n)
$N_i = \text{count}(R_i)$
$N = \text{count}(R)$
for i = 1, ..., n **do**
    weight += $\frac{1}{N_i \times N}$
end for
Reward_reporter(H, n) = $\alpha \times (\omega + \text{p}) \times$ weight
willing_to_report(H) += 1% $\times$ Reward_reporter(H, n)

report events in a blockchain network. It starts by counting how many reports were made for each event and in total. The algorithm then calculates a weight for each report, giving more importance to reports that are less common. This weight is based on how unique each report is compared to others. The final reward is calculated by combining this weight with some preset values that might represent things like the importance of the report or the current state of the network. After determining the reward, the algorithm slightly increases the reporter's likelihood of reporting again in the future. This approach encourages participants to report events, especially those that others might miss, helping to keep the network well-informed and active.

The Supervisory Reward Calculation Algorithm 3 determines rewards for participants who monitor network activities. It begins by counting how many times each participant has monitored and the total number of monitoring instances across all participants. The algorithm then calculates a weight for each participant's monitoring effort, giving more value to those who monitor more frequently relative to others. This weight is based on how much each participant contributes to the overall monitoring effort. The final reward is calculated by combining this weight with preset factors that might represent the importance of monitoring or current network conditions. After determining the reward, the algorithm slightly increases the participant's likelihood of monitoring again in the future. This approach encourages consistent network monitoring, helping to maintain the network's security and reliability by rewarding those who actively contribute to oversight.

Algorithm 3 Supervisory Reward Calculation Algorithm

Input: H, n, monitor(H) = $\{M_1, \ldots, M_n\}$
Output: Reward_monitor(H, n)
$L_i = \text{count}(M_i)$
$L = \text{count}(M)$
for $i = 1, \ldots, n$ **do**
    weight += $\frac{1}{L_i \times L}$
end for
Reward_monitor(H, n) = $\beta \times (\omega + p) \times$ weight
willing_to_monitor(H) += 1% $\times$ Reward_monitor(H, n)

4. Experiments

In this section, we conduct simulation experiments based on the TRON developer documentation, which adopts the DPoS consensus protocol. We implemented the core components of the DPoS consensus mechanism using Python, defining a series of complex data structures including candidates, voters, and spontaneous supervisors, along with their respective voting, reporting, and supervisory records. After constructing the experimental framework, we validated the results by adjusting various node configurations and system parameters to assess the performance and stability of the proposed incentive mechanism.

4.1. Simulation setup and configuration

We used probabilistic methods to simulate the voting behavior of nodes, including the strategic interactions between voter nodes, candidate nodes, and spontaneous supervisory nodes. Based on TRON's current voting architecture, we configured the simulation with 1000 potential voter nodes, 400 candidate nodes, and 500 potential supervisory nodes. During the main voting phase, 400 nodes are selected as candidates, while in the previous election phase, 27 candidate nodes were chosen as committee members. Nodes ranked 28 to 127 serve as committee partners, with the remaining nodes classified as candidate reserves. Since only committee members receive block production rewards, the high level of centralization in blockchain governance may incentivize improper behavior from committee partners or members aiming to secure block production rewards.

4.2. Preliminary experiment and results

In the preliminary experiment, we established a set of hypothetical parameters to simulate the voting process. Specifically, we assume that at the start of voting, potential voter nodes have a 50% probability of participating. Within a 20-round election cycle, 10 candidate nodes have a 50% likelihood of engaging in bribery, while bribed voter nodes have a 20% probability of reporting such behavior, and spontaneous supervisors have a 10% probability of intervening. Based on this configuration, the experiment compared and analyzed the number of nodes actively participating in voting under both the original DPoS consensus mechanism and the improved DPoS consensus mechanism with the incentive mechanism proposed in this study.

Figure 4 illustrates the average number of voter nodes across 10 simulations. As shown, in the original DPoS model (labeled "original"), approximately 50% of potential voters choose to participate in each round. However, in the improved DPoS model (labeled "improved") with the incentive mechanism based on participation and voting performance, the number of potential voter nodes actively participating in voting gradually increases. By the end of the 20th election round, the proportion of nodes converting into actual voters consistently rises to over 60%. This indicates that the proposed incentive mechanism effectively boosts node participation in voting, thereby enhancing the governance security of the DPoS blockchain.

Figure 4. Number of participating voting nodes before and after improvement. It shows an increase in voter participation in the improved DPoS model compared to the original model. By the 20th round, the improved model consistently achieves over 60% node participation, indicating the effectiveness of the proposed incentive mechanism.

Table 4. Voters and spontaneous supervisors eliminate the number of malicious nodes.

simulation	Round																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
M1	0	2	2	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
M2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0	1	0	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
M3	0	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	0	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M4	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0
	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M5	1	2	2	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

4.3. Analysis of malicious node removal

Figure 5 presents the cumulative number of nodes removed across different simulation runs (M1–M5 represent various simulations) as the election rounds progress. Each line represents a distinct simulation run, and the y-axis shows the cumulative number of bribed nodes removed, while the x-axis indicates the voting rounds. In the early rounds (1–5), there is a sharp increase in the number of bribed nodes removed across all simulations. This indicates that the initial detection and removal mechanisms are highly effective at identifying a large proportion of malicious nodes quickly. Between rounds 5 and 10, the rate of node removal starts to slow down. This plateau suggests that the most obvious bribed nodes have already been removed, and the remaining malicious nodes are either less active or more adept at evading detection. By rounds 15 to 20, the cumulative removal curves start to flatten out, reaching a total of 10 nodes removed in each simulation. This stabilization indicates that the system has effectively identified and eliminated the majority of bribed nodes, with minimal new detections in the final rounds. Throughout the 20 rounds of voting, effective collaboration between voters and supervisors led to the successful identification and removal of 10 candidate nodes suspected of engaging in bribery before the conclusion of the voting process.

This outcome confirms the simulation model's stability and underscores its reproducibility and reliability. Furthermore, it underscores the model's potential effectiveness in detecting and eliminating malicious behavior within the blockchain governance system.

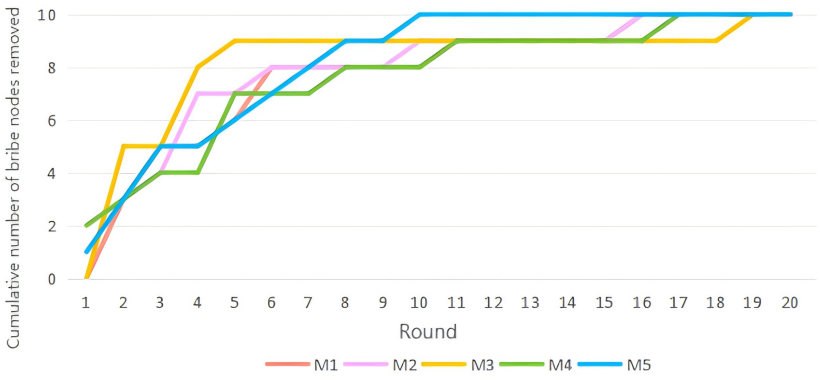

Figure 5. Cumulative node removal in simulations. Cumulative removal of malicious nodes in different simulation runs (M1–M5) over 20 voting rounds, highlighting the stability and effectiveness of the blockchain governance model in detecting and eliminating bribery.

Table 4 presents the number of bribery nodes detected by voters (indicated in white) and spontaneous supervisors (displayed in gray) during the corresponding rounds across five simulations.

4.4. Role analysis of reporting and supervision

Next, we analyze the distinct roles of reporting and supervision in the model. To separately observe their effects, we conducted simulations where the probabilities of voters choosing to report and spontaneous supervisors choosing to supervise were set to zero. Under the scenario without reporting, the average number of bribery nodes removed across 10 simulations was 8.9. In the scenario without supervision, this average dropped slightly to 8.3. These results highlight that effective detection of improper behavior requires the combined efforts of both reporting and supervision mechanisms.

At the end of 10 simulations, the average voting intention of potential voters increased to 66.13%, and the average supervision intention of supervisor nodes rose to 10.88%. However, the reporting probability of voter nodes unexpectedly decreased to 19.42%. A closer examination of the reporting intentions among all voters revealed that 3% of voters had a reporting intention below 20%, 96% of voters maintained a reporting intention at exactly 20%, and only 1% of voters had a reporting intention above 50%. This pattern can be attributed to the assumptions and simulation design: given the limited supervisory capacity of spontaneous supervisor nodes each round and the relatively low reporting intention among voter nodes, collusion between candidate nodes and voter nodes occurred without adequate oversight. Furthermore, since voters adopting an acceptance strategy were assumed to exhibit reduced reporting intentions, this phenomenon was observed. To further validate this hypothesis, we conducted simulations with initial reporting probabilities set at 30% and 10%. In the 10-round simulation with an initial reporting probability of 30%, the average reporting intention was 29.71%, whereas in the simulation with an initial reporting probability of 10%, the average reporting intention dropped to 9.43%. These results indicate that the incentive mechanism proposed in this paper can provide some resistance against bribery attempts by candidate nodes.

We identified the potential voter node with the highest voting intention in the system, whose voting behavior can be represented as [True, True, True, True, True, True, True, True, True, True]. After 10 simulation rounds, this node's voting intention reached as high as 99.16%. This demonstrates that the incentive mechanism proposed in this paper can effectively motivate both potential voting nodes and spontaneous supervisor nodes, directly or indirectly increasing participation in the blockchain governance process.

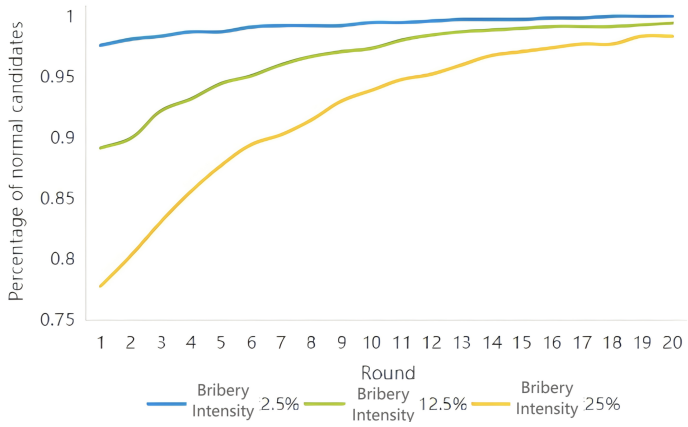

Figure 6. Normal candidate proportions across bribery intensities. The blue line (2.5% intensity) remains nearly flat, indicating a high, stable percentage of normal candidates. The green (12.5%) and yellow lines (25%) show gradual increases, with the system demonstrating resilience and recovery despite higher bribery levels.

4.5. *Impact of bribery intensity on voting performance*

Then, we examine the voting performance under different levels of bribery intensity. We define varying bribery intensities as the ratio of the number of candidate nodes likely to engage in bribery to the total number of candidate nodes. For comparison, we set three levels of bribery intensity: 2.5%, 12.5%, and 25%.

Figure 6 shows the percentage of normal candidates across 20 rounds under three different bribery intensity levels: 2.5%, 12.5%, and 25%. The blue line, representing the lowest bribery intensity (2.5%), starts close to 1 and remains nearly flat, indicating a consistently high percentage of normal candidates. The green line, corresponding to 12.5% bribery intensity, shows a gradual increase from around 0.9 to close to 1, indicating a moderate recovery of normal candidates over time. The yellow line, representing the highest bribery intensity (25%), starts lower at approximately 0.8 but also increases steadily, although it remains below the other two lines. This suggests that higher bribery intensity initially reduces the proportion of normal candidates, but over time, the system shows resilience, gradually improving the percentage of normal candidates.

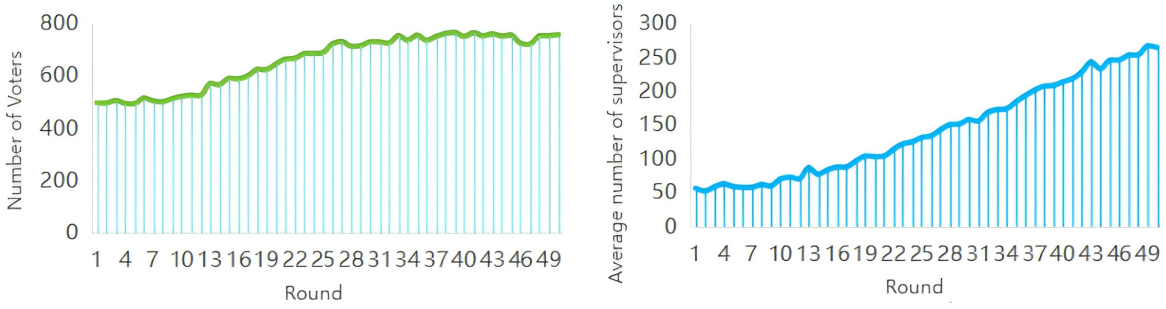

Figure 7. Number of voters and spontaneous supervisors under different levels of bribery. The left chart shows a steady rise in the number of voters, starting from approximately 400 and reaching nearly 800, indicating increasing engagement over time. The right chart highlights the growth in average supervisor numbers, beginning around 50 and progressively climbing to nearly 300. Both charts demonstrate a consistent upward trend in participation across the rounds.

Figure 7 consists of two line charts that illustrate the trends in the number of voters and the average number of supervisors over multiple rounds under a 25% bribery intensity scenario. In the left chart, the number of voters starts at approximately 400 and gradually increases to nearly 800, showing a steady upward trend across the 50 voting rounds. In the right chart, the average number of supervisors begins at around 50 and rises steadily to nearly 300. Both charts indicate a consistent increase in participation over time, reflecting a growing engagement from both voters and supervisors despite the 25% bribery intensity.

In this experiment, we simulate a network with 400 potential nodes, of which 25% (100 nodes) are initially bribed. We choose the 25% bribery intensity as it represents a significant threat that could potentially destabilize the system, allowing us to demonstrate TriGuard's effectiveness against substantial bribery attempts. Figure 8 displays the cumulative number of bribed nodes removed over a series of 50 rounds under a 25% bribery intensity scenario. Initially, the number of removed nodes increases rapidly, reaching around 80 by the 15th round. After this point, the growth rate slows down, with the cumulative total gradually approaching 100 nodes by the end of the 50 rounds. The curve flattens out towards the later rounds, indicating that the removal of bribed nodes becomes less frequent as the rounds progress. This pattern suggests that while the initial strategy is effective, it encounters diminishing returns over time. Additionally, as the number of bribed nodes in the system decreases, the curve's slower convergence implies that more rounds are required to eliminate the remaining bribed

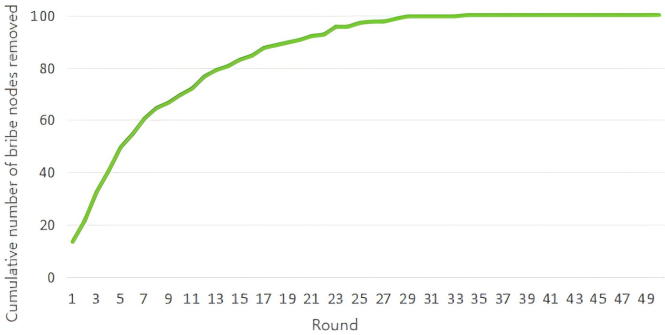

Figure 8. Accumulated removal of malicious nodes at 25% bribery intensity. The flattening of the curve toward later rounds indicates that the frequency of removing bribed nodes decreases over time, highlighting the effectiveness of early strategies with diminishing returns in subsequent rounds.

nodes. The slowing growth rate and flattening curve in later rounds indicate that as the number of remaining bribed nodes decreases, they become increasingly difficult to detect and remove, possibly due to more sophisticated concealment strategies employed by the remaining bribed nodes.

In summary, our experimental results demonstrate the significant improvements and robustness of the proposed incentive mechanism in enhancing DPoS blockchain governance. The improved model showed a marked increase in voter participation, rising from an initial 50% to over 60% by the 20th election round. The collaborative effort between voters and supervisors proved highly effective, successfully identifying and removing all 10 candidate nodes suspected of bribery within 20 voting rounds. The mechanism's resilience was evident even under high bribery intensity scenarios, where the proportion of normal candidates increased from 80% to nearly 90% over 20 rounds in a 25% bribery intensity setting. Long-term simulations revealed sustained growth in engagement, with the number of active voters doubling and supervisors increasing six-fold over 50 rounds. The combined impact of reporting and supervision mechanisms was crucial, removing an average of 8.9 bribery nodes when only supervision was active and 8.3 when only reporting was employed. By the end of the simulations, the average voting intention of potential voters increased to 66.13%, while the average supervision intention of supervisor nodes rose to 10.88%, underscoring the effectiveness of our incentive mechanism. These results collectively demonstrate the robustness, efficacy, and long-term stability of our proposed incentive mechanism in enhancing DPoS blockchain governance.

5. Related Work

5.1. Incentive mechanisms

Token-based incentive mechanisms are the most common in blockchain systems, driving participant behavior by issuing cryptocurrencies or tokens. These mechanisms include mining rewards, staking rewards, and transaction fee dividends, which encourage activities like verifying transactions, maintaining network security, and participating in governance. Wang *et al.*[18] proposed a DPoS consensus mechanism incorporating improved reward distribution through economic incentives based on HK clustering. Similarly, Almusaylim *et al.*[19] developed a K-anonymous location privacy protection scheme, using general tokens as incentives in an Ethereum-based experimental environment.

Incentive mechanisms play a two-fold role in facilitating governance. Firstly, as suggested by De Filippi *et al.* [20], incentive mechanisms attract individuals to participate in governance issues. Secondly, according to Wright Jr [21], incentive mechanisms also enable different interest groups to make decisions as a whole.

Reputation-based incentive mechanisms assess user behavior to form a reputation table, which influences node privileges. Higher reputations lead to greater rewards, while lower reputations reduce privileges. For example, Chen *et al.*[22] incorporated reputation rewards in the DPoS consensus mechanism, where consistent validators receive fixed reputation increases, influencing the distribution of transaction fees and deposits. Luo *et al.*[23] proposed a reputation-based election scheme to counter bribery in DPoS blockchain elections, ensuring that candidates with high reputations are elected.

Reciprocity-based incentive mechanisms focus on the contributions made by user nodes, enforcing reciprocity to prevent free-riding strategies. Shin[24] introduced T-Chain, a distributed cooperative computing fairness incentive mechanism that ensures nodes that do not contribute are unable to use public resources, thus promoting fairness. Table 5 provides a comprehensive overview of key studies in incentive mechanisms, summarizing their main contributions, unique features, and limitations, thus offering a concise reference for the current state of research in this field.

Table 5. Incentive Mechanisms Research.

Author(s)	Focus	Main Contribution	Unique Features	Limitations
Wang *et al.*[18]	DPoS consensus	Improved reward distribution	HK clustering-based economic incentives	Limited to DPoS systems
Almusaylim *et al.*[19]	Privacy protection	K-anonymous location privacy scheme	Ethereum-based experimental environment	Focused on location privacy only
De Filippi *et al.*[20]	Governance participation	Incentive mechanisms attract participation	Highlights role of incentives in governance	Lack of empirical data to support its claims
Wright Jr[21]	Decision-making	Incentives enable collective decision-making	Focus on interest group dynamics	Potentially undermine the equity of democratic processes
Chen *et al.*[22]	DPoS consensus	Reputation-based rewards	Incorporation of validator consistency	Limitation in addressing deeper issues
Luo *et al.*[23]	DPoS elections	Anti-bribery scheme	Reputation-based candidate election	Potential for reputation manipulation
Shin *et al.*[24]	Cooperative computing	T-Chain fairness mechanism	Prevents free-riding	Not Strictly Fair

5.2. Blockchain governance

Blockchain governance involves decision-making rights, accountability, and incentives, as highlighted by Beck *et al.*[25]. They describe a framework where decision-making pertains to authority within the blockchain, accountability assigns responsibility for outcomes, and incentives drive participant behavior.

In the consensus layer, which is critical for blockchain security, Bao *et al.*[26] and Du *et al.*[27] explore consensus protocols, their security issues, and their application scenarios, offering insights into selecting appropriate algorithms.

For DPoS-based blockchain governance, Chao Li *et al.*[13] analyzed the resistance to takeovers in DPoS systems, showilng how token-based voting governance impacts resistance levels. Liu *et al.*[28] conducted a systematic review identifying challenges in blockchain governance, while Singh *et al.*[29] discussed the formation of DAOs to formalize and enforce governance policies. Kim[30] introduced a stochastic game framework for identifying strategies to prevent network failures caused by attackers.

In terms of attack resilience, Eyal *et al.*[31] introduced the concept of selfish mining in Bitcoin, questioning its incentive compatibility. This concept was further optimized by

Sapirshtein *et al.*[32], who proposed algorithms to quantify the resources required for selfish mining to be profitable. Gervais *et al.*[33] developed a framework to counter double-spending and selfish mining in PoW systems.

Other attacks include blocking attacks introduced by Kwon *et al.*[34], where large mining pools could dominate by attacking others without falling into a "miners' dilemma." Gao *et al.*[35] studied Power Adjustment Withholding (PAW) and Bribery Selfish Mining (BSM), showing PAW's effectiveness in avoiding dilemmas and BSM's increased attack risks. Gazi *et al.*[36] further analyzed how resource centralization affects Bitcoin's security thresholds, offering insights into maintaining network integrity.

The security of decentralized governance has recently garnered significant attention. Jeong *et al.* [37] conducted a theoretical study on determining the optimal number of votes per account in DPoS blockchains using the approval voting rule. Stroponiati *et al.* [38] examined the governance of six DAOs, where decisions are made through stake-weighted votes. Their findings revealed that these projects were highly centralized.

6. Conclusion

This paper presents TriGuard, an enhanced governance mechanism designed to tackle bribery and collusion in DPoS systems, which are critical threats to the decentralization of Web 3.0. By integrating a tripartite evolutionary game model with targeted incentive mechanisms, TriGuard fosters fair participation, encourages the reporting of bribery, and strengthens supervisory actions. Our simulations and analysis show that TriGuard effectively increases voter engagement, reduces the influence of malicious actors, and enhances the overall security and decentralization of blockchain governance. TriGuard tackles bribery and enhances the long-term sustainability of decentralized ecosystems by fostering resilient governance. Future work will focus on refining TriGuard's adaptability to changing network conditions and exploring its applicability to other consensus mechanisms, further advancing decentralized governance.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2023YFB2704100), the National Natural Science Foundation of China (No. 62202038, 62472022) and the Aeronautical Science Foundation of China (No. 2022Z0660M1001).

Conflicts of interests

The authors declared that they have no conflicts of interests.

Authors' contribution

Conceptualization, L.J.Y.; Methodology, Formal analysis, Software, L.J.Y., S.X.C., L.C.; Visualization, Investigation, L.J.Y., S.X.C.; Data curation, Software, X.Y.F., Y.Y.F.; Writing–Original draft preparation, S.X.C., X.Y.F.; Writing– Reviewing and Editing, L.J.Y., L.C., Y.Y.F., Y.W.Y.; Supervision, L.C., Y.W.Y.; All authors have read and agreed to the published version of the manuscript.

References

[1] Alabdulwahhab FA. Web 3.0: the decentralized web blockchain networks and protocol innovation. In *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia, 04–06 April, 2018, pp. 1–4.

[2] Zheng J, Chen Q, Su C, Huang H. BrokerFi: A DeFi dApp Built upon Broker-based Blockchain. In *2023 IEEE 29th International Conference on Parallel and Distributed*

Systems (ICPADS), Dan Zhou, China, 17–21 December, 2023, pp. 1817–1825.
[3] Fritsch R, Müller M, Wattenhofer R. Analyzing Voting Power in Decentralized Governance: Who controls DAOs? *Blockchain: Res. Appl.* 2024, 5(3):100208.
[4] Rehman W, e Zainab H, Imran J, Bawany NZ. NFTs: Applications and challenges. In *2021 22nd International Arab Conference on Information Technology (ACIT)*, Muscat, Oman, 21–23 December, 2021, pp. 1–7.
[5] Zheng P, Han M, Kuang H, Yuan H, Chen X. Analysis and prospects for blockchain testing and evaluation. *Blockchain* 2024, 2(1):29–58.
[6] Yang F, Ding Z, Yu Y, Sun Y. Interaction mechanism between blockchain and IPFS. *Blockchain* 2023, 1(2):24–25.
[7] Li C, Xu R, Duan L. Characterizing Coin-Based Voting Governance in DPoS Blockchains. In *Proceedings of the International AAAI Conference on Web and Social Media*, New York, USA, 3–6 June, 2024, pp. 1148–1152.
[8] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review* 2008 .
[9] Li C, Xu R, Palanisamy B, Duan L, Shen M, *et al.* Blockchain Takeovers in Web 3.0: An Empirical Study on the TRON-Steem Incident. *ACM Trans. Web* 2024 .
[10] Xu B, Luthra D, Cole Z, Blakely N. EOS: An architectural, performance, and economic analysis. *Retrieved June* 2018, 11(2019):41.
[11] Li C, Xu R, Duan L. Liquid democracy in DPoS blockchains. In *Proceedings of the 5th ACM international symposium on blockchain and secure critical infrastructure*, Melbourne, Australia, 10 – 14 July, 2023, pp. 25–33.
[12] Li J, Zhang W, Zhang L. Research and Implementation of Improved DPoS Consensus Mechanism. In *2023 3rd International Signal Processing, Communications and Engineering Management Conference (ISPCEM)*, Montreal, Canada, 25–27 November, 2023, pp. 150–156.
[13] Li C, Palanisamy B, Xu R, Duan L, Liu J, *et al.* How hard is takeover in dpos blockchains? understanding the security of coin-based voting governance. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, Copenhagen, Denmark, 26–30 November, 2023, pp. 150–164.
[14] von Neumann J. *Theory of Games and Economic Behavior*, Princeton University Press2024.
[15] Xu Y, Wu W, Gong Y, Wang KY, Hu C, *et al.* Frustum: achieving high throughput in blockchain systems through hierarchical and pipelined sharding. *Blockchain* 2024, 2(1):30–53.
[16] Huang Z, Zhu J, Huang Z, Xu Y, Yen J, *et al.* Safeguarding the unseen: a study on data privacy in DeFi protocols. *Blockchain* 2023, 1(2):48–66.
[17] Deng J, Pan H, Zhang S, Zou B. Mean-Variance Tradeoff of Bitcoin Inverse Futures. *Blockchain* 2024, 2(1):99–110.
[18] Wang L, Xu P, Su W, Li Y, Chen X. Research on Improvement of blockchain DPOS consensus mechanism based on HK clustering. In *China Automation Congress*, Beijing, China, 22–24 October, 2021, pp. 1167–1172.
[19] Almusaylim Z, Jhanjhi NZ. Comprehensive review: Privacy protection of user in location-aware services of mobile cloud computing. *Wireless Pers. Commun.* 2020 111(1):541–564.
[20] De Filippi P, Loveluck B. The invisible politics of bitcoin: governance crisis of a decentralized infrastructure. *Internet policy rev.* 2016, 5(4):1–32.
[21] Wright Jr D. Quadratic voting and blockchain governance. *UMKC L. Rev.* 2019, 88:475.
[22] Chen Y, Liu F. Research on improvement of DPoS consensus mechanism in collaborative governance of network public opinion. *Peer peer. Netw. Appl.* 2022, 15(4):1849–1861.
[23] Luo Y, Chen Y, Chen Q, Liang Q. A new election algorithm for DPoS consensus

mechanism in blockchain. In *The 7th International Conference on Digital Home*, Guilin, China, 30 November – 01 December , 2018, pp. 116–120.

[24] Shin K, Joe-Wong C, Ha S, Yi Y, Rhee I, *et al.* T-chain: A general incentive scheme for cooperative computing. *IEEE/ACM Trans. Networking* 2017, 25(4):2122–2137.

[25] Guan C, Ding D, Guo J. Web3.0: A Review And Research Agenda. In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, Ho Chi Minh City, Vietnam, 20–22 December, 2022, pp. 116–120.

[26] Bao Q, Li B, Hu T, Sun X. A survey of blockchain consensus safety and security: State-of-the-art, challenges, and future work. *J. Syst. Software* 2023, 196:111555.

[27] M D, X M, Z Z, X W, Q C. A review on consensus algorithm of blockchain. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, Banff, Canada, 05-08 October, 2017, pp. 2567–2572.

[28] Liu Y, Lu Q, Yu G, Paik HY, Zhu L. Defining blockchain governance principles: A comprehensive framework. *Inf. Syst.* 2022, 109:102090.

[29] Singh M, Kim S. Blockchain technology for decentralized autonomous organizations. *Adv. Comput.* 2019, 115:115–140.

[30] Kim SK. Blockchain governance game. *Comput. Ind. Eng.* 2019, 136:373–380.

[31] Eyal I, Sirer EG. Majority is not enough: Bitcoin mining is vulnerable. *Commun. ACM* 2018, 61(7):95–102.

[32] Sapirshtein A, Sompolinsky Y, Zohar A. Optimal selfish mining strategies in bitcoin. *Financial Cryptography and Data Security: 20th International Conference* 2017, 515–532.

[33] Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, *et al.* On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, Vienna, Austria, 24 – 28 October, 2016, pp. 3–16.

[34] Kwon Y, Kim D, Son Y, Vasserman E, Kim Y. Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, Dallas Texas, USA, 30 October–3 November, 2017, pp. 195–209.

[35] Gao S, Li Z, Peng Z, Xiao B. Power adjusting and bribery racing: Novel mining attacks in the bitcoin system. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, London, United Kingdom, 11–15 November, 2019, pp. 833–850.

[36] Gaži P, Kiayias A, Russell A. Tight consistency bounds for bitcoin. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Virtual Event, USA, 9–13 November, 2020, pp. 819–838.

[37] Jeong SE. Centralized decentralization: Does voting matter? simple economics of the dpos blockchain governance. *Simple Economics of the DPoS Blockchain Governance* 2020 .

[38] Stroponiati K, Abugov I, Varelas Y, Stroponiatis K, Jurgeleviciene M, *et al.* Decentralized governance in DeFi: Examples and pitfalls. *Tech. rep.* 2020 .