Towards a General Theory of Consensus: Probabilistic Distributed Fault Tolerance Consensus, Decentralized Voting in DAOs, and A Unified Consensus Framework

Yixuan Fan

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

July 2025

University of Glasgow

College of Science & Engineering

Statement of Originality

Name: Yixuan Fan **Registration**
Number: XXXXXXX

I certify that the thesis presented here for examination for a PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that if any issues are raised regarding good research practice based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature:

Date: August 25, 2025

Abstract

Consensus serves as a foundational mechanism in both social coordination and distributed technical systems. While machine consensus research in engineering focuses on fault tolerance and synchronization, social science emphasizes human deliberation, participation, and governance. However, the increasing convergence of human and machine decision-making, exemplified by decentralized autonomous organizations (DAOs), intelligent agents, and cyber-physical social systems, demands a more integrated and theoretically robust understanding of consensus. This thesis addresses this interdisciplinary gap by investigating consensus across three interconnected dimensions: probabilistic fault-tolerant consensus systems, human-driven voting mechanisms in DAOs, and a unified conceptual framework bridging human and machine consensus.

The first part of the thesis focuses on distributed fault-tolerant consensus in uncertain environments. Traditional approaches often rely on deterministic assumptions about node failures and fixed quorum rules. These assumptions may fail to reflect real-world systems where node behaviour is influenced by heterogeneous reliability and probabilistic failures. To address this limitation, a probabilistic modelling framework is proposed, treating node reliability as a stochastic variable. Within this framework, consensus outcomes are classified into three categories: safe, risky, and compromised. A new concept, referred to as the reliability quorum, is introduced to provide a more flexible threshold for achieving consensus based on targeted reliability levels. This model enables system designers to tailor fault tolerance according to specific reliability requirements, providing both analytical clarity and practical adaptability.

The second part investigates consensus in decentralized systems primarily driven by human-oriented agents, using DAO voting as a representative case. In contrast to deterministic coordination among machines, DAO consensus arises from voluntary participation, heterogeneous voting power, and non-uniform approval conditions. To guide the analysis, the thesis introduces the DAO governance triangle alongside the SEED framework, which qualitatively evaluates voting mechanisms across four dimensions: Security, Efficiency, Effectiveness, and Decentralization. Building on this conceptual foundation, the study proceeds

to a quantitative investigation of two key SEED dimensions. For decentralization, a stochastic process model is proposed to capture probabilistic participation and power distribution, leading to the formulation of the Consistency Rate and the Decentralization Coefficient as quantitative indicators. For efficiency, the model is further extended to characterize the interactions among participation probability, voting duration, and approval rate, enabling a formal evaluation of voting responsiveness and resource usage. Simulation results support both aspects of the analysis, revealing how power concentration, turnout behaviour, and mechanism design jointly influence decentralization and efficiency in DAO voting.

The third part presents a unifying conceptual framework to analyse consensus across human, machine, and human-machine hybrid systems. Despite disciplinary differences, the thesis identifies three core components of any consensus process: participants (the actors of agreement), communication (the medium of exchange), and state (the evolving representation of agreement). Framing consensus as an entropy-reduction process that resolves cognitive or informational divergence, this abstraction enables comparative analysis across diverse systems. The framework also distinguishes among human consensus, machine consensus, and human-machine hybrid consensus, and offers design guidelines aligned with the characteristics and limitations of each.

Together, these three threads construct a comprehensive theory of consensus that connects distributed computation, social governance, and emerging hybrid collectives. By integrating modelling, evaluation, and abstraction, this thesis contributes a multi-layered foundation for understanding and designing consensus mechanisms that are robust, scalable, and trustworthy in increasingly decentralized and intelligent environments.

Contents

List of Publications

Journal

- **Y. Fan**, L. Zhang, R. Wang, M. A. Imran, ***Insight into Voting in DAOs: Conceptual Analysis and A Proposal for Evaluation Framework***
  in IEEE Network, doi: 10.1109/MNET.137.2200561.

- **Y. Fan**, H. Wu, Z. Dong, Z. Li, L. Zhang, ***From Human to Machine Networks: A Framework for Integrating Consensus Approaches***
  in IEEE Network, doi: 10.1109/MNET.2025.3559378.

- **Y. Fan**, K. Ma, L. Zhang, X. Wang, ***Outcome Uncertainty Analysis and Quorum Redesign in Probabilistic Consensus***
  Submitted to IEEE Transactions on Emerging Topics in Computing.

- **Y. Fan**, L. Zhang, Y. Sun, X. Lin, Y. Fang, ***Decentralized Autonomous Organizations (DAOs) Voting: Modeling and Analysis of Decentralization Performance***
  Submitted to Information Sciences.

- H. Xu, **Y. Fan**, W. Li, L. Zhang, ***Wireless Distributed Consensus for Connected Autonomous Systems***
  in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2022.3229746.

- Y. Li, **Y. Fan**, L. Zhang, J. Crowcroft, ***RAFT Consensus Reliability in Wireless Networks: Probabilistic Analysis***
  in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2023.3257402.

- X. Lin, **Y. Fan**, L. Zhang, K. Ma, Y. Sun, A. Tukmanov, Q. Abbasi, and M. A. Imran, ***Resource Allocation for RIS-aided mmWave System with Cooperative and Non-cooperative Base Stations***
  in IEEE Transactions on Vehicular Technology, doi:10.1109/TVT.2024.3493772.

Conference

- **Y. Fan**, Z. Zhou, Z. Qiao, Y. Sun, L. Zhang, ***Efficiency Analysis of Decentralized Autonomous Organization (DAO) Voting Mechanisms***
  in IEEE Globecom Workshop, 2024.

- **Y. Fan**, Z. Zhou, Z. Qiao, L. Zhang, ***Decentralized Governance and Technology Integration in DAOs within the Web3 Ecosystem***
  in IEEE Global Blockchain Conference, 2024.

- Z. Zhou, **Y. Fan**, X. Lin, L. Zhang, M. A. Imran, O. Onireti, ***Base Station-enabled PBFT Consensus Network: An Outlook and Performance Analysis***
  in IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2024.

- C. Guo, Z. Zhou, H. Xu, **Y. Fan**, X. Zhang, L. Zhang, ***Besharing: A Copyright-aware Blockchain-enabled Knowledge Sharing Platform***
  in IEEE 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), 2022.

- Z. Zhou, C. Guo, H. Xu, X. Zhang, **Y. Fan**, L. Zhang, ***Be-dns: Blockchain-enabled Decentralized Name Services and P2P Communication Protocol***
  in IEEE 9th World Forum on Internet of Things (WF-IoT), 2023.

Survey

- H. Wu, C. Yue, **Y. Fan**, Y. Li, D. Flynn, L. Zhang, ***Half a Century of Distributed Byzantine Fault-Tolerant Consensus: Design Principles and Evolutionary Pathways***
  submitted to ACM Computing Surveys, arXiv preprint arXiv:2407.19863.

List of Figures

List of Acronyms

AI	Artificial Intelligence
AMM	Automated Market Maker
BFT	Byzantine Fault Tolerance
CAV	Connected and Autonomous Vehicle
CFT	Crash Fault Tolerance
CBDC	Central Bank Digital Currency
DAO	Decentralized Autonomous Organization
DApp	Decentralized Application
DC	Distributed Consensus
DEX	Decentralized Exchange
DLT	Distributed Ledger Technology
ETH	Ethereum
GEN	Genesis Token (used in Holographic Consensus)
IoT	Internet of Things
IIoT	Industrial Internet of Things
MAS	Multi-Agent System
ML	Machine Learning
NFT	Non-Fungible Token
PBFT	Practical Byzantine Fault Tolerance
PoA	Proof of Authority
PoS	Proof of Stake
PoW	Proof of Work
RAFT	RAFT Protocol
RM	Relative Majority
AM	Absolute Majority
VT	Voting Power Threshold
QT	Quorum Threshold
SEED	Security, Efficiency, Effectiveness, Decentralization

TBQ	Token-Based Quorum
V2V	Vehicle to Vehicle
V2X	Vehicle to Anything
WDC	Wireless Distributed Consensus

Glossary

Safety	No two correct processes decide differently; once a value is decided it cannot be contradicted later.
Liveness	Every correct process eventually reaches a decision under the assumed timing model.
FLP Impossibility	In a fully asynchronous system with possible crash failures, no deterministic consensus protocol can guarantee both safety and liveness.
Partial Synchrony	A system model where message delays and processing times are eventually bounded, enabling progress in practical consensus protocols.
State Machine Replication (SMR)	Technique for building fault-tolerant services by replicating deterministic state machines and enforcing a total order of operations.
Quorum	The minimum number of confirmations or votes required for a decision to be valid.
Intersection Quorum	A quorum rule requiring that any two quorums overlap in at least one correct node, ensuring agreement.
Reliability Quorum	The minimal confirmation threshold selected to achieve a target probability of consensus success under probabilistic assumptions.
Safe State	A state where reliable nodes form a sufficient majority to guarantee consistent and correct consensus outcomes.
Risky State	A state where consensus is still possible but not guaranteed, as reliable nodes hold a majority without securing full protection against inconsistencies.
Compromised State	A state where consensus cannot be achieved, either because reliable nodes fall below the required threshold or the quorum settings prevent agreement.

Fault Tolerance	The ability of a system to continue operating correctly despite a bounded number of component failures.
Byzantine Fault	Arbitrary or adversarial behaviour of nodes, including sending conflicting or false messages.
Probabilistic Consensus	A consensus paradigm that models node or link reliability probabilistically and achieves high reliability under uncertainty.
Decentralized Autonomous Organization (DAO)	An organization governed by smart contracts where decisions are made collectively by members on a blockchain.
Decentralization Coefficient	A quantitative indicator measuring how decision power and participation are distributed across DAO members.
SEED Metrics	A four-dimensional evaluation of DAO governance: Security, Efficiency, Effectiveness, and Decentralization.
Smart Contract	On-chain program that autonomously enforces predefined rules and executes actions once conditions are met.
Distributed Ledger	An append-only, tamper-evident record replicated across nodes that stores transactions and state transitions.

Acknowledgements

First and foremost, I would like to express my heartfelt and profound gratitude to my supervisor, Professor Lei Zhang, for his unwavering guidance, consistent encouragement, and invaluable support throughout every stage of my doctoral journey. His deep expertise, constructive feedback, and patient mentorship not only shaped the direction of my research but also profoundly influenced my academic thinking and professional growth.

I am also deeply thankful to my second supervisor, Dr. Yao Sun, for his insightful advice, technical guidance, and generous help during my research.

I am profoundly grateful to my parents for their unwavering love, understanding, and encouragement. Their enduring strength provided the steady foundation upon which I was able to build and complete this journey.

To my love, Zhixiang, thank you for your love, companionship, and all the care and understanding you have shown throughout these years. Your presence has been a constant source of joy and a profound sense of peace in my life.

I also wish to thank my friends and colleagues, whose company and kindness brought joy and unforgettable memories to my doctoral life. Your support made this journey more meaningful, fulfilling, and full of laughter.

Finally, thanks to my little dog Russell. You are silly, clever, and surprisingly thoughtful, and you always made sure I took more breaks than I planned to.

Chapter 1

Introduction

The word “consensus" is explained as “Agreement in opinion, feeling, or purpose among a group of people, especially." in the Oxford English dictionary [3]. The broad discussion on consensus in human society is an ancient topic. Initially, from the early functionalist perspective that emerged in the late 19th and early 20th centuries, consensus was viewed as a manifestation of social cohesion and collective consciousness [4]. By the mid-20th century, the focus shifted to social constructivism, which viewed consensus as constantly constructed and reconstructed through social interaction and communication [5]. Further developments in the mid and late 20th centuries led to the discussions on the methodologies of forming consensus [6]. The exploration of consensus from a social sciences perspective has always been developing and changing, continually influenced by the evolving structures of societal networks.

With the rise of computer science in the 1940s, the research of consensus has spanned widely from social science to technology. The concept of consensus soon began to be explored within distributed networks due to the need for distributed computing nodes to reach a unified understanding of the state, sequence, or outcome of certain operations [7]. Since the 1960s, aerospace control systems have used replicated processors for error detection, posing a challenge in achieving consistent decisions across processes and initiating the research of Distributed Fault-Tolerance (DFT) consensus [8]. This was significantly advanced by Leslie Lamport’s 1982 “Byzantine Generals Problem," which established a framework for achieving consensus in systems despite the presence of malicious nodes [9]. By 2000, Internet companies began to use distributed servers, which required advanced consensus algorithms to synchronize data across databases, greatly advancing consensus technology. The emergence of Bitcoin in 2008 triggered a decade of intense blockchain development. As its core technology, the consensus algorithm has made great progress and broadened the research scope in the field of consensus technology. Additionally, multi-agent systems (MAS) represents

another well-known form of consensus, focusing on coordinated behaviour and decision-making among autonomous agents. Over time, the integration of artificial intelligence (AI) has largely driven the development of MAS, enabling these systems to handle more complex tasks and optimize performance [10].

The forms of consensus vary widely, depending on the participants, methods, objects, etc. For a long time, experts in various fields have carried out consensus research in their respective professional fields. In the social sciences, consensus typically involves subjective reasoning, cognitive negotiation, and value alignment among individuals [11]. In contrast, engineering approaches focus on maintaining consistency across distributed systems through algorithmic protocols under assumptions of faults and delays [12]. While these field-specific discussions on consensus effectively address the needs of their respective areas, the lack of an overall understanding of consensus may fail to meet the continuously emerging new consensus requirements driven by societal and technological progress. For example, new challenges such as advanced nodes and dynamic consensus networks, have been introduced to distributed autonomous systems when vehicle-to-vehicle networks are implemented, complicating the consensus process [13] [14]. Traditional paradigms such as DFT consensus provide a foundational framework for many consensus networks involving advanced nodes [13]. These models were originally designed for systems composed of nodes performing basic operations like reading, writing, and execution. As intelligent systems become increasingly complex and adaptive, there is growing interest in exploring alternative consensus approaches. In particular, methods inspired by human consensus paradigms may offer complementary perspectives for understanding and facilitating consensus among highly autonomous and cognitively capable nodes [15].

A typical example of a cross-field consensus need can be seen in the recent guidance provided by Industry 5.0 [16] [17] and Society 5.0 [18], which emphasize enhanced human-machine collaboration rather than pursuing solely machine-based systems with advanced features [19]. A paradigm of hybrid consensus that integrates human and machine consensus is a promising trend in manufacturing and also in everyday societal interactions [19] [20]. This complex interaction requires not only the algorithmic design of machine operation in consensus but also a grasp of the psychological and sociological factors that influence human behaviour in consensus [20].

As a broad concept, consensus holds the same core characteristic, i.e., a network of participants developing an apparent consistency or agreement. Commonalities and key elements are evident, even though consensus research spans different disciplines. Integrating discussions from various consensus disciplines can lead to several significant benefits. First, different consensus research can learn from each other. For example, some typical MAS consen-

sus algorithms draw inspiration from consensus mechanisms observed in animal behaviour [10]. Second, by clarifying and understanding the features of the core elements of consensus, researchers can better estimate the limits and applicability of various consensus strategies. For instance, the fault-tolerance threshold, which is well-discussed in DFT consensus, may also be applicable to general consensus research by following the quorum intersection rules. Third, adopting a comprehensive perspective on consensus helps develop consensus strategies that are flexible and not confined to specific disciplines, enabling them to address new consensus challenges across various fields.

Therefore, this thesis presents a study of consensus mechanisms from multiple perspectives. Rather than treating consensus as a problem confined to either human or machine domains, this work explores it as a general process of agreement formation, applicable across decentralized, intelligent systems. The thesis conducts in-depth investigations in three interconnected directions: (1) probabilistic modelling and analysis of fault-tolerant consensus systems under Byzantine node behaviour, (2) qualitative and quantitative evaluation of decentralized governance through DAO voting mechanisms, and (3) proposal of a conceptual framework for understanding consensus processes across human, machine, and human-machine hybrid systems. These three threads collectively aim to deepen our understanding of consensus, not only as an algorithmic protocol but also as a structural and behavioural phenomenon.

1.1 Motivation and Challenge

As distributed systems evolve toward greater autonomy, intelligence, and increasingly human-machine collaboration, the nature of consensus mechanisms in these systems is undergoing a fundamental paradigm shift. Traditional consensus mechanisms, typically designed for homogeneous and predictable environments, are now faced with complex scenarios. These scenarios include not only coordination between traditional computing nodes, but also coordination between advanced intelligent agents with heterogeneous capabilities, varying reliability, and dynamic behaviours. From the adaptive protocols required for autonomous robots in vehicle-to-vehicle networks and uncertain cyber-physical systems to the nuanced collective decision-making in blockchain-based governance systems such as DAOs, where consensus is generated by intelligent agents or humans, the requirements for consensus are unprecedented. The two broad paradigms of robust machine coordination with basic functions and dynamic advanced agent decision-making represent distinct but equally important frontiers in modern distributed consensus. Achieving consensus effectively in such a multifaceted and evolving environment requires a comprehensive understanding of how consensus is defined, evalu-

ated, and ultimately achieved in systems where participants can fail probabilistically, behave strategically, or participate voluntarily. To address these emerging challenges, this paper is explicitly driven by three core research goals, each of which aims to address the challenges that consensus faces under different new paradigms.

1) Advancing Consensus under Probabilistic Fault Conditions

The first motivation arises from the critical need to extend DFT consensus to more realistic and uncertain settings where node behaviour is inherently probabilistic. Traditional DFT consensus algorithms typically assume a deterministic fault model, such as requiring fewer than one-third of the nodes to be faulty. While effective in static and predictable environments, this rigid assumption fails to accurately reflect real-world systems, where node reliability varies due to dynamic communication conditions, heterogeneous hardware, external interference, or even sophisticated cyber threats. For instance, in an autonomous driving fleet, vehicles may differ substantially in their computational power, sensor precision, and connectivity quality, making a uniform fault threshold either unnecessarily conservative for high-capability nodes or insufficiently protective when weaker nodes degrade under stress. This scenario illustrates how deterministic models result in conservative over-design or, conversely, a lack of robustness under realistic heterogeneity and uncertainty. A probabilistic perspective addresses these shortcomings by treating node reliability as a stochastic variable, thereby capturing heterogeneity across nodes and enabling consensus criteria that adapt to context-specific reliability conditions. This more flexible and engineering-oriented view not only aligns with how real distributed systems are designed and operated, but also provides a principled foundation for developing consensus mechanisms that can balance safety and efficiency under genuine uncertainty.

Recent research has indeed begun to model node reliability as a probabilistic parameter, recognizing the limitations of deterministic assumptions. However, a significant gap remains: most existing probabilistic approaches still primarily focus on system performance metrics while overlooking the profound implications of probabilistic node behaviour on the core guarantees and operational dynamics of the consensus mechanisms themselves. They often continue to rely on fixed quorum rules derived from deterministic models, thus failing to fully capitalize on the flexibility and accuracy offered by probabilistic assumptions.

To address this fundamental limitation, this thesis is motivated to provide a comprehensive theoretical framework that systematically analyses the full spectrum of consensus outcomes under stochastic failure scenarios. In this framework, consensus outcomes are

precisely categorized into three states, safe[1], risky, and compromised, each defined by the likelihood of achieving consistency and correctness given probabilistic node behaviour. Furthermore, the thesis introduces the novel concept of a reliability quorum, which replaces traditional fixed-size quorums with probability-driven thresholds. This allows for consensus protocols to be optimally designed and adaptively configured based on specific, application-driven reliability requirements. Through rigorous mathematical derivation and analysis, this work enables the design of consensus mechanisms that are both robust and adaptive in the presence of real-world uncertainty, marking a crucial step towards more practical and reliable distributed systems.

2) Analysing Human-Centric Consensus in DAOs: From Conceptual Frameworks to Quantitative Performance

The second motivation for this thesis focuses on the critical domain of human-centric consensus, specifically within the rapidly evolving landscape of Decentralized Autonomous Organizations (DAOs). As DAOs gain significant traction as a novel form of blockchain-based governance within the Web3 ecosystem, on-chain voting has emerged as the central mechanism for collective decision-making. However, unlike machine consensus, which is strictly governed by pre-defined algorithmic rules, DAO consensus critically depends on voluntary human participation, complex social dynamics, and often an unequal distribution of underlying token-based voting power. These characteristics introduce significant uncertainty and make it profoundly difficult to objectively evaluate whether a voting process is truly decentralized, fair, or effective. In practice, these limitations have already led to tangible governance challenges: in several high-profile DAOs, voting outcomes have been disproportionately influenced by a small number of large token holders, while low turnout rates have further exacerbated the problem, resulting in decisions that, although procedurally valid, were widely regarded as unrepresentative of the broader community. Such distortions risk undermining trust in governance processes and, in extreme cases, have triggered disputes and crises that threaten the stability of the DAO itself.

Addressing these complexities necessitates a multi-faceted approach, moving beyond anecdotal observations to systematic analysis. Initially, this thesis is motivated by the need to develop a foundational conceptual understanding of DAO voting mechanisms. Given the nascent stage and inherent complexity of DAOs, a structured qualitative analysis is essential to identify the critical role of voting, categorize its varying decentralization levels, and

[1]The notion of "safe state" in this thesis is distinct from the classical safety property in distributed consensus, which denotes agreement consistency. Here it represents a probabilistic classification of consensus outcomes.

define key performance dimensions. Existing studies often lack such systematic conceptual frameworks. This work therefore establishes a comprehensive conceptual model (e.g., the DAO governance triangle and a five-tier decentralization scheme) and proposes an evaluation framework to thoroughly analyse the characteristics and challenges of diverse DAO voting mechanisms.

Building upon this conceptual foundation, the thesis is further motivated to develop rigorous quantitative methods to assess the decentralization and efficiency performance of human-driven consensus in DAOs. A fundamental challenge arises from a stark contradiction: while DAOs are built upon a core philosophy of decentralization, their existing voting mechanisms frequently exhibit a strong tendency toward centralized control, often due to concentrated token ownership or low voter participation. Despite the growing popularity and widespread adoption of DAO voting systems, there is a critical absence of formal metrics or universally accepted standards to quantitatively assess their true decentralization or comprehensively evaluate their governance efficiency. Current evaluations largely rely on qualitative assessments, informal reasoning, or anecdotal evidence, which inherently lack reproducibility, precision, and comparability across different DAOs.

To address this pressing quantitative gap, this thesis is strongly motivated to establish a robust and reproducible framework for analysing DAO consensus. This involves constructing the first stochastic process model for DAO voting that accurately captures both individual voting preferences and probabilistic participation, enabling a deeper understanding of real-world governance dynamics. Based on this novel model, new, formal metrics are proposed to quantify decentralization, including the consistency rate (reflecting individual voter influence) and the decentralization coefficient (providing a comprehensive, system-wide measure of decentralization). Concurrently, this work is motivated to quantitatively describe and analyse critical metrics related to DAO voting efficiency, including participation probability, voting period duration, and approval rate, detailing their interrelationships within the developed model. These analytical tools collectively enable a more objective and robust evaluation of diverse DAO voting mechanisms, directly supporting the design of governance protocols that truly align with decentralized ideals and operate with optimal efficiency, thereby bringing much-needed clarity, measurability, and analytical rigour to the study of voting-based consensus in human-centric distributed systems.

3) Establishing a Unified Framework for Human and Machine Consensus

The third motivation originates from the profound and growing need to understand consensus processes that seamlessly involve both human and machine participants. In many modern

and emerging systems, such as human-robot collaboration in industrial settings, autonomous transportation networks, smart grid management, and cyber-physical social networks, consensus increasingly emerges through complex interactions between agents with fundamentally different capabilities, behaviours, and underlying logic. However, existing research predominantly tends to analyse human consensus and machine consensus separately, employing distinct theoretical models, terminologies, and analytical approaches that are inherently difficult to compare, contrast, or integrate. This disciplinary fragmentation creates a significant analytical gap, hindering the development of truly comprehensive and interoperable consensus solutions for hybrid systems.

For instance, the vision of Society 5.0 highlights human-centric production and services, where intelligent systems must coordinate with human decision-makers in domains such as healthcare, manufacturing, and urban management. Likewise, in applications such as earthquake search and rescue, autonomous robots can rapidly collect and process environmental data, but final decisions on prioritisation and ethical trade-offs must be reached jointly with human coordinators. These examples illustrate the need for hybrid consensus mechanisms that align machine efficiency with human judgment. These scenarios illustrate the importance of developing a unified framework that abstracts beyond individual domains, so that both routine human–machine integration and emergency responses can be analysed under a consistent structure.

This thesis is thus strongly motivated to respond to this critical challenge by proposing a novel and broadly applicable conceptual framework for analysing consensus across these disparate types of systems. Rather than focusing on granular implementation details of specific algorithms or social mechanisms, the framework abstracts and identifies three essential, indispensable elements common to all consensus processes: the participants (as the entities reaching agreement), the communication structures (as the channels through which information flows), and the evolution of shared state (describing the transformation from divergence to agreement). These abstract elements provide a unified vocabulary, a consistent analytical structure, and a coherent lens through which diverse consensus scenarios, from human deliberation to machine coordination, can be systematically analysed and understood.

Furthermore, within this unified framework, the thesis redefines the consensus process fundamentally as a mechanism to eliminate or reduce cognitive differences among participants regarding a shared object, conceptualizing it as an entropy-reduction process. This profound understanding guides the framework's utility, enabling it to classify consensus into three distinct categories, human consensus, machine consensus, and hybrid consensus. This comprehensive classification facilitates meaningful comparative studies and provides invaluable guidance for designing robust, adaptive, and ethically aligned consensus strategies that

are appropriate for the unique demands of various future intelligent, distributed, and cooperative systems. The framework's ability to bridge analytical divides is expected to inspire future cross-disciplinary consensus research and foster more effective mechanism design.

1.2 Original Contribution

This thesis investigates key challenges in distributed consensus through three complementary perspectives: **a rigorous probabilistic analysis of consensus outcomes and a novel reliability quorum model for DFT consensus protocols**, **a comprehensive framework for consensus in DAO governance that combines qualitative analysis with novel quantitative methods, including models for understanding voting mechanisms, a foundational stochastic analysis, and metrics for assessing decentralization and efficiency**, and **a conceptual consensus framework for analysing consensus across human and machine systems**. These contributions span theoretical abstraction, protocol-level modelling, and empirical evaluation, with a focus on applications in autonomous systems and decentralized organizations such as DAOs. The original contributions of this thesis are summarized as follows:

- **A Probabilistic Model for DFT Consensus and Uncertainty Analysis of Consensus Outcomes:**
  - The first comprehensive theoretical analysis of consensus outcomes in a probabilistic model for fault-tolerant consensus systems is presented, explicitly accounting for probabilistic node failures. By mapping these outcomes to every possible configuration of faulty and non-faulty nodes, together with key non-deterministic parameters, a robust theoretical model is developed to evaluate outcome probabilities under varying Byzantine behaviour and reliability-quorum assumptions. The resulting mathematical derivations provide insights into optimising quorum size and node reliability for stringent system-level requirements.
  - A novel classification of consensus outcomes into three states, *safe*, *risky*, and *compromised* is introduced. These states are defined by the interaction between non-deterministic counts of faulty and non-faulty nodes and the specified quorum threshold. Computing the probability of each state enables rapid estimation of consensus outcomes and direct assessment of overall system reliability.
  - The concept of a *reliability quorum*, defined as the minimal subset of nodes required to achieve a target reliability level, is formulated. A weighting-based selection strategy permits flexible prioritisation of consensus outcomes to meet diverse

application demands. Analytical results further show that the reliability quorum maximising the safe-state probability coincides with the traditional $2f_{\max}+1$ rule, confirming its validity under probabilistic Byzantine fault assumptions.

  - Wireless Distributed Consensus (WDC) is proposed to evaluate the reliability of consensus systems in wireless environments. Comprehensive models of WDC based on PBFT, incorporating essential synchronization processes, are derived. Specifically, analytical expressions are developed to quantify the reliability of PBFT under conditions of node failure, link failure, and combined node-link failures.

- **Comprehensive Quantitative Framework for Decentralized Collective Decision-Making in DAOs:**

  - The *DAO governance triangle* is proposed to precisely locate the pivotal role of the voting mechanism within Decentralized Autonomous Organizations. Within this model, the mutual constraints and interdependencies between DAO voting mechanisms and underlying smart contracts are discussed in comprehensive detail.
  - Key performance metrics for DAO voting mechanisms are abstracted and formalized, encompassing Security, Efficiency, Effectiveness, and Decentralization (SEED). This framework enables systematic and holistic evaluation of performance and robustness across diverse DAO voting mechanisms.
  - Based on the SEED framework, a detailed analysis of seven typical DAO voting mechanisms is presented, including summaries of their operational procedures and performance evaluations aligned with the SEED principles, thereby completing a qualitative and framework-level assessment.
  - A novel stochastic model for the DAO voting process is constructed, accurately capturing a wide range of voting schemes and approval conditions. This model provides a foundational analytical framework for the quantitative evaluation of DAO voting mechanisms.
  - Within the stochastic voting model, the concept of *Consistency Rate* is defined to quantitatively represent each voter's effective control over the voting process, establishing a fundamental parameter for assessing the overall level of decentralization in DAO voting systems.
  - Building on the Consistency Rate, a new metric called the *Decentralization Coefficient* is introduced as the first quantitative measure to evaluate decentralization

in DAO voting. This coefficient enables robust demonstration and comparison of decentralization performance across entire voting systems.

  - Three critical metrics related to DAO voting efficiency are quantitatively described: participation probability, voting period duration, and approval rate. Their interrelationships within the developed DAO voting model are also detailed.
  - Extensive simulations to analyse the impact of key factors (voting power distribution, participation rate, and the voting process) on decentralization performance through the Decentralization Coefficient for representative voting cases are conducted. Additionally, the efficiency performance of typical voting mechanisms is evaluated by providing simulations for participation probability, voting period duration, and approval rate in representative voting scenarios.

- **Rigorous Validation and Empirical Analysis:**
  - All proposed theoretical models, analytical methods, and new metrics throughout this thesis are rigorously validated. This includes extensive analytical derivations, comprehensive simulations for probabilistic outcome estimations, reliability quorum calculations, and detailed evaluations of the decentralization and efficiency performance of representative DAO voting mechanisms in various scenarios.

- **A Framework for Integrating Consensus Approaches Across Human and Machine Systems:**
  - A novel unified framework is presented to describe consensus across human, machine, and human-machine hybrid systems. This framework identifies three core, indispensable components: participants as the carriers of consensus, communication as the bridge, and state descriptions marking the transformation from chaos to consistent cognition. This abstraction enables consistent analysis of consensus dynamics across diverse domains.
  - The distinct characteristics of human consensus (HC) and machine consensus (MC) are extracted and contrasted. Building on this comparison, a new form of consensus arising from human-machine collaboration is introduced, termed *Hybrid Consensus* (HBC). This interdisciplinary perspective offers a comprehensive understanding of the nature of agreement, transcending traditional disciplinary boundaries.
  - Based on the unified consensus framework, a refined definition of the consensus process is formulated. Fundamentally, this process aims to eliminate or reduce

cognitive differences among participants regarding the consensus object, conceptualizing consensus as an entropy-reduction process. This understanding can inform the design of more effective consensus mechanisms.

- Examples of consensus mechanism design strategies are demonstrated, illustrating their development in alignment with the proposed framework and the fundamental goals of the consensus process.

1.3 Thesis Outline

This thesis consists of six chapters. Chapter 1 introduces the main research problems and motivations in the field of distributed consensus. It also summarizes the key contributions of the work.

Chapter 2 reviews the foundations of distributed fault-tolerant consensus, with a focus on recent developments in probabilistic modelling. It also provides an in-depth examination of the DAO ecosystem, particularly its governance mechanisms and associated challenges.

Building on this foundation, Chapter 3 presents a comprehensive theoretical analysis of consensus outcomes in probabilistic models for fault-tolerant systems. It introduces new classifications of consensus states, including safe, risky, and compromised, and proposes a theoretical framework for estimating their probabilities. In addition, a reliability quorum model is developed to improve system robustness under uncertainty.

Chapter 4 proposes a multi-dimensional framework for analysing decentralized collective decision-making. It begins with qualitative analysis and conceptual modelling, including the DAO governance triangle and a five-level decentralization structure. This is followed by the introduction of a novel stochastic model for DAO voting, which forms the basis for defining quantitative performance metrics and evaluating decentralization and efficiency.

Chapter 5 establishes a unified framework for analysing consensus in both human and machine systems. It is structured around three core components: participants, communication, and state. The chapter examines the distinctive features of human and machine consensus, and interprets the consensus process as a mechanism for reducing uncertainty, offering guidance for the design of future consensus protocols.

Chapter 6 concludes the thesis by summarizing the main findings and original contributions. It also discusses broader implications, addresses limitations, and suggests potential directions for future research in the evolving domain of consensus studies.

Chapter 2

Literature Review

In this chapter, a literature review of the two types of consensus that form the core focus of this thesis is presented. The first type is DFT consensus in the field of computer science and engineering. This line of research aims to enable a network of distributed nodes to reach agreement on a shared state or decision, even in the presence of potential failures or asynchronous communication. The second type is voting-based consensus among human agents, particularly within DAOs. It is rooted in social science theories of collective decision-making, such as voting and governance structures, but realized through computational mechanisms including smart contracts and distributed ledgers. This form of consensus emphasizes how collective decisions are reached through structured procedures while maintaining decentralized governance.

2.1 Distributed Consensus

With the rise of computer science in the 1940s, the concept of consensus began to attract attention in distributed networks, driven by the need for computing nodes to reach a unified understanding of system states, operation sequences, or computational outcomes [7]. In the 1960s, aerospace control systems began employing replicated processors for fault detection, which posed significant challenges for achieving consistent decisions across processes and catalysed early research into DFT [8]. A landmark development came in 1982 with Leslie Lamport's formulation of the "Byzantine Generals Problem," which established a formal model for consensus under adversarial conditions, allowing for the presence of arbitrary or malicious faults [9]. By the early 2000s, large-scale Internet services adopted distributed servers, requiring advanced consensus protocols to synchronize distributed databases and logs [21]. This trend greatly accelerated consensus research and deployment. In 2008, the introduction of Bitcoin initiated a decade of intensive development in blockchain systems,

where consensus algorithms became foundational for ensuring consistency and security in decentralized environments [22, 23].

2.1.1 Fundamental Concepts

The distributed consensus (DC) problem addresses the core challenge of how multiple autonomous computing nodes can reach agreement on a single, consistent value despite the presence of faults and unpredictable communication delays. Consensus is essential for ensuring data consistency, fault tolerance, and coordination in distributed environments, ranging from replicated databases to distributed ledgers and control systems. Formally, a consensus protocol aims to satisfy the following three properties.

- **Safety (Agreement)**: No two correct processes decide differently. Once a value is decided, it cannot be contradicted later. Safety is typically enforced through quorum intersection. For crash faults, a simple majority suffices, while for Byzantine faults at least $2f+1$ out of $n=3f+1$ replicas are required. Deterministic state machine execution and well-defined commit rules also contribute to preventing conflicting certificates.
- **Liveness (Termination)**: Every correct process must eventually reach a decision, assuming the timing model provides sufficient guarantees. In a fully asynchronous system, the FLP result shows that deterministic consensus cannot achieve liveness in the presence of faults. To make progress in practice, protocols adopt partial synchrony, where mechanisms such as leader election, view change, and bounded timeouts ensure termination once the system stabilizes.
- **Validity**: If all correct nodes propose the same initial value v, then v must be the value decided by all correct nodes. This property ensures that the decided value is meaningful and related to the inputs provided by the participants, preventing arbitrary or irrelevant decisions.

These properties collectively define the necessary conditions for robust coordination in distributed environments, underpinning the reliability of critical applications such as replicated state machines, blockchain systems, and fault-tolerant control mechanisms.

2.1.2 Distributed System Assumptions

The design and analysis of distributed consensus protocols are deeply rooted in the assumptions made about the underlying system environment, formally captured by the system model. These assumptions define the characteristics of communication, timing, and node behaviour,

Table 2.1: System models and their implications for distributed consensus

Model Type	Category	Implications for Consensus
Timing Model	Synchronous	Predictable execution; simplifies liveness, allows higher fault tolerance
	Asynchronous	Unpredictable; FLP impossibility applies (safety and liveness cannot coexist deterministically)
	Partially Synchronous	Balanced trade-off; enables liveness and safety in practice (e.g., PBFT, Raft)
Fault Model	Crash Fault	Fail-stop model; easiest to tolerate (e.g., $f < N/2$)
	Byzantine Fault	Unpredictable and deceptive; hardest to tolerate (e.g., $f < N/3$), requires stronger protocols

and they directly shape the feasibility, robustness, and efficiency of consensus algorithms. A typical distributed system consists of n interconnected nodes that coordinate through message passing and may operate under various timing constraints and fault conditions.

Timing Assumptions

Timing assumptions are traditionally categorized into three types. In the synchronous model, both message delays and process execution speeds are known and bounded [24]. This model significantly simplifies consensus protocol design and allows for stronger guarantees, such as tolerating up to $f < n/2$ faulty nodes [25]. However, it is often considered unrealistic in wide-area networks where such rigid bounds cannot be enforced. At the other extreme, the asynchronous model assumes no bounds on communication delays or processing times [26]. While more representative of real-world systems, it introduces significant theoretical limitations. The well-known FLP impossibility result [26] (explained in Sec. 2.1.4) shows that no deterministic consensus protocol can guarantee both safety and liveness in a fully asynchronous environment with even a single fault. To reconcile these extremes, the partially synchronous model was proposed by Dwork, Lynch, and Stockmeyer [27]. It assumes the system behaves asynchronously for an unbounded initial period but eventually becomes synchronous after an unknown global stabilization time (GST). This model strikes a practical balance, enabling progress under realistic assumptions and forming the basis of many widely used protocols such as PBFT and Raft.

Failure Assumptions

Alongside timing, assumptions about node failures are equally important. The crash fault model assumes that nodes may abruptly halt and cease participation, but they do not act mali-

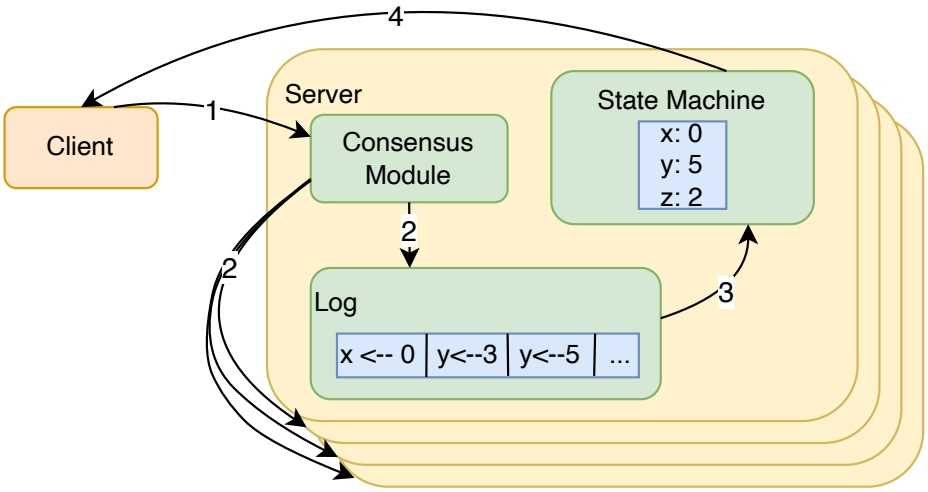

Figure 2.1: The SMR workflow [1]

ciously [28]. It is the simplest fault model and is commonly addressed by Crash Fault Tolerant (CFT) protocols, which can often tolerate up to $f < n/2$ faulty nodes [25, 26]. In contrast, the Byzantine fault model is significantly more challenging, allowing nodes to behave arbitrarily or maliciously [9]. Such nodes may lie, collude, or act inconsistently, thus requiring stronger assumptions and more resilient mechanisms to maintain safety. The Byzantine Generals Problem formalized this model [9], and protocols that address it, known as Byzantine Fault Tolerant (BFT) protocols, typically operate under stricter bounds, such as $f < n/3$ [25, 29].

These foundational assumptions determine the theoretical limits and practical trade-offs of consensus protocols. Table 2.1 summarizes the key characteristics of timing and fault models and their implications for consensus design.

2.1.3 State Machine Replication (SMR)

State Machine Replication (SMR) is a foundational and widely adopted technique for constructing fault-tolerant distributed services by replicating a deterministic service across multiple nodes [28]. The core principle of SMR is that if multiple replicas of a service are identical and execute the same sequence of operations in the same order, they will maintain identical states. This allows the system to continue operating correctly even if a subset of replicas fail.

The efficacy of SMR hinges on two critical requirements. First, the state machine itself must be deterministic. This means given the same initial state and input, it must always produce the same output and transition to the same next state. This determinism ensures that all replicas, when processing the same sequence of commands, will evolve consistently. Second,

all replicas must receive and apply client requests in a consistent global order. Achieving this total order broadcast is precisely the role of distributed consensus protocols, which coordinate agreement on the sequence of operations to be executed across all participating nodes [30]. A visual overview of the SMR workflow is illustrated in Fig. 2.1. The diagram shows a client sending a request to one of the replicated servers. This request is proposed to the consensus module, which establishes a total order of operations (log entries). Once the order is agreed upon, each replica applies the operations to its local deterministic state machine, ensuring consistent state across all nodes.

Consensus protocols serve as the underlying mechanism to establish and agree upon a total order of commands among the replicas. In common leader-based SMR implementations (such as Raft [31] and PBFT [29]), a designated leader node proposes client commands. Follower nodes then agree upon this proposed order through the consensus protocol. Once a command is committed (i.e., consensus is reached on its position in the log), all replicas apply the command to their local state machine. This ensures that even if the leader fails, a new leader can be elected, and the system can continue processing commands consistently.

2.1.4 FLP Impossibility Result

The FLP impossibility result, published by Fischer, Lynch, and Paterson in 1985 [26], represents one of the most profound and influential theoretical findings in distributed computing. This result formally demonstrated that, in a fully asynchronous distributed system where even a single process may fail by crashing, no deterministic consensus algorithm can simultaneously guarantee both *safety* (Agreement property) and *liveness* (Termination property).

The core intuition behind the FLP result lies in the inherent inability of processes in an asynchronous system to distinguish between a very slow message or process and a truly crashed message or process. In an asynchronous environment, there are no timeouts or bounds that can reliably determine if a message has been lost or if a process has simply halted. This fundamental uncertainty prevents any deterministic protocol from making progress (liveness) while simultaneously ensuring that all correct nodes decide on the same value (safety). If a protocol tries to guarantee liveness, it runs the risk of divergence. If it tries to guarantee safety, it runs the risk of being stuck indefinitely.

The FLP impossibility result has had profound implications for the design of practical distributed systems, forcing researchers and engineers to adopt strategies to circumvent its limitations. Here are the primary approaches. Relaxing Timing Assumptions is the most common approach is to move from a purely asynchronous model to a partially synchronous model [27]. Most practical strong-consistency consensus algorithms, including PBFT [29]

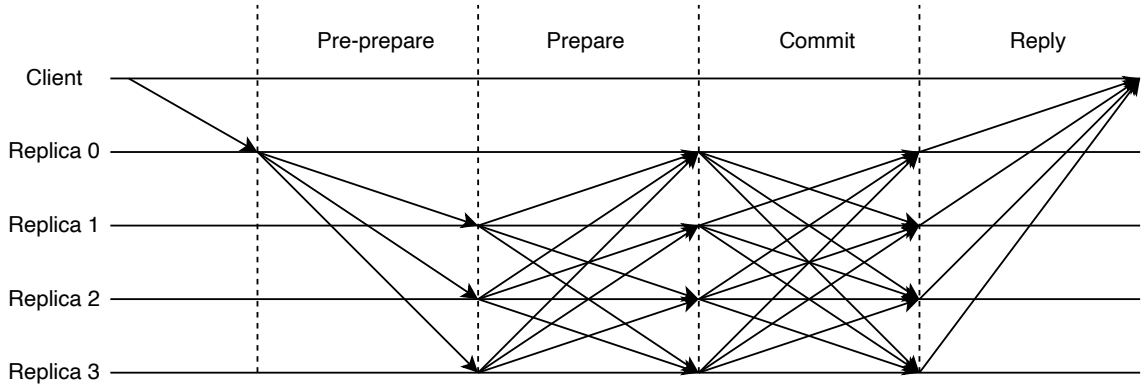

Figure 2.2: PBFT normal operation

and Raft [31], rely on this assumption. An alternative approach is to introduce randomization into the consensus algorithm. Randomized protocols can achieve consensus in fully asynchronous systems with crash failures by using probabilistic techniques to break symmetry and ensure termination with probability 1 [32]. However, these often come with higher complexity or a non-deterministic termination time. Another method is weakening safety or liveness guarantees. For instance, allowing for eventual consistency (weakening safety temporarily) is a common pattern in large-scale distributed databases where availability is prioritized over immediate consistency [33, 34]. Alternatively, systems might accept periods of unavailability (weakening liveness) during network partitions or failures to preserve strong consistency [35]. However, for strong consensus aiming for immediate agreement, compromising these properties is generally not the goal.

2.1.5 Representative Distributed Consensus Mechanism

Based on the failure model, consensus mechanisms can generally be classified into two categories: BFT algorithms [36], which tolerate arbitrary (including malicious) faults, and CFT algorithms, which only consider fail-stop behaviors [37]. In the following, two representative mechanisms are presented: one classical BFT algorithm, Practical Byzantine Fault Tolerance (PBFT), and one widely adopted CFT algorithm, Raft.

Practical Byzantine Fault Tolerance (PBFT)

Practical Byzantine Fault Tolerance (PBFT), introduced by Castro and Liskov in 1999 [29], marked a pivotal advancement in consensus protocol design by making Byzantine fault tolerance feasible in asynchronous and adversarial distributed environments. Earlier Byzantine agreement protocols often suffered from prohibitive message complexity or relied on unrealistic assumptions of network synchrony. In contrast, PBFT provided a practically deployable

solution with polynomial communication complexity, enabling robust consensus even when up to $1/3$ of the participating replicas exhibit arbitrary or malicious behaviour.

PBFT operates through a three-phase protocol: pre-prepare, prepare, and commit. In the pre-prepare phase, a designated primary (leader) node receives a client request and proposes it to the other replicas by broadcasting a pre-prepare message, which includes a sequence number and a cryptographic digest of the request. In the prepare phase, each replica echoes the proposal to all other replicas via a prepare message. A replica considers the request prepared once it has received matching pre-prepare and $2f$ prepare messages (where f is the maximum number of faulty replicas). This ensures that a quorum of $2f+1$ replicas has seen the same request at the same sequence number. Finally, in the commit phase, replicas exchange commit messages, and upon receiving $2f+1$ matching commits, each replica safely executes the request. This final phase guarantees that all non-faulty replicas will eventually execute the same request in the same order, even in the presence of Byzantine faults.

PBFT emphasizes both safety, ensuring that non-faulty nodes agree on the same order of operations, and liveness, guaranteeing that the system eventually makes progress under the assumption of eventual message delivery. One of the major contributions of PBFT is its practical performance, achieving consensus with $O(n^2)$ communication complexity where n is the number of replicas [7]. This made PBFT a foundational model for the design of many subsequent Byzantine fault-tolerant systems, including permissioned blockchain platforms such as Hyperledger Fabric [38] and Tendermint [39].

Despite its strengths, PBFT also has several notable limitations. The quadratic communication cost becomes a bottleneck as system size increases, limiting its scalability to small or moderate-sized networks [40] [41]. Moreover, PBFT assumes static membership, requiring a fixed set of known participants, which makes it unsuitable for open or permissionless environments [42]. Additionally, while PBFT guarantees liveness under partial synchrony, it can stall under prolonged network partitions or under active denial-of-service attacks targeting leader nodes [7]. Recent research has proposed various improvements and extensions to PBFT, including techniques such as optimistic execution [43] [44], and integration with trusted execution environments to reduce communication overhead [45]. Nonetheless, PBFT remains a critical foundation in the study of Byzantine fault-tolerant consensus, and continues to influence both theoretical advances and practical system designs in distributed computing.

Raft

Raft is a consensus algorithm introduced by Ongaro and Ousterhout in 2014 [31], designed to manage replicated logs in distributed systems. Unlike BFT protocols such as PBFT, Raft assumes a CFT model, where nodes may fail by stopping but not by behaving arbitrarily or

maliciously. Its primary goal is to offer both safety (agreement on a consistent log order) and liveness (continued progress in the presence of failures) under partial synchrony.

Raft divides the consensus process into three key components: leader election, log replication, and safety guarantees. In each term, one node is elected as the leader through a randomized timeout and voting process. Once elected, the leader handles all client requests, appending them to its local log and replicating them to follower nodes. A log entry is considered committed once it is stored on a majority of nodes and the leader has appended it to its own log. This approach simplifies the reasoning about correctness by centralizing control, while still ensuring fault tolerance.

The leader election process is illustrated in Fig. 2.3, where followers independently initiate an election upon timeout, request votes from peers, and a new leader is elected after receiving a majority of votes. Once elected, the leader maintains its authority by periodically sending heartbeat messages (empty `AppendEntries` RPCs) to followers to prevent them from starting new elections.

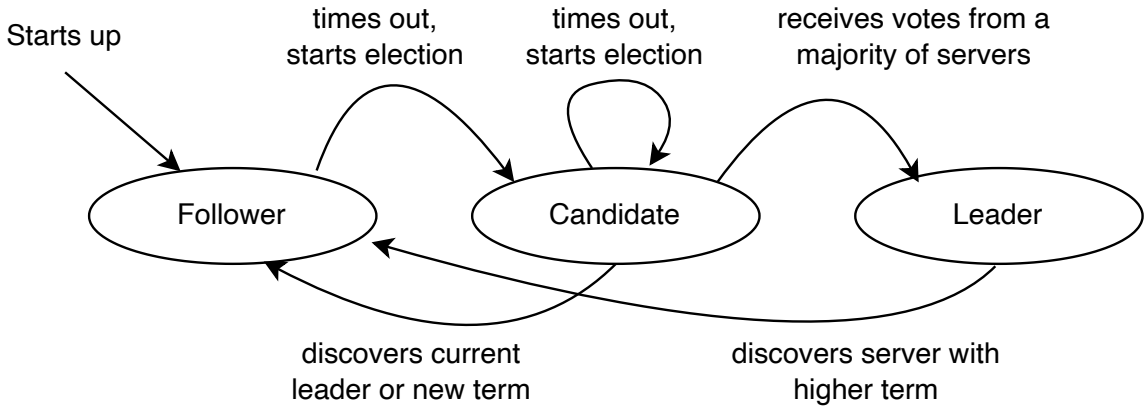

Figure 2.3: Raft leader election process [2]

The log replication process, illustrated in Fig. 2.4, begins when the elected leader receives a new client command and appends it as a log entry to its local log. The leader then issues `AppendEntries` RPCs to all followers, which contain the new entries along with metadata specifying the index and term of the entry preceding the new entries (for consistency checks). Upon receiving the `AppendEntries` request, a follower will verify that its log matches the leader's at the specified previous index and term. If the consistency check passes, it appends the new entries to its own log and acknowledges the leader. Once the leader receives successful acknowledgments from a majority of the servers (including itself), it safely advances its commit index to include the newly replicated entries. These committed entries are then applied to the replicated state machines of the leader and followers. Heartbeat messages are also used to carry commit information and maintain log synchronization even when no new

entries are being proposed.

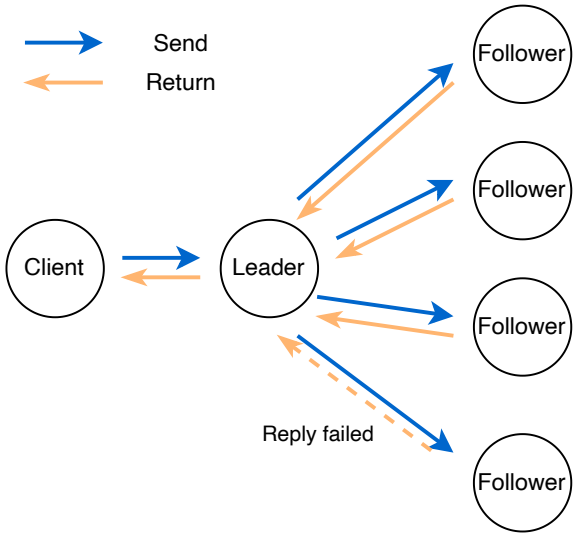

Figure 2.4: Raft log replication

Raft's design emphasizes understandability and modularity. Each component (e.g., elections, log replication) is designed to be separable and independently understandable. This has made Raft especially popular in industry and education, and it is widely used in production systems such as etcd [46] and RethinkDB [47].

2.1.6 Consensus Mechanisms in the Era of Blockchain

The emergence of blockchain technology, beginning with Bitcoin in 2008 [48], ushered in a groundbreaking era for distributed consensus. This innovation fundamentally redirected the focus from traditional centralized or permissioned settings, where participants are known and often trusted, to open, decentralized, and potentially untrustworthy environments [49, 50]. Unlike classic consensus protocols like PBFT, designed for a closed group of known participants [29], blockchain consensus mechanisms strive to achieve agreement among a vast and anonymous network of nodes [22]. This profound shift necessitated novel approaches to overcome challenges like Sybil attacks [51, 52], ensure security without central authority [53], and incentivize broad participation through crypto-economic mechanisms [54]. In the following sections, two primary categories of blockchain consensus algorithms, Proof-based mechanisms (e.g., Proof-of-Work, Proof-of-Stake) and BFT variants adapted for blockchain contexts [36] are briefly introduced.

Proof-based Consensus

Proof-based mechanisms, exemplified by Bitcoin's Proof-of-Work (PoW), rely on participants expending computational resources (or other scarce resources) to demonstrate their eligibility to propose and validate blocks [48].

In PoW, nodes (miners) compete to solve a cryptographic puzzle. The first to find a solution proposes a new block to the network [48]. This process, often referred to as Nakamoto Consensus, provides a probabilistic guarantee of agreement and security against malicious actors, assuming a majority of computational power is controlled by honest participants [53, 55]. The "longest chain rule" ensures eventual consistency: honest nodes always extend the longest valid chain, which implicitly achieves agreement on the transaction order [48, 56]. While highly robust against Byzantine faults in open networks [57], PoW suffers from high energy consumption [58, 59] and limited transaction throughput [60]. Other proof-based mechanisms include Proof-of-Stake (PoS), where validators are chosen based on the amount of cryptocurrency they "stake" as collateral [61, 62], and Proof-of-Authority (PoA), where a limited set of trusted validators are pre-approved [63]. PoS aims to be more energy-efficient and scalable than PoW, but introduces new challenges like "nothing-at-stake" attacks [64] and stake centralization concerns [57, 65].

Table 2.2: Comparison of Representative Distributed Consensus Mechanisms

Feature	**PBFT**	**Raft**	**PoW**	**BFT-Blockchain**
System Model	Partially synchronous	Partially synchronous	Asynchronous (probabilistic)	Partially synchronous
Fault Tolerance	BFT	CFT	BFT	BFT
Max Faulty Nodes	$\lfloor (n-1)/3 \rfloor$	$\lfloor (n-1)/2 \rfloor$	>50% honest compute	$\lfloor (n-1)/3 \rfloor$
Consistency	Strong (deterministic)	Strong (deterministic)	Eventual finality	Strong finality
Communication Complexity	$O(N^2)$	$O(N)$	$O(N)$	$O(N^2)$
Scalability	Small N	Moderate N	Low throughput	Moderate to high
Leader Election	Fixed primary with view change	Randomized timeout	Mining competition	Rotating primary
Network Environment	Permissioned	Permissioned	Permissionless	Permissioned
Key Mechanism	3-phase commit, quorums	Log replication, heartbeats	Cryptographic puzzle	Multi-round voting
Energy Efficiency	High	High	Low	High
Example Systems	Hyperledger, Zyzzyva	etcd, ZooKeeper, Consul	Bitcoin, Ethereum	Tendermint, Cosmos, Polkadot

BFT-inspired Consensus in Blockchains

While classical BFT protocols like PBFT were originally designed for permissioned environments, many modern blockchain systems, particularly permissioned blockchains and some public blockchains, have adapted and evolved BFT principles to achieve stronger consistency and higher throughput than proof-based methods [66, 67]. These BFT-inspired protocols typically operate with a known, often smaller, set of validators, which allows for deterministic finality and higher performance [66, 68]. Examples include Tendermint [39], which employs a variant of BFT for block finalization, and protocols used in Hyperledger Fabric [38]. These systems often trade off the full decentralization of Nakamoto Consensus for improved performance and immediate transaction finality, making them suitable for enterprise applications [69]. Challenges include managing validator sets [70, 71], ensuring robust decentralization in permissioned settings [69], and scaling their communication complexity for large validator groups [69, 72]. The innovation brought by blockchain consensus mechanisms has significantly expanded the landscape of distributed consensus, introducing new trade-offs between decentralization, scalability, and security [23, 60, 66].

While each consensus mechanism presents distinct advantages and trade-offs, a comparative analysis is essential to understand their applicability in different distributed settings. Table 2.2 offers a detailed comparison of the representative consensus algorithms discussed in this section, delineating their performance, security, and scalability properties under varying network and fault assumptions.

2.2 Probabilistic Models for Distributed Consensus

Traditional distributed consensus (DC) protocols are based on deterministic fault tolerance assumptions, which typically impose a strict upper bound on the number of faulty nodes. For example, BFT protocols assume that no more than $1/3$ of the nodes may behave arbitrarily (maliciously or erroneously), while CFT protocols assume that no more than $1/2$ of the nodes may fail by crashing. However, in practical distributed systems such as wireless networks [13] or connected autonomous systems [14], node behaviour is often probabilistic due to hardware variability, environmental interference, and unpredictable communication conditions [73]. These factors make rigid deterministic assumptions increasingly inadequate for modelling real-world system behaviour.

In response to these challenges, a growing line of research focuses on probabilistic models for distributed consensus, which represent node reliability in probabilistic terms rather than relying on fixed fault thresholds. In such models, each node is assigned a probability

Table 2.3: Compare conventional and probabilistic model for distributed consensus

	Fault Tolerance	**Aim of Consensus**	**Typical Scenarios**
Conventional Model	System level: $N \geq 3f+1$ (BFT)	Keep quorum alive, consistency	Database
Probabilistic Model	Node level: Node reliability, $1-p_b$	Reach high consensus reliability	Autonomous systems, decentralized IIoT

N is the total number of nodes. f is the number of faulty nodes tolerated. p_b is the probability that a node behaves non-faulty.

of correct behaviour during the consensus process, and the overall system reliability is assessed based on the aggregate probability distribution across all nodes. This approach shifts the focus from achieving deterministic guarantees through strict quorum rules to optimizing consensus reliability under uncertainty. A conceptual comparison between traditional deterministic models and probabilistic models is summarized in Table 2.3.

Several notable works contribute to the development of this probabilistic modelling paradigm. [74] first introduced a probabilistic reliability framework for Raft-based consensus, estimating the overall consensus success rate from node failure probabilities. [75] proposed a dynamic fault model for PBFT using Bernoulli-distributed node behaviour to calculate the likelihood of successful agreement. [13] extended these ideas by incorporating probabilistic modelling for both nodes and communication links, analysing reliability, latency, and throughput in wireless implementations of Raft and PBFT. [76] further examined how probabilistic node failures influence consensus reliability in wireless environments. In addition, [77] relaxed the assumption of uniform node reliability by introducing heterogeneous reliability distributions, enabling a more realistic and flexible modelling framework. These studies reflect a growing recognition of the limitations of deterministic assumptions and lay the foundation for more adaptable and robust probabilistic approaches to distributed consensus in uncertain environments.

Building on this body of research, this thesis further investigates two fundamental problems in probabilistic modeling of distributed consensus in Chapter 3: (1) the analysis of non-deterministic consensus outcomes caused by probabilistic node reliability, with a focus on consistency and correctness, and (2) the design of a reliability-aware quorum selection strategy that guarantees a target level of system reliability.

2.3 DAO Governance and Typical voting mechanisms

While fault-tolerant consensus mechanisms focus on ensuring consistency and reliability among machines through deterministic coordination, human-centric decision-making processes follow fundamentally different principles. These processes often involve voluntary participation, heterogeneous preferences, and collective judgment. With the advancement of blockchain technology and the emergence of Web 3 [78] [79], DAOs have emerged as a new organizational form that embodies these human-oriented characteristics within digital environments [80]. They have attracted increasing attention across a wide range of application domains. Initially introduced in the Ethereum white paper, DAOs are blockchain-based organizations that operate through smart contracts and collective voting rather than centralized control. Since the launch of the first DAO in 2016 [81], more than 10,000 DAOs [82] have been created for purposes such as investment, philanthropy, and decentralized platform governance [83–86]. Examples include *LAO* [83] and *MetaCartel Ventures* [84] for venture capital, *Ecorise* [85] for environmental stewardship, and *Decentraland* [86] for community-managed virtual platforms. These use cases reflect the broad applicability of DAOs and their ability to replace traditional hierarchical structures with decentralized, autonomous alternatives. In contrast to traditional hierarchical institutions, DAOs are structured in a non-hierarchical manner, with ownership and authority distributed among all participants [87]. Major activities, including the creation and modification of rules as well as the management of shared funds, are collectively determined by the members [88] [89]. This decentralized model of governance provides the foundation for a range of voting mechanisms that support decision-making within DAOs.

As an emerging organizational paradigm in the Web3 era, DAOs enable diverse forms of community-driven collaboration and decision-making. To understand how these decentralized entities function and evolve, it is essential to examine both their broader ecosystem structure and the governance mechanisms that support collective action. The following sections provide a structured overview of the DAO ecosystem within Web3, focusing on the core of DAO governance, the voting mechanisms, by presenting a unified framework and several representative implementations.

2.3.1 DAO Under Web3 Infrastructure

DAOs represent a pivotal component of the Web3 ecosystem, with their operations fundamentally underpinned by a robust technical architecture. As illustrated in Figure 2.5, the overall structure of a DAO can be abstracted into several critical layers, each serving specific functions and collectively supporting the decentralized operational paradigm of DAOs. The

synergy among these layers ensures the transparency, immutability, and autonomy of DAOs, distinguishing them from traditional centralized organizational structures.

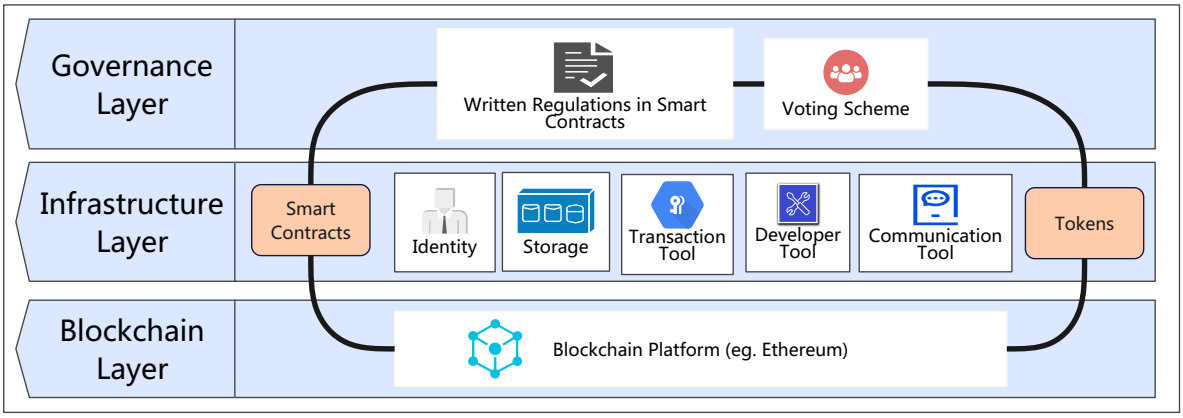

Figure 2.5: DAOs in Web 3 ecosystem

Blockchain Layer

The blockchain layer forms the foundational stratum of the DAO ecosystem and is the primary enabler of its decentralized characteristics. This layer is generally categorized into Layer 1 and Layer 2 solutions.

Layer 1 refers to the foundational layer of a blockchain, essentially the main chain itself. It encompasses the blockchain's native protocols and network infrastructure, responsible for core transaction processing, data storage, and the execution of consensus mechanisms. A blockchain fundamentally operates as a shared distributed ledger, cryptographically secured to ensure its unforgeable, traceable, and transparent nature. Unlike traditional systems that rely on a central institution to establish trust, blockchain platforms build trust in a trustless environment through a decentralized network of nodes executing these consensus algorithms, thereby ensuring system consistency [50].

Consequently, blockchain platforms possess distributed computing power and are not controlled by any single entity. As such, the blockchain platform serves as the bedrock for DAOs, responsible for storing, verifying, and protecting transactions and data, thereby ensuring their immutability and traceability [90]. These inherent properties form the fundamental characteristics of all blockchain applications, including DAOs. Ethereum stands as the most prominent and widely adopted scalable blockchain platform, with a substantial number of DAOs built upon it. Ethereum's introduction of smart contracts brought programmability and extensibility to blockchain, fostering the emergence of numerous Turing-complete blockchain platforms [62]. As detailed in Buterin's whitepaper [62], one of the earliest DAO instances, *The DAO*, was constructed on Ethereum via smart contracts. DAOs maintain a de-

centralized model by encoding their organizational rules directly into smart contracts, which then execute autonomously based on pre-defined conditions. Transaction records and asset transfer histories are permanently recorded on the blockchain. The immutable and traceable nature of the blockchain prevents users from forging their own assets or tampering with others' assets, thus providing a foundational layer of security.

Layer 2 refers to network layers built on top of Layer 1, designed to enhance efficiency and scalability by processing transactions and data off the main chain. Layer 2 solutions achieve this without directly modifying the underlying blockchain protocol, instead employing additional protocol layers [91]. Exemplary Layer 2 technologies include sidechains, state channels [92], Plasma [93], and Rollups (such as Optimistic Rollups and ZK-Rollups) powered by Zero-Knowledge Proofs [94]. For instance, sidechains enable asset transfers and interactions with the main chain through a two-way peg mechanism. They process a large volume of transactions off-chain and then periodically aggregate or submit the final results back to the main chain, thereby alleviating the load on the main chain and increasing overall throughput. These technologies are crucial for DAOs in scenarios requiring high-frequency, low-value transactions or faster finality, significantly reducing transaction costs and improving user experience.

Infrastructure Layer

The infrastructure layer is built upon the blockchain platform, comprising various tools and components meticulously designed to facilitate the development and operation of DAOs. Many of these essential tools are Decentralized Applications (DApps) built on blockchain technology, which not only enhance DAO functionality but also uphold the core tenets of blockchain, such as transparency, security, and decentralization. A summary of these key components is presented in Table 2.4.

For instance, identity and authentication tools are paramount for securing transactions and managing users' digital identities and private keys within a DAO. Common examples include various cryptocurrency wallets like MetaMask, Ledger, and Trezor, which form the bedrock of participation in DAO activities [50]. Beyond these, emerging technologies such as Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) promise more private and sovereign identity management solutions for DAO members [95].

When it comes to storage solutions, DAOs commonly leverage distributed file systems such as the InterPlanetary File System (IPFS) [96] for non-transactional data like documents and media, enabling decentralized storage and retrieval through content addressing and a peer-to-peer network. For more complex queries and large volumes of structured data, some DAOs may also opt for decentralized database solutions like BigchainDB [97] or Arweave

[98], which combine traditional database efficiency with blockchain's immutability and permanence.

Transaction and asset management tools are equally critical, providing functionalities for executing and managing on-chain transactions, including automated execution, tracking, and auditing. These are vital for ensuring a DAO's financial transparency and accountability. This category notably includes multi-signature wallets [99], alongside integrations with various DeFi protocols for efficient treasury management [100]. Meanwhile, developer tools are indispensable for creating, testing, and deploying smart contracts, offering environments and libraries such as Truffle, Hardhat, and OpenZeppelin Contracts [101]. Some of these have evolved into comprehensive DAO development platforms like DAOstack [102] and Aragon [103], simplifying DAO creation and management for even non-technical users.

For effective DAO governance, communication and collaboration tools are pivotal. While traditional options like Discord and Telegram are common, more Web3-native decentralized communication protocols and platforms, are emerging to provide censorship-resistant and privacy-preserving environments. Finally, oracles serve as crucial conduits, allowing blockchains to securely interact with external data. In many DAO operations, smart contracts rely on real-time external information, like market prices or event outcomes, to trigger behaviour. Oracles, such as Chainlink [104], securely obtain and verify this off-chain data in a decentralized manner, transmitting it on-chain to extend the functional boundaries of smart contracts and enable DAOs to respond to real-world events.

All these tools and components are meticulously designed around the core principles of decentralization, leveraging blockchain infrastructure to ensure transparent and traceable operations while adhering to the spirit of decentralization, meaning no single central entity exerts control. Through such a robust infrastructure layer, DAOs can operate within a distributed structure, realizing a self-managing and decentralized organizational framework.

Governance Layer

The foundation of a DAO begins with the creation of smart contracts, which are essentially programs stored on a blockchain that run when predetermined conditions are met [105]. Programmers develop these contracts, embedding within them the rules of the organization as well as the mechanisms for executing decisions autonomously. Once the foundational smart contracts are established, a DAO typically initiates funding through a token sale, a common mechanism for decentralized capital formation within the Web3 ecosystem [106]. These tokens represent voting power and potentially a share in the profits of the DAO [107]. Ownership of these governance tokens directly grants members the right to vote on important decisions, such as the usage of funds, project development, and changes to the DAO's rules,

Table 2.4: Web3 Ecosystem Components for DAOs

Category	Description	Examples / Key Technologies
Identity & Authentication	Tools for managing digital identities, private keys, and securing interactions within a DAO.	Cryptocurrency wallets (MetaMask, Ledger, Trezor), Decentralized Identifiers (DIDs), Verifiable Credentials (VCs)
Storage Solutions	Decentralized systems for storing non-transactional data (documents, media) and structured data.	InterPlanetary File System (IPFS), BigchainDB, Arweave
Transaction & Asset Management	Functionalities for executing, tracking, and auditing on-chain financial operations and managing DAO treasuries.	Multi-signature wallets, DeFi protocol integrations, asset management dashboards
Developer Tools	Environments, frameworks, and libraries for building, testing, and deploying smart contracts and DAO functionalities.	Truffle, Hardhat, OpenZeppelin Contracts, DAOstack, Aragon
Communication & Collaboration	Platforms and protocols enabling decentralized and secure communication and collective decision-making among DAO members.	Status, decentralized messaging DApps
Oracles	External data feeds that securely bring off-chain information onto the blockchain for smart contract execution.	Chainlink

forming the bedrock of token-based governance [108]. This voting process is critical as it embodies the decentralized governance model of DAOs, where no single entity has control, and decisions are made collectively by the community [109].

The operation of a DAO is heavily reliant on these tokens and the associated smart contracts. When a proposal is made within a DAO, token holders cast their votes proportional to their holdings. These votes are then tallied automatically by the smart contracts, and if consensus is achieved, the contracts execute the decision autonomously on-chain. This might include releasing funds, starting new projects, or altering the DAO's operational framework. All actions taken within a DAO are recorded transparently on the blockchain, providing a tamper-proof ledger of decisions and executed tasks. This level of transparency ensures that all members have access to the same information and can trust in the immutability of the records, which reinforces security and accountability within the organization.

Although the DAO voting process is fundamentally part of the governance layer, its successful implementation relies on the support of both the blockchain and infrastructure layers. Blockchain infrastructure ensures the immutability, security, and automation of voting through smart contracts, while tools in the infrastructure layer, such as wallets, DApps, and oracles, provide the necessary interfaces and data connectivity. Therefore, DAO voting

mechanisms are deeply integrated across the full technical stack, with their logic residing in the governance layer and their operations enabled by the lower layers. Building upon these fundamental principles, the following section delves into the specific framework governing voting mechanisms within DAOs, detailing how these concepts translate into actionable decentralized decision-making processes.

2.3.2 Voting Mechanism Framework

While the voting mechanism serves as the core decision-making component in DAOs, its structure and operational flow are distinct from traditional voting systems due to the integration of decentralization principles and blockchain infrastructure. Fig. 2.6 demonstrates the interaction of the DAO voting mechanism with the blockchain infrastructure. First, DAO members start proposing and voting. Once a proposal is successfully submitted, it remains open for voting until it either attains the required approval condition or reaches a specified time limit. After the voting period ends, the ballot is uploaded to the blockchain for storage. Then, the ballot is checked by the written smart contracts to verify whether it satisfies the proposal approval requirements and to determine the voting result, and the voting result is uploaded to the blockchain. Finally, implementations, including writing new smart contracts, are carried out according to the voting result.

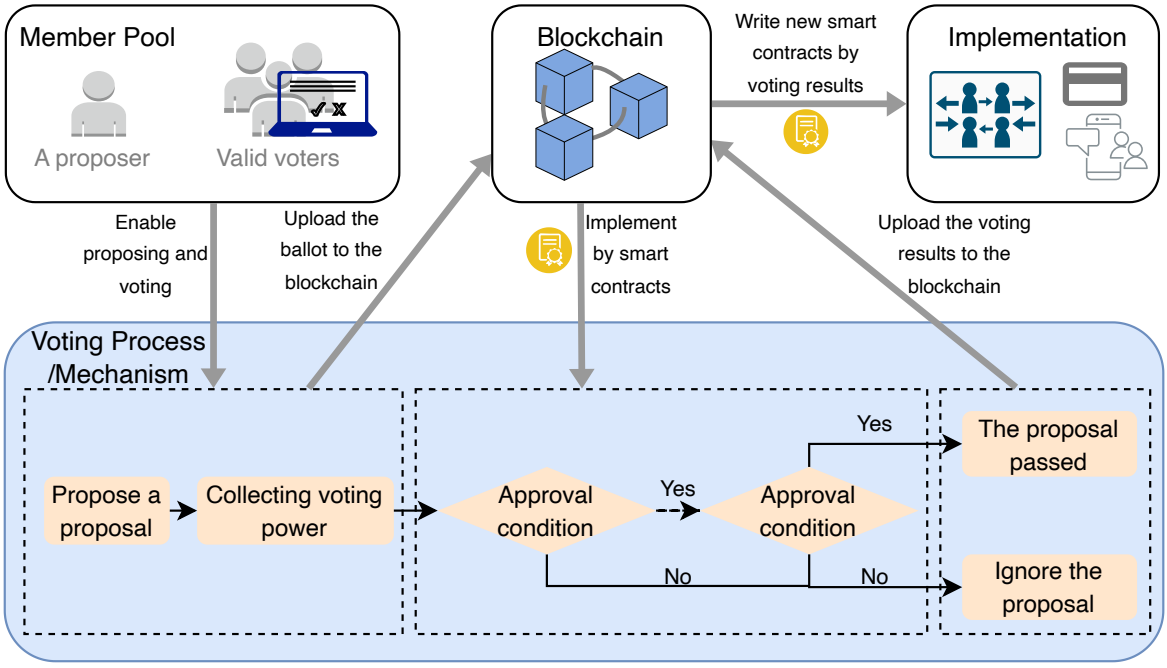

Figure 2.6: Framework of the DAO voting system

Membership and Proposal: DAOs are typically characterized by their permissionless and inclusive nature, allowing anyone from the entire network to participate. Although a few

DAOs have more stringent entry requirements like MetaCartel [84], in most DAOs, membership is tied to token ownership, meaning that possessing the native tokens of a DAO automatically grants one membership. The rights to make proposals and cast votes are typically linked to membership, which is a common feature. Proposals serve as the direct subjects of voting, encompassing all the ideas, viewpoints, plans, and actions presented by DAO members regarding rules and activities within the organization. The scope of proposals is extensive, covering aspects such as management, financial expenditures, and the development of new community initiatives. Given the diverse and rigorous nature of proposals, some DAOs categorize them and employ distinct voting procedures for different categories. In the majority of DAOs, all members have the privilege to submit proposals.

Voting Power: Voting power is the number of votes of a member, and the voting power allocation largely impacts the voting results in DAOs [110]. One person one vote (1P1V) is a typical voting power allocation scheme that assigns equal voting rights/powers to each voter but is rarely used in DAOs. Although 1P1V aligns with the decentralization advocated by DAOs, it increases the risk of low-cost manipulation of DAOs in a permissionless entry environment. It can easily lead to irresponsible voting behavior by voters with few tokens.

In contrast with 1P1V, the most widely use method is one token one vote (1T1V), which is mostly applied in current DAOs due to the financial support requirements in a large number of proposals. The most widely use method is one token one vote (1T1V) which means that 1 token can be converted into 1 unit of voting power. Although this method seems fair, effective, and simple, plutocracy in which a large number of voting power is in the hands of a few members is basically inevitable under this scheme [110], which runs counter to the DAO's pursuit of decentralization. In order to compare the decentralized effectiveness of the most commonly used DAO voting mechanisms, the 1T1V is used in the analysis of this paper. However, by adding a conversion relationship between tokens and voting power, our model is also applicable to evaluating other voting power conversion schemes.

Approval Conditions: Fig. 2.6 illustrates a common voting process in DAO with approval conditions as criteria for the approval of proposals. The criteria for approval in the voting process differ among various voting mechanisms. In this context, common approval conditions are presented, which will be illustrated as examples in the voting model discussion.

Decision-making thresholds are essential for evaluating whether a proposal should be approved in various voting mechanisms. Two commonly used conditions are the relative majority (RM) and the absolute majority (AM). The relative majority (RM) is defined as the number of voting power for which a majority (more than 50%) of all voting power (excluding abstentions) favours [111]. RM is effective but also carries risks, as it can lead to the approval

of harmful proposals without the knowledge of the majority of members. In contrast, the absolute majority (AM) is defined as the amount of the cast vote power in favour of a proposal surpassing half of the total eligible voting power (including abstentions) [112]. This method is a safer option but can be more challenging to achieve, especially on large-scale platforms.

Promoting adequate participation is an effective strategy for maintaining the decentralization effectiveness of DAO voting. Unlike decision-making thresholds, which are feasible for a voting process, attention thresholds do not always exist in all DAO voting mechanisms, but attention thresholds play a crucial role in ensuring that a proposal garners sufficient attention from the required number of voters to be approved. There are two common attention thresholds: the Voting Power Threshold (VT) and the Quorum Threshold (QT). VT sets the minimum voting power required for a proposal to pass, while QT specifies a specific number of voters needed to participate in the voting. Both attention thresholds aim to increase the participation rate, but they may also impact the efficiency of the voting process [113].

2.3.3 Representative DAO Voting Mechanisms

Then, several representative DAO voting mechanisms are presented, highlighting their underlying design principles, as well as their respective strengths and limitations.

Permissioned Relative Majority (PRM): PRM is the simplest and the most widely used voting mechanism so far. The mechanism is remarkably highly efficient, clarifying the approval condition as a majority supporting voting power, which means a voting pass threshold is clearly notified as 50% among all the voted power spent in this voting. However, the mechanism can be easily manipulated when dangerous proposals do not get enough attention from members. Without any attention requirement, PRM has a high-security risk of the slip-through passed proposal threat proposed in Sec. 4.2.2. This problem may be alleviated in small-scale organizations with high activity, but its low security will certainly affect the long-term development of the organization.

Token Based Quorum Mechanism (TBQ): TBQ is another major mechanism applied in DAOs. The core rule is the same as PRM, which is to listen to the majority, but it requires a higher level of participation from the organization. Participation can be enhanced by adding an attention threshold to a relative majority or simply substituting the relative majority threshold with the absolute majority threshold. The participation requirement largely reduces the slip-through passed proposal threat which increases the security of the voting with the member's attention playing a safeguard role. However, it is a trade-off that the voting process likely extends the time to collect sufficient attention, and the proposal passing rate may be reduced due to the more strict approval conditions, thereby, the voting efficiency will largely

decline compared to PRM. In addition, the attention requirement involves more members in voting which increase the decentralization feature to some extent.

Quadratic Voting: *Quadratic Voting* is an improved voting power allocation scheme that balance 1T1V and 1P1V with a marginal cost increase design. The original quadratic voting was proposed by Edward H. Clarke [114] and applied in democratic politics. *Quadratic voting* in DAOs applied the same method that Edward proposed: the marginal cost increases as a user repeatedly votes on the same option [114]. For example, for the same option voting choice from one user, 1 vote requires 1 token, 2 votes require 4 tokens, 3 votes require 9 tokens, etc. The number of votes from each user equals the square root of the tokens paid. Compared with the commonly used 1T1V, the decentralized metric is significantly improved by mitigating the overwhelming voting power of a small number of enormous token holders. At the same time, it is more reasonable and flexible compared to the 1P1V.

Liquid Democracy: *Liquid Democracy* is a voting scheme commonly discussed in political science, which can also benefit voting in DAOs. In *Liquid Democracy* scheme, voters are allowed to vote directly or delegate their voting rights to a representative who is usually an expert in the community, which can improve the effectiveness of the voting. It is also notable that the delegation is allowed to be multi-level, which means representatives can also delegate to other representatives with all the votes they have been delegated as well. The representative is normally much more active than ordinary voters, so the efficiency of voting will be greatly improved due to the rapid processing of a large number of delegated votes. However, delegation may cause a setback for decentralization since a delegation voting structure tends to be hierarchical and centralized. Fortunately, the liquid feature is designed to emphasize that voters can change their delegation at any time, and voters can delegate different issues to different experts or partially delegate their voting rights. Thereby the trend of centralization can be alleviated. However, the impact of liquid democracy on decentralization cannot be ignored, which is a trade-off for improved efficiency.

Weighted Voting: *Weighted Voting* is a typical way to increase effectiveness. As the name suggests, *Weighted Voting* add extra weight to the calculation of the utility of the voting power of each member. Usually, reputation and knowledge are the main sources of weight calculation. *Knowledge-Extractable Voting* is an example that gives experts in a certain field more voting power by increasing their voting weight, which is decided by the knowledge token each user has [89]. The knowledge tokens will reward users if their voting choices match the winning result [89]. On the contrary, the knowledge tokens will deduce if their voting choice is different from the winning result [89]. Basically, the *Weighted Voting* tends to allocate more voting power to experts or experienced members, which increases the possibility of making informed decisions. Therefore, the effectiveness of voting is greatly improved.

Rage Quitting: *Rage Quitting* is very popular in investment DAOs as the mechanism ensures that everyone's interests are not harmed by others. In these DAOs, members are allowed to withdraw from the organization at any time and retrieve their funds in tokens. In *Rage Quitting*, a passed proposal will stay in a grace period before the voting results are executed. In this grace period, the members who are extremely unsatisfied with the result can withdraw from the organization in anger. Giving members a more relaxed right to choose is also a manifestation of the decentralization in voting. Each member is not bound by a collective choice, and it is difficult for an owner with overwhelming voting power to control the assets of others which increases the decentralization of members' rights. However, a grace period clearly slows down the voting process, which has a noticeable negative impact on voting efficiency.

Holographic Consensus: *Holographic Consensus* was developed by DAO stack, an open-source full software stack for building and running DAOs [115]. *Holographic Consensus* associates each proposal with a prediction market and introduces a betting token GEN specifically for prediction markets [115]. DAO members or the general public can bet on proposals they think will pass or fail by up staking or down staking GENs [115]. Bettors who make predictions consistent with the voting results gain more GENs for reward [115]. *Holographic Consensus* is designed to believe that the voting participation threshold can be reasonably reduced when obtaining great attention from the prediction market. Therefore, proposals have two paths to reaching a valid voting result. In one path, the proposals collecting advocating GENs above a threshold are boosted and enter the boosted state [115]. Then the proposals are only required a relative majority to pass. On the other path, proposals without collecting enough advocating GENs stay in Queued state and require an absolute majority in all voting power voting to pass [115].

The use of GEN is a financial incentive mechanism, which makes the voting flexible and efficient. In terms of decentralization, GENs help people to show their opinions on proposals that they do not have large voting power. However, *Holographic Consensus* is not flawless. Although the quorum participating in voting is very likely to be different from the quorum betting, it is inevitable that people who pay GENs on a proposal have higher motivation to vote on a pass, which may distort the voting results. High approval in the prediction market is considered confidence in the proposal. However, it is difficult to determine whether this confidence reflects an assessment of the bettor's own benefits or an evaluation of the organization's development.

Table 2.5: A comparison of DAO voting and conventional voting

DAO Voting	**Conventional Voting**
Equal position among members.	Hierarchical positions.
Generally, proposals can be initiated by any member in any aspect.	Proposals are for typical issues and most members do not have the right to make proposals.
Usually, all members have voting rights.	Voting may not be open for everyone in the organization.
Voting processes are automated through smart contracts.	Voting processes usually require manual handling and rely on internal trust.
Voting powers are usually token-based and democratic distributed.	Voting powers are heavily influenced by status and wealth.
Voting processes are all transparent.	Voting processes are limited to the public.

2.3.4 Compare DAO Voting and Conventional Voting

DAO voting shares similarities with voting in conventional organizations, such that they have similar decision-making goals and similar entities. However, following the ethos of decentralization, DAO voting is very different from conventional voting. First of all, affected by decentralization, DAO voting is featured with equal positions among members, while conventional voting usually has hierarchical positions. Second, proposals in DAO voting can be initiated by any member in any aspect generally. In contrast, most members do not have the right to make proposals in conventional voting and proposals are limited to typical issues. Usually, all members have voting rights, while voting may not be open for everyone in conventional organizations. In terms of voting powers, they are usually token-based and democratically distributed in DAO voting, while voting powers are heavily influenced by status and wealth in conventional voting. Supported by distributed ledgers, voting processes are transparent in DAO voting, while they are limited to the public in conventional voting. A comparison is shown in TABLE 2.5.

2.3.5 Existing Evaluations of DAO Voting Mechanisms

While existing literature has primarily focused on categorizing DAO voting mechanisms and describing their operational principles, comparative analyses of their governance performance have also emerged. Existing studies have increasingly sought to compare and evaluate DAO voting mechanisms across multiple dimensions of governance performance. A common line of analysis focuses on decentralization and fairness. Empirical investiga-

tions into leading DeFi DAOs, including Uniswap, Aave, and Compound, consistently reveal that despite their claims of decentralization, voting power tends to be highly concentrated in practice [116]. This concentration has led scholars to describe a "decentralization illusion," where a small number of large stakeholders dominate outcomes and expose DAOs to collusion risks and governance attacks [117]. Beyond fairness, comparative research has also examined efficiency and participation. Cross-platform analyses show significant divergence: while DAOs in the Internet Computer ecosystem exhibit participation rates above 60% due to the absence of gas costs, DAOs operating on Ethereum frequently report participation rates below 10% [118], reflecting the prohibitive impact of transaction fees on small token holders. High proposal approval rates, often exceeding 90% across platforms, have been interpreted as evidence of community alignment but may also reflect low contestation, strategic proposal filtering, or disengaged voter bases [119]. These findings highlight that DAO voting outcomes are shaped not only by mechanism design but also by the underlying blockchain infrastructure and incentive models. However, despite these efforts, existing comparative studies remain fragmented, lack standardized evaluation metrics, and often fail to capture the interplay between decentralization, efficiency, and security. This motivates the need for a more systematic and reproducible framework, as developed in Chapter 4.

Chapter 3

Probabilistic Model for DFT Consensus

With rapidly growing scale, connectivity and inherent heterogeneity, distributed architectures play a crucial role as the foundational structure in a variety of modern connected systems. Examples include autonomous systems [120] [121], Web 3 infrastructure [122] [123], decentralized Industrial Internet of Things (IIoT) [124] [74], and decentralized supply chain networks [125]. Each entity in such connected systems is often designed to operate individually while they remain interconnected, making reliable information exchange and synchronization with other entities essential for system-level cohesive operation and effective coordination. Centralized control can be applied to distributed structures but may potentially lead to bottlenecks such as critical single-point failures due to the underlying network dynamics and unpredictable communication performance [126], as well as delayed response from the server due to its resource constraint [127]. In contrast, distributed synchronization and joint decision-making eliminate reliance on any single point, enhancing system resilience, performance and flexibility [13]. Distributed consensus (DC), as a key enabling technique for distributed operations, is crucial for maintaining coordination for a system in dynamic, decentralized environments.

DC mechanisms, has been extensively studied in computer networks, from early work on fault-tolerant distributed databases to recent advancements in blockchain technology [128] [129]. However, using traditional DC mechanisms in these distributed systems leads to several new challenges. DC mechanisms usually have a theoretical assumption on the maximum number of faulty nodes that can be tolerated. Specifically, BFT and CFT protocols are designed to tolerant less than $1/3$ [9] [7] and $1/2$ [130] faulty nodes, respectively. However, faults and failures are common and unpredictable in real-world connected systems, where no node can be guaranteed to remain reliably operational at all times. This unpredictability is especially problematic in safety-critical scenarios and makes deterministic assumptions about the maximum number of faulty nodes in entire systems impractical. Moreover, deterministic

fault tolerance assumptions are inflexible. Such fixed boundaries can neither guarantee absolute security nor ensure the efficiency of consensus. In highly reliable systems, the fault tolerance design may incur unnecessary computational and communication overheads. Conversely, in less reliable systems, the number of faulty nodes may exceed the threshold, leading to system failures.

To address this challenge, probabilistic model has been proposed to change the fault tolerance assumption from a deterministic limit on the number of faulty nodes to a probabilistic model of node reliability, which is evaluated as the probability of non-faulty behaviour. Despite extensive studies [74] [13] [76] [75] [77] analysing the performance metrics of the probabilistic model, they fail to address how the assumption of probabilistic node failures fundamentally challenge the design of consensus protocols, as all existing research relies on original consensus principles. This leads us to propose two critical issues that have been previously ignored. First, the full range of outcomes from the execution of the probabilistic model for distributed consensus has not been thoroughly examined. Since each node operates with a certain level of reliability, extreme conditions may lead to a significant number of faulty nodes, making consensus achievement not always guaranteed. This can result in undesirable outcomes such as partial consensus, consensus forks, or even incorrect consensus. Previous studies have only focused on the probability of reaching an ideal consensus and neglected the discussion of other consensus outcomes.

To address this, in this chapter, the entire range of potential consensus outcomes across the dimensions of correctness and consistency is systematically analyzed. Consistency ensures agreement among participating nodes, while correctness ensures that consensus decisions reflect the opinions of non-faulty nodes and are not affected by faulty nodes. To narrow the scope of possible outcomes, the interaction between different numbers of faulty and non-faulty nodes with various quorum values is examined. This analysis yields three consensus states: safe (where all non-faulty nodes consistently reach correct consensus), risky (where correct consensus cannot always be guaranteed), and compromised (where correct consensus can never be achieved). This state-based view provides a rapid way to estimate possible outcomes. Building on that, an analytical approach is developed to calculate the precise probabilities of each consensus outcome.

The second issue arises from changes in the fault tolerance assumption, which critically affect quorum designs that depend on it. Quorum refers to the minimum subset of nodes required to make a decision or verify an operation in a distributed system. In BFT, quorum is traditionally defined as $2f_{\max} + 1$, where $f_{\max} = \left\lfloor \frac{n-1}{3} \right\rfloor$ is the maximum number of faulty nodes. With the $1/3$ maximum faulty node boundary, the quorum ensures the consistency of the decision of all non-faulty nodes. However, in the probabilistic model, this quorum rule is

no longer effective, as the probabilistic assumption about the faulty behaviour of each node can result in an nondeterministic faulty node number and the number can exceed $f_{\max}$. Unfortunately, previous studies on probabilistic model for distributed consensus directly adopted this conventional quorum value $2f_{\max}+1$ in the protocol as a key indicator for determining whether a node has received enough confirmations from other nodes, which is inappropriate.

To address this issue, the **reliability quorum** is introduced to define the minimum subset of nodes required to ensure a specified level of system reliability in the consensus process, rather than enforcing a rigid, deterministic threshold. The optimal reliability quorums are derived either by maximizing the probability of a safe consensus or by assigning weights to different consensus outcomes, thereby tailoring the quorum choice to application-specific priorities. The reliability quorum provides a more reasonable threshold for nodes to evaluate whether they have received adequate confirmation from other nodes during the consensus process. Our conclusions on the two reliability quorum approaches are broadly applicable to most BFT algorithms.

Beyond the probabilistic nature of node failures, many modern distributed systems are increasingly deployed in environments where wireless communication is indispensable, yet inherently unreliable. Consider safety-critical applications like cooperative autonomous driving, drone swarms for search and rescue, or decentralized industrial control systems. In these scenarios, distributed consensus is crucial for joint decision-making, but its robustness is directly tied to the underlying communication channels. Unlike stable wired connections, wireless links are prone to unpredictable and dynamic impairments: signal fading, interference, path loss, and intermittent connectivity are common occurrences that can lead to packet loss, increased latency, and even complete link disruptions.

While our analysis of probabilistic node reliability offers a more nuanced understanding of consensus under faulty nodes, it has implicitly assumed perfectly reliable or ideal communication channels. This simplification is often inadequate for accurately evaluating and designing robust distributed systems that operate over wireless links. The unique challenges of wireless propagation introduce an independent and significant layer of uncertainty that traditional consensus models, even those considering probabilistic node failures, often overlook. Therefore, understanding the interplay between probabilistic node behaviour and probabilistic link reliability is paramount for achieving genuine reliability in wireless distributed consensus (WDC).

This chapter not only analyses how probabilistic node failures impact consensus outcomes and quorum design but also independently extends this framework to comprehensively incorporate the probabilistic nature of wireless communication links. By jointly considering both probabilistic node behaviour and probabilistic link reliability, a more holistic and accurate

model for distributed consensus in truly heterogeneous and dynamic environments is provided. This dual perspective is critical for designing robust and resilient distributed systems in challenging wireless settings. While the framework offers a comprehensive understanding of consensus behaviour under probabilistic assumptions, it should be noted that the focus of this chapter remains at the analytical level rather than algorithmic implementation. This chapter therefore aims to provide an improved theoretical foundation: it formalises reliability conditions and examines their implications, clarifying the limitations and potential of existing approaches. No new algorithmic implementation is introduced at this stage; instead, the models and propositions presented are intended to prepare the ground for future algorithmic design.

3.1 Probabilistic Byzantine Node Model

A distributed network of entities is considered, characterized by nodes with relatively independent and autonomous control, yet requiring consensus for collaboration.

3.1.1 Nodes Behaviour

The consensus system is assumed to contain faulty nodes, adopting the worst-case Byzantine behaviour model, in which nodes may exhibit arbitrary errors or malicious actions. Nodes in the network exhibit either reliable or Byzantine behaviour. Reliable nodes make consistent evaluations and follow protocol rules, while Byzantine nodes behave arbitrarily, whether due to faults or malicious intent. During request proposal, reliable nodes submit valid requests, whereas Byzantine nodes may submit invalid ones, disrupting consensus. Throughout the consensus process, reliable behaviour follows the protocol and ensures evaluations align with those of other reliable nodes. In contrast, Byzantine behaviour includes any malicious or faulty actions that deviate from the expected protocol norms.

Let $\Omega = \{N_1, N_2, \ldots, N_n\}$ be the set of n nodes in a consensus network. Each node N_i exhibits either Byzantine ($B_i = 1$) or reliable ($B_i = 0$) behaviour. Byzantine behaviour is modelled probabilistically by defining $P_{B,i} \sim D$ as a random variable representing the probability that node N_i is Byzantine, i.e., $P(B_i = 1) = p_{B,i}, P(B_i = 0) = 1 - p_{B,i}, p_{B,i} \in [0, 1]$. Each $P_{B,i}$ follows an independent distribution with a probability density function $f_{P_{B,i}}(p_{B,i})$, chosen based on application needs. The primary node, initiating the consensus request, is always denoted as N_1, with a Byzantine probability of $P_{B,1}$. Since nodes fail probabilistically, the total number of Byzantine nodes, F, is a random variable determined by all $P_{B,i}$. The sample space is defined as $\mathbf{S_f} = \{\mathcal{S}_{f,1}, \ldots, \mathcal{S}_{f,\binom{n}{f}}\}$, where each subset $\mathcal{S}_{f,j} = \{N_i \mid B_i = 1\}$ contains

exactly f Byzantine nodes. The total number of possible Byzantine sets is $|\mathbf{S_f}| = \binom{n}{f}$. The probability mass function (PMF) of F is then given by

$$P(F = f) = \int_0^1 \cdots \int_0^1 P(F = f \mid p_{B,1}, p_{B,2}, \ldots, p_{B,n}) \prod_{i=1}^{n} f_{P_{B,i}}(p_{B,i})\, dp_{B,1}\, dp_{B,2} \cdots dp_{B,n}, \tag{3.1}$$

where

$$P(F = f \mid p_{B,1}, p_{B,2}, \ldots, p_{B,n}) = \sum_{\mathcal{S}_{f,j} \in \mathbf{S_f}} \prod_{N_i \in \mathcal{S}_{f,j}} p_{B,i} \prod_{N_i \in \Omega \setminus \mathcal{S}_{f,j}} (1 - p_{B,i}). \tag{3.2}$$

Since the behaviour of the primary node profoundly impacts consensus, the probability that the system contains $F = f$ Byzantine nodes while the primary node remains reliable is of particular interest and can be derived as

$$P((F = f) \cap (B_1 = 0)) = \begin{cases} 0, & \text{if } f = n, \\ \int_0^1 \int_0^1 \cdots \int_0^1 \sum_{\mathcal{S}'_{f,j} \in \mathbf{S'_f}} \prod_{N_i \in \mathcal{S}'_{f,j}} p_{B,i} \prod_{N_i \in \Omega \setminus \mathcal{S}'_{f,j}} (1 - p_{B,i}) & \text{otherwise,} \\ \qquad \times \prod_{i=1}^{n} f_{P_{B,i}}(p_{B,i})\, dp_{B,1}\, dp_{B,2} \cdots dp_{B,n}, \end{cases} \tag{3.3}$$

where $\mathcal{S}'_{f,j}$ denotes a subset containing exactly f Byzantine nodes selected from $\Omega \setminus \{N_1\}$ (with $B_1 = 0$, so that the primary node is reliable), and $\mathbf{S}'_f = \{\mathcal{S}'_{f,j} : 1 \le j \le \binom{n-1}{f}\}$ is the sample space of all such subsets.

Similarly, the probability that the system has $F = f$ Byzantine nodes while the primary node is Byzantine can be written as

$$P((F = f) \cap (B_1 = 1)) = \begin{cases} 0, & \text{if } f = 0, \\ \int_0^1 \int_0^1 \cdots \int_0^1 \sum_{\mathcal{S}''_{f-1,j} \in \mathbf{S''_{f-1}}} \prod_{N_i \in \mathcal{S}''_{f-1,j}} p_{B,i} \prod_{N_i \in \Omega \setminus \mathcal{S}''_{f-1,j} \setminus \{N_1\}} (1 - p_{B,i}) \times p_{B,1} & \text{otherwise,} \\ \qquad \times \prod_{i=1}^{n} f_{P_{B,i}}(p_{B,i})\, dp_{B,1}\, dp_{B,2} \cdots dp_{B,n}, \end{cases} \tag{3.4}$$

where $\mathcal{S}''_{f-1,j}$ denotes a subset containing exactly $f-1$ Byzantine nodes selected from N_2 to N_n (i.e., excluding the primary node N_1), and $\mathbf{S}''_{f-1} = \{\mathcal{S}''_{f-1,j} : 1 \le j \le \binom{n-1}{f-1}\}$ is the sample space of all such subsets.

3.1.2 Reliability Quorum and Intersection Quorum

As discussed, the concept of reliability quorum, denoted as Q_r, is proposed. It is defined as the minimum quorum size necessary to ensure a specified level of system reliability, serving as the threshold for nodes to wait for sufficient confirmations during the consensus process. Q_r can be an integer within $[0, n]$. In contrast, the conventional quorum is typically defined by the intersection rule, ensuring that any two quorums share at least one reliable node. This interaction rules results in the conventional BFT quorum being $\left\lceil \frac{n+f+1}{2} \right\rceil = 2f+1$ [131], when $n = 3f+1$. While the focus is placed on the reliability quorum for consensus, the conventional quorum remains important for analysing consensus states (see Sec. 3.2.1). For clarity, the conventional quorum is referred to as the *intersection quorum*, denoted as Q_{int}. Since the number of Byzantine nodes is a variable F, the intersection quorum is also treated as a variable,

$$Q_{int} = \left\lceil \frac{n+F+1}{2} \right\rceil. \tag{3.5}$$

From (3.5), the crucial inequality can be easily derived as

$$Q_{int} > F. \tag{3.6}$$

The intersection quorum is used for analysing the consensus state and reliability but does not directly impact the consensus protocol design. The reliability quorum serves as the threshold for nodes to collect enough confirmation messages during the consensus process.

3.1.3 Consistency Threshold

Probabilistic failures disrupt State Machine Replication (SMR) in conventional BFT [28], making it very likely that in probabilistic model non-Byzantine nodes reach an inconsistent result. Theoretically, based on the relationship between Q_r and Q_{int}, the following property on system-wide consensus is given.

Specifically, when $Q_r \geq Q_{int}$, consensus reached by any reliable node implies that system-wide consensus is guaranteed. In contrast, when $Q_r < Q_{int}$, consensus reached within a subset of reliable nodes does not necessarily extend to the whole system, creating the possibility of a consensus fork. This observation is consistent with classical consensus theory and is included here to validate that the proposed probabilistic framework remains aligned with established deterministic principles. The reasoning follows from the fact that, if two sets of nodes confirm consensus with sizes satisfying $Q_r \leq |\mathcal{D}_1| < Q_{int}$ and $Q_r \leq |\mathcal{D}_2| < Q_{int}$, then in the worst-case scenario all Byzantine nodes may be distributed between these two sets. As a result, the sets may contain no common reliable node, allowing them to reach different outcomes and

thus jeopardising system-wide consensus. Conversely, if both sets satisfy $|\mathcal{D}_1| \geq Q_{int}$ and $|\mathcal{D}_2| \geq Q_{int}$, then by definition they must share at least one reliable node, ensuring consistent results across the system.

While this result provides a theoretical foundation for analysing consensus under probabilistic failures, it is not directly applicable to system design because Q_{int} is a random variable and its exact relationship with Q_r is uncertain. To address this, we introduce the *consistency threshold* t, defined as the minimum number of reliable nodes required for successful consensus. In practice, t can be tuned according to system requirements, but is typically set above half of the total nodes, i.e., $t \geq \lfloor \frac{n}{2} \rfloor + 1$, in order to prevent Byzantine nodes from dominating the decision process. This assumption ensures that at least half of the nodes share a consistent state, thereby preserving consensus validity.

3.1.4 BFT Consensus under Probabilistic Model

Based on the assumption of Byzantine behaviour from an individual perspective, the operation of distributed consensus under a probabilistic model is described within the framework of BFT algorithms, using PBFT [29] as an illustrative example. Note that the similar approaches in this thesis apply to most BFT algorithms.

Request

Since consensus is designed for self-operating systems requiring collaboration or group decision-making, any node can initiate a consensus request, which is then broadcast across the network. The initiating node acts as the primary node, while others evaluate the request's validity, as defined below.

Definition 1. ***Validity*** *is the property of a request in a consensus system that ensures it conforms to system regulations, is correctly formatted, and is authenticated as legitimate and compliant with operational rules.*

Note that it is assume the standards and understanding of request validity are consistent among at least all the reliable nodes in the system. Any faulty behaviour caused by deviations from these validity standards is also regarded as a Byzantine fault. Following the consensus protocol, each node interacts with others to determine whether to approve the request.

Consensus Process

Fig. 3.1 illustrates the consensus process, which consists of three phases: pre-prepare, prepare, and commit. The primary node initiates the process by broadcasting a pre-prepare mes-

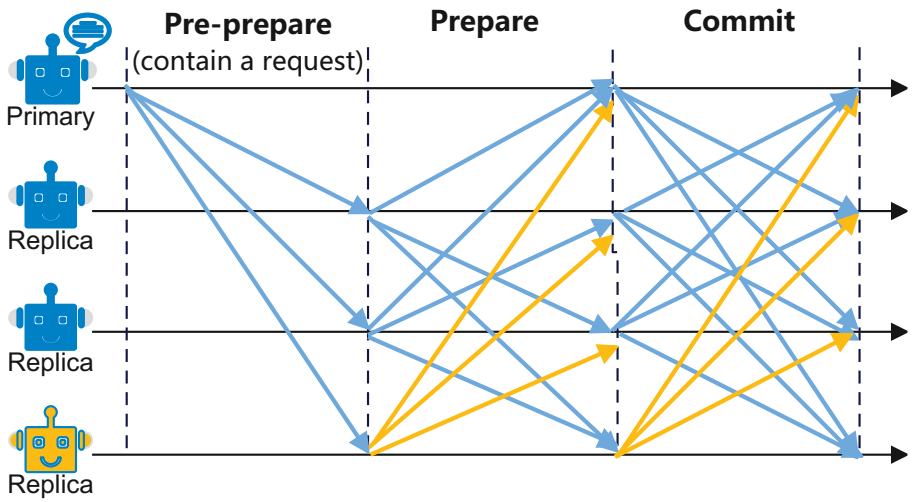

Figure 3.1: PBFT-based probabilistic consensus

sage with the consensus request. Replicas verify its validity and reject invalid messages. If the message is valid, they proceed to the prepare phase, where each node (except the primary) broadcasts a prepare message to all other nodes. A node enters the commit phase when it receives at least the reliability quorum Q_r of identical and valid prepare messages. During this phase, all nodes, including the primary, broadcast commit messages. Consensus is reached when a node receives at least Q_r identical and valid commit messages. However, in probabilistic model, individual nodes may reach different conclusions on consensus. System-wide consensus is considered achieved when at least the consistency threshold t of nodes reach the same outcome. Throughout the consensus process, reliable nodes strictly follow validity rules and the protocol, while Byzantine nodes may exhibit arbitrary behaviour due to faults, such as incorrect requests, or malicious actions, such as disrupting consensus with fraudulent messages.

3.2 Analysis of Non-Deterministic Outcomes in Probabilistic Consensus

This section begins with a qualitative analysis of consensus outcomes under various scenarios. To enable more efficient quantitative analysis, the notion of consensus states is introduced, allowing for rapid estimation of consensus outcomes based on node and quorum configurations. Finally, a detailed method is presented for quantitatively calculating the probabilities of different consensus outcomes. Key parameters used in quantitative analysis are listed in Table 3.1. Note that, we adhere to a specific notational convention where lowercase letters denote invariant parameters, and uppercase letters represent random variables to

clearly distinguish between system design constants and variables for our analysis. f_{max} represents the maximum number of Byzantine nodes the system can tolerate, which is a constant determined by the protocol's fault-tolerance properties. F is a random variable representing the actual number of Byzantine nodes present in a given simulation run, a value that is drawn from a probability distribution within the range of 0 to N. This approach allows us to systematically evaluate the system's performance under various fault conditions.

Table 3.1: Frequently used notations

Notation	Definition
$p_{B,i}$	The probability of node N_i is a Byzantine node
n	Number of nodes in the consensus network
F	Number of Byzantine nodes
Q_r	Reliability Quorum
Q_{int}	Intersection Quorum
p_s	The probability that a Byzantine node sends an invalid message rather than no message
t	Consistency Threshold
$\mathcal{E}_S$	The event of safe state
$\mathcal{E}_{R1}$	The event of risky state with $F < Q_r < \min(Q_{int}, n-F)$
$\mathcal{E}_{R2}$	The event of risky state with $Q_r \leq F < t \leq n-F$ and $t \leq n-F$
$\mathcal{E}_{C1}$	The event of compromised state with $F < t \leq n-F < Q_r$
$\mathcal{E}_{C2}$	The event of compromised state with $n-F < t$
$\mathcal{O}_{\text{CA}}$	The event of CA
$\mathcal{O}_{\text{CR}}$	The event of CR
$\mathcal{O}_{\text{CF}}$	The event of CF
$\mathcal{O}_{\text{CFk}}$	The event of CFk

3.2.1 Qualitative Analysis of Consensus Outcomes

The consensus outcome of the probabilistic model can be evaluated along two dimensions: correctness and consistency. Correctness, from an individual perspective, refers to whether each reliable node agrees on a valid consensus outcome or correctly rejects an invalid request. Consistency, from a system-wide perspective, refers to whether all reliable nodes reach the same consensus outcome and meet the consistency threshold required by the system, which may vary depending on system requirements. Based on these two dimensions, four types of consensus outcomes are categorized, illustrated in Fig. 3.2. Since correctness is a property of each node and consistency is a system property, when Fig. 3.2 evaluates outcomes from the system's perspective, correctness is represented as a spectrum (with the quadrants divided by the dashed line), while consistency is binary (with the quadrants divided by the solid line).

The four types of outcomes are explained in detail below.

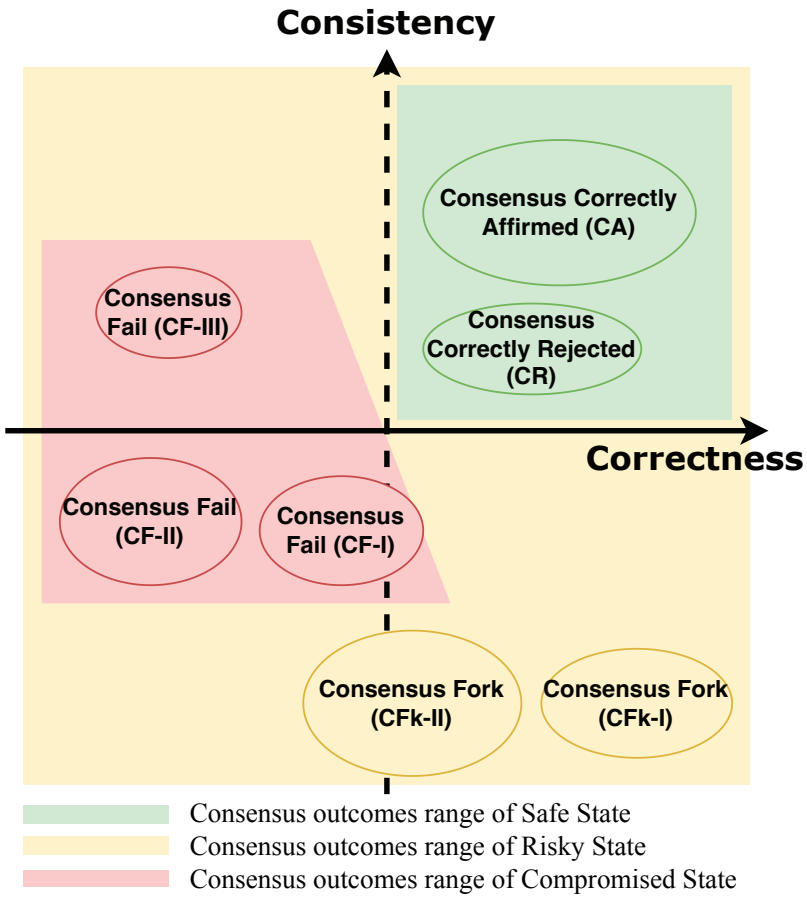

Figure 3.2: Consensus outcomes of probabilistic consensus in nondeterministic autonomous system

1. **Consensus Correctly Affirmed (CA)** is satisfied if: (1) all reliable nodes reach a consistent consensus on the same valid request, and (2) reliable nodes meet the consistency threshold of the network.

2. **Consensus Correctly Rejected (CR)** is satisfied if the invalid or inconsistent request is unanimously rejected by all reliable nodes, who also meet the consistency threshold of the network.

3. **Consensus Fork (CFk)** is satisfied if the reliable nodes meet the consistency threshold of the network but arrive at inconsistent conclusions.

4. **Consensus Fail (CF)** is satisfied if (1) Byzantine nodes exceed the required consistency threshold for controlling the consensus process, or (2) none of the reliable nodes affirm a valid consensus request, even though the number of reliable nodes surpasses the consistency threshold for consensus.

It is important to note that, apart from the CA and CR outcomes, which fully guarantee both correctness and consistency, CFk and CF exhibit varying behaviors in terms of correctness, as the correctness of the system cannot always be classified in a binary manner. For CFk, there are cases where the consensus outcomes of all reliable nodes are valid but inconsistent (CFk-I), as well as cases where the consensus outcomes are partially valid and partially invalid but inconsistent (CFk-II). The CF can be categorized into three scenarios in terms of correctness and consistency, as illustrated in Fig. 3.2. The first scenario (CF-I) occurs when most reliable nodes fail to reach consensus or reach an incorrect consensus outcome, while a small portion of reliable nodes correctly accept or reject the consensus request even in Byzaninte nodes are in majority control. The second scenario (CF-II) is when all nodes either obtain incorrect consensus outcomes or fail to reach consensus altogether. The third scenario (CF-III) is less common, where an incorrect but consistent consensus is reached under the influence of Byzantine nodes.

3.2.2 Rapid Estimation of Consensus Outcomes Using Consensus States

Non-determinism in probabilistic consensus largely comes from the uncertain relationship between reliable nodes ($n-F$), Byzantine nodes (F), the reliability quorum (Q_r), and the intersection quorum (Q_{int}). To better assess consensus outcomes, the concept of consensus states—namely safe, risky, and compromised, is introduced. These states enable rapid estimation of outcomes based on quorum and node configurations without requiring analysis of the full consensus process. This approach simplifies evaluation and provides a foundation for a more precise probability analysis of consensus outcomes.

Firstly, consensus outcomes can be determined by whether reliable nodes meet the consistency threshold, t. Typical cases $t \geq \lfloor \frac{n}{2} \rfloor + 1$ are taken for this detail analysis. Therefore, when $n - F < t$, the system will always result in CF. For all cases where $n - F \geq t$, the PBFT protocol requires each node to collect a number of valid and consistent messages that at least reach the threshold (Q_r). It can be easily observed that a CF outcome always occurs when $F < t \leq n - F < Q_r$. However, if $Q_r \leq F < t \leq n - F$, all four consensus outcomes are possible because the threshold Q_r for each node can be met either by a collection of valid messages or invalid messages, depending on what constitutes the first sufficient messages for a reliable node to make a judgment. Additionally, as discussed earlier, system-wide consensus requires $Q_r \geq Q_{int}$ to avoid the risk of inconsistent outcomes or consensus forks, CFk in this case becomes an unavoidable scenario. For cases where Q_r lies between F and $n - F$ when $t \leq n - F$, if $F < Q_{int} \leq Q_r \leq n - F$, consensus is always reliable, and CA and CR are the only possible outcomes. However, if $F < Q_r < \min(Q_{int}, n - F)$, CFk may occur. The results of

Table 3.2: Classification of consensus outcomes based on relationships between n, Q_r, t, and F

Consensus State	Consensus Outcome	Condition	Relation	Explanation
Safe State	CA, CR	$F < Q_{int} \leq Q_r \leq n - F$ $t \leq n - F$	Valid PBFT conditions.	Sufficient messages to tolerate F Byzantine nodes, ensuring safe results.
Risky State	CA, CR, CFk, CF	$F < Q_r < min(Q_{int}, n - F)$ $t \leq n - F$	Q_r is between reliable and Byzantine nodes but below the quorum.	Depends on whether the primary node sends an inconsistent request.
		$Q_r \leq F < t \leq n - F$	Q_r is less than or equal to both reliable and Byzantine nodes.	Depends on the first Q_r messages and primary node's behaviour.
Compromised State	CF	$F < t \leq n - F < Q_r$	Q_r exceeds both the number of reliable and Byzantine nodes.	Q_r always consists of a mix of reliable and Byzantine nodes.
		$n - F < t$	Reliable nodes cannot meet the consistency threshold.	Byzantine nodes control consensus.

Note: Q_r and F are non-negative integers where $0 \leq Q_r \leq n$, $0 \leq F \leq n$, $Q_{int} = \left\lceil \frac{n+F+1}{2} \right\rceil > F$, and $t \geq \left\lfloor \frac{n}{2} \right\rfloor + 1$.

this exploration are summarized in Table 3.2.

According to Table 3.2, before executing the consensus algorithm, the system's key parameters help narrow the range of possible outcomes. The conditions are categorized into three states: safe, risky, and compromised. The relationship of outcomes and the states is indicated in Fig. 3.2.

Safe State

A Safe State ensures all reliable nodes consistently reach a valid consensus or reject invalid requests, with their count exceeding the consistency threshold. In this state, consensus outcomes are always reliable, correct, and consistent. Specifically, CA and CR are the only possible outcomes. The Safe State corresponds to the relationship $F < Q_{int} \leq Q_r \leq n - F$ with $t \leq n - F$. Let $\mathcal{E}_S$ represent the event where the system achieves consensus in the Safe State. According to the relationship constraint, the probability of system maintaining Safe State, denoted as the **safe state probability** is derived as

$$P(\mathcal{E}_S \mid Q_r = q) = \sum_{f=0}^{n-t} P(\mathcal{E}_S \mid F = f, Q_r = q) P(F = f), \tag{3.7}$$

where $P(F = f)$ is defined in equation (3.1), and

$$P(\mathcal{E}_S \mid F = f, Q_r = q) = \begin{cases} 1 & \text{if } \left\lceil \frac{n+f+1}{2} \right\rceil \leq q \leq n - f, \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

Risky State

Risky State is a state when neither a valid consensus nor the rejection of an invalid request can be consistently ensured. However, since reliable nodes meet the consistency threshold, a correct consensus remains possible. This state arises when reliable nodes are the majority, but the reliability quorum is not set or cannot exceed the intersection quorum, leading to potential inconsistencies among reliable nodes. Notably, this state encompasses all possible consensus outcomes, corresponding to the conditions $F < Q_r < \min(Q_{int}, n - F)$, denoted as $\mathcal{E}_{R1}$ and $Q_r \leq F < t \leq n - F$ with $t \leq n - F$, denoted as $\mathcal{E}_{R2}$. The probability of the system being in the Risky State, denoted as the **risky state probability**, is

$$P(\mathcal{E}_{R1} \cup \mathcal{E}_{R2} \mid Q_r = q) = \sum_{f=0}^{n-t} P(\mathcal{E}_{R1} \cup \mathcal{E}_{R2} \mid F = f, Q_r = q) P(F = f), \tag{3.9}$$

where $P(F = f)$ is defined in equation (3.1), and

$$P(\mathcal{E}_{R1} \cup \mathcal{E}_{R2} | F = f, Q_r = q) = \begin{cases} 1 & \text{if } 0 \leq q < \left\lceil \frac{n+f+1}{2} \right\rceil, \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

Compromised State

Compromised State is a state when valid consensus is impossible, typically because reliable nodes fail to meet the consistency threshold, allowing Byzantine influence. It can also arise if the reliability quorum Q_r is set too high, preventing consensus and paralyzing decision-making. Compromised State only results in the CF outcome. The events in which the system enters the Compromised State are defined as $\mathcal{E}_{C1}$ when $F < t \leq n - F < Q_r$, and as $\mathcal{E}_{C2}$ when $n - F < t$. The probability that the system is in the Compromised State, referred to as the **compromised state probability**, can be derived as

$$P(\mathcal{E}_{C1} \cup \mathcal{E}_{C2} \mid Q_r = q) = \sum_{f=0}^{n-t} P(\mathcal{E}_{C1} \mid F = f, Q_r = q) \times P(F = f) + \sum_{f=n-t+1}^{n} P(F = f), \tag{3.11}$$

where $P(F = f)$ is defined in equation (3.1), and

$$P(\mathcal{E}_{C1} \mid F = f, Q_r = q) = \begin{cases} 1 & \text{if } q > n - f, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

3.2.3 Quantitative Analysis of Consensus Outcomes

Based on the rapid estimation enabled by the three consensus states, the detailed probability of each consensus outcome can be calculated.

Consensus Correctly Achieved Probability

Let the event that consensus has a CA outcome as $\mathcal{O}_{CA}$. The $P(\mathcal{O}_{\text{CA}} | Q_r = q)$ under different $Q_r = q$ is derived as

$$\begin{aligned} P(\mathcal{O}_{\text{CA}} | Q_r = q) = \sum_{f=0}^{n-t} [P(\mathcal{E}_S \cup \mathcal{E}_{R1} \mid F = f, Q_r = q) + P(\mathcal{E}_{R2} | F = f, Q_r = q) \\ \cdot P(\mathcal{O}_{\text{CA}} | \mathcal{E}_{R2} \cap (B_1 = 0) \cap (F = f) \cap (Q_r = q))] P((F = f) \cap (B_1 = 0)), \end{aligned} \tag{3.13}$$

where

$$P(\mathcal{E}_S \cup \mathcal{E}_{R1} | F = f, Q_r = q) = \begin{cases} 1 & \text{if } f < q \leq n - f, \\ 0 & \text{otherwise,} \end{cases} \tag{3.14}$$

$$P(\mathcal{E}_{R2} | F = f, Q_r = q) = \begin{cases} 1 & \text{if } 0 \leq q \leq f, \\ 0 & \text{otherwise,} \end{cases} \tag{3.15}$$

are the two indicator functions specify the valid ranges for $\mathcal{E}_S \cup \mathcal{E}_{R1}$ and $\mathcal{E}_{R2}$ respectively. $P((F = f) \cap (B_1 = 0))$ is provided in (3.3). $P(\mathcal{O}_{\text{CA}} \mid \mathcal{E}_{R2} \cap (B_1 = 0) \cap (F = f) \cap (Q_r = q))$ represents the condition in the Risky State $\mathcal{E}_{R2}$ where consensus can only be correctly affirmed if all reliable nodes consistently receive a sufficient number of valid messages before receiving enough invalid messages. Therefore,

$$\begin{aligned} &P(\mathcal{O}_{\text{CA}} | \mathcal{E}_{R2} \cap (B_1 = 0) \cap (F = f) \cap (Q_r = q)) \\ &= G_{\text{count}}(M_p = n - f, M_c = n - f, F = f, Q_r = q), \end{aligned} \tag{3.16}$$

where $G_{\text{count}}(M_p = m_p, M_c = m_c, F = f, Q_r = q)$ indicate the probability in $\mathcal{E}_{R2}$ with m_p alive nodes in prepare stage and m_c alive nodes in commit stages respectively. The detail of G_{count} is given in Appendix A.1.

Consensus Correctly Rejected Probability

Let the event that consensus has a CR outcome as $\mathcal{O}_{\text{CR}}$. As the safe state and the risky state result in CR when the primary node is the Byzantine node, the total probability of $\mathcal{O}_{\text{CR}}$ is given by

$$P(\mathcal{O}_{\text{CR}} | Q_r = q) = \sum_{f=0}^{n-t} [P(\mathcal{E}_S | f, q) + P(\mathcal{E}_{R1} \cap \mathcal{E}_{R2} | f, q)] \cdot P((F = f) \cap (B_1 = 1)), \tag{3.17}$$

where $P(\mathcal{E}_S | f, q)$, $P(\mathcal{E}_{R1} \cap \mathcal{E}_{R2} | f, q)$ and $P((F = f) \cap (B_1 = 1))$ are given in (3.12) (3.9) (3.4).

Consensus Fail Probability

Let the event that consensus has a CF outcome as $\mathcal{O}_{\text{CF}}$. As the compromised state always result in CF and the risky state can result in CF when the primary node is reliable but message exchange fails, the total probability of $\mathcal{O}_{\text{CF}}$ is

$$\begin{aligned} &P(\mathcal{O}_{\text{CF}} | Q_r = q) = P(\mathcal{E}_{C1} \cap \mathcal{E}_{C2} | Q_r = q) \\ &+ \sum_{f=0}^{n-t} P(\mathcal{O}_{\text{CF}} | \mathcal{E}_{R2} \cap (B_1 = 0) \cap (F = f) \cap (Q_r = q)) \cdot P((F = f) \cap (B_1 = 0)), \end{aligned} \tag{3.18}$$

where $P(\mathcal{E}_{C1} \cap \mathcal{E}_{C2} | Q_r = q)$ and $P((F = f) \cap (B_1 = 0))$ is given in (3.11) and (3.3). $P(\mathcal{O}_{\text{CF}} | \mathcal{E}_{R2} \cap (B_1 = 0) \cap (F = f) \cap (Q_r = q))$ is derived as primary node is reliable but all the reliable nodes fail in event $\mathcal{E}_{R2}$,

$$P(\mathcal{O}_{\text{CF}} | \mathcal{E}_{R2} \cap (B_1 = 0) \cap (F = f) \cap (Q_r = q)) = G_{\text{count}}(M_p = m_p, M_c = 0, F = f, Q_r = q). \tag{3.19}$$

Consensus Fork Probability

Let the event that consensus has a CFk outcome as $\mathcal{O}_{\text{CFk}}$. The $P(\mathcal{O}_{\text{CFk}} | Q_r = q)$ is

$$P(\mathcal{O}_{\text{CFk}} | Q_r = q) = 1 - P(E_{CA} | Q_r = q) - P(\mathcal{O}_{\text{CR}} | Q_r = q) - P(\mathcal{O}_{\text{CF}} | Q_r = q). \tag{3.20}$$

3.3 Quantitative Analysis of Reliability Quorum

From Table 3.2, it can be observed that how the reliability quorum Q_r impacts consensus outcomes and states. To design a high-reliability distributed consensus under probabilistic model, this section quantitatively analyses the optimal Q_r settings for robust consensus. The optimal Q_r can be determined by balancing the probabilities of different consensus outcomes and states, with variations depending on the chosen optimization strategy. An example is first presented to demonstrate the optimization of Q_r through the assignment of different weights to consensus outcomes. Subsequently, a theoretical proof is provided showing that the commonly used intersection quorum $(2f_{\max} + 1)$, under the setting $n = 3f_{\max} + 1$, remains a strong choice for distributed consensus within the probabilistic model.

3.3.1 Balancing Consensus Outcomes for an Optimized Reliability Quorum

Determining the optimal reliability quorum requires balancing correctness, consistency, and efficiency based on system requirements. An example approach is to assign different weights to consensus outcomes, tailoring the quorum selection to specific application needs. An objective function incorporating weighted probabilities of all possible outcomes is defined as

$$\begin{aligned} q_{a,\text{opt}} = \arg\max_{q} \Big[& w_1 P(\mathcal{O}_{CA} \mid Q_r = q) + w_2 P(\mathcal{O}_{CR} \mid Q_r = q) \\ & - w_3 P(\mathcal{O}_{CFk} \mid Q_r = q) - w_4 P(\mathcal{O}_{CF} \mid Q_r = q) \Big], \end{aligned} \tag{3.21}$$

where w_1, w_2, w_3, w_4 denote the weights assigned to the outcomes CA, CR, CFk, and CF, respectively. These weights represent the system's priorities by quantifying the relative importance of different consensus outcomes. For safety-critical applications such as autonomous systems, minimizing consensus forks is crucial, since conflicting decisions can lead to serious consequences. To reduce this risk, a higher CFk penalty (w_3) can be set to optimize the reliability quorum. For example, setting $w_1 = w_2 = 1, w_3 = 5, w_4 = 1$ prioritizes consistency by strongly discouraging forks. This emphasis on stability helps avoid conflicting states that would compromise the safety of the system. Conversely, efficiency-focused systems (e.g., non-safety-critical data synchronization) may prioritize achieving consensus quickly. Assigning a higher weight to CA (w_1) while moderating penalties for other outcomes (e.g., $w_1 = 3, w_2 = 1, w_3 = 1, w_4 = 1$) optimizes for consensus success. This approach enables flexible quorum selection tailored to specific reliability and efficiency trade-offs. Several cases with different weight choice are provided in Sec. 4.4.5.

3.3.2 Intersection Quorum Remain a Safe Choice

This section proves that the optimal reliability quorum coincides with the intersection quorum $(2f_{\max} + 1)$ under the condition $n = 3f_{\max} + 1$, when adopting the strategy of maximizing the safe state probability. Specifically, $q_{s,\text{opt}}$ is defined as the value of Q_r that maximizes $P(\mathcal{E}_S \mid Q_r = q)$.

$$\begin{aligned} q_{s,opt} &= \arg\max_{q} P(\mathcal{E}_S | Q_r = q) \\ \text{s.q.} \quad & q \in \{0, 1, 2, \ldots, n\} \end{aligned} \tag{3.22}$$

Proposition 1. *For BFT-like consensus protocol, the optimal reliability quorum q_{opt} for maximizing Safe State reliability is*

$$q_{s,opt} = \begin{cases} \frac{2n}{3} \text{ or } \frac{2n}{3} + 1, & \text{if } n \bmod 3 = 0, \\ \frac{2n+1}{3}, & \text{if } n \bmod 3 = 1, \\ \frac{2n+2}{3}, & \text{if } n \bmod 3 = 2. \end{cases} \tag{3.23}$$

Proof. According to (3.9), $P(\mathcal{E}_S \mid Q_r = q)$ is a sum of $P(\mathcal{E}_S, F = i \mid Q_r = q)P(F = i)$, where $P(\mathcal{E}_S, F = i \mid Q_r = q)$ is an indicator function. Each $P(\mathcal{E}_S, F = i \mid Q_r = q)$ represents a scenario where, for each value of $F = i$, the condition $Q_{p_i} \le Q_r \le n - F_i$ is satisfied.

The corresponding parameters is denoted as F_i, Q_{p_i}, and $n - F_i$ when $F = i$. For $i < j$, the relationships $F_i < F_j$, $Q_{p_i} < Q_{p_j}$, and $n - F_i > n - F_j$ always hold. Therefore, if (3.12) always yields $P(\mathcal{E}_S \mid F = i, Q_r = q) = 1$ whenever $Q_{p_i} \le Q_r \le n - F_i$, and $P(\mathcal{E}_S \mid F = i, Q_r = q) = 0$ otherwise, the effective range for Q_r of each summation term $P(\mathcal{E}_S, F = i \mid Q_r = q)P(F =$

i) extends outward, independent of the value of $P(F = i)$. This effective interval always encompasses the narrowest interval. When $n \bmod 3 = 1$ or 2, the narrowest interval lies between $[Q_{p_i}, n - F_i]$, where $Q_{p_i} = n - F_i$ holds. In this case, $q_{s,opt} = \arg\max_t P(\mathcal{E}_S \mid Q_r = q)$ when $q_{s,opt} = Q_{p_i} = n - F_i$. When $n \bmod 3 = 0$, the narrowest interval lies between $[Q_{p_i}, n - F_i]$, where $Q_{p_i} + 1 = n - F_i$ holds. In this case, $q_{s,opt} = \arg\max_t P(\mathcal{E}_S \mid Q_r = q)$ when $q_{s,opt} = Q_{p_i}$ or $n - F_i$. Therefore, (3.23) is proved.

□

This alignment of $q_{s,\text{opt}}$ with the intersection quorum arises because the intersection rule inherently ensures absolute consistency. Thus, under certain conditions, the BFT quorum selection ($n \leq 3f + 1$) is not only optimal for its Byzantine fault tolerance assumptions but also remains effective for maximizing the safe state probability under more relaxed Byzantine node assumptions.

3.4 Probabilistic Model for Wireless Distributed Consensus (WDC)

While our probabilistic modelling of node behaviour offers a significant step towards more accurate reliability analysis, real-world distributed systems often contend with an additional, independent, and equally critical source of unreliability: the communication medium itself. This is particularly true for many modern applications where wireless communication is indispensable yet inherently unreliable.

WDC specifically refers to distributed consensus mechanisms operating where inter-node communication primarily occurs over wireless channels. This is critical because, unlike more stable wired connections, wireless links are susceptible to signal fading, interference, path loss, and intermittent connectivity. These factors frequently result in message loss, increased latency, and even complete link disruptions, introducing a significant layer of uncertainty that profoundly impacts consensus achievement.

In WDC, the underlying consensus protocol, such as PBFT, must operate robustly despite these unreliable wireless links. As illustrated in Fig. 3.3, PBFT consensus relies heavily on frequent, multi-phase inter-node communications (Pre-prepare, Prepare, Commit, Reply, and Sync). The success of each phase, and thus the overall consensus, is directly tied to the reliability of these numerous wireless message exchanges. For example, in scenarios like cooperative autonomous driving, where backup sensors (potentially faulty or "Byzantine-like" due to perception errors) provide conflicting readings, the PBFT quorum must make a Byzantine-proof decision requiring reliable communication even when facing misinformation. While

traditional PBFT tolerates less than $1/3$ Byzantine nodes, its performance and safety in a wireless environment are further compounded by unreliable links, which can mimic or exacerbate node failures from a communication perspective.

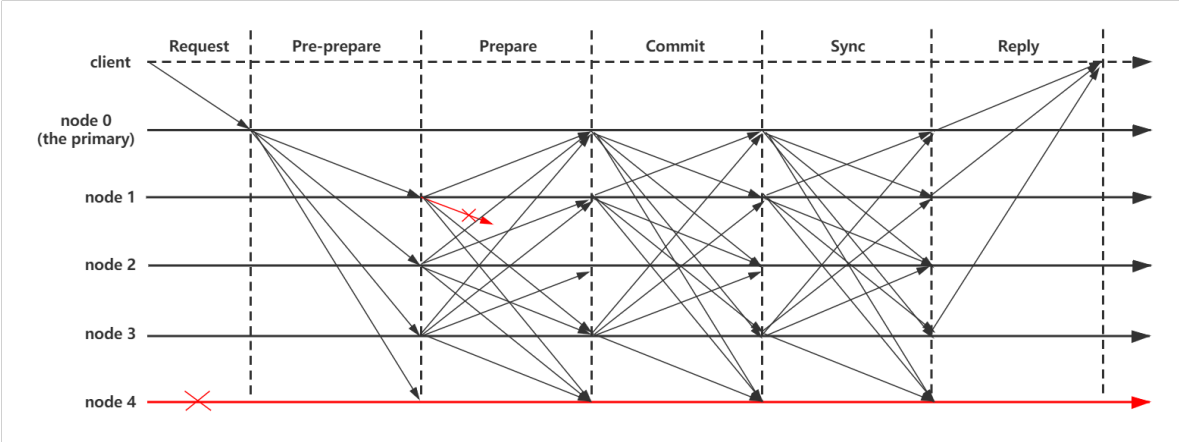

Figure 3.3: PBFT-based wireless distributed consensus

In this section, the reliability of WDC based on PBFT consensus under non-perfect communication links and nodes is provided. The reliability of such a system is critical for mission-critical decision-making, where the process consists of many components, subsystems, and external elements, e.g., computing nodes, physical connectors, and wireless channel quality for wireless connected systems. Specifically, in WDC, the following three cases are considered, node failure only, communication link failure only, and both node and communication link failure. A node failure is described as Byzantine-like behaviour in PBFT in probability, and the communication link reliability is defined as the statistical probability of success of the point-to-point wireless communication at the given time, which is also a simplified value for a set of network and traffic environment with considerations on channel modelling and interference. As a general assumption, each **node's reliability** P_n is considered under a node failure model with a given total of n nodes, along with the **communication link probability of success** P_l for each wireless channel between nodes. It is assumed that the leader node is always available at the first round of communication for reliability analysis for simplicity, upon the assumption that this very node is the initiator of the consensus.

3.4.1 Consensus reliability based on PBFT

PBFT consensus system provides safety and liveness against malicious attacks up to $f = \lfloor \frac{n-1}{3} \rfloor$ faulty nodes [29], where f is the number of faulty nodes, among number of n nodes. According to the consensus requirements of the original PBFT protocol, the number of nodes that successfully participated in all consensus phases should not be less than $n - f$ [29]. The

consensus reliability model for PBFT consists of PBFT node failure (P-N) model, PBFT link failure (P-L) model, and PBFT node and link failure (P-N-L) model, respectively, detailed in the following subsections.

Theorem 1. *Given P_n, P_l and n, the probability of success of the DC P_{P-N-L} can be obtained by*

$$\begin{aligned} P_{P-N-L} = \sum_{m=m_{pp}}^{n} \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} (P_{node}(n,m) \cdot \\ P_{pp}(m,m_{pp}) \cdot P_p(m_{pp},m_p) \cdot P_c(m_p,m_c)), \end{aligned} \tag{3.24}$$

where $P_{node}(n,m)$, $P_{pp}(m,m_{pp})$, $P_p(m_{pp},m_p)$ and $P_c(m_p,m_c)$ are given below.

$$P_{node}(a,b) = \binom{a-1}{b-1} P_n^{b-1} (1-P_n)^{a-b} \tag{3.25}$$

$$P_{pp}(a,b) = \binom{a-1}{b-1} P_l^{b-1} (1-P_l)^{a-b} \tag{3.26}$$

$$P_c(a,b) = \binom{a}{b} P_s(a)^b (1-P_s(a))^{a-b} \tag{3.27}$$

$$\begin{aligned} P_p(a,b) =& P_s(a) \binom{a-1}{b-1} P_s(a-1)^{b-1} (1-P_s(a-1))^{a-b} \\ &+ (1-P_s(a)) \binom{a-1}{b} \times P_s(a-1)^b (1-P_s(a-1))^{a-1-b} \end{aligned} \tag{3.28}$$

P_s in equation (3.27) and (3.28) are denoting

$$P_s(a) = \sum_{k=2f}^{a-1} \binom{a-1}{k} P_l^k (1-P_l)^{a-1-k}. \tag{3.29}$$

PBFT Model with Node and Link Failure (P-N-L)

In this model, node failure (i.e., $1-P_n$) and link failure (i.e., $1-P_l$) are considered at the same time. The following theorem is provided to show the relationship between them and the consensus reliability P_{P-N-L}, as seen in Theorem 1.

Theorem 1 provides a precise equation of the overall reliability of the PBFT system in the real world, where the reliability of nodes or communication links is not guaranteed. Derived by successive summation and multiplication in Theorem 1, the computational complexity of the P-N-L model is $O(n^5)$. The high computational complexity is justified as the derivation equations are precise, and the computational complexity cannot be reduced as the accurate equations cannot be optimized. The proof of Theorem 1 is given in *Appendix A.2*.

PBFT Model with Node Failure (P-N)

In the P-N model, only the node failure rate P_n is considered. In contrast, the P-N-L model assumes that communication channels within the consensus system are always reliable when $P_l = 1$. By substituting $P_l = 1$ into equation (3.24), the following remark establishes the relationship among the node reliability P_n, the number of nodes n, and the WDC reliability.

Remark 1. *Given P_n, n and $P_l = 1$, the probability of successful consensus of the system P_{P-N} can be obtained by*

$$P_{P-N} = \sum_{m=n-f}^{n} P_{node}(n,m). \tag{3.30}$$

The expression of $P_{node}(n,m)$ is denoted in equation (3.25) in Theorem 1.

Remark 1 provides a straightforward answer to node failure mode for PBFT consensus, and it is useful to estimate the reliability of the system while the communication link is stable, e.g., wired connected scenario or when sufficient spectrum resource is used in wireless communications (e.g., repeat transmission).

PBFT Model with Link Failure (P-L)

For P-L model, it is also a special case that all the nodes are reliable in P-N-L model. By taking $P_n = 1$, the relation between link probability of success P_l, the number of nodes in the system n and the consensus rate of P-L model P_{P-L} is illustrated.

Remark 2. *Given P_l, n and $P_n = 1$, the probability of successful consensus of the system can be calculated in the following equation:*

$$\begin{aligned} P_{P-L} = \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} [P_{pp}(m,m_{pp}) \cdot \\ P_p(m_{pp},m_p) \cdot P_c(m_p,m_c)]. \end{aligned} \tag{3.31}$$

The expressions of $P_{pp}(m,m_{pp})$, $P_p(m_{pp},m_p)$ and $P_c(m_p,m_c)$ are denoted in equation (3.26), equation (3.28) and equation (3.27) in Theorem 1.

The analytical and simulated results of the relationship of consensus failure rates $1 - P_{P-N}$, $1 - P_{P-L}$ or $1 - P_{P-N-L}$ and total number of nodes n are detailed in Section 3.5.4, it provides a theoretical mitigation on weak nodes, weak communication links or combined scenarios. Note that in the case of perfect communication links or nodes, Remark 1 and 2 are the special cases of Theorem 1.

3.4.2 Reliability of Full Consensus with Synchronization

A complete and successful round of consensus requires all non-faulty nodes to sync up to actuate the outcome of consensus. However, the faulty nodes of the current round will be left out and prohibited from entering the next consensus round. Hence, it is important to sync up the previously failed nodes to maintain the liveness of the whole system, shown as Sync in Fig. 3.3. To achieve full consensus, a synchronization phase is added to help all the nodes who failed due to link failures to update the latest log from the successful nodes and ready them for future requests. To extend the protocol, an alive-node broadcast phase, similar to the original *commit* phase, is added as *sync* phase to PBFT. The following is a detailed description of the *sync* phase in PBFT.

In the case of PBFT, for the nodes which do not experience any node or communication link failures during the consensus process, they enter the synchronization phase by multicasting synchronization messages *sync* to all other nodes. When a node receives *sync* messages, it will check if it has both $prepare$ and $commit$ certificates with the same view number, sequence number and request digest as the synchronization message provided. If the request in the synchronization message has not been committed, it accepts the message and waits for a weak certificate via *sync* consensus, which requires fewer message counts than the normal consensus process with at least $f+1$ *sync* messages with the same view, sequence number and request's digest from different nodes. Otherwise, the node remains unchanged. This weak certificate is named as the synchronized certificate. Nodes with this certificate execute the request and update their logs without replying to the client. Similar to the reply certificate in PBFT model, the synchronized certificate with $f+1$ messages from different nodes aims to ensure the synchronization operation is valid since there is at least one reliable message which indicates that the request has been accepted by a quorum. To calculate the reliability of full consensus with synchronization, the following remarks show the probability of successful consensus in the P-N-L model with *sync* phase.

Remark 3. *Given P_n, P_l and n, the probability of success of the DC with sync phase in P-N-L model can be calculated by*

$$\begin{aligned} P_{P-N-L} = \sum_{m=m_{pp}}^{n} \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} [P_{node}(n,m)\cdot \\ P_{pp}(m,m_{pp}) \cdot P_p(m_{pp},m_p) \cdot P_c(m_p,m_c) \cdot P_{syn}(m,m_c)], \end{aligned} \tag{3.32}$$

where $P_{node}(n,m)$, $P_{pp}(m,m_{pp})$, $P_p(m_{pp},m_p)$ and $P_c(m_p,m_c)$ are provided in Theorem 1

while $P_{syn}(m, m_{pp})$ is given below.

$$P_{syn}(a,b) = (\sum_{k=f+1}^{b} \binom{b}{k} P_l^k (1-P_l)^{b-k})^{a-b} \tag{3.33}$$

Added *sync* phase adds additional requirements to the consensus completion, which means synchronization has a negative impact on probability of successful consensus. The analysis of the impact is described in Section 3.5.4.

By defining the probability of success of consensus synchronizations, the overall probability of success can be concluded. Note that, in all scenarios, when one of the components (node or link) reliability reaches 1 or failure rate reaches 0, it matches up with the conclusion stated in Remark 3.

3.5 Numerical Results Analysis

This section validates the theoretical analysis presented in Sections 3.2.2 and 3.2.3 through a series of Monte Carlo simulations. These simulations are designed to accurately model the consensus process and provide a statistical estimation of the system's performance across various typical scenarios. For each scenario, we performed iterations to ensure statistical significance, with key parameters such as the number of replica nodes (N) and Byzantine nodes (f) being systematically varied.

The core of our simulation models a PBFT-based consensus protocol, with a specific focus on the prepare phase. To reflect the standard PBFT requirements and the integrity of the protocol, our simulation incorporates a critical condition: a replica node cannot transition to the next state without receiving confirmations from at least Q_r other nodes, where we set the threshold to $Q_r \geq 3$. This condition is strictly enforced throughout all simulated scenarios.

By repeatedly executing the consensus process, we were able to statistically estimate the probability of the system being in each consensus state and the probability of achieving each consensus outcome. Furthermore, the simulations were used to examine the performance of our proposed optimal reliability arbitration parameter, $q_{a,\text{opt}}$, under different weighting strategies and in the presence of varying Byzantine behaviours. The results and detailed analysis of these simulations will be presented in the following subsections.

3.5.1 Consensus State Probability Analysis

The probabilities of the three consensus states: safe, risky, and compromised are illustrated. In Fig. 3.4 and Fig. 3.5, the reliability of Byzantine and reliable nodes is modelled using dif-

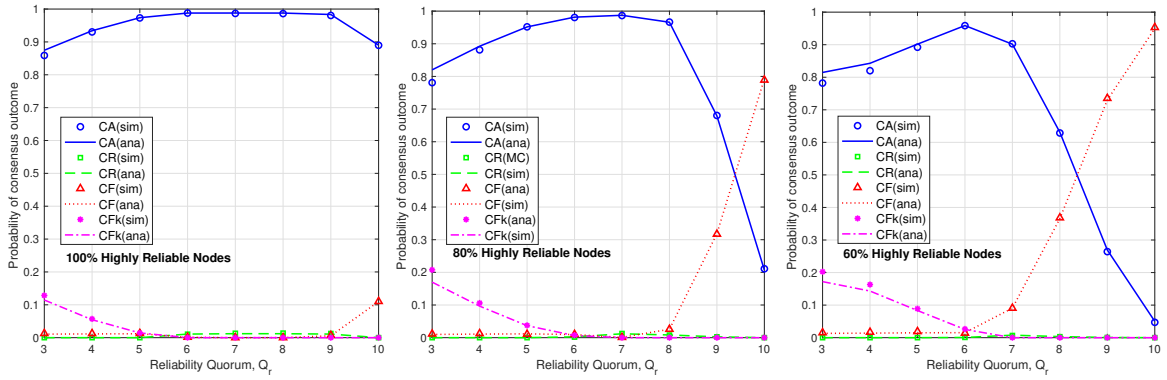

Figure 3.4: Consensus outcome probabilities with varying byzantine node proportion. ($n = 10$, highly reliable nodes: $\mu_{P_B} = 0.01, \sigma_{P_B} = 0.005$, highly Byzantine nodes: $\mu_{P_B} = 0.5, \sigma_{P_B} = 0.05$)

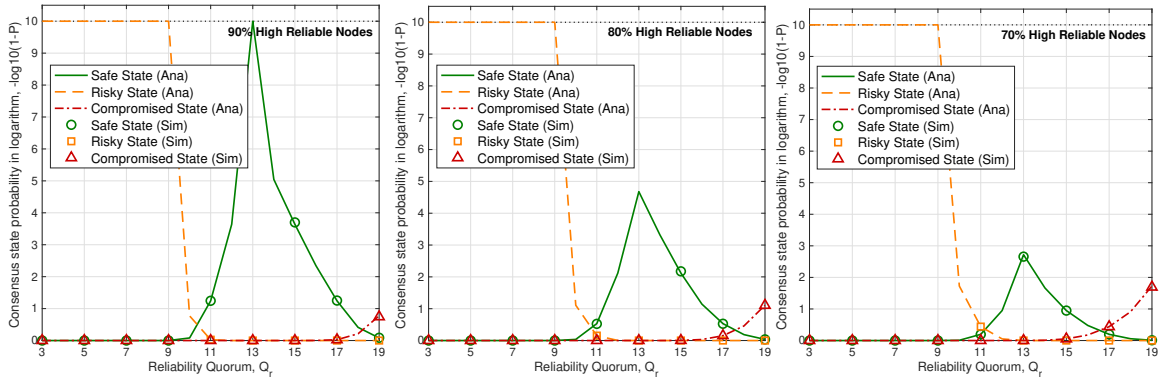

Figure 3.5: Consensus state probabilities with varying byzantine node proportion. ($n = 19$, highly reliable nodes: $\mu_{P_B} = 0.01, \sigma_{P_B} = 0.005$, highly Byzantine nodes: $\mu_{P_B} = 0.5, \sigma_{P_B} = 0.05$)

ferent values of the mean (μ_{P_B}) and variance (σ_{P_B}) of their failure probability distributions. The highly reliable nodes are assigned lower mean values ($\mu_{P_B} = 0.01, \sigma_{P_B} = 0.005$) with relatively small variance to reflect their generally stable and reliable behaviour, while highly Byzantine nodes are assigned larger mean values ($\mu_{P_B} = 0.5, \sigma_{P_B} = 0.05$) with larger variance to capture the uncertainty and unpredictability of adversarial behaviour. These choices are consistent with the modelling assumptions commonly adopted in probabilistic reliability studies of distributed consensus and wireless systems [13, 74, 77], where reliable nodes are considered predominantly reliable and Byzantine nodes exhibit much lower probability of correct operation. Three scenarios are analysed: (1) 90% highly reliable nodes and 10% highly Byzantine nodes, (2) 80% highly reliable nodes and 20% highly Byzantine nodes, and (3) 70% highly reliable nodes and 30% highly Byzantine nodes. The simulation results closely align with the analytical results, confirming the accuracy of the derived probability expressions for consensus states (in Sec. 3.2.2).

As shown Fig. 3.5, an increasing proportion of Byzantine nodes reduces the probability of reaching a safe state. However, the highest safe state probability is achieved at the optimal reliability quorum, $q_{s,opt} = \frac{2n+1}{3} = 13$, verifying Proposition 1. At this point, the probabilities of risky and compromised states remain minimal. Additionally, state transitions become more pronounced as the reliability quorum (q) deviates from $q_{s,opt}$. A lower q results in a significantly higher probability of the risky state, while an excessively high q increases the likelihood of the compromised state. This analysis highlights that consensus state reliability is highly dependent on node composition, but appropriate quorum selection can enhance system stability and maintain a reliable safe state.

3.5.2 Consensus Outcome Probability Analysis

The probabilities of the four consensus outcomes, namely CA, CR, CFk, and CF, are illustrated in Fig. 3.4. These outcome probabilities are compared across three reliability scenarios in a network consisting of $n = 10$ nodes, where the proportion of highly reliable nodes to highly Byzantine nodes is varied. Highly reliable nodes are characterized by a low Byzantine failure rate ($\mu_{P_B} = 0.01, \sigma_{P_B} = 0.005$), while highly Byzantine nodes exhibit a significantly higher failure rate ($\mu_{P_B} = 0.5, \sigma_{P_B} = 0.05$). Three specific configurations are examined in detail. First, all nodes are highly reliable. This setup reflects a highly reliable, homogeneous system, where the likelihood of Byzantine behaviour is minimal. In the second, 80% are highly reliable and 20% highly Byzantine, and in the third, 60% are highly reliable and 40% highly Byzantine. These two systems simulate a realistic, heterogeneous environment where both trustworthy and malicious nodes coexist. As shown in Fig. 3.4, increasing the propor-

tion of Byzantine nodes significantly reduces the probability of reaching CA, since a larger number of highly Byzantine nodes introduces greater disruption into the negotiation process.

Meanwhile, the probability of CFk becomes more prominent when the reliability quorum Q_r is set too low, whereas CF is more pronounced when Q_r is set too high. This aligns with the intuition that an excessively small Q_r triggers more inconsistencies (partial agreement among some but not all reliable nodes). In contrast, an overly large Q_r makes it difficult for any node to collect enough valid confirmations, thus exacerbating CF. CR occurs more frequently only if the primary node happens to be Byzantine; in scenarios where most nodes are reliable, the CR probability remains relatively low. Our simulation results closely match the theoretical curves, confirming the accuracy of the derived outcome probabilities. Moreover, these findings underscore that the precise choice of Q_r and the system's node composition can significantly affect each of the four consensus outcomes.

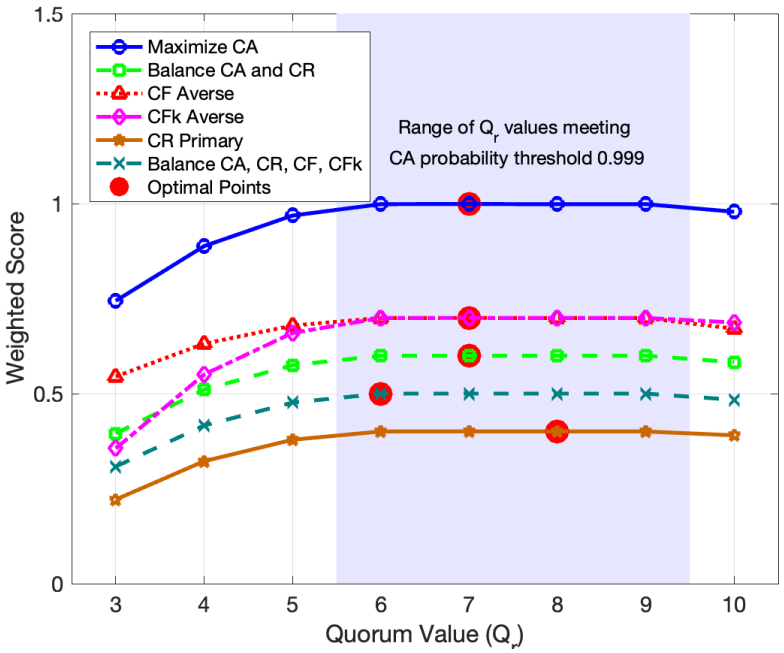

Figure 3.6: Reliability quorum selection based on weights, $n = 10$

3.5.3 Reliability Quorum Results under Weighted Priorities

The impact of different priority settings on the selection of Q_r is evaluated in a system comprising $n = 10$ nodes, each with a failure probability of $p_{B,i} = 0.001$. For each candidate value of Q_r, the probabilities of four consensus outcomes (CA, CR, CF, and CFk) are computed. Since achieving CA is often a primary objective in consensus systems, any value of Q_r that results in a CA probability below 0.999 is excluded to ensure a high likelihood of valid consensus. Among the remaining candidates, a weighted score is calculated using equation (3.21),

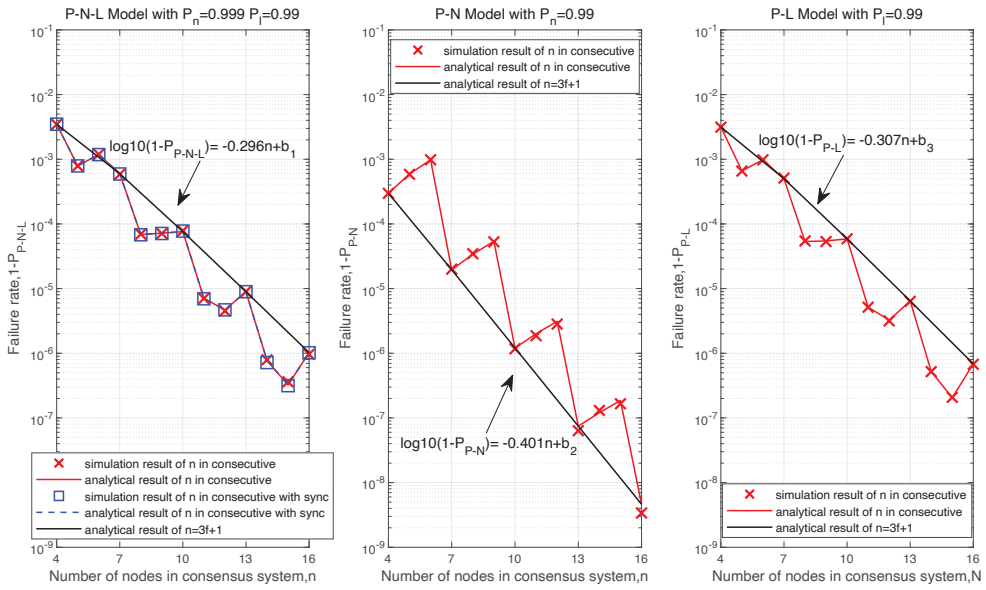

Figure 3.7: Reliability performance of PBFT consensus with combined failure rate (a: left), node failure rate (b: middle), link failure rate (c: right)

under six different weight configurations. They are Maximize CA $(1.0, 0.0, -1.0, -1.0)$, Balance CA and CR $(0.6, 0.4, -1.0, -1.0)$, CF Averse $(0.7, 0.0, -2.0, -0.5)$, CFk Averse $(0.7, 0.0, -0.5, -2.0)$, CR Primary $(0.4, 0.6, -0.5, -1.0)$, Balance CA, CR, CF, CFk $(0.5, 0.5, -1.0, -1.0)$. Figure 3.6 plots these weighted scores against Q_r, with each scenario shown as a separate curve and red markers denoting the optimal Q_r that achieves the highest overall score while maintaining CA above 0.999. As shown, even though different priorities shift the peak location, most scenarios favour a moderately high Q_r. In safety-critical settings (e.g., Maximize CA or CF/CFk Averse), the optimal Q_r tends to be higher, ensuring a more stringent level of consensus correctness. Overall, these results demonstrate selecting the quorum that best balances correctness, consistency, and efficiency for each application context.

3.5.4 Simulations of WDC Reliability

To verify Theorem 1 and it special cases of Remarks 1 and 2. Successive sets of values for n, P_n, and P_l are configured to evaluate the performance of each model and to illustrate the relationship between the consensus failure rate and the number of nodes. The results for PBFT are shown in Fig. 3.7.

In the P-N model, it is assumed that a failed node does not respond in any phase of the consensus process. The outcome of each simulated consensus instance is determined by verifying whether the number of failed nodes exceeds the upper limit $\lfloor \frac{n-1}{3} \rfloor$, which is the fault

tolerance threshold of PBFT. The consensus failure rate, $1 - P_{P-N}$ can be seen in Fig. 3.7 (b), where the reliability of PBFT nodes are assumed to be 0.99 and 0.9 respectively (a realistic assumption for the consumer-grade product, e.g., iPhone 6 in 2017 has been reported with less than 700 hours Mean Time Between Failures (MTBF) of non-self recoverable failures, and less than 10 hours of MTBF for temporarily occurred glitches [132]). The proven remarks of P-N can be used in a fast validation of WDC when the wireless links are not involved.

In P-L model, a link failure leads to the failure in the corresponding communication phase, and only the live nodes enter the next round of consensus, as illustrated Remark 2. According to the behaviour of the PBFT at each step of consensus processes, each link is applied with a uniformly distributed random number to simulate the uncertainty of every transmission. The number of valid messages is counted for every live node in each phase. If the number of the live nodes after the commit phase is more than $n - f$, the consensus process is successful. In the simulation, $P_l = 0.99$ is set for PBFT, as seen in Fig. 3.7 (c). Note that the P_l values are carefully crafted for a comparable WDC reliability range according to earlier results of P-N model instead of the consistent values from P-N-L simulation. The P-L model can be used in a fast validation of WDC when the nodes are considered reliable, which is particularly useful for transient WDC evaluations that only involve a small amount of time. It can also be deduced that transient evaluations are not sensitive to node reliability but wireless link reliability.

To build the simulation of P-N-L model, the same method is leveraged as before and combine P-N and P-L models. The relationship between the number of nodes n in WDC and the consensus failure rate in P-N-L model $1 - P_{P-N-L}$ is shown in Fig. 3.7 (a). In these two figures, $P_l = 0.99$ and $P_n = 0.999$ are set for P-N-L model. The results are of significant importance to the practical system design for two reasons. Firstly, given the communication network or node reliability, the size of the consensus network can be adjusted to achieve the required consensus reliability in different application scenarios. Moreover, the confident correlation between analytical and simulated results guides the future distributed network deployment where less reliable COTS and wireless connection may be adopted for high-reliability applications.

It can be seen from Fig. 3.7 that the simulation and analysis results match each other, indicating that the analysis of node and link failure by each model is reliable. It's worth noting that the number of nodes in different cases are grouped by a consistent upper boundary (the worst case). The zigzag shape of red lines are induced by the different remainder of security threshold, for instance, $n = 3f + 1$, $n = 3f + 2$ and $n = 3f + 3$ for PBFT. The number of faulty nodes cannot be divided into consecutive integers, which leads to the discontinuity in the trend of the failure rate of the consensus. According to the relation of the security threshold

f and the total number of nodes n in each group with the same remainder, it can be observed that the proportion of f in n grows as n increases, which means the faulty tolerance of the consensus system increases as the total number of nodes increases. This matches the tendency of the case in Fig. 3.7, i.e., the failure rate of a consensus process decreases gradually as the number of nodes increases. Moreover, the group with $n = 3f + 1$ in Fig. 3.7 is indicated, as the number of nodes shows a linear relationship with the consensus failure rate in log scale.

To analyse the impact of *sync* on the reliability of consensus systems, the analytical results by applying equation (3.32) in Remarks 3 is shown, with a set of simulations. To compare the result of scenarios with *sync* phase with P-N-L, the same P_n, P_l are set, and n as in the original P-N-L model and plot a set of analysis and simulation results in Fig. 3.7 (a) using lines coloured in blue.

Comparing the lines of cases with and without *sync* phase, it is clear that the final probability of successful consensus of P-N-L does not significantly decrease. As the *sync* phase shown in Fig. 3.3 and protocol details described in Section 3.4.2, all live nodes broadcast to help sync up in PBFT case to yield a high chance of success for each link-failed node sync up.

3.6 Case Study: Autonomous Systems

The case of autonomous systems is used as an example to analyse situations where the probabilistic model is suitable for explicit consensus requirements. Autonomous systems, such as robotic swarms, connected and autonomous vehicle (CAV) networks, and underwater exploration systems, operate in complex environments with sensor noise, communication delays, and entity failures. Consensus in these contexts facilitates cooperation and improves decision accuracy by integrating diverse perspectives and data. Deterministic consensus, with its fixed fault assumptions, can be ineffective in adapting to uncertainty and rapid system changes, risking exceeding fault-tolerance limits. In contrast, the probabilistic model offers a more flexible and efficient solution in following ways.

The the probabilistic model can adjust consensus criteria based on dynamic changes in the consensus network. For example, on the network side, issues such as physical obstructions or distance may weaken or block wireless communication [133] [134], and the probabilistic model can adapt to changes in communication quality by dynamically adjusting consensus thresholds when safety is not a critical concern. On the node side, nodes may unpredictably join or leave the network, altering its topology [135], and the probabilistic model can adjust consensus conditions based on different network structures and evaluate node reliability through probabilistic models to ensure consensus stability.

The the probabilistic model consensus framework can also offer different types of consensus strategies based on varying needs. Taking the CAV network as an example, for non-safety-critical tasks (such as route optimization or traffic flow coordination), an efficiency-oriented consensus strategy [136] can be adopted to improve overall system efficiency, even if suboptimal or incorrect decisions may be reached in some cases. For safety-critical tasks (such as emergency obstacle avoidance or collision prevention), a safety-oriented consensus strategy must be employed, even if it requires longer decision-making times or additional communication overhead[137].

Furthermore, the the probabilistic model model in this paper can assign different reliability values to each node, thereby better addressing the heterogeneity of nodes in consensus systems. Nodes may differ in their ability to perceive changing conditions accurately, respond in a timely manner, and process complex information, leading to varying levels of reliability [138]. For example, in CAV systems, vehicles may have different levels of autonomous driving capabilities, ranging from fully automation to those with limited automation [139]. This diversity in capabilities highlights the importance of the ability in the probabilistic model to adapt to node reliability changes and support diverse reliability levels in decision-making. On the other hand, the introduction of AI increases the complexity of consensus in autonomous systems [140]. While AI enhances decision-making and adaptability, it also introduces more diverse consensus strategies [20]. Entities with varying intelligence levels may behave unpredictably, adding to consensus complexity. The node heterogeneity fault-tolerance feature offers a reliability-based consensus framework that ensures adaptability and resilience in dynamic environments.

Here several autonomous system scenarios are presented that have a clear consensus demand and their features different from traditional distributed consensus.

3.6.1 Consensus in Vehicle to Vehicle Network

Although current autonomous driving primarily relies on individual vehicles making autonomous judgments and decisions, with the advancement of vehicle-to-everything (V2X) technology, it is becoming increasingly promising for vehicles to connect with each other and make collective decisions on certain actions [13] [141]. For example, decisions related to lane changes, speed adjustments, or route planning could be more efficient when coordinated among multiple vehicles rather than being handled by a single vehicle. Such collective decision-making demands cannot be met without a robust consensus mechanism. It is worth noting that the consensus process faces challenges due to the high mobility of vehicles and the dynamic changes in network composition as vehicles enter and exit the network. The

behaviour of nodes may become unpredictable, and in this open and evolving network, there could be an uncertain number of faults or even Byzantine behaviours. For instance, autonomous driving systems might experience sensor malfunctions, leading to incorrect object detection, or software glitches that result in erroneous decision-making, such as a vehicle mistakenly accelerating when it should brake. Therefore, a reliable consensus mechanism must account for these variables to ensure that all vehicles reach an agreement on critical decisions, despite potential inconsistencies or failures within the network.

3.6.2 Consensus in Drone Swarms

A drone swarm is another typical example of a non-deterministic autonomous system with a need for consensus [142]. In operations such as search and rescue missions, drones must work together to cover large areas, avoid collisions, and optimize their search patterns. Although most current drone swarms rely on centralized control, where a central controller coordinates the actions of all drones, this approach can encounter limitations in dynamic environments or unexpected situations. For example, if the central controller fails or communication is interrupted, the drone swarm may no longer be able to operate cohesively. In such cases, adopting a distributed consensus mechanism would allow the drones to autonomously reach an agreement and continue their mission without the central controller. Therefore, even though drone swarms currently depend on centralized control, it is still important to consider a consensus mechanism to enhance the system's flexibility and robustness. The dynamic environment of a drone swarm, along with the possibility of unforeseen obstacles or environmental changes, demands a flexible and robust consensus protocol. Similar to vehicular networks, drones may experience malfunctions or malicious attacks, which requires the consensus mechanism to handle non-deterministic faults.

3.6.3 Consensus between Intelligent Robots

With the continuous advancement of AI and machine learning (ML), machines are increasingly equipped with intelligent capabilities [143], and the rise of intelligent robots is a prime example. In the future, collaboration among these robots is expected to become a major trend in both production and daily life. For instance, in intelligent manufacturing environments, groups of robots will need to coordinate their actions to accomplish complex tasks, requiring consensus on issues such as task allocation, path planning, and collision avoidance [20].

The autonomy granted by AI significantly amplifies the challenge of reaching consensus. Unlike traditional automation, intelligent robots can independently assess and adapt to their environments, leading to diverse data interpretations and strategic decisions. This au-

tonomy introduces complexity into the consensus process compared to traditional models. Furthermore, AI may endow nodes with strategic behaviour, making their actions more complex and less predictable. From a decision-making perspective, robots might make different choices based on varying circumstances or encounter unforeseen interactions. These factors demand more sophisticated consensus mechanisms to address the complex challenges posed by intelligent, autonomous agents working together.

The model presented in this paper is not a complete solution to all the potential issues mentioned above. Instead, it is a foundational reliability consensus framework designed to address probabilistic fault among consensus nodes and link within the system.

3.7 Conclusion

This chapter addresses key challenges in applying probabilistic model to distributed systems under probabilistic Byzantine behaviour assumptions. Two main research gaps are tackled: analysing non-deterministic consensus outcomes and redesigning the reliability quorum to align with probabilistic fault tolerance assumptions. A spectrum of non-deterministic outcomes is established and a theoretical model is built to evaluate probabilities of each outcomes. To enhance system reliability, a quorum selection framework is introduced tailored to probabilistic Byzantine behaviour and propose an optimal quorum computation based on a weighted scheme. Furthermore, the probabilistic framework is extended to account for unreliable communication links inherent in WDC, illustrating how node failures and link unreliability jointly affect overall system reliability. Our findings, validated through simulations, provide a foundation for more resilient and adaptable consensus mechanisms in dynamic, real-world wireless distributed systems.

Chapter 4

Decentralized Voting in DAOs

In decentralized systems, consensus is not confined to algorithmic coordination among machines. It also takes the form of collective decisions made by human participants. DAOs reflect this broader understanding of consensus. Rather than relying on distributed fault-tolerant protocols, DAOs primarily achieve consensus through voting among members. While smart contracts [144] and distributed ledgers [145] are essential for automating operations and maintaining transparency, voting mechanisms play a central role in determining decisions and guiding organizational changes. However, compared to these technical components, voting mechanisms have received much less systematic analysis. In particular, their role and performance in supporting decentralization, which is a fundamental feature of DAO governance, has not been thoroughly examined.

This chapter aims to address these gaps by systematically analysing voting as a form of consensus in DAOs. First, a governance triangle is introduced to clarify the interplay between voting, smart contracts, and distributed ledgers. Next, a stochastic model of DAO voting is developed to quantify decentralization performance, and the Decentralization Coefficient is proposed to provide a technical method for analysing and comparing governance mechanisms in real-world DAOs. Finally, the SEED metrics (Security, Efficiency, Effectiveness, Decentralization) are defined to support comprehensive evaluation of DAO voting schemes.

4.1 DAO Governance Triangle

In the extensive discussion of DAOs, voting receives little attention and is recognized as a decision-making method that naturally emerges from decentralization. In contrast, smart contracts and distributed ledger technology applied in DAOs have received much attention. However, as the primary decision-making method of DAOs, voting deeply affects the soundness of the entire DAO network. Therefore, the role and impact of the voting mechanism

within the overall DAO system are first discussed, and a DAO governance triangle is proposed.

The two prominent governance functional entities in DAO are the smart contract and the distributed ledger. Featured as automatic execution, predictable outcomes, public records, privacy protection, and visible terms, smart contracts can convert the rules and contracts based on human maintenance in conventional organizations into programs writing contracts as codes that are automatically executed. The distributed ledger is responsible for recording key activities in DAOs to the blockchain, ensuring that the recorded information is immutable.

The smart contract and the distributed ledger release the human maintenance requirements and reputation-based trust in the governance. However, they are not sufficient for all governance tasks, especially when it comes to decision-making. On the one hand, the requirements and transactions of DAOs are complex and vary over time and environment. Smart contracts are rules for predictable situations that are hard to cope with changing circumstances. On the other hand, due to the strict execution of smart contracts, leaving no room for change can create enormous pressure when contracts are set. Therefore, human decisions are inevitable and crucial in most DAOs to deal with unpredictable events and bring flexibility to the written stipulations in smart contracts. To maintain the decentralization in DAOs, all events requiring human decisions are decided by collective votes managed by voting mechanisms in current DAOs.

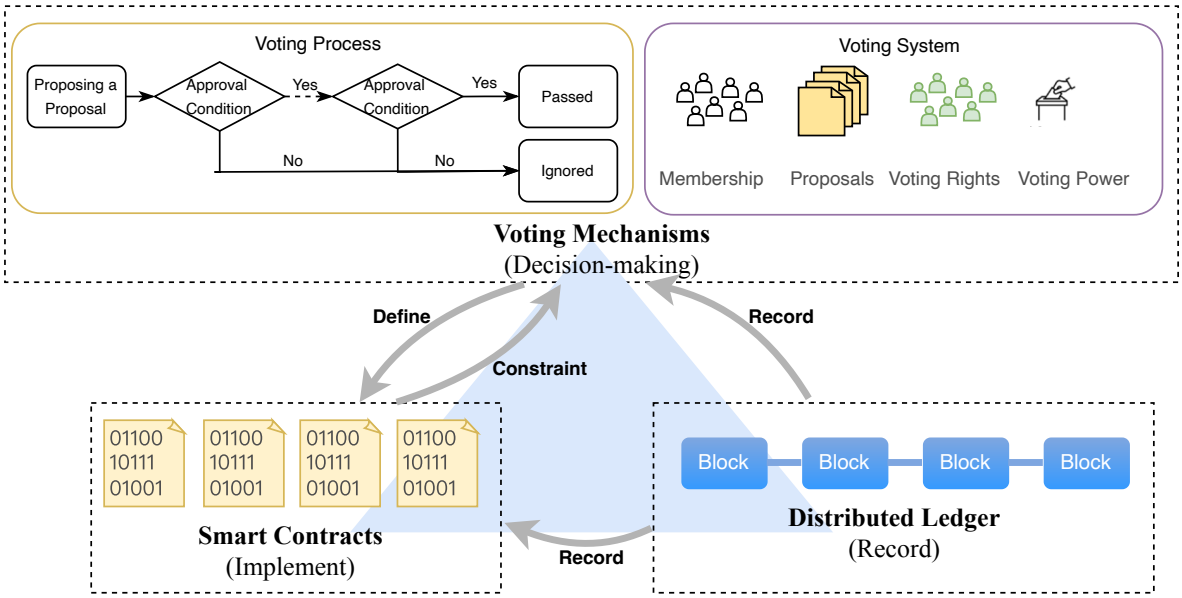

Figure 4.1: The DAO governance triangle: the relationship between distributed ledger, smart contracts and voting mechanisms in DAOs

Voting mechanisms importantly complement the smart contract to maintain non-hierarchy governance in DAOs. The governance architecture of DAOs can be described as a triangle

between distributed ledger, smart contracts, and voting mechanisms. Their relationship is shown in Fig. 4.1. Smart contracts are responsible for overseeing and implementing all the rules and contracts in DAOs, which of course, include rules and procedures in voting mechanisms. However, conversely, voting is the only formal way to change the rules already defined by smart contracts and add new rules to smart contracts. Smart contracts and voting mechanisms not only jointly build management in DAOs, but also co-constraint each other, and they are both critical to the healthy operation of DAOs. Critical activities performed by voting mechanisms and smart contracts are recorded in a distributed ledger database. The distributed ledger is not only aligned with the requirements of decentralization but also provides consistent and immutable records. The interactions and constraints between distributed ledger, smart contracts, and voting mechanisms form a stable governance system for DAOs.

4.2 SEED: A Multi-Dimensional Metric for Evaluating DAO Voting

Decentralization is a foundational principle in DAO voting, but it alone does not fully capture the qualities required for a robust and sustainable governance mechanism. To support effective decision-making in diverse DAO environments, additional performance dimensions must be considered. This study proposes a comprehensive evaluation framework comprising four critical aspects: Security, Efficiency, Effectiveness, and Decentralization, collectively referred to as the SEED framework. As illustrated in Fig. 4.2, this multi-dimensional metric provides a structured approach for analysing and comparing different voting mechanisms, offering both theoretical guidance and practical benchmarks for design.

4.2.1 Decentralization

Although DAO voting mechanisms may take inspiration from traditional voting systems, they cannot adopt them without significant adaptation. The essential difference lies in the principle of decentralization, which must be carefully preserved throughout the design and implementation of DAO governance. While DAO voting is often regarded as inherently decentralized due to its open participation, individual autonomy, and tolerance for partial engagement, these characteristics alone do not ensure truly decentralized outcomes.

In practice, various structural components of DAO voting carry the risk of introducing centralization. Aspects such as the allocation of voting power, the criteria for membership, and the rules governing decision thresholds can all produce imbalances that undermine the intended collective nature of governance. These risks are frequently overlooked when tech-

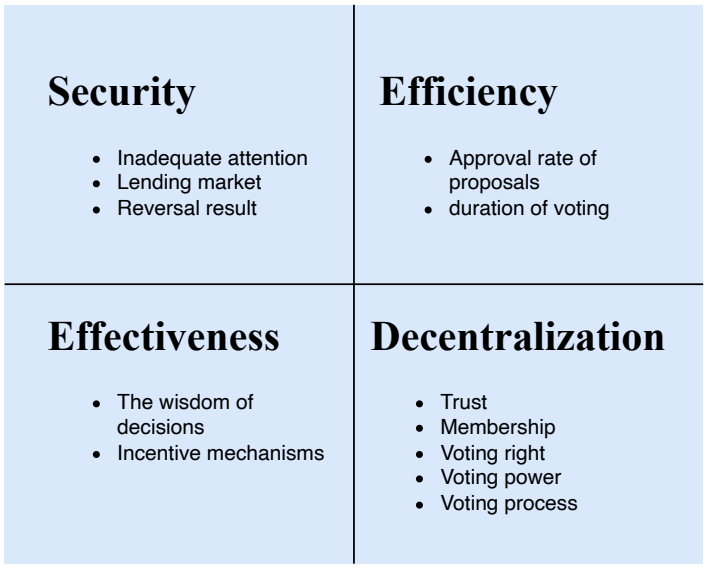

Figure 4.2: SEED: four dimensions to evaluate DAO voting

nical decentralization—such as transparent smart contracts and public ledgers, is assumed to be equivalent to governance decentralization. For example, although voting rights are typically granted to all token holders, empirical evidence reveals severe inequality in influence. In many DAOs employing a one-token-one-vote model, fewer than 1% of members control over 90% of the total voting power [146]. This concentration of power directly contradicts the non-hierarchical ideals that DAOs aim to uphold.

Some alternative mechanisms, such as quadratic voting [114], have been proposed to alleviate such imbalances. However, their effectiveness in practice remains uncertain due to a lack of consistent evaluation criteria. Recent studies have identified the persistence of centralized control in DAO governance as one of the key shortcomings of existing voting designs [110, 147]. Many existing improvements target isolated governance elements without addressing the broader systemic interactions that shape voting outcomes.

To address these challenges, it is necessary to move beyond qualitative discussion and adopt a more formal approach to evaluating decentralization. The following Sec. 4.3 introduces a quantitative framework that models DAO voting as a stochastic process and defines measurable indicators for assessing decentralization performance. This framework aims to provide a more objective foundation for comparing voting mechanisms and informing future design choices.

4.2.2 Security

For organizations like DAOs that own digital assets and conduct a large number of transactions, security is an indispensable basic condition [148]. The notion of security in DAO

voting considered in this study does not involve the security of transactions and verification of digital assets, as these concerns are typically addressed by underlying cryptocurrency and non-fungible token (NFT) infrastructures. Instead, the focus is on potential security risks arising from vulnerabilities in the voting mechanism and the possible ways in which an attacker may manipulate voting outcomes. In conjunction with the current design of DAOs, three representative security issues are identified.

The first threat is the slip-through passed proposals, meaning that a proposal may pass without the knowledge of most members. Since DAOs are causal participant communities, it can be difficult for members to keep an eye on every proposal, especially in large-scale DAO organizations with a large number of proposals. An attacker could exploit this vulnerability to pass harmful proposals. If the harmful proposal is related to public digital assets, the consequences are severe for DAOs. The security threat of the slip-through passed proposals is often caused by voting mechanisms not designed to require sufficient voting participation and attention. The most typical example is *Permissioned Relative Majority* (details refer to Sec. 2.3.2), which only relies on the relative majority to decide whether a proposal is approved or not without any requirement on the participant amount in each voting process. Adding a requirement for adequate attention in approval conditions is a straightforward method to avoid the slip-through passed proposals. Several specific methods can refer to Sec. 2.3.2.

The second security issue is voting power lending. Although the voting power in DAOs cannot be transferred, a secondary voting power lending market may be created. An attacker can gain a large number of voting powers in a short period by renting them. To avoid such a situation, some token-based DAOs link voting power conversion with time, which means the longer a member holds a token, the higher the corresponding voting power of the token.

The reversal result is the third security issue, which indicates a dramatic reversal of voting results at the end of the voting period. While a reversal of voting results before the end of the voting period may be reasonable, it cannot be ruled out that malicious members deliberately retain substantial voting power to change the voting results near the end of the voting period without allowing other members to act on the reversed results. Therefore, requiring the voting result to remain unchanged for a period before the voting closes to be a valid result is a safety measure that can be considered.

Although these three security issues may not cover all the vulnerabilities of DAO voting, it is obvious that the current design of the DAO voting mechanism has obvious security loopholes. Since the consequences of security issues are serious, security should be the primary consideration when designing a DAO voting mechanism. A secure voting mechanism is the basis of the stability in DAOs.

4.2.3 Efficiency

The efficiency of the voting mechanism can be measured from two aspects: the approval rate of proposals and the duration of voting. The quality of proposals in current DAOs is mixed, so a high proposal approval rate is unreasonable and cannot be used as a symbol of high efficiency. However, a low proposal approval rate does affect the decision-making efficiency of an organization works. Although the overall quality of the proposal and the voting habits of members can affect the proposal approval rate, a low voting participation rate with a high attention threshold and the limited attention of each member on the large number of proposals in large-scale DAOs increase the difficulty for proposals to pass.

The duration of voting is usually specified, but a reasonable voting duration is affected by the level of activity of DAOs. As organizations with no obvious entity, DAOs cannot easily maintain close ties between members and the organization. An inadequate voting duration may largely decrease the approval rate of proposals. However, the duration of voting directly affects efficiency, and short-duration voting is necessary for time-sensitive and opportunistic decision-making. Since voting is the most important way to adjust and promote DAOs, insufficient and untimely changes caused by low proposal approval rates and long voting duration can profoundly limit the development of the organizations. The efficiency quantification analysis is provided in Sec. 4.4.

4.2.4 Effectiveness

Decentralization, security and efficiency in voting do not guarantee the decision voted by the member is a good decision for the development of the organization. Therefore, effectiveness which emphasizes the quality of the voting is also indispensable for DAOs. Limited by perceptions of information, knowledge, and expertise, it is difficult to make effective judgments when members are faced with issues in unfamiliar territory. Therefore, it is not convinced that collective decision-making made under a decentralized, secure and efficient voting mechanism is wise, especially when voting right is easily available in most DAOs. Therefore, only informed and professional decision-making is more likely to make effective decisions that can sustainably contribute to the development of DAOs. A feasible way to improve the effectiveness of voting is to assign more voting weight to professional members in proposals related to their expertise, which is a typical approach used in *Knowledge-Extractable Voting* [149] to solve this problem.

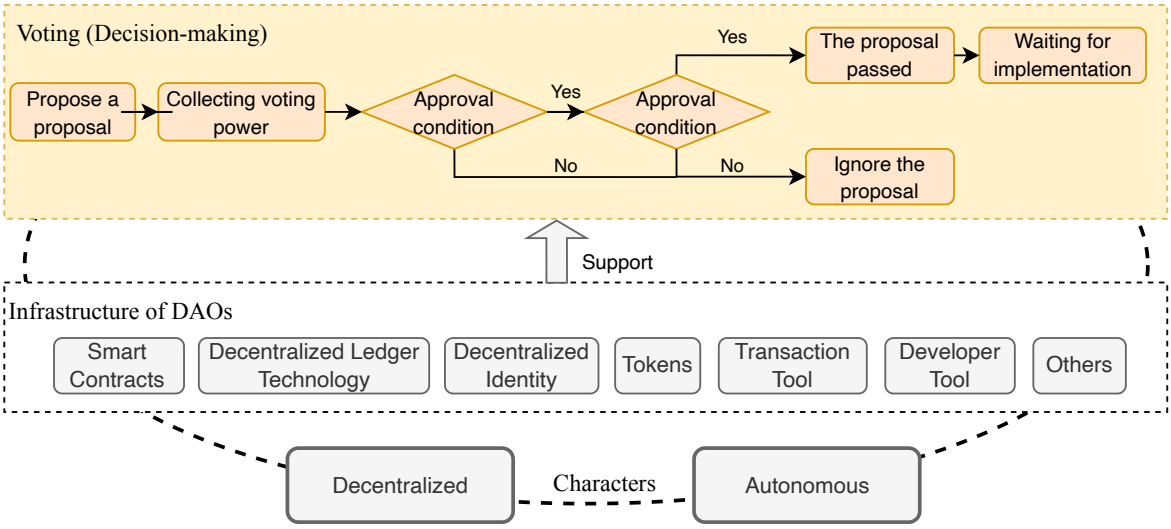

Figure 4.3: Voting system based on DAO Infrastructure

4.3 Quantify Decentralization Performance

Building on the SEED framework, which identifies key performance dimensions of DAO voting mechanisms, this section introduces a formal model to enable quantitative analysis of decentralization performance based on the DAO voting process. The voting process varies across different DAOs, and new voting mechanisms are continuously created in personalized DAO projects. However, proposing a proposal, collecting votes, verifying with approval conditions, and executing results should be the backbone of all DAO voting processes (detailed in Sec. 2.3.2).

Specifically, each voting process starts with a proposal. After the proposal is successfully submitted, it remains open for voting until it reaches the target voting power threshold or the time limit. The final voting result is automatically generated according to the proposal passing criteria, referred to as approval conditions, which are written into the smart contract. The voting mechanisms discussed in this paper are all based on this fundamental voting process. A generalized model representing the DAO voting process is developed to capture core elements such as voter behaviour, token-weighted influence, and probabilistic participation, following the voting process of DAOs introduced in Fig. 4.3.

The model assumes a set of n eligible voters, denoted by the set $\mathbf{U} = \{N_1, N_2, N_3, \ldots, N_n\}$. For simplicity, a one-token-one-vote (1T1V) scheme is adopted, in which each voter's voting power is proportional to the number of tokens held. The voting power of voter i is represented by v_i, and the set of voting powers is denoted as $\mathbf{V} = \{v_1, v_2, v_3, \ldots, v_n\}$. The key parameters used in the quantitative analysis are summarized in Table 4.1.

Table 4.1: Frequently used notations

Notation	Definition
n	Total number of voters in the DAO voting process
$\mathbf{U}$	Set of all voters, i.e., $\{N_1, N_2, ..., N_n\}$
v_i	Voting power held by voter N_i
$\mathbf{V}$	Voting power distribution vector, i.e., $\{v_1, v_2, ..., v_n\}$
X_i	Voting preference of voter i, with $X_i = 1$ (support), -1 (oppose)
Y_i	Voting participation choice of voter i, with $Y_i = 1$ (participate), -1 (abstain)
q_i	Probability that voter i supports a proposal
p_i	Probability that voter i participates in the vote
$\mathbf{Q}$	Voting preference probability vector, i.e., $\{q_1, q_2, ..., q_n\}$
$\mathbf{P}$	Participation probability vector, i.e., $\{p_1, p_2, ..., p_n\}$
ω_j	A specific voting profile in the sample space Ω
Ω	Sample space of all possible voting profiles
Ω_v	Valid sample space satisfying all approval conditions
R	Voting result: $R = 1$ (proposal approved), $R = -1$ (rejected)
P_i^C	Consistency Rate: probability that voter i's preference aligns with the result
C_i	Controlling Ability of voter i, normalized from P_i^C
D	Decentralization Coefficient measuring overall decentralization
$A_k(\omega_j)$	Evaluation of the k-th approval condition on voting profile ω_j
$F(\omega_j)$	Overall approval function determining whether ω_j is valid
h_v	Voting power threshold (used in VT)
h_q	Quorum threshold (used in QT)

4.3.1 The Stochastic Process of DAO Voting

Given the option for voters to abstain from voting, the voting behaviour of each participant is partitioned into two distinct aspects. The first is their voting preference choice, while the second is their voting participation choice. It is assumed that each individual has a unique voting preference, irrespective of their participating choices in the voting procedure. This voting preference is encapsulated by their binary selection between supporting (*Yes*) and opposing (*No*), quantified as numerical values 1 and -1 correspondingly. Hence, the set of available preference choices is denoted as $\mathbf{C_1} = \{1, -1\}$. As for the engagement, similarly, the set of participating choices is established as $\mathbf{C_2} = \{1, -1\}$, where 1 indicates active participation and -1 indicates abstention.

For each voter i, let the random variable X_i, $X_i = \{x_i | x_i \in \mathbf{C_1}\}$ represents the voter's voting preference. For each voter i, let the random variable Y_i, $Y_i = \{y_i | y_i \in \mathbf{C_2}\}$ represents the voter's participating choice. The voting behaviour of all voters make up the overall voting profile denoted as an n-tuple $\mathbf{B} = (\{X_1, Y_1\}, \{X_2, Y_2\}, ..., \{X_n, Y_n\})$. Clearly, there are four types of voting behaviours: support and participate ($X_i = 1, Y_i = 1$), oppose and participate ($X_i = -1, Y_i = 1$), support and abstain ($X_i = 1, Y_i = -1$), and oppose and abstain ($X_i = -1, Y_i = -1$). It is assumed that each user behaves independently. It is also defined the set of voting supporting preference probability and the set of participating probability are $\mathbf{Q} = \{q_1, q_2, ..., q_n\}$ and $\mathbf{P} = \{p_1, p_2, ..., p_n\}$ respectively, where the probability for each voter i to have a supporting voting preference is q_i, and the probability for each voter i to participate in the voting is p_i.

A sample space is denoted as $\Omega = \{\omega_1, \omega_2, ..., \omega_{4^n}\}$, with 4^n basic events describing all the cases of the voting profile in one voting process. Each basic event ω_j of the sample space Ω describes a unique voting profile numbered j. According to the four types of behaviour in voting, voters are divided into four sets $\mathbf{M}^Y_{\omega_j} = \{i | X_i(\omega_j) = 1, Y_i(\omega_j) = 1, i \in \mathbf{U}\}$, $\mathbf{M}^N_{\omega_j} = \{i | X_i(\omega_j) = -1, Y_i(\omega_j) = 1, i \in \mathbf{U}\}$, $\mathbf{M}^{AY}_{\omega_j} = \{i | X_i(\omega_j) = 1, Y_i(\omega_j) = -1, i \in \mathbf{U}\}$, $\mathbf{M}^{AN}_{\omega_j} = \{i | X_i(\omega_j) = -1, Y_i(\omega_j) = -1, i \in \mathbf{U}\}$ containing voters in event ω_j who are in favour and participated, against and participated, in favour and abstained, and against and abstained respectively. Therefore, the probability of event ω_j can be written as

$$\begin{aligned} Pr(\omega_j) &= Pr(\mathbf{M}^Y_{\omega_j}, \mathbf{M}^N_{\omega_j}, \mathbf{M}^{AY}_{\omega_j}, \mathbf{M}^{AN}_{\omega_j}) \\ &= \prod_{a \in M^Y_{\omega_j}} q_a p_a \prod_{b \in M^N_{\omega_j}} (1 - q_b) p_b \prod_{c \in M^{AY}_{\omega_j}} q_c (1 - p_c) \prod_{d \in M^{AN}_{\omega_j}} (1 - q_d)(1 - p_d). \end{aligned} \tag{4.1}$$

It is assumed that there will always be a validated voting result, which is whether the proposal is passed or not. This voting result is denoted as a random variable following the Bernoulli distribution as $R = \{1, -1\}$, where 1 indicates the proposal is approved and -1

indicates the proposal is rejected. The proposal passing rate is the sum of the probability of all voting profile cases that enable the proposal to pass. Obviously, not all the voting profiles in Ω satisfied the approval conditions in the voting process. Therefore, those voting profiles can satisfy all the conditions in the voting process are concluded in a valid sample space Ω_v ($\Omega_v \subseteq \Omega$). Therefore, the probability for a proposal to pass in one voting process is

$$Pr(\Omega_v) = \sum_{\omega_j \in \Omega_v} Pr(\omega_j). \tag{4.2}$$

With the expression of the proposal passing rate, the probability mass function (*p.m.f*) of voting result R is

$$f_R(r) = \begin{cases} Pr(\Omega_v) & (r = 1) \\ 1 - Pr(\Omega_v) & (r = -1). \end{cases} \tag{4.3}$$

Since the valid sample space Ω_v cannot be defined unless the voting process, approval conditions and the token distribution are clear. To provide a specific analysis of the sample spaces, the typical voting procedures is discussed as follows.

4.3.2 The Analysis of the Valid Sample Space

To determine the sample space Ω_v, each valid event ω_j in Ω_v has to be evaluated by specific approval conditions in the voting process. The key to evaluating whether an event ω_j meets the approval conditions is the comparison of the number of voters or the amount of voting power in the different voter groups divided by their voting behaviours, such as participating voters, abstaining voters, participating voters who support, and participating voters who oppose. Such groups are named as critical voter clusters. Therefore, to assess whether an event ω_j meets approval conditions, the critical voter clusters are key analysis objects.

In most cases, the critical voter clusters and their corresponding number and voting power are the keys to determining whether the approval conditions in a voting process are met. The four critical voter clusters are the participated voter sets, the abstained voter sets, the participating and supporting voter sets and the participating and opposing voter sets. To determine the valid sample space is to evaluate the critical voter clusters according to approval conditions. To be consistent with the voting result $R = \{1, -1\}$, a sign function is defined to indicate all the results in binary numbers as

$$S(r) = \begin{cases} 1 & r > 0 \\ 0 & r \leq 0. \end{cases} \tag{4.4}$$

The number and voting power of the critical voter clusters of a voting profile case ω_j can be summarized as TABLE 4.4.

Table 4.2: Number and voting power of critical voter clusters

Critical groups	**Number of voters**	**Sum of voting power**
Participated voters	$\sum_{i \in \mathbf{U}} S(Y_i(\omega_j))$	$\sum_{i \in \mathbf{U}} S(Y_i(\omega_j))v_i$
Abstained voters	$\sum_{i \in \mathbf{U}} S(-Y_i(\omega_j))$	$\sum_{i \in \mathbf{U}} S(-Y_i(\omega_j))v_i$
Favour in participated voters	$\sum_{i \in \mathbf{U}} S(X_i(\omega_j))S(Y_i(\omega_j))$	$\sum_{i \in \mathbf{U}} S(X_i(\omega_j))S(Y_i(\omega_j))v_i$
Against participated voters	$\sum_{i \in \mathbf{U}} S(-X_i(\omega_j))S(Y_i(\omega_j))$	$\sum_{i \in \mathbf{U}} S(-X_i(\omega_j))S(Y_i(\omega_j))v_i$

In the valid sample space Ω_v, all the voting profiles of $\omega_j \in \Omega_v$ must satisfy each approval condition. According to the voting process in Fig. 4.3, approval conditions are added in sequence during the voting process. Each approval condition is denoted in the voting process as A_k, where k refers to its sequence in the voting process. In Fig. 4.3, each link of the approval condition is a A_k. Each A_k is also a function of ω_j, representing whether the ω_j satisfy the approval condition A_k. In one voting process, all the approval conditions are denoted in the approval condition set $\mathbf{A} = \{A_1, A_2, ...A_k\}$. To be consistent with the definition of voting result R, the fulfilled approval condition is defined as $A_k = 1$, while failing to achieve approval condition $A_k = 0$.

Most approval conditions can be expressed with the number and the voting power from critical voter clusters. Four examples of typical approval conditions are provided, including RM, AM, VT and QT and list their expressions in TABLE 4.5.

Table 4.3: Encoded expressions of approval conditions

Approval Condition	**Expression**
RM	$A_{rm}(\omega_j) = S(\sum_{i \in \mathbf{U}} S(X_i(\omega_j))S(Y_i(\omega_j))v_i - \sum_{i \in \mathbf{U}} S(-X_i(\omega_j))S(Y_i(\omega_j))v_i)$
AM	$A_{am}(\omega_j) = S(\sum_{i \in \mathbf{U}} S(X_i(\omega_j))S(Y_i(\omega_j))v_i - 50\% \sum_{i \in \mathbf{U}} v_i)$
VT	$A_{vt}(\omega_j) = S(\sum_{i \in \mathbf{U}} S(Y_i(\omega_j))v_i - h_v)$
QT	$A_{qt}(\omega_j) = S(\sum_{i \in \mathbf{U}} S(Y_i(\omega_j)) - h_q)$

h_v is voting power threshold and h_q is quorum threshold.

Combining all the approval conditions in the voting process, an approval function $F(\omega_j)$ is defined to judge whether the voting profile in ω_j can enable the proposal to pass, which means $\omega_j \in \Omega_v$. The approval function is expressed by connecting all the encoded approval conditions in the voting process as

$$F(\omega_j) = A_1(\omega_j) \cdot A_2(\omega_j) \cdot \ldots \cdot A_k(\omega_j). \tag{4.5}$$

Therefore, the valid sample space can be defined as $\Omega_v = \{\omega_j | F(\omega_j) = 1\}$.

With the approval function, equation (4.2) can be expressed as

$$P^A \triangleq Pr(\Omega_v) = \sum_{\omega_j \in \Omega_v} \prod_{a \in M^Y_{\omega_j}} q_a p_a \prod_{b \in M^N_{\omega_j}} (1-q_b) p_b \times \prod_{c \in M^{AY}_{\omega_j}} q_c (1-p_c) \prod_{d \in M^{AN}_{\omega_j}} (1-q_d)(1-p_d). \tag{4.6}$$

where $\Omega_v = \{\omega_j | F(\omega_j) = 1\}$. Here, P^A denotes the approval rate, i.e., the probability that a proposal is approved in a single voting process, and also helps to define the Bernoulli distribution in the voting result $R \sim \mathcal{B}(1, Pr(\Omega_v))$.

This voting model provides a versatile framework for quantitative analysis of DAO voting. The impact of different voting processes and conditions on the entire voting system can be integrated into the model for analysis. In addition, it can reflect the performance of the entire voting system rather than a single mechanism since the model includes a description of the entire voting process.

4.3.3 Consistency Rate and Controlling Ability

To evaluate the decentralization performance of the overall system, the initial step is quantifying the impact of each voter's preference on the collective voting outcome. This influence can be equated with the degree of control each voter holds over the voting results. Therefore, the probability that each voter's voting preference (X_i) aligns with the final voting result (R) is first analysed. This probability is referred to as the *Consistency Rate*. The Consistency Rate for voter i is denoted by P_i^C, defined as the sum of two probabilities: the probability that voter i prefers to support the proposal when the final result is passed, and the probability that voter i prefers to oppose the proposal when the final result is rejected. Accordingly, the expression for P_i^C is given as

$$P_i^C = Pr(R = 1, X_i = 1) + Pr(R = -1, X_i = -1). \tag{4.7}$$

As for the $Pr(R = 1, X_i = 1)$, a sample space $\Omega^{pass}_{i_c}$ is assumed containing all cases

of the voting profile that the proposal passes while the voting preference of voter i_c is in favour. Each basic event in the sample space $\Omega_{i_c}^{pass}$ is $\omega_{i_c,j}^{pass}$, j is used to distinguish different events. Due to the additional condition that the voter i_c is in favour of the proposal, $\Omega_{i_c}^{pass} = \{\omega_{i_c,j}^{pass} | F(\omega_{i_c,j}^{pass}) \cdot X_{i_c}(\omega_{i_c,j}^{pass}) = 1\}$. All the voters in each event $\omega_{i_c,j}^{pass}$ are divided into the voter set that is in favour and participated $\mathbf{M}^{Y}_{\omega_{i_c,j}^{pass}}$, the voter set that is against and participated $\mathbf{M}^{N}_{\omega_{i_c,j}^{pass}}$, the voter set that is favour and abstained $\mathbf{M}^{AY}_{\omega_{i_c,j}^{pass}}$, and the voter set that is against and abstained $\mathbf{M}^{AN}_{\omega_{i_c,j}^{pass}}$. Then, the probability of each basic event $\omega_{i_c,j}^{pass}$ is

$$\begin{aligned} Pr(\omega_{i_c,j}^{pass}) &= Pr(\mathbf{M}^{Y}_{\omega_{i_c,j}^{pass}}, \mathbf{M}^{N}_{\omega_{i_c,j}^{pass}}, \mathbf{M}^{AY}_{\omega_{i_c,j}^{pass}}, \mathbf{M}^{AN}_{\omega_{i_c,j}^{pass}}) \\ &= \prod_{a \in M^{Y}_{\omega_{i_c,j}^{pass}}} q_a \cdot p_a \prod_{b \in M^{N}_{\omega_{i_c,j}^{pass}}} q_b \cdot p_b \prod_{c \in M^{AY}_{\omega_{i_c,j}^{pass}}} q_c \cdot (1 - p_c) \prod_{d \in M^{AN}_{\omega_{i_c,j}^{pass}}} q_d \cdot (1 - p_d), \end{aligned} \tag{4.8}$$

where $\mathbf{M}^{Y}_{\omega_{i_c,j}^{pass}} = \{i | X_i(\omega_{i_c,j}^{pass}) = 1, Y_i(\omega_{i_c,j}^{pass}) = 1\}$, $\mathbf{M}^{N}_{\omega_{i_c,j}^{pass}} = \{i | X_i(\omega_{i_c,j}^{pass}) = -1, Y_i(\omega_{i_c,j}^{pass}) = 1\}$, $\mathbf{M}^{AY}_{\omega_{i_c,j}^{pass}} = \{i | X_i(\omega_{i_c,j}^{pass}) = 1, Y_i(\omega_{i_c,j}^{pass}) = -1\}$, $\mathbf{M}^{AN}_{\omega_{i_c,j}^{pass}} = \{i | X_i(\omega_{i_c,j}^{pass}) = -1, Y_i(\omega_{i_c,j}^{pass}) = -1\}$.

The probability of $Pr(R = 1, X_i = 1)$, which is also the probability of all the basic event in the sample space $\Omega_{i_c}^{pass}$ is

$$Pr(R = 1, X_i = 1) = Pr(\Omega_{i_c}^{pass}) = \sum_{\omega_{i_c,j}^{pass} \in \Omega_{i_c}^{pass}} Pr(\omega_{i_c,j}^{pass}), \tag{4.9}$$

where $\Omega_{i_c}^{pass} = \{\omega_{i_c,j}^{pass} | F(\omega_{i_c,j}^{pass}) \cdot X_{i_c}(\omega_{i_c,j}^{pass}) = 1\}$.

For the calculation of $Pr(R = 0, X_i = 0)$, a sample space $\Omega_{i_c}^{rej}$ is assumed containing all cases of the voting profile that the proposal is rejected while the voter i_c is against the proposal. Each basic event for the sample space $\Omega_{i_c}^{rej}$ is $\omega_{i_c,j}^{rej}$, j is used to distinguish different events. Due to the additional condition that the voter i_c is against the proposal, $\Omega_{i_c}^{rej} = \{\omega_{i_c,j}^{rej} | F(\omega_{i_c,j}^{rej}) = 0, X_{i_c}(\omega_{i_c,j}^{rej}) = -1\}$. All the voters in each event $\omega_{i_c,j}^{rej}$ are divided into the voter set that is in favour and participated $\mathbf{M}^{Y}_{\omega_{i_c,j}^{rej}}$, the voter set that is against and participated $\mathbf{M}^{N}_{\omega_{i_c,j}^{rej}}$, the voter set that is in favour and abstained $\mathbf{M}^{AY}_{\omega_{i_c,j}^{rej}}$, and the voter set that is against and abstained $\mathbf{M}^{AN}_{\omega_{i_c,j}^{rej}}$. Then, the probability of each basic event $\omega_{i_c,j}^{rej}$ is

$$\begin{aligned} Pr(\omega_{i_c,j}^{rej}) &= Pr(\mathbf{M}^{Y}_{\omega_{i_c,j}^{rej}}, \mathbf{M}^{N}_{\omega_{i_c,j}^{rej}}, \mathbf{M}^{AY}_{\omega_{i_c,j}^{rej}}, \mathbf{M}^{AN}_{\omega_{i_c,j}^{rej}}) \\ &= \prod_{a \in M^{Y}_{\omega_{i_c,j}^{rej}}} q_a \cdot p_a \prod_{b \in M^{N}_{\omega_{i_c,j}^{rej}}} q_b \cdot p_b \prod_{c \in M^{AY}_{\omega_{i_c,j}^{rej}}} q_c \cdot (1 - p_c) \prod_{d \in M^{AN}_{\omega_{i_c,j}^{rej}}} q_d \cdot (1 - p_d), \end{aligned} \tag{4.10}$$

where $\mathbf{M}^{Y}_{\omega^{rej}_{ic,j}} = \{i|X_i(\omega^{rej}_{ic,j}) = 1, Y_i(\omega^{rej}_{ic,j}) = 1\}$, $\mathbf{M}^{N}_{\omega^{rej}_{ic,j}} = \{i|X_i(\omega^{rej}_{ic,j}) = -1, Y_i(\omega^{rej}_{ic,j}) = 1\}$, $\mathbf{M}^{AY}_{\omega^{rej}_{ic,j}} = \{i|X_i(\omega^{rej}_{ic,j}) = 1, Y_i(\omega^{rej}_{ic,j}) = -1\}$, and $\mathbf{M}^{AN}_{\omega^{rej}_{ic,j}} = \{i|X_i(\omega^{rej}_{ic,j}) = -1, Y_i(\omega^{rej}_{ic,j}) = -1\}$.

The probability of $Pr(R = -1, X_i = -1)$, which is also the probability of all the basic event in the sample space $\Omega^{rej}_{i_c}$ is

$$Pr(R = -1, X_i = -1) = Pr(\Omega^{rej}_{i_c}) = \sum_{\omega^{rej}_{ic,j} \in \Omega^{rej}_{i_c}} Pr(\omega^{rej}_{ic,j}), \tag{4.11}$$

where $\Omega^{rej}_{i_c} = \{\omega^{rej}_{ic,j} | F(\omega^{rej}_{ic,j}) = 0, X_{i_c}(\omega^{rej}_{ic,j}) = -1\}$. With (4.9) and (4.11), the *Consistency Rate* P^C_i of voter i_c can finally obtained by (4.7).

The derivation of the *Consistency Rate* takes into account the voting process, providing a comprehensive basis for evaluating the overall controlling ability of each voter on the voting results. The *Consistency Rate* reflects the alignment of each voter's voting preference with the final result, regardless of whether they participated in the voting. Since the analysis does not target a specific proposal (voting object) when evaluating the decentralized performance of the voting mechanism, it is assumed that each voter's preference is neutral. Specifically, for each voter i, the probability of supporting the proposal is set to $q_i = 0.5$. With this assumption, even if a voter has no ability to influence the voting at all, their *Consistency Rate* remains at 0.5. On the other hand, a dominant voter holding all the voting power approaches a *Consistency Rate* of 1 (when a dominant voter is assumed to have a very small probability of abstaining). Therefore, $P^C_i \in [0.5, 1)$.

To more intuitively define *Controlling Ability* of each voter over DAO voting, the min-max normalized Consistency Rate is introduced. By applying min-max normalization to the *Consistency Rate*, the values to a range between 0 and 1 is rescaled. The min-max normalized *Consistency Rate* serves as a clear and standardized measure of the *Controlling Ability* of each voter in the DAO voting system. The *Controlling Ability* C_i is defined as

$$C_i = \frac{P^C_i - 0.5}{1 - 0.5} = 2(P^C_i - 0.5), \tag{4.12}$$

where $C_i \in [0, 1)$. A *Controlling Ability* score of 1 indicates that the voter has the highest ability to affect the voting outcome, while a score of 0 suggests that the voter has no influence on the voting.

4.3.4 Decentralization Coefficient

The *Controlling Ability* can evaluate the ability of individual voters to control and influence the vote in a voting process. However, to further evaluate the degree of decentralization

in voting, it is necessary to conduct an overall assessment of the *Controlling Ability* of all voters within the voting process. The Gini coefficient, first proposed by Corrado Gini in 1912, is extensively employed in economic and social science research to assess the level of equality [150]. In the context of DAOs, where voting serves as the primary decision-making mechanism, the equality level of the *Controlling Ability* can be interpreted as the level of decentralization. In other words, the more equitable the distribution of controlling ability among voters, the higher the decentralization of the voting process in the DAO. Therefore, the decentralization performance of DAO voting can be assessed by quantifying the equality level of each voter's *Controlling Ability*, using the Gini coefficient.

The Gini coefficient is usually defined with the help of the Lorenz curve [151], sorting the accumulated resource allocation ratio from low to high, but it can also be simply summarized as the formula [151],

$$G = \frac{\sum_{a=1}^{n} \sum_{b=1}^{n} |S_a - S_b|}{2n^2 \overline{S}}, \tag{4.13}$$

where $S_{a/b}$ represents each partial of the resource allocated to a or b, and $\overline{S}$ is the average of $S_{a/b}$.

To express the decentralized performance of DAO voting, an indicator, the decentralization coefficient is defined as the Gini coefficient of the *Controlling Ability* to quantify the decentralization performance of DAO voting mechanisms, which can be expressed as

$$D = \begin{cases} \frac{\sum_{i_1=1}^{n} \sum_{i_2=1}^{n} |C_{i_1} - C_{i_2}|}{2n^2 \overline{C}} & (\overline{C} \neq 0) \\ 0 & (\overline{C} = 0) \end{cases} \tag{4.14}$$

where $\overline{C}$ is the average of all the C_i of all voters, and C_{i_1} and C_{i_2} denote arbitrary C_i.

Boundary of Decentralization Coefficient

The value of the Gini coefficient is always between 0 and 1. To determine the specific boundaries of the decentralization coefficient, the decentralization coefficient (D) is calculated for the most decentralized and most centralized voting cases.

In the most decentralized voting case, all voters hold an equal amount of voting power, resulting in identical *Controlling Abilities*. As a result, the decentralization coefficient of the most decentralized voting case is the lower boundary, $D_{min} = 0$. In the most centralized voting case, a dominant voter possesses a significant amount of voting power, while all other voters have no voting power, rendering their voting preferences ineffective. In this case, only the dominant voter has the non-zero *Controlling Ability*, $C_d = \lambda, (0 < \lambda < 1)$ (the value of the *Controlling Ability* depends on the participating rate of the dominant voter), and the

Controlling Abilities of all other voters are $C_i = 0$. Accordingly, based on the definition of the decentralization coefficient (4.14), the decentralization coefficient of the most centralized voting case is the upper boundary, $D_{max} = \frac{n-1}{n}$. Therefore, the decentralization coefficient $D \in [0, \frac{n-1}{n}]$ encompasses all possible values between the completely decentralized and completely centralized voting scenarios. A smaller decentralization coefficient indicates a more pronounced decentralization.

The following sections analyse how various factors influence the decentralization performance in DAO voting and illustrate the decentralization characteristics of several representative voting cases.

4.3.5 Simulation Results

Based on the DAO voting model, the decentralization performance of the voting system measured by the decentralization coefficient $D(\mathbf{V}, \mathbf{P}, \mathbf{A})$ is influenced by three key factors: the distribution of tokens ($\mathbf{V}$), the participating rate of each voter ($\mathbf{P}$) and the voting process made of approval conditions ($\mathbf{A}$). To enable separate analysis of the three factors, the evaluation begins with the assumption that all voters participate in the voting process ($\mathbf{P} = \{1\}^n$), allowing for an isolated examination of the impact of voting power distribution. The factor of participation is then introduced to enable a more comprehensive analysis. Finally, various voting processes with different approval conditions are compared.

Based on the commonly used approval conditions described in Sec. 2.3.2, four representative voting scenarios are considered: approval conditions based solely on RM, those based solely on AM, those combining RM and QT, and those combining both RM and VT. These typical cases are selected for detailed analysis and simulation-based evaluation. The choice of a small value for n does not alter the fundamental trends of our results.

Impact of Voting Power Distribution

To focus on voting power distribution, it is assumed all voters always participate in the voting ($\mathbf{P} = 1^n$). When voters fully participate, the AM becomes the same condition as the RM under full participation, rendering the two attention thresholds (QT and VT) unnecessary, which means the four typical voting schemes yield similar effects. Therefore, in the analysis of fully participating cases, the RM is exclusively applied as the approval condition.

According to the investigation of [146], in the 10 major DAOs they investigated, more than 90% voting power is held by less than 10% voters, which reflects that most of the DAOs are facing the centralized threats of the dominant voting power control. To describe the adjustable dominant token distribution, the $Zipf$ distribution is applied to define the proportion

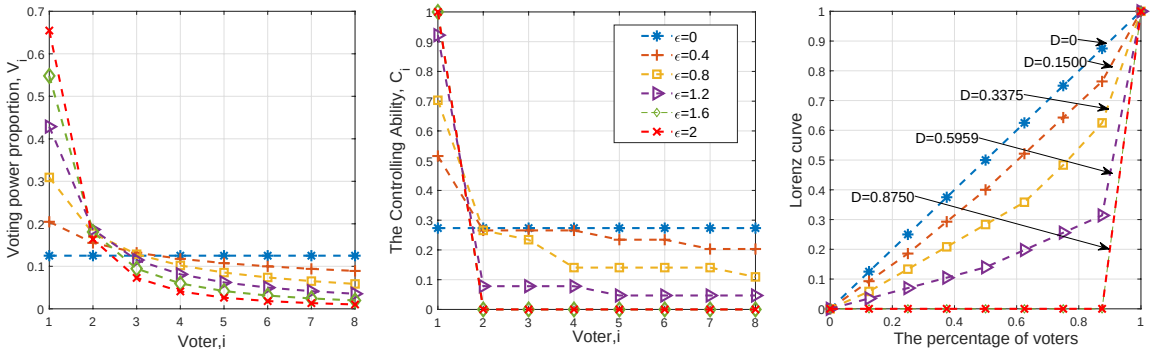

Figure 4.4: The voting power proportion (a: left), the controlling ability (b: middle), the Lorenz curve of the controlling ability (c: right) with different centralization levels of voting power distribution when $n = 8$

of voting power, V held by each voter. Let $V \sim Zipf(\epsilon, n)$,

$$V(y) = \frac{1}{y^{\epsilon} \sum_{i=1}^{n} (1/i)^{\epsilon}} \quad y = 1, 2, ..., n \tag{4.15}$$

where ϵ ($\epsilon \geq 0$) denotes the centralized level of the voting power distribution, and n is the number of voters. The larger the ϵ, the more centralized/dominant the voting power distribution. Here, $\epsilon \geq 0$ denotes the Zipf exponent that controls the skewness of the ranked voting-power distribution. When $\epsilon = 0$, the distribution reduces to the uniform case $V(y) = 1/n$. As ϵ increases, voting power becomes increasingly concentrated among top-ranked voters, leading to higher centralization; in the limit $\epsilon \to \infty$, the top-ranked voter dominates the outcome.

The decentralization performance of typical voting power distribution cases, defined by (4.15), is compared under varying levels of centralization, indicated by $\epsilon = \{0, 0.4, 0.8, 1.2, 1.6, 2\}$, with the number of voters fixed at $n = 8$. The comparison results are demonstrated in Fig. 4.4. To facilitate the display of the results, the voters in the figure are deliberately sorted from largest to smallest according to their voting power. Fig. 4.4 (a) shows the voting power proportion of each voter with different ϵ. When $\epsilon = 0$, the voting power is completely decentralized, and all the voters hold the same amount of voting power. The centralization/dominance of the voting power distribution increases with ϵ. As shown in Fig. 4.4 (b), the controlling ability dominant trend increases as the voting power distribution is more centralized. However, the change in controlling ability is step-wise compared with the smooth continuous change of the initial assumption of voting power proportion, and voters in a certain range of voting power proportion have the same controlling ability. In other words, when voters increase or decrease their voting power, sometimes within a certain range, there will be no change in anyone's controlling ability (unless it is at a critical value that brings about changes). Correspondingly, decentralization performance declined when the voting power distribution is more centralized. As twice the area between the Lorenz curve and the diago-

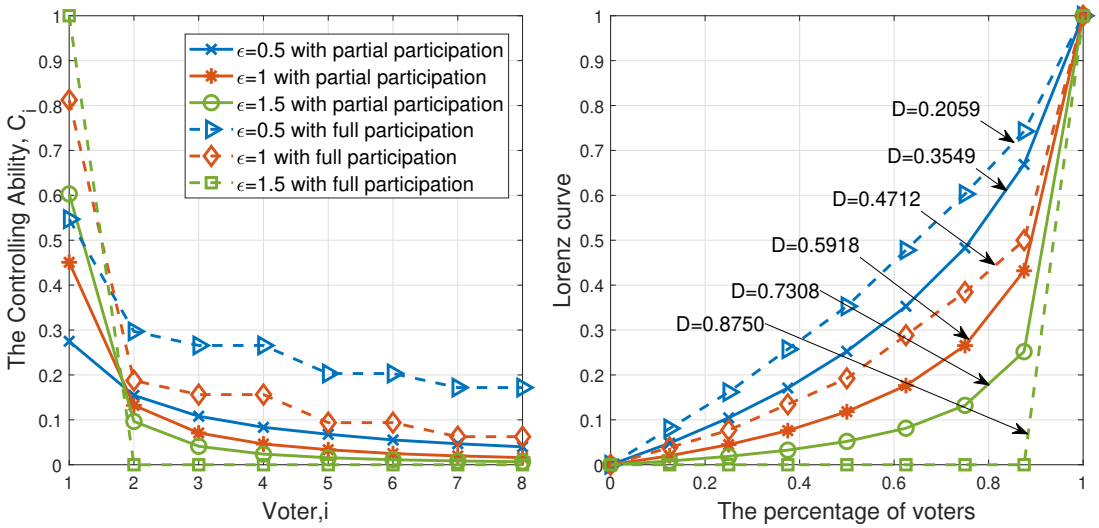

Figure 4.5: Comparison of the partial participation and the full participation: the controlling ability (a: left), the Lorenz curve of the controlling ability (b: right) with different centralization levels of voting power distribution when $n = 8$

nal line is another way to calculate the Gini coefficient, the Lorenz curves of the controlling ability are illustrated in Fig. 4.4 (c). The Lorenz curves are getting farther away from the diagonal line (also the $\epsilon = 0$), representing absolute decentralization as the ϵ increases. The values of the decentralization coefficient are indicated as D in Fig. 4.4 (c). It is observed that the decentralization coefficient values increase until the voting power of the voter with the largest amount of power to vote has exceeded the threshold of the absolute majority (0.5). Since the most dominant voter can absolutely control the voting result, the decentralized performance impacted by the voting power distribution has reached the worst case, which explains the Lorenz curve of $\epsilon = 1.6$ and $\epsilon = 2$ overlap with each other, and their decentralization coefficient equals to the upper limit $\frac{n-1}{n} = 0.8750$ (when $n = 8$). A comparison between the voting power proportion and the decentralization coefficient, as indicated by the Lorenz curve of controlling ability, suggests that voting power distribution can partially reflect the decentralization performance of DAO voting. However, for a comprehensive evaluation, the decentralization coefficient provides a more accurate and reliable measure. The distribution of voting power plays a crucial role in the decentralized performance of the voting system. As the concentration of voting power increases, the system tends to centralize, potentially undermining the ideal of decentralized decision-making. Conversely, a more decentralized distribution of voting power promotes a higher degree of decentralization.

Remark 4. *In general, as the concentration of voting power increases, the DAO voting tends towards centralization and vice versa.*

Impact of Participating Rate

Participation plays a crucial role in the controlling ability and overall decentralization performance of DAO voting since the participation of a DAO voting will naturally affect the result of the voting. In most cases, a significant voting power proportion can motivate voters to actively participate in the voting process. Conversely, voters with low voting power might feel their preferences have little influence on the voting outcomes, leading to reduced engagement and potential withdrawal from voting. Overall, there is a positive correlation between voting participation and voting rights. To analyse the impact of participation on the decentralization performance of typical voting cases, a logarithmic relationship is assumed in the simulation to describe the tendency of participation rate affected by the voting power,

$$p_i = log_2^{v_i+1}. \tag{4.16}$$

(4.16) presents a method that estimates the probability of a voter's participation in voting using the proportion of their voting power. The positive correlation between voting power and participation, as empirically observed in [152], supports our theoretical assumption of a monotonic relationship. Our model further refines this by incorporating a logarithmic function, which aligns with the principle of diminishing marginal returns on incentives [153].

To demonstrate the decentralization performance with the participation rate taken into account, three voting power distribution cases are considered as examples: the low decentralized case ($\epsilon = 0.5$), the moderately centralized case ($\epsilon = 1$), and the highly centralized case ($\epsilon = 1.5$). Fig. 4.5 illustrates the controlling abilities and their Lorenz curves, along with the denoted decentralization coefficients. Comparing the voting cases of the full participation RM and the RM with possible abstains, it is obvious from the comparison in Fig. 4.5 (a) that all the voters in the full participation cases in low and moderate centralized voting power distribution cases have a higher value on the *Controlling Ability*, which means they have higher control to the voting. However, when the voting power of the dominant voter is over half of the voting power, all the non-dominant voters have a slightly higher controlling ability to vote in the possible abstain case, and the situation is contrary for the dominant voter. This is because the possible abstain of the dominant voter gives a small chance to the non-dominant voter's voting power to become effective.

Remark 5. *Besides the extreme centralized cases with a dominant voter holding more than half of the voting power, higher participation can increase the influence of a voter on voting.*

From the comparison in Fig. 4.5 (b), it is observed that in cases where the voting power distribution is low centralized or moderately centralized (no voter has more than half of the

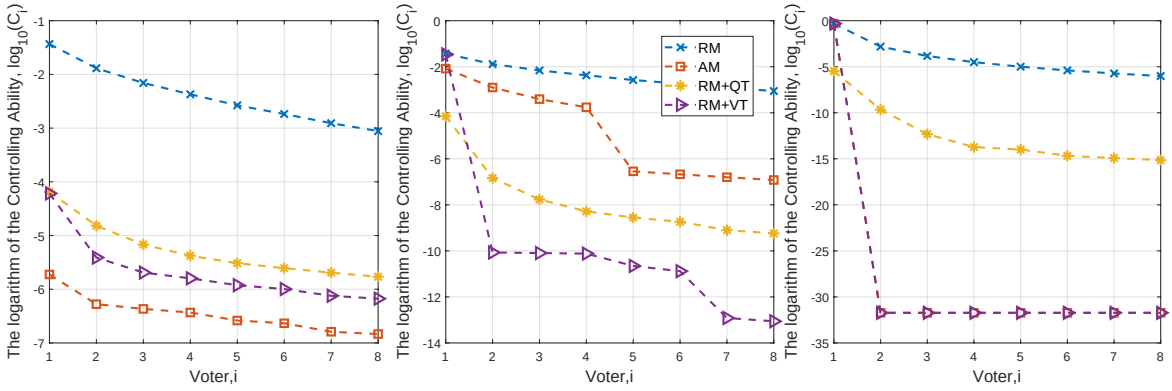

Figure 4.6: The Controlling Ability of four different voting mechanisms with low centralization voting power distribution $\epsilon = 0.4$ (a: left), moderate centralization voting power distribution $\epsilon = 1.2$ (b: middle), and high centralization voting power distribution $\epsilon = 2$ (c: right)

voting power) with participation taken into account, the decentralization coefficients become larger than those with full participation, which means the voting participation habit weakens the decentralization performance. This finding is reasonable since highly motivated large voting power holders may gain more control over the voting process, while lower engagement among low voting power holders can lead to reduced decentralization. The only different case is when a dominant voter has more than half of the voting power (highly centralized case), the decentralization coefficient decreases, and the decentralization performance slightly increases. However, the increase is attributed to the dominant voter having a small chance to withdraw voting. Regardless of the small chance of the withdrawal of the dominant voter, the lack of full participation of low voting power voters consistently leads to a decline in the overall voting decentralization performance. Their abstention results in more controlling ability being concentrated in the hands of dominant voters.

Considering participation rates is crucial when evaluating decentralization performance. It reveals the intricate interplay between voting power distribution and engagement levels, emphasizing the importance of fostering active participation among all voters to maintain a decentralized decision-making process.

Compare Voting Process with Different Approval Conditions

Although voting power allocation and participating rate largely affect the decentralization performance of DAO voting. However, the primary design flexibility lies in the voting process since the distribution of tokens and the participating rate of each voter are not easily controlled in the voting mechanism design. A well-designed process can contrast or relieve the dominant control caused by token allocation to some extent. The three typical voting processes, the AM, the RM with the QT, and the RM with the VT, are claimed to boost the security of the voting by getting adequate voters involved in the voting, which also improves

the decentralization performance compared to the essential widely use voting process with the RM as the only approval condition to some extent. However, it is not reliable to analyse the possible benefits of an optimized voting process only from a qualitative point of view. Surprisingly, the analysis of the simulation results in this section found non-consistent results in the decentralization performance of these advanced voting processes compared to some qualitative discussions. For comparison purposes, RM is considered the fundamental voting process, while AM, RM with QT, and RM with VT are regarded as advanced voting processes.

Fig. 4.6 compares the controlling abilities of voters with low centralization ($\epsilon = 0.4$), moderate centralization ($\epsilon = 1.2$) and high centralization ($\epsilon = 2$) voting power distribution. Fig. 4.7 compares the performance of the *Decentralization Coefficient* of the four voting mechanisms when $n = 8$, and $\epsilon = [0, 0.4, 0.8, 1.2, 1.6, 2]$ respectively. The values of the thresholds of the QT and the VT, in this case, are $n/2$ (half of the number of voters) and 0.5 (half of the total voting power), respectively. The values of *Decentralization Coefficient* of the four voting mechanisms cases are illustrated in Fig. 4.7. In general, comparing the controlling ability values of each voting process, the RM always has the highest controlling ability in all cases, which implies that RM is the easiest voting process to control for all voters. The *Decentralization Coefficient* always increases as ϵ increases, which confirms the analysis in Sec. 4.3.5 that the more centralized the voting distribution is, the worse the impact on the overall decentralization of voting will be. However, interesting findings are illustrated in the comparison of different voting processes. The values of the *Decentralization Coefficient* of the advanced voting processes are higher than the essential case RM in a large number of cases, which is not consistent with the intuition of these so-called advanced approval conditions. The detailed analysis is as follows.

AM: Based on Fig. 4.7, it can be observed that the *Decentralization Coefficient*, denoted as D, for the AM scheme is lower than that of the RM scheme when the value of ϵ remains below 1.2. Conversely, for the cases of ϵ exceeding 1.2, the D value for the AM scheme surpasses that of the RM scheme. This implies that the AM scheme only exhibits improved decentralization performance in scenarios where voting power distribution is moderately centralized or low. However, in highly centralized voting power distribution cases, the AM scheme's decentralization performance deteriorates in comparison to the RM scheme.

Based on the aforementioned simulation results, it becomes evident that the trend can be attributed to the following reasons. The AM demands greater attention in the voting process compared to the RM. This is achieved by requiring more than half of the voting power for a proposal to pass, unlike the RM scheme, which has no requirement for voting engagement. However, the attention requirement only benefits decentralization in voting

when there is no obvious dominant voter. When the voting power is distributed relatively equally among voters, this scheme ensures a broader representation of voter preferences, which improves the decentralization performance. However, when the dominant voter exists (especially when the voter holds more than half of the voting power), the AM scheme actually benefits the dominant voter since the AM still can easily determine the voting results with the overwhelming voter power and not obstructed by the requirement of half of the voting power requirement in AM. Accordingly, the other voter with a very small amount of voting power can hardly ever make an effectual opposite voting choice to the preference of the dominant voter. Especially when all of the other voters' voting power is less than half, they can never achieve a valid approval choice as the AM scheme mandates a greater-than-half threshold for approval. The new finding regarding the AM is concluded in the following remark.

Remark 6. *The AM scheme improves decentralization in low or moderately centralized voting power distribution but falters against the RM scheme in highly centralized cases.*

RM with QT: The voting process with RM and QT is adding an attention threshold QT to the essential decision scheme RM, requiring the least amount of quorum to be involved in the voting. According to the simulation result illustrated in Fig. 4.7, surprisingly, the value of D is quite close in low centralized voting power distribution cases, while in the highly centralized voting power distribution cases, the value of D for RM with QT is obviously larger than that of RM alone. This reveals that QT does not benefit decentralization performance and even worsens performance in highly centralized voting power distribution cases. The results from the simulation are contrary to the function suggested in some of the voting process designs with the QT as the QT commonly aims at getting adequate voters involved in the voting and, along with some of the discussion that it can help with the decentralization feature to some extent. However, with our simulation results, the qualitative discussion may overlook the factor of participation. When the voting power is relatively equally held by the voters, their participation rates are also similar, which results in a similar chance among voters to influence the voting result compared to the essential case with RM alone. However, when the voting power distribution becomes largely centralized, the incline to participate in the voting among voters becomes considerably different. The voter with a large amount of voting power is more likely to participate in the voting, which means they are more likely to appear in the quorum that passes the QT condition. Consequently, the high motivation of dominant voters to participate increases their influence on the voting result. Although RM has similar decentralization performance with the RM with QT in the low centralized voting power distribution cases, the controlling ability of all voters in the RM with QT is much lower than that in the RM in the same cases according to Fig 4.6, which implies that the extra

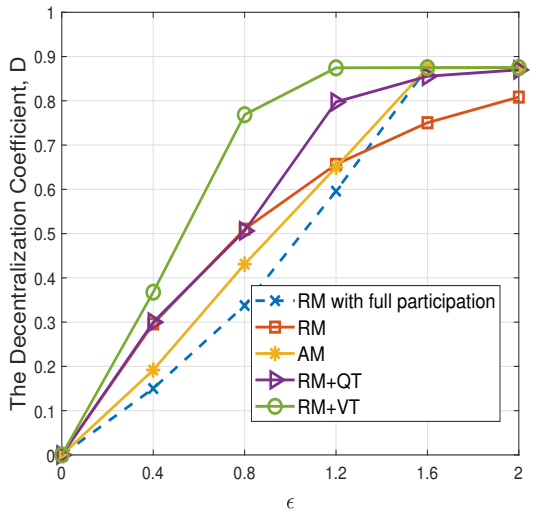

Figure 4.7: The Decentralization Coefficient of different voting mechanisms

approval condition QT does not improve the decentralization performance but largely limit the control of voting of all voters. The new finding regarding the RM with QT is concluded in the following remark.

Remark 7. *The RM scheme with QT demonstrates similar decentralization performance to RM alone in cases of low centralized voting power distribution. However, in highly centralized scenarios, the decentralization performance of RM with QT diminishes, primarily due to variations in participation rates.*

RM with VT: RM with VT is adding the limit of the least amount of voting power involved in the voting. However, in the case of the limit of 50% voting power, the RM with VT has the worst decentralization performance compared to the other three voting processes based on the results in Fig. 4.7. The effect of the VT is similar to the effect of the AM since they all have extra requirements on the least of voting power involvement. However, the difference is that the voting power involvement of AM is directly set for the approval of voting power, but the VT only generally targets all the participation no matter whether it is approved or against. Therefore, similar to AM, the approval condition tends to encourage the involvement of a large amount of voting power holder since their involvement makes the proposal easily pass the VT. Within the cases that fulfil the VT, there is a large probability that a large amount of voting power holders participate in the voting. Consequently, the voting result is more likely to follow the dominant voters' preference. Therefore, the VT actually benefits the voters with a large proportion of voting power and deteriorates the voters with a small proportion of voting power, and this impact is obvious in all levels of centralization of the voting power distribution.

Remark 8. *The voting process of the RM with VT generally worsens the decentralization performance compared to the RM in all levels of centralization.*

4.4 Quantify Efficiency Performance

In conventional voting systems, voting efficiency is commonly evaluated based on voter turnout, processing time, resource utilization, accuracy, transparency, accessibility, cost, simplicity, etc [154]. However, DAO infrastructure providers, such as DAO stack [102] provide automated, simplified, low human maintenance voting systems that leverage blockchain technologies such as smart contracts and DLT [155]. This result in highly efficient resource utilization, transparency, accessibility, cost and simplicity. Therefore, when analysing efficiency in DAO voting, the main focus is on voter turnout, processing time and accuracy. Regarding voter turnout, decentralized and non-hierarchical management, and voluntary voting participation in DAOs bring greater obstacles to motivating voter participation. In terms of processing time, since smart contracts automate the voting process, the primary time factor depends on the voting period. Determining the right voting period, one that's sufficient but not excessive is a crucial aspect of the efficiency of DAO voting mechanisms. Due to the uncertainty of voter turnout numbers and frequency, a lower voter participation probability may necessitate a longer voting period, and vice versa. The accuracy of voting refers to the voting results accurately showing the preference of voters. In a two-option (yes/no) voting system, an accurate approval rate should be 50% if all voters are assumed to be neutral. A close-to-accurate approval rate is likely to have adequate voter turnout, which means the participation probability and the voting period both affect the approval rate.

To gauge the efficiency analysis of the DAO voting mechanism in terms of voter turnout, processing time and accuracy, a model that analyses approval rates of different voting mechanisms regarding the participation probability and the voting period duration is built. We assume that n voters are valid for participating in the voting, with the voters are indicated in the set $\mathbf{\Omega} = \{N_1, N_2, N_3, ..., N_n\}$. The amount of tokens own by each voter i ($i = N_1, N_2, N_3, ..., N_n \in \Omega$) is denoted as t_i, while the voting power proportion is v_i. Then, the voting power proportion owned by each voter can be indicated in the set $\mathbf{V} = \{v_1, v_2, v_3, ..., v_N\}$.

4.4.1 Participation Behaviour in a Poisson Process

Since voting is not compulsory and voters voluntarily participate in the voting, we conduct a preliminary analysis of the participation behaviour of the voters. We denote the probability of voter i participate in the voting as P_i^p. Our assumption is that a voter will participate in the

voting for an open-for-voting proposal once they engage with the proposal pool in the DAO platform. This implies that the frequency with which a voter checks the DAO platform and reviews current proposals is a crucial factor influencing their participation probability P_i^p. To model this behaviour of checking and responding to open voting proposals, we assume the probability of the voter i engaging with the proposal pool during the very short time interval Δt is a constant, named as engagement probability, denoted as λ_i ($\lambda_i \in (0,1)$), and the probability of multiple engagements happening in Δt is negligible. During the voting period t, we represent the number of times voter i checks and engages with the proposal pool as $E(t)$. We denote the probability of exactly k engagements as $P(E(t) = k)$. As this process conforms to a Poisson process, we can express $P(E(t) = k)$ as a Poisson distribution as

$$P(E(t) = k) = \frac{e^{-\lambda}(\lambda_i t)^k}{k!}. \tag{4.17}$$

We assume that once a voter have checked the open-for-voting proposal at least once, the voter will participate in the voting. Therefore, the only case in which a voter does not participate is when $E(t) = 0$, while in all other cases, we assume the voter participated. Accordingly, the participation probability of a voter i can be expressed as

$$P_i^p(t) = 1 - P(E(t) = 0) = 1 - e^{-\lambda_i t}. \tag{4.18}$$

Based on (4.18), we can clearly observe that increasing both engagement probability λ_i and voting period duration t enhances participation probability P_i^p, consequently improving the efficiency metric voter turnout performance. However, shorter voting period duration is also remarked as an efficiency performance metric, Thus, (4.18) reveals a distinct trade-off between a shorter voting period and higher participation probability in pursuit of greater efficiency.

4.4.2 The Approval Rate

To calculate the approval rate of the voting, we build a stochastic model of the voting process. The behavior of each voter can be decomposed into two steps, choosing whether to participate in voting and deciding to vote for (yes) or against (no) the proposal. In the initial participation step, we assume that for a group of voters, the complete set of all possible combinations of participated voters is denoted as the full sample space $\mathbf{S^{full}}$. We define the sub-sample space, $\mathbf{S^p}$ ($\mathbf{S^p} \subseteq \mathbf{S^{full}}$), consisting of participating voter sets that have the potential to approve the proposal with their participation. We refer to this sub-sample space as the participating-valid sample space. Note that the final decision on whether the proposal passes or not ultimately

depends on the specific voting choices made.

Each element of the participating-valid sample space $\mathbf{S^p}$ is a set with participated voters, denoted as M^p. The participating choice of each voter i is represented as a random variable X_i, $X_i = \{1,0\}$, representing participation or non-participation respectively. $X_i \sim b(1, P_i^p)$ follows the Bernoulli distribution where the participation probability, P_i^p is defined in (4.18). The participating choices of all voters form the participating behavior profile represented as an n-tuple $\mathbf{X} = \{X_1, X_2, ..., X_n\}$. Thus, $M^p = \{i | i \in \mathbf{\Omega}, x_i = 1\}$. The probability that all the voters in a set M^p participated in the voting can be expressed as

$$Pr(M^p) = \prod_{a \in M^p} P_a^p \prod_{b \in \complement_{\mathbf{\Omega}} M^p} (1 - P_b^p). \tag{4.19}$$

For each participated voter, two voting choices are available to them, i.e, for or against. In this second step, we define a sample space $\mathbf{S^f}$ where each event in the sample space $\mathbf{S^f}$ is denoted as M^f. Each event M^f represents a set of voters from a same participating voter set M^p who voted for (yes), and having in favour votes from all voter in M^f ($M^f \subseteq M^p$) can lead to the approval of a proposal. Such sample spaces $\mathbf{S^f}$ are named as voting-valid sample spaces. It is important to note that each participant set M^p corresponds to a voting-valid sample space $\mathbf{S^f}$, because the sets of voters who voted for (yes) within the same voting-valid sample space $\mathbf{S^f}$ all originate from the same participating voter set M^p.

Similarly, the voting choice of each voter i can be denoted as a random variable Y_i, and $Y_i = \{1,0\}$, indicating the for and the against choices, respectively. Each voting choice $Y_i \sim b(1, q_i)$ follows the Bernoulli distribution, and q_i indicates the probability of voter i to vote, namely, the probability of voting preference. Thus, $M^f = \{i | i \in \mathbf{\Omega}, y_i = 1, x_i = 1\}$. The voting choices of all voters form the voting behavior profile represented as an n-tuple $\mathbf{Y} = \{Y_1, Y_2, ..., Y_n\}$. With above definition, the conditional probability that all the voters in set M^f voted for when voters in set M^p participated in the voting can be expressed as

$$Pr(M^f | M^p) = \prod_{a \in M^f} q_a \prod_{b \in \complement_{M^p}^{M^f}} (1 - q_b). \tag{4.20}$$

Therefore, the probability the proposal pass with voters in M^p participated in the voting and voters in M^f voted for can be expressed as

$$Pr(M^p, M^f) = Pr(M^p) Pr(M^f | M^p). \tag{4.21}$$

We consider the voting result, which is whether the proposal is passed or not, is denoted as a random variable following the Bernoulli distribution as $R \sim b(1, P^A)$, where P^A indicates

the approval rate of the proposal. The proposal approval rate is the sum of the probability of all cases in the participating-valid sample space $\mathbf{S^p}$, and each voting-valid sample space $\mathbf{S^f}$, which is expressed as

$$P^A = \sum_{M^p \in \mathbf{S^p}} \sum_{M^f \in \mathbf{S^f}} Pr(M^p, M^f). \tag{4.22}$$

Since the sample space $\mathbf{S^p}$ and $\mathbf{S^f}$ cannot be defined unless the voting process, approval criteria and the token distribution are clear. To provide specific analysis of the sample spaces, we discuss in typical voting procedures as follows.

4.4.3 The analysis of valid sample spaces

To express sample spaces $\mathbf{S^p}$ and $\mathbf{S^f}$, the critical voter clusters including the participated voter sets, the abstained voter sets, the voter in favour sets and the against voters sets and their corresponding number and voting power have to be defined, since they are widely used to judge whether the approval conditions in a voting process are met.

According to assumption of the system model, the participating choice and the voting choice of each voter i are denoted as the random variable $X_i = \{1, 0\}$, $Y_i = \{1, 0\}$ respectively. Therefore, the critical values used to check whether an approval set satisfies the approval conditions can be summarized as TABLE 4.4.

Table 4.4: Number and voting power of critical groups

Critical Groups	**Number of Voters**	**Sum of Voting Power**
Participated voters	$\sum_{i \in \Omega} x_i$	$\sum_{i \in \Omega} x_i v_i$
Abstained voters	$\sum_{i \in \Omega} (-(x_i - 1))$	$\sum_{i \in \Omega} (-(x_i - 1))v_i$
In favour voters	$\sum_{i \in \Omega} x_i y_i$	$\sum_{i \in \Omega} x_i y_i v_i$
Against voters	$\sum_{i \in \Omega} x_i(-(y_i - 1))$	$\sum_{i \in \Omega} x_i(-(y_i - 1))v_i$

According to the voting process in Fig 4.3, approval conditions are added in sequence during the voting process. We denote each approval condition as A_j where j refers to its sequence in the voting process. To find all the element participating sets in the participating-valid sample space, each participating set must satisfy all the approval conditions if we assume all the voters in participating rate voted in favour. To be consistent with the pass (1) or fail (0) denotations in the voting result set R, we defined the fulfilled approval condition

$A_j = 1$, while a fail to achieved approval condition $A_j = 0$. The sign function defined in (4.4) encode the results into binary numbers.

With the number and the voting power from typical groups, most approval conditions can be expressed. We provide four examples of typical approval conditions introduced in Sec. 2.3.2 and listed in TABLE 4.5.

Table 4.5: Encoded Expressions of Approval Conditions

Approval Condition	**Expression**
RM	$A_{rm} = S(\sum_{i\in\Omega} v_i x_i y_i - \sum_{i\in\Omega} v_i x_i(-(y_i - 1)))$
AM	$A_{am} = S(\sum_{i\in\Omega} v_i x_i y_i - 0.5)$
VT	$A_{vt} = S(\sum_{i\in\Omega} v_i x_i - h_v)$
QT	$A_{qt} = S(\sum_{i\in\Omega} x_i - h_q)$

h_v is the value of the voting power threshold.
h_q is the value of the quorum threshold.

To find the valid approval sets, all the approval conditions in the voting process should be satisfied. An approval function, denoted as F is defined to determine whether the approval sets are valid in a specific case. To indicate the final voting result, the approval function, F, can be expressed by all the encoded approval conditions in the voting process connected by logical conjunctions as

$$F(M^P, M^f) = A_1 \wedge A_2 \wedge ... \wedge A_j, \tag{4.23}$$

where $\wedge$ is the logical conjunction, **AND** operator.

For each elements in $\mathbf{S^p}$ and $\mathbf{S^f}$, when $F(M^P, M^f) = 1$, the pairing elements in M^P and M^f are verified as approval sets. When $F(M^P, M^f) = 0$, the pairing elements are not approval sets. Therefore, in the approval function, the sample spaces $\mathbf{S^p}$ and $\mathbf{S^f}$ can be expressed as $\mathbf{S^p}, \mathbf{S^f} = \{M^P, M^f | F(M^P, M^f) = 1\}$.

4.4.4 Interrelationship of Efficiency Factors

A high participation probability, a short voting period, and a close-to-neutral approval rate are key performance indicators of a highly efficient DAO voting mechanism. The interrelationships of these three factors are separately analyzed.

The Relationship Between Voting Period Duration and the Participation Probability

(4.18) shows a negative correlation between increasing the participation probability P^p and decreasing the voting period duration t. This suggests that there is a trade-off between the efficiency requirements for processing time and voter turnout if the engagement rate is constant.

The Relationship Between the Participation Probability and the Approval Rate

(4.22) is a general equation to calculate the approval rate P^A in all situations with a general assumption of the participation probability. Since each voter has their own participation probability, the combined effect of these probabilities from all voters on the overall voting process is not singularly dependent. To focus on the efficiency analysis of the voting mechanisms, as the proposal approval rate does not refer to any specific proposal, we assume each voter's voting preference is neutral to all proposals, which means $q_i = 0.5$ for all voters. Under this assumption, an accurate outcome with entirely neutral voting preferences should yield a 50% approval rate. This scenario occurs in the case of full participation, where all voters have $P_i^p = 1$, as described by (4.22). Therefore, we infer that a higher participation probability leads to a closer approximation of a neutral approval rate, indicating a positive relationship between voter turnout and accuracy in achieving high efficiency. The further verification is provided through simulations in Sec. 4.4.5.

The Relationship of the Voting Period Duration and the Approval Rate

An interdependent relationship between the approval rate P^A and the voting period t can be uncovered by substituting the participation probability P^p with its function in terms of the voting period, denoted in (4.18). With (4.18), we can extend the relationship from the participation probability P^p to both the engagement probability λ and the voting period t in the context of the approval rate P^A. Then, we yield

$$P^A(t) = \sum_{M^p \in \mathbf{S^p}} \sum_{M^f \in \mathbf{S^f}} \prod_{a \in M^p} (1 - e^{-\lambda_a t}) \prod_{b \in \complement_{\mathbf{\Omega}} M^p} e^{-\lambda_b t} 0.5^{|M^p|} \tag{4.24}$$

To fulfill the high efficiency, both a shorter voting period (t) and an adequate voting approval rate (P^A) are required. According to (4.24), we can observe that an increase of t can lead to an increase of P^A, and vice versa. This signifies the presence of a trade-off between these two efficiency metrics. Detailed analysis of this trade-off is shown in Sec. 4.4.5.

4.4.5 Simulation Analysis

As the voting power proportion and the engagement probability are required in simulation, we make assumptions for them. Given the observed centralization trend in the distribution of voting power in most DAOs [146], the $Zipf$ distribution is applied to describe the proportion of voting power featured as dominant controlled, v_i held by each voter. Let $V \sim Zipf(\alpha, n)$,

$$v_i = \frac{1}{i^\alpha \sum_{k=1}^{n} (1/k)^\alpha} \quad i = 1, 2, ..., n \tag{4.25}$$

where α ($\alpha \geq 0$) denotes the centralized level of the voting power allocation, i is the sequence of voters, and n is the number of voters. The larger the α, the more dominant the voting power distribution is. It is also assumed that the engagement probability λ_i is positively correlated with the proportion of voting power v_i, as voters holding a larger share of voting power are generally more motivated to stay informed about the development of the DAO organization [110, 156]. Accordingly, the engagement probability λ_i and the voting power v_i are assumed to follow an approximately linear relationship.

$$\lambda_i(v_i) = \beta v_i + \epsilon, \tag{4.26}$$

where ϵ denotes the residual term. Since an increase in β leads to higher voter turnout, β is referred to as the *voter turnout indicator*.

Since (4.18) indicates a direct correlation between the voting period duration t, and the participation probability P^p, this analysis primarily focus on simulating the correlation between the participation probability and the approval rate, as well as the correlation between the voting period duration and the approval rate.

Simulation of the Participation Probability and the Approval Rate

According to (4.18), with the same t, an increase in the value of λ_i can be assumed to correspond to an increase in P_i^p. To simulate various levels of participation probabilities, six sets of λ_i values are defined by adjusting the voter turnout indicator β, with $\beta = [0.5, 1, 1.5, 2, 2.5, 3]$, based on equation (4.26). Figure 4.8 presents the resulting approval rates P^A for four common DAO voting mechanisms: RM, AM, RM with QT, and RM with VT. These approval rates are plotted against the voter turnout indicator β, with $n = 6$ and $\alpha = 0.5$. It is evident that an increase in the voter turnout indicator consistently leads to an approval rate closer to the neutral result of 0.5. This observation affirms the critical positive correlation between participation probability and approval rate, emphasizing the need to coordinate these factors to achieve high DAO voting efficiency.

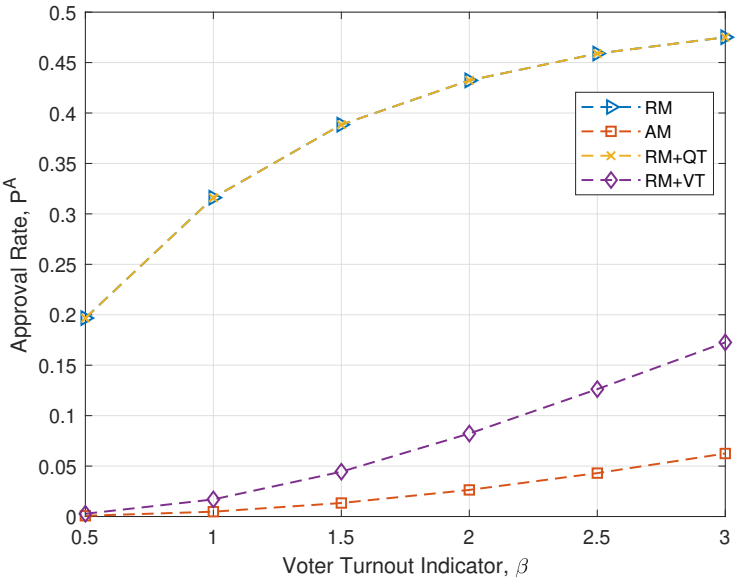

Figure 4.8: The approval rate versus the voter turnout indicator when $n = 6$, $\alpha = 0.5$

Simulation of the Voting Period Duration and the Approval Rate

Fig. 4.9 compares the performance of the approval rate to the voting period duration t of the four typical voting mechanisms when $n = 8$, with $\alpha = 0.5$. The values of the thresholds of the QT and the VT in this case are $n/2$ (half of the number of voters) and 0.5 (half of the total voting power), respectively. Based on the simulation results in Fig. 4.9, the approval rate converges to 0.5, which is recognized to be accurate, as t grows. Therefore, to achieve an accurate approval rate within a short voting period, an optimal balance between these two metrics can be identified, which defines an efficient voting process.

4.5 Future Work

Future validation of the proposed models and the SEED framework can be conducted using real governance data. An on-chain/off-chain dataset may be constructed by parsing public governance contracts (e.g., GovernorBravo-style) and Snapshot spaces, collecting for each proposal voter addresses, delegated voting power at the snapshot block, casting timestamps, outcomes (passed/failed/executed), and protocol parameters (quorum/threshold/delegation rules). Based on these data it is possible to (i) estimate the Zipf exponent ϵ (per proposal or per epoch) via maximum-likelihood fitting and assess the goodness-of-fit for a rank-based power law; (ii) derive metrics such as Efficiency (proposal lifetime from creation to execution and average casting latency) and Decentralization (rank-inequality/entropy of voting weights) directly from raw logs; (iii) test the simulation predictions by correlating ϵ with

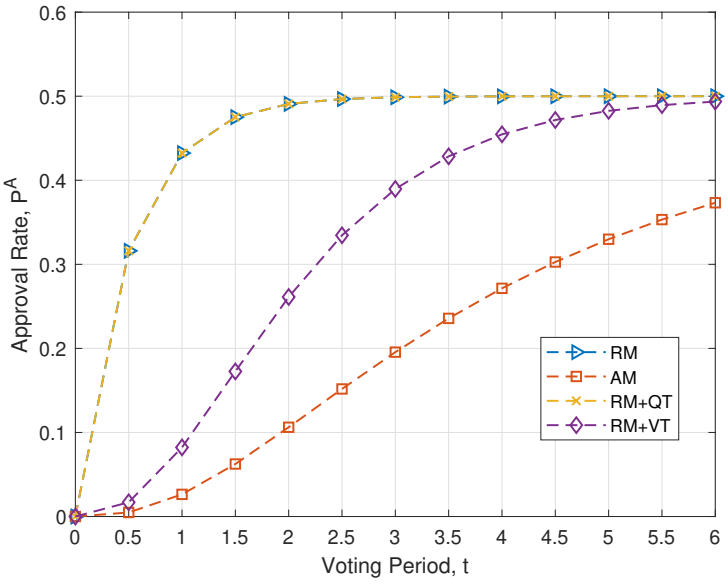

Figure 4.9: The approval rate of typical voting mechanisms when $n = 8$

SEED metrics using cross-DAO panel regressions (DAO and time fixed effects) and by conducting out-of-sample checks across proposals with electorates ranging from small (e.g., 8 voters) to very large (e.g., 8000 voters) to evaluate the generalisability of the observed trends.

4.6 Conclusion

This chapter examined voting as a fundamental mechanism of consensus in DAOs, emphasizing its structural role alongside smart contracts and distributed ledgers. The DAO governance triangle clarifies the interdependence among voting, smart contracts, and distributed ledgers, and emphasizes the irreplaceable role of human decisions in adapting to uncertain or evolving contexts. Based on this foundation, the SEED metric system is proposed to evaluate decentralization from multiple dimensions. Building upon these concepts, a stochastic model of DAO voting is developed to capture the dynamics of individual participation and power distribution. The model enabled us to define the Decentralization Coefficient, providing the first quantitative method for measuring the decentralization performance of DAO voting mechanisms. This framework offers both a theoretical tool and practical guidance for analysing, comparing, and designing DAO voting schemes with greater clarity and rigour.

Chapter 5

A Consensus Framework: From Human to Machine

In the introduction, the growing need to rethink consensus in light of emerging systems that integrate both human and machine agents was highlighted. As intelligent tasks and distributed coordination become more complex, traditional domain-specific mechanisms are no longer sufficient. To better support interdisciplinary analysis, this chapter proposes two discussion points: **Can the indispensable fundamental elements that primarily affect the consensus process across both human and machine systems be identified in a unified way? And what are the primary objectives of a consensus process, and what are the primary factors affecting the achievement of consensus?** To answer these two questions, a unified framework is established that spans from society to technology to discuss the common characteristics of consensus. Within this framework, three indispensable and fundamental components applicable to consensus analysis across various fields are proposed: participants as the carriers of consensus, communication as the bridge, and state descriptions as markers of the transformation from chaos to consistent cognition. The terms "algorithm", "protocol", and "mechanism" are used interchangeably throughout this paper to describe the underlying processes of achieving consensus.

5.1 Consensus from Society to Technology

Before building a unified consensus framework, various consensus fields are analysed to compare well-known consensus. The classification of consensus types in this paper depends on the participants involved. Consensus is categorized into Human Consensus (HC), characterized by human participants; Machine Consensus (MC), marked by computational entities such as computer nodes, agents, or intelligent machines; and a promising new emerging form,

Hybrid Consensus (HBC), which involves interactions between humans and machines.

5.1.1 Human Consensus (HC)

HC is typically studied as a process that leads a group of people within a social network to reach an agreement on various issues, ranging from social norms to political affairs. While early discussions of consensus in social sciences focused on theoretical understandings of group dynamics and social cohesion, it was not until the mid and late-20th centuries that specific methodologies for forming consensus among people were proposed. Notable among these mechanisms are the Delphi Method, Nominal Group Technique (NGT), and Round-Robin Discussion [6]. These methods are commonly used in fields such as public policy development, organizational management, and social research, where gathering diverse opinions and reaching a collective agreement are essential. While those mechanisms are primarily designed for structured environments, the HC also manifests in various voting systems and in the consensus state of culture, social norms, and ethical standards within social networks. TABLE 5.1 outlines the main characteristics of HC. Moreover, consensus is not exclusive to humans in the biological realm. It is also prevalent in animal behaviours such as migration, foraging choices, and the selection of group territories and defence strategies among birds, insects, fish, and mammals. In this work, the focus is placed on humans as consensus examples of living organisms.

5.1.2 Machine Consensus (MC)

MC refers to all forms of consensus exclusively involving machines, such as computer nodes, agents, robots, and other automated participants in a system that relies on a communication network to achieve consensus. Generally, there are three well-known MC domains: DFT consensus, Blockchain consensus, and MAS Consensus. DFT consensus plays a crucial role in distributed computing systems. Typically, Byzantine Fault Tolerance (BFT) algorithms are designed to achieve consensus on a valid value within a system that tolerates a limited number of malicious nodes. Crash Fault Tolerance (CFT) algorithms, on the other hand, only assume a limited number of nodes that may crash but no malicious behaviour. Blockchain technology, which emerged later, adopts similar principles of consistency from DFT but introduces innovations in achieving consensus. This is particularly evident through a consistency log maintained as a “chain" in a widely accessible network. The third type, MAS consensus, adopts a distinct approach to consensus compared to the other two. Operating under the assumption of autonomous agents, its key algorithms, such as average consensus and interactive consistency, draw parallels to consensus behaviours observed in biological populations

[10], which enable MAS systems to reach reliable consensus outcomes through cooperative decision-making among agents. TABLE 5.1 outlines the main characteristics and comparisons of MC.

5.1.3 Hybrid Consensus (HBC)

While DFT and blockchain consensus mechanisms prioritize consistency and fault tolerance, MAS consensus focuses on adaptive decision-making and collaborative problem-solving, reflecting the dynamic interaction and collective intelligence of autonomous agents. This makes MAS consensus research extend to higher intelligent agent consensus with the recent rise of AI technology. The autonomy and intelligence level of machines holds immense potential to operate in an unframed and versatile goal mode, similar to human behaviour. The intelligent upgrade of nodes and the increasing demand in the environment have revealed the fusion goal of integrating fault tolerance and preferences, which is commonly recognized in HC. On the other hand, as highly intelligent machines become integrated into human life and take on more flexible work, the interaction and cooperation between high-intelligence machines and humans are inevitable [20]. Humans and machines generally have relatively obvious independence in HBC scenarios. The logic of consensus can be specified in advance by algorithms or autonomously negotiated in a relatively open environment. The following is an example of an existing HBC. In an L3 autonomous driving system, when an unexpected obstacle is encountered during driving, humans and autonomous driving car agents negotiate decisions. The vehicle system and the human driver each retain independent decision-making perspectives. The vehicle agent AI will provide optimization solutions based on the perception data, and humans will select or fine-tune these solutions based on situational awareness and ethical considerations that go beyond pure data analysis. Ultimately, the consensus is not simply reached by letting people or machines take over completely but by cooperating within a safe and controllable range. Cases like this will blur the boundary between consensus paradigms of machines and humans. Considering both technological and societal dimensions in consensus is very necessary. To build a dialogue of consensus from society to technology, a unified consensus system is established in the following sections to condense the common important elements and characteristics of different forms of consensus. Examples and their features of HBC are also listed in TABLE 5.1.

Table 5.1: Comparison of consensus from technology to society

Types	Examples of Consensus Forms	Main Participants	Consensus Object	Mechanism Examples
Human Consensus	Voting, Discussion	Humans	Defined/Implicit Proposals	Majority Vote, Delphi Method
Machine Consensus	Distributed fault-tolerance consensus Blockchain consensus Multi-agent system consensus	Nodes (Processes) Nodes (Miners) Autonomous agents	Client requests Transactions Actions, Decisions	PBFT, Raft PoW, PoS Average consensus, Voting
Hybrid Consensus	Healthcare decision support system, Autonomous driving network	Intelligent agents and Humans	Cooperation tasks, Shared data	Not clear

Table 5.2: Features of Participants

Consensus Type	Honesty Assumption	Cognitive Ability	Cognitive Differences
Human Consensus	Unpredictable	Very high	Moderate to very high
Ordinary Machine Consensus	Byzantine nodes: low Crashed nodes: Absolute honest Good nodes: Absolute honest PoX: Conditional (constrained by incentive mechanism)	Low	BFT: low CFT: low (Only crashed/Byzantine nodes have different cognition) PoX: generally low (the standard to verify a valid transaction among different nodes is highly consistent between nodes)
Advanced Machine Consensus	Undetermined	Moderate to high	Moderate to high
Hybrid Consensus	Undetermined	Moderate to high	Moderate to high

5.2 A Generalized Consensus Framework

Despite the diverse forms of consensus spanning from society to technology, commonalities in the consensus process can be observed, and the essential constraints can be summarized into three key elements: participants, state, and communication, which are the primary factors affecting the achievement of consensus. Based on these elements, a generalized consensus framework is proposed, as illustrated in Figure 5.1, laying the groundwork for a generalized consensus process. This framework explains that consensus is a transformation process from a chaotic initial state to a consensus state of significant agreement among participants regarding a consensus object, facilitated by an implicit or explicit consensus mechanism that depends on information exchange through communication networks. Essentially, the consensus process relies on participants utilizing the communication network to interact effectively and exchange information. The participants' characteristics (cognition and integrity), the communication network conditions, and the consensus state requirements largely determine the feasibility and method of achieving consensus. Therefore, clarifying these three elements is essential before establishing or selecting any specific consensus mechanism. Here, these three elements and their impact on the consensus system are explained in detail.

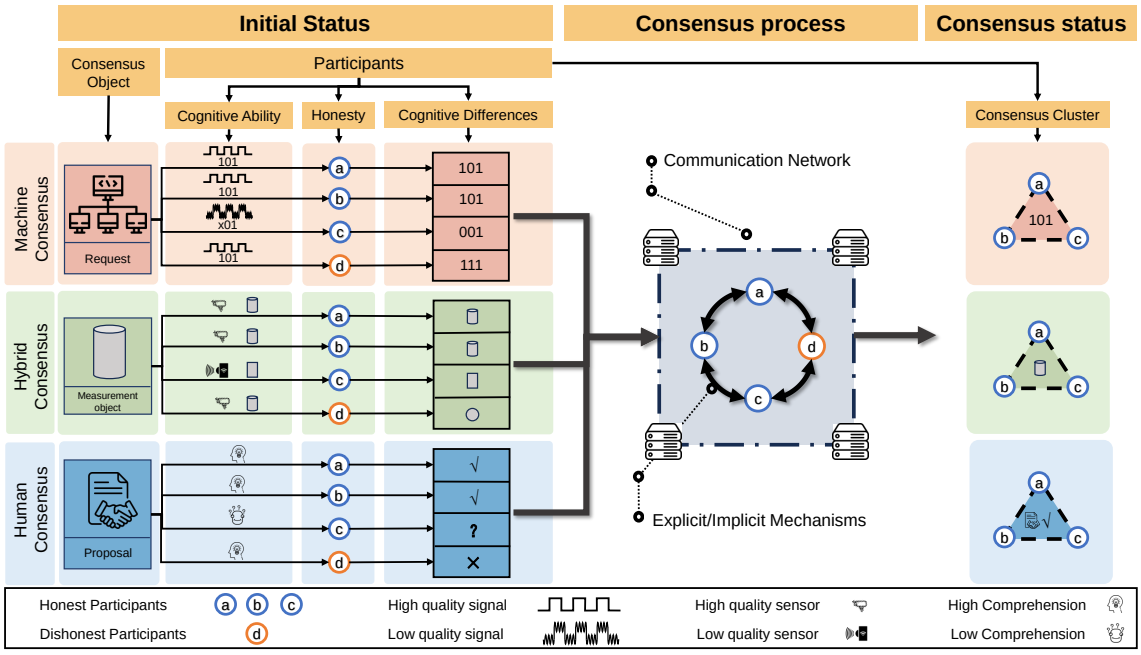

Figure 5.1: Consensus framework

5.2.1 Participants

Participants are the entities involved in the consensus process, acting as carriers of consensus. They can be human individuals, computational nodes, autonomous agents, or hybrid entities. The cognitive features of participants, along with their honesty, significantly influence the intricacy of the consensus process. The following subsections will explore them in detail and compare them across different types of consensus, as summarized in TABLE 5.2.

Cognitive Ability

Cognitive ability is an anthropomorphic expression of the consensus participants' ability to receive, understand, evaluate and transmit information and to make decisions based on received information in the consensus process. It is also strongly influenced by participants' openness to external assistance and sensitivity to environmental impacts. For a machine system, cognitive ability can refer to the capability to receive and process information, such as signal acquisition (e.g., sensor inputs) and data evaluation (computational analysis). For instance, each node aggregates and evaluates data from multiple sources, determines potential consensus targets, and adjusts its decision logic based on detection and inference processes. The cognitive abilities of participants significantly influence the complexity of the consensus process by introducing more intricate decision-making factors. For example, in human voting-based consensus, participants can employ tactical voting to conceal their true preferences in order to achieve a collective voting outcome that is closer to their own inclinations.

Here, MC is further categorized based on the cognitive abilities of the participants into two types: Ordinary Machine Consensus (OMC) and Advanced Machine Consensus (AMC). OMC includes consensus systems with low cognitive abilities, such as DFT and blockchain consensus algorithms, which typically offer simpler functionality focused on read-write consistency. AMC includes consensus systems with relatively high cognitive abilities, such as MAS consensus systems, particularly those involving intelligent agents. The cognitive abilities of agents in AMC rely on their level of intelligence but are generally higher than OMC due to their advanced functions, such as AI-enhanced sensing and complex decision-making. Generally, humans, as participants, show high cognitive abilities. The cognitive abilities of different consensus types are listed in TABLE 5.2.

Cognitive Differences

Cognitive abilities are relative to individual participants' measurement standards. Since consensus must be processed within clusters, the diversity in how participants process, respond to, and interpret information serve as a critical collective metric because eliminating cognitive

disparities towards consensus objects within these clusters is the primary goal of consensus. This metric is named as cognitive differences. Cognitive differences are influenced not only by the levels of participants' pre-existing knowledge but also by their distinct data-processing methods, characteristics, and operational modes for handling incoming information. Low cognitive differences are exemplified by DFT consensus systems, where the majority of nodes are non-faulty and exhibit highly consistent, programmed reactions to the consensus process. Conversely, nodes with different behaviours, such as crash nodes and Byzantine nodes, typically represent a minority and show cognitive inconsistencies. In cases of low cognitive differences, the priority of the consensus process is to reach an agreement by minimizing differences among participants regarding the consensus object and tolerating the minority of inconsistent participants. An illustrative example of high cognitive differences is a project meeting where participants have different priorities, with some focusing on profit and others on environmental considerations. These significant cognitive differences can pose substantial challenges to reaching a consensus. In such cases, consensus groups usually require more centralized decision-making methods, such as majority voting that relies on a central platform, to achieve consensus.

TABLE 5.2 presents the cognitive consistency comparisons of MC, HC and HBC. In HC, humans often exhibit high cognitive differences due to varied information processing and reasoning. HBC similarly tends toward high differences because humans and machines may prioritize decisions differently. Note that in scenarios involving clearly defined consensus choices, specific instances of HC may sometimes demonstrate moderate cognitive differences. Cognitive similarities in consensus can stem from shared consensus backgrounds within the group, such as culture, shared life experiences, etc. As for OMC, the cognitive differences of DFT consensus have been discussed. In most PoX-type blockchain algorithms, the standards for verifying a transaction between nodes are highly consistent due to the verification methods being standardized, showing low cognitive differences. AMC featured participants with higher levels of intelligence (such as intelligent agents) than those in MC but generally lower than humans in HC. In such systems, cognitive differences depend on the uniformity of participants' design features. For instance, if the same or similar algorithms are applied to identical agents, then their cognitive differences are low. However, due to the potential difference in agent systems and the varying designs of their operational mechanisms, diversities in cognition about the consensus object can arise. Therefore, overall, the cognitive differences in AMC are considered to be moderate to high.

Honesty

The consensus process involves information exchange among participants, where not only cognitive differences but also the integrity of participants significantly impact the outcome. Given that honesty assumptions vary widely across different scenarios, honesty is specifically addressed within the context of explicit mechanism examples. Conclusive comparisons are listed in TABLE 5.2, while detail analysis are as follows. The BFT algorithm tolerates a limited number of dishonest nodes by allowing some malicious behaviour. In contrast, the CFT algorithm assumes nodes may fail (crash) but do not act maliciously, implying overall honesty, even among failed nodes. In PoX blockchain consensus mechanisms, nodes are generally considered unreliable. However, these algorithms use incentives to encourage honesty and limit dishonest behaviour, making their honesty conditional. In HC and HBC systems involving humans, honesty is unpredictable due to strategic and complex thinking, such as tactical voting. In AMC and HBC, the assumption of intelligent agents may be defined case by case and waiting for further research. Therefore, honesty is not assumed in Table 5.2.

5.2.2 State

States represent the level of agreement in the consensus process, marking the transition from an initial state of disparity to a consensus state of agreement. These states are fundamental and universally applicable across all forms of consensus. The initial state can range from a set of clear preferences (e.g., agree/disagree, choosing among multiple options) to being entirely unclear and lacking a definite direction (e.g., in iterative or convergent consensus). For consensus state, different consensus systems have varying criteria around the consistency requirement. Common consensus states may include agreement on a single value, multiple valid values, or even a range of values. In addition to the state of the consensus object, the degree of consistency can also vary. The agreement could involve all participants, only non-faulty participants, a majority of participants, or even just key participants. Time requirements are also considered as a dimension of the consensus state, including whether the consensus process is a one-time event or continuous and whether there are specific time series requirements. Additionally, some consensus algorithms define strong and weak consistency. Strong consistency requires all participants to have the same view simultaneously, while weak consistency allows temporary discrepancies but ensures eventual convergence [7]. Moreover, in certain cases, consensus is designed to be achieved with a very high probability [13]. Table 5.3 illustrates the common initial state and consensus state of the process.

The consensus process, which converges from an initial state of disorder to a state of unanimity, can be viewed as a process of entropy reduction. The message transmission and

decision-making in the consensus process all contribute to this entropy reduction. This state transition process can occur through explicit methods, such as thoughtful negotiations or structured decision-making procedures. Often, these explicit processes are accompanied by well-defined consensus mechanisms, such as voting, DFT algorithms, or blockchain consensus algorithms. Alternatively, the process can occur through implicit methods, which evolve gradually from a shared culture or a tacit understanding of group behaviour without clear coordination.

Table 5.3: Initial state and consensus state

Category	**Details**
Initial state	Binary choice / Multiple choice Open proposal
Consensus state	Strong consistency / Weak consistency Single value / Multivalue / Range All the good nodes / All the nodes reach consistency Consensus in majority / Critical members Single consensus / Continuous consensus Time series requirement Deterministic / Probabilistic consensus

5.2.3 Communication

According to the state transition idea of consensus, information exchange is essential for establishing agreement among participants. Only through communication can participants interact and eliminate cognitive differences. Communication here refers to the means and channels through which participants exchange information during the consensus process. The environment of the communication network is subject to objective scene restrictions and can also be crafted during the consensus process to create more favorable channels for consensus. The mode of communication can significantly impact the feasibility, speed, consistency, and even the outcome of consensus formation.

Communication Across Different Consensus Types

In the process of HC, communication often relies on language interaction or the use of informational platforms. Some of these communication methods are straightforward and transparent, such as real-time messaging through person-to-person oral communication or modern digital channels like texting or online chatting. However, communication can also occur

through more complex and layered means, such as public media, culture expressions and arts. They convey messages for broader consensus in subtler and often more profound ways.

Furthermore, the integration of humans and intelligent machines in HBC introduces additional communication challenges. In addition to establishing a strong communication network infrastructure, this hybrid interaction also requires human-machine interaction technology to assist communication. For example, natural language processing enables the comprehension and generation of human language, facilitating seamless textual interactions. Speech recognition and synthesis technologies support verbal communication, allowing for natural and intuitive exchanges. Additionally, tactile feedback mechanisms provide a tangible dimension to interactions, enhancing the physical and sensory experience [19].

Table 5.4: Communication assumptions overview

Category	**Assumptions**
Timing	Synchronous/Asynchronous
Network Structure	Centralized/Distributed; Fully/Partially connected
Communication Mode	HC: Verbal/Non-verbal communication MC: Signal/Data transmission

Communication Assumptions

Here, some assumptions of communication are listed in TABLE 5.4. Under the timing assumption, synchronous communication among machines bears resemblance to live workshop discussion, where all the participants knows when the discussion starts and ends by the announcement of the chair. This entails the transmission of messages within predefined temporal constraints universally acknowledged by all participants. Put differently, each participant has a synchronized global time, enabling discernment that if a message remains undelivered within a specified time frame, it has not been dispatched at all. Conversely, in asynchronous scenarios, communication parallels distant individuals employing foot messengers, where the transmission may be delayed. Consequently, recipients are unable to distinguish whether the delay is attributable to the messenger's departure or the absence of a messenger.

The network structure of communication can both impose limitations and aid in forming consensus. For example, a fully connected topology facilitates the transfer of messages among participants, while a poorly connected structure may cause message blockages or singular channels for some participants, reducing the reachability of consensus messages. On the other hand, if the communication network architecture is highly centralized, it results in consensus messages being predominantly controlled by a central entity. For example, in

wireless communication, a centralized approach entails nodes transmitting their information to a central control station. This central station is responsible for making final decisions and sending instructions back for execution. Examples can also be found in HC, such as traditional news dissemination, which relies on public media. Although the centralized method often has higher consensus efficiency, the overall consensus outcome more significantly relies on the central entity, and this influence can be either positive or negative.

While the communication network builds a backbone, the communication mode decides how communication happens. Examples are retransmission mode and non-retransmission mode according to the Quality of Service (QoS) based on the inherent communication framework. Another example is granted communication requires explicit permission for entities to interact within a system, enhancing security, while non-granted communication allows open interaction without prior authorization, prioritizing ease of access and convenience.

5.2.4 The Consensus Process

Based on the proposed consensus three key elements, a perspective on the definition of the consensus process is proposed in Remark 9.

Remark 9. *The consensus process involves overcoming cognitive differences and dishonesty among participants, transitioning their cognition of the consensus object from chaos to significant agreement. Fundamentally, it aims to eliminate or reduce cognitive differences among members regarding the consensus object.*

Remark 9 is motivated by the fundamental need for consensus, which arises when participants have differing or unclear understandings of the consensus object. As discussed in Sec. 5.2.1, participants with significant cognitive differences face greater challenges in achieving consensus. Furthermore, participant dishonesty can negatively impact the dissemination of messages that facilitate consensus, thereby impeding the consensus process and introducing uncertainty. The primary objective of a consensus process is to eliminate or reduce cognitive differences and dishonesty among participants, guiding their cognition of the consensus object from disorder to significant agreement. This definition is exploratory and may reveal its incompleteness in future research. Nevertheless, it is hoped that this viewpoint can inspire research on consensus across various domains.

5.3 Consensus Strategies

Typical examples of strategies in established consensus systems are presented here to demonstrate how these primary obstacles are overcome.

5.3.1 Strategy Examples of Overcoming Cognitive Differences

Identifying cognitive differences among participants in a consensus system can help in selecting more suitable consensus strategies. For example, in systems where participants share a clear, unified task orientation and operate within a limited set of actions, consensus can typically be achieved by synchronizing trusted information. For instance, in DFT and blockchain consensus mechanisms to ensure that a consensus value, based on single input consensus, is either accepted or rejected by a sufficient number of participants. For example, in CFT, a request is processed depending on whether it receives adequate support from nodes. In blockchain, the longest chain rule decides which chain is confirmed. However, in scenarios involving an open consensus object that participants might hold significantly different values, participants often need to change information iteratively to converge either to a single value or to values within an acceptable margin. In systems with greater cognitive difference, a more open and multidimensional decision space may require deliberate convergence mechanisms over multiple rounds. For instance, an consensus approach in [157] uses AI mediation to iteratively generate "group statements," helping participants converge on shared perspectives while incorporating both majority and minority views.

5.3.2 Strategy Examples of Overcoming Dishonesty

Honesty assessment of nodes in a consensus system is a crucial factor in selecting and establishing an appropriate consensus method. For example, in a DFT consensus system, if all nodes are assumed to be non-malicious (no dishonest nodes), a CFT algorithm can be used, which each quorum has $f+1$ ($f=\frac{n}{2}$) such that at least one node is correct and it is responsible for reducing the cognitive inconsistency. However, if the system is expected to include malicious (Byzantine) nodes, a BFT consensus is necessary, requiring $2f+1$ nodes ($f=\frac{n}{3}$) so that the number of honest nodes ($f+1$) is always at least one more than the potential Byzantine nodes (up to f). The quorum threshold design in protocols ensures that there are enough honest nodes to outweigh any dishonest ones (Byzantine nodes), allowing participants to identify and rely on information from honest nodes to resolve cognitive differences. This example illustrates how the honesty conditions of the system can increase the complexity of reaching a consensus, requiring different strategies.

Some other examples of blockchain consensus are provided to display their strategies to overcome dishonesty. In terms of mitigating the dishonesty of participants, public key cryptography could be used to authenticate the identity of participants and constrain the ability to act as someone else. Reliable broadcast [32] can force a sender to tell the same story to everyone. Typically, blockchain consensus generally uses digital signature (public key

cryptography) to guarantee that every participant cannot pretend to be another node.

In token-integrated blockchain, token-based consensus might be used, including PoS and PoW. In these consensus, token are used as the incentives for reaching consensus. In PoW, miners compete to solve a complex math problem for the reward token, the winner's block will be validated and accepted. In PoS, the more token a participant holds, the more likely it could be selected to validate a block. Because a stake holder tend to increase its money rather than losing value, including its stake and the price of its stake in the whole system, stake-holders are more likely to maintain the system honestly.

5.4 Conclusion

This chapter compares and analyses consensus mechanisms across technological and social science domains, establishing a unified consensus framework. This framework, derived from a broad perspective on consensus, incorporates three core elements: the participants as carriers, the state transitions in the consensus process, and communication serving as the connecting bridge. This consensus framework contributes to a more comprehensive understanding of consensus and proposes a systematic approach to evaluate and integrate consensus mechanisms in different contexts. Based on this framework, the nature of the consensus process is concluded as eliminating cognitive differences among participants. The consensus framework and definition proposed in this paper aim to promote consensus mechanisms to adapt to complex and dynamic real-world scenarios and transcend traditional limitations.

The consensus framework introduced in this chapter provides a structured basis for cross-domain analysis, but further research is needed to operationalise and evaluate it. Promising directions include the formal modelling of state transitions and these dynamics under varying participant behaviours and communication conditions. In addition, classical consensus protocols can be embedded into the framework as case studies, and new protocols may be proposed that are specifically tailored to its structure. Statistical analysis of synthetic or real-world data can further support the calibration of model parameters and the validation of the framework's generality. These extensions would move the framework from a purely conceptual construct toward a practical foundation for systematic modelling and simulation of consensus phenomena.

Chapter 6

Conclusion and Future Trend

6.1 Conclusion

This thesis presented a comprehensive and multi-dimensional study of consensus, motivated by the increasing convergence of human and machine decision-making in decentralized and intelligent systems. Addressing the limitations of existing models that typically focus either on fault-tolerant machine coordination or human-centric governance, this work bridges the gap through a unified investigation of three interconnected aspects: probabilistic consensus in distributed systems, DAO voting as a decentralized governance mechanism, and a conceptual framework unifying human and machine consensus.

The first part of the thesis developed a probabilistic model for distributed fault-tolerant consensus under uncertainty. Unlike conventional deterministic models, this approach treats node reliability as a stochastic variable, allowing consensus outcomes to be classified into safe, risky, and compromised states. A new construct, the reliability quorum, was introduced to support adaptive consensus design based on desired reliability thresholds. This framework enables fine-grained control over fault tolerance and provides analytical tools for system-level reliability estimation and optimization.

The second part examined decentralized consensus in human-driven systems, using DAO voting as a representative example. In contrast to deterministic coordination among machines, DAO voting involves voluntary participation, varied voting power, and flexible approval conditions. To evaluate this, the thesis introduced the DAO governance triangle and the SEED framework, capturing four critical performance dimensions: Security, Efficiency, Effectiveness, and Decentralization. Building upon this, a quantitative stochastic model was proposed, leading to the formulation of metrics such as the Consistency Rate and Decentralization Coefficient. Simulation-based evaluations further revealed how voting rules, power concentration, and participation behaviour affect overall governance outcomes.

The final part of the thesis proposed a unifying abstraction of the consensus process, applicable across human, machine, and hybrid systems. This framework identifies three core components of any consensus scenario: participants, communication, and state. By framing consensus as a process of entropy reduction, where system-wide divergence is gradually resolved into coherent agreement. It offers a cross-domain analytical lens to compare, interpret, and design consensus mechanisms. The framework also distinguishes among three forms of consensus, including human, machine, and human-machine hybrid, and offers guiding principles suited to each context.

Together, these contributions provide a theoretical and practical foundation for understanding and designing consensus mechanisms that are resilient, scalable, and trustworthy across diverse application domains. The models and frameworks developed in this thesis are especially relevant in the context of Web3 ecosystems, decentralized governance, and emerging intelligent systems, offering guidance for future research and system implementation in increasingly heterogeneous and decentralized environments.

6.2 Future Trends

6.2.1 Extension of Current Researches

This thesis lays the groundwork for understanding consensus across various aspects. Building on this foundation, several directions emerge for future exploration.

- **How does network topology impact the performance of distributed consensus? How should consensus mechanisms be adapted to match different topological structures?** Most consensus models assume uniform communication or global reliability knowledge. In contrast, real-world systems often operate over heterogeneous or dynamically changing networks. Mobile agents, unreliable links, and partial visibility all affect consensus performance. A promising extension involves studying how topological properties, such as graph connectivity, clustering, and trust propagation, affect convergence, delay, and fault resilience.

- **Can consensus mechanisms support open-ended and multi-dimensional decision processes rather than fixed-value agreement?** Most existing consensus protocols are designed to reach agreement on a single, predefined value or system state, such as a transaction log or a binary choice. However, many real-world settings, especially in decentralized governance, collaborative planning, or policy negotiation, involve complex issues that evolve over time and cannot be reduced to a single decision point. In

these cases, the goal of consensus shifts from final agreement to ongoing alignment of diverse preferences and iterative coordination. Future research could explore models that allow partial agreement, adaptive quorum rules, or continuous deliberation, while still preserving coherence and stability across the system.

- **Can the Decentralization Coefficient be generalized to complex governance designs?** The Decentralization Coefficient introduced in this thesis captures control concentration in token-based DAO voting systems. However, DAO governance increasingly involves mechanisms such as vote delegation [156], time-based token vesting, and reputation scoring [158]. Extending the coefficient to reflect multi-layered influence or dynamic control may require graph-based or agent-based modelling. Future studies could also evaluate how proposed anti-centralization mechanisms affect decentralization over time.

- **How should consensus mechanisms evolve in the presence of intelligent autonomy?** The probabilistic model of distributed fault-tolerance consensus developed in this thesis effectively addresses uncertainty in static, homogeneous machine networks. However, as nodes acquire autonomous capabilities and adaptive learning behaviours, the assumptions underlying traditional consensus mechanisms begin to break down. Intelligent agents may exhibit evolving strategies, heterogeneous utility functions, or non-stationary risk profiles. These characteristics challenge existing convergence guarantees and suggest the need for consensus protocols that accommodate behavioural dynamics. Future research could integrate stochastic game theory, adaptive learning, and belief update mechanisms to support strategic negotiation and stability under uncertainty.

- **What consensus paradigms are suitable for supporting human-machine collectives in the era of AI?** The integration of intelligent machines into collective decision-making introduces new requirements for consensus design. In human-machine collectives, the distribution of authority, interpretation of social norms, and resolution of role ambiguity become critical. Unlike conventional distributed systems, these settings involve cognitive, ethical, and temporal asymmetries between agents. Future systems may require hybrid architectures incorporating hierarchical delegation, participatory control, or semantic alignment layers that mediate between human intuition and machine optimization. Research into explainable AI and social computing may offer essential tools for bridging this gap.

- **How can distributed consensus contribute to building trust in distributed systems?**

A core motivation behind distributed consensus is to enable coordination without centralized control. Beyond achieving agreement, consensus mechanisms can also serve as a structural foundation for establishing trust among participants. By providing verifiable, tamper-resistant, and collectively validated decision outcomes, decentralized consensus can reduce reliance on individual entities and mitigate risks stemming from misbehaviour or failure. Future research should explore how consensus protocols can be designed not only for correctness and fault tolerance, but also as trust-generating infrastructures, supporting transparency, accountability, and long-term cooperation in decentralized environments.

6.2.2 Promising Future Direction

How AI Will Affect Consensus

AI represents a transformative force that extends advanced intelligence to machine groups, fundamentally redefining the nature of machine consensus. While traditional machine consensus primarily focuses on fundamental tasks like information synchronization, state consistency, and fault-tolerant data verification, the advent of high-intelligence agents introduces a new paradigm. Consensus among these intelligent agents significantly elevates the demands on decision-making, moving beyond simple agreement to complex collective intelligence. Each intelligent agent may exhibit distinct learning strategies, diverse risk preferences, varied objective functions, and heterogeneous computational capabilities [159]. These inherent differences can amplify divergences within the machine group, complicating coordination, potentially leading to local optima, slower convergence, or even deadlocks [160]. Consequently, future consensus mechanisms must transcend basic state replication and data synchronization, evolving to incorporate sophisticated negotiation, bargaining, and belief fusion processes. This will likely necessitate the adoption of complex game-theoretic approaches, as well as principles from multi-agent reinforcement learning (MARL) [161] and distributed optimization, to effectively balance diverse, potentially conflicting interests and optimize resource allocation within a dynamic, multi-agent system.

Simultaneously, the profound advancement of AI technology has intensified the critical need for human-machine consensus. As AI systems gain sufficient intelligence and autonomy to collaborate intimately with human operators in critical domains, such as autonomous driving with human oversight [162], or AI-assisted medical diagnosis [163] [164], new layers of considerations emerge far beyond fundamental technical aspects. These include designing intuitive and effective human-AI interaction mechanisms, aligning potentially divergent decision-making priorities between humans and machines, and establishing clear, dynamic

role and authority allocations. Key challenges here encompass designing transparent and efficient human-machine collaboration frameworks, which involves integrating insights from cognitive science and human factors engineering to create interfaces that foster trust and understanding, enabling humans to comprehend AI's reasoning and vice versa. Furthermore, ensuring a balanced and adaptive distribution of authority in critical scenarios requires developing mechanisms for dynamic delegation of control, where human oversight can be effectively maintained, and authority can seamlessly shift based on situational awareness, trust calibration [165], and pre-defined ethical AI principles. Finally, defining responsibilities and clear boundaries among multiple human and AI decision-making entities necessitates addressing complex ethical, legal, and accountability questions, ensuring clear lines of responsibility for collective actions and establishing robust fallback mechanisms in cases of disagreement or failure [166].

To facilitate such sophisticated cooperative consensus, whether purely among machines or in human-machine collectives, the establishment of robust and dynamic trust mechanisms is paramount. A fundamental challenge is ensuring that AI's decisions are not only reliable and robust but also transparent, explainable, and comprehensible to all participants [167], particularly as AI agents assume increasingly central roles in distributed decision-making. Enhancing explainability in AI decision-making, including methods like post-hoc explanations [168], and interpretable models [169], is therefore a critical research direction. For consensus specifically, the challenge extends to explaining collective decisions and ensuring consistency in explanations across heterogeneous agents. Additionally, trust fundamentally depends on the predictability and consistency of both human and AI participants in adhering to established agreements and exhibiting expected behaviours. This necessitates developing robust mechanisms for behavioural modelling and ensuring adherence to agreed-upon protocols. To mitigate risks associated with malicious, unreliable, or uncooperative actions (whether intentional or emergent due to AI's unpredictable nature), various regulatory and incentive mechanisms can be explored. These include designing sophisticated game-theoretic models for incentive and punishment systems to encourage cooperative behaviour and penalize deviation, tailored to the unique learning and objective functions of AI agents [170]. Furthermore, developing dynamic, verifiable reputation and credit scoring models that track the performance and trustworthiness of individual AI agents and human participants over time can influence their participation and authority in future consensus processes [171]. Finally, leveraging advancements in distributed ledger technologies like blockchain can enhance data integrity, ensure immutable transaction traceability, and provide verifiable credentials for AI agents, thereby significantly reducing concerns about data manipulation and information asymmetry and strengthening mutual trust among all stakeholders in the consen-

sus process. Ultimately, the integration of AI into distributed consensus opens up vast new research avenues, demanding interdisciplinary approaches that combine distributed systems theory, AI, game theory, human-computer interaction, and ethics to build the next generation of resilient, intelligent, and trustworthy autonomous collective decision-making systems.

Exploring the Potential Influence of Consensus on Future Financial Systems

The evolution of consensus mechanisms holds the potential to fundamentally reshape future financial systems, transitioning them from centralized, intermediary-dependent structures towards more decentralized, efficient, and resilient paradigms. Traditional finance is characterized by reliance on trusted third parties, slow settlement times, and limited global accessibility, often leading to high transaction costs and systemic vulnerabilities. Advanced consensus protocols, particularly those underpinning DLTs, could enable direct, peer-to-peer transactions and atomic settlement, paving the way for instantaneous cross-border payments, fractional ownership of illiquid assets, and the seamless creation of tokenized securities [53]. This transformation may necessitate the development of highly efficient and secure consensus algorithms capable of handling massive transaction volumes and ensuring rapid finality, driving innovation in areas like DLT-based payment rails, decentralized exchanges (DEXs) [172], and automated market makers (AMMs) [173], which rely on consensus for price discovery and liquidity provision.

Furthermore, consensus mechanisms are poised to play a pivotal role in potentially enhancing the overall resilience and stability of financial systems by mitigating centralized points of failure and increasing transparency. By providing a shared, immutable, and auditable record of transactions, distributed consensus could significantly reduce fraud, improve regulatory oversight, and streamline reconciliation processes, thereby bolstering trust and potentially reducing operational risks across the financial ecosystem [174]. In decentralized finance (DeFi), robust consensus is critical for the functioning of decentralized lending and borrowing platforms, ensuring the integrity of collateralization mechanisms and the execution of complex financial primitives like flash loans [175]. During periods of financial stress or crisis, distributed ledgers, secured by advanced consensus, might offer a real-time, shared source of truth for clearing and settlement, potentially enabling more effective crisis management and faster recovery compared to fragmented, opaque legacy systems [176].

The profound potential influence of consensus on future financial systems may also necessitate a re-evaluation of existing regulatory and institutional frameworks. Regulators could face the challenge of adapting to decentralized, borderless financial activities, requiring the development of new approaches to digital asset regulation, consumer protection, and anti-money laundering compliance within a distributed context [177]. Concurrently, the inte-

gration of DLT-based financial infrastructure with traditional finance may require the establishment of interoperability standards, the exploration of Central Bank Digital Currencies (CBDCs) based on consensus protocols [178], and the development of institutional-grade DeFi solutions. Addressing new forms of financial crime and market manipulation, such as front-running in DEXs or oracle manipulation, will also become critical research areas, demanding novel detection and mitigation strategies rooted in the principles of decentralized consensus and cryptoeconomics [179]. Ultimately, the future of finance is likely to be deeply intertwined with the advancements in consensus, requiring unprecedented collaboration among technologists, economists, legal experts, and regulators to construct a potentially more efficient, inclusive, and resilient global financial architecture.

Appendix A

Derivation of Theorems

A.1 Proof of function G_{count}

We express the probability that M_p reliable nodes are alive in the prepare phase and M_c in the commit phase as $G_{\text{count}}(M_p = m_p, M_c = m_c, F = f, Q_r = q)$ under $\mathcal{E}_{R2}$. Using PBFT as an example, we show the calculation of G_{count}.

With a reliable primary node, consensus passes the pre-prepare phase. For the prepare and commit phases, nodes exchange messages to verify consensus validity. Each reliable node must receive at least $Q_r = q$ identical valid messages. The next received message is denoted as X, where receiving a valid or invalid message is represented by $X = \{1, 0\}$.

The probability of a reliable node receiving a valid or invalid message after receiving u valid and v invalid messages is

$$\begin{aligned} P(X=1) &= \frac{n-f-u-1}{(n-f-u-1)+(f-v)p_s}, \\ P(X=0) &= \frac{(f-v)p_s}{(n-f-u-1)+(f-v)p_s}, \end{aligned} \tag{A.1}$$

where p_s is the fixed probability that a Byzantine node sends an invalid message. The probability of a reliable node receiving u valid and v invalid messages is given by

$$P_{rec}(u,v) = P(X=1)P_{rec}(u-1,v) + P(X=0)P_{rec}(u,v-1), \tag{A.2}$$

where $u \in \{1, ..., n-f-1\}$, $v \in \{0, ..., f\}$, $P_{rec}(1,0) = 1$.

A reliable replica node must receive at least $q-2$ valid messages (including its own and the primary node's), while a reliable primary node requires at least $q-1$ valid messages

before accumulating q invalid messages, leading to

$$P_o = \frac{\sum_{v=0}^{q-1} P_{rec}(q-1,v)}{\sum_{v=0}^{q-1} P_{rec}(q-1,v) + \sum_{u=0}^{q-2} P_{rec}(u,q)}, \tag{A.3}$$

$$P_r = \frac{\sum_{v=0}^{q-1} P_{rec}(q-2,v)}{\sum_{v=0}^{q-1} P_{rec}(q-2,v) + \sum_{u=0}^{q-3} P_{rec}(u,q)}. \tag{A.4}$$

In the commit phase, all nodes require at least $q-1$ messages (excluding their own). The probability of passing the commit phase follows the same expression as (A.3). The final expression for G_{count} is

$$G_{\text{count}}(M_p = m_p, M_c = m_c, F = f, Q_r = q) = \sum_{m_p=\lfloor \frac{n}{2} \rfloor + 1}^{n-f} \sum_{m_c=\lfloor \frac{n}{2} \rfloor + 1}^{m_p} P_{pre}(m_p) P_{com}(m_p, m_c), \tag{A.5}$$

where

$$P_{pre}(m_p) = P_o P_r^{m_p - 1} (1-P_r)^{n-f-m_p} + (1-P_o) P_r^{m_p} (1-P_r)^{n-f-m_p-1}, \tag{A.6}$$

and

$$P_{com}(m_p, m_c) = P_o^{m_c} (1-P_o)^{m_p - m_c}. \tag{A.7}$$

A.2 Proof of Theorem 1

To prove Theorem 1, we use equations denoted as $P_{pp}(a,b)$, $P_p(a,b)$, $P_c(a,b)$ to calculate the probability of success of $pre-prepare$ phase, $prepare$ phase and $commit$ phase separately by giving the number of successful nodes a before entering the phase and b after completing the phase. Similarly, the rate of failed nodes in a certain number is calculated in $P_{node}(a,b)$ where a denotes the total number of nodes while b is the number of non-faulty nodes. According to the communication principle shown in Fig. 3.3 and the assumption of node failure, it is easy to apply binomial distribution method to calculate $P_{pp}(a,b)$ and $P_{node}(a,b)$ by equation (3.26) and equation (3.25). As for $P_p(a,b)$ and $P_c(a,b)$, the probability of success of each node should be calculated according to P_l first since the number of broadcast messages each node receives determines whether it can proceed to the next phase.

Therefore, by applying binomial distribution with minimum valid messages required, i.e., $2f$ messages (without itself) from different nodes, we have equation (3.29) to calculate the probability of success of each node in $prepare$ and $commit$ phases. With the probability of success for each node calculated in $prepare$ and $commit$ phase, equation (3.28) and equation (3.27) for $P_p(a,b)$ and $P_c(a,b)$ can be regarded as node failure calculation by using binomial distribution as in $P_{node}(a,b)$.

To calculate the probability of successful consensus of a complete $P-N-L$ model with unknown number of success nodes in each phase, we use intermediate notations to replace index a and b in $P_{pp}(a,b)$, $P_p(a,b)$, $P_c(a,b)$ and $P_{node}(a,b)$. m is the number of non-faulty nodes and m_{pp}, m_p, m_c are used to represent the number of success nodes after $pre-prepare$ phase, $prepare$ phase and $commit$ phase. Therefore, we obtain the rate to complete an entire consensus process with a known number of success nodes in each phase as

$$\begin{aligned} P_{P-N-L_{sub}} =& P_{node}(n,m) \cdot P_{pp}(m,m_{pp}) \cdot \\ & P_p(m_{pp},m_p) \cdot P_c(m_p,m_c). \end{aligned} \tag{A.8}$$

The probability of successful consensus of $P-N-L$ is actually adding all the possible cases of equation (A.8). To sum up, all the cases that are able to achieve consensus successfully, we have

$$P_{P-N-L} = \sum_{m=m_{pp}}^{n} \sum_{m_{pp}=m_p}^{n} \sum_{m_p=m_c}^{n} \sum_{m_c=n-f}^{n} P_{P-N-L_{sub}}, \tag{A.9}$$

and the final expansion equation is equation (3.24).

The core idea of the equation (3.24) is that if a node fails, it does not participate in the rest of the consensus phase. As we know, failed nodes can be divided into two types. One is caused by node failure, which means the node is unavailable in the entire consensus process, and the other is caused by link failure since the number of messages a node collects in one phase cannot support it entering the next phase. Based on this analysis, we can conclude that, for a successful consensus process, $n \geq m \geq m_{pp} \geq m_p \geq m_c \geq n-f$.

Bibliography

[1] X. Yan, An introduction to the system architecture abstraction of the replicated state machine, accessed: July 2, 2025 (August 2022).
URL https://www.alibabacloud.com/blog/an-introduction-to-the-system-architecture-abstraction-of-the-replicated-state-machine_599279

[2] V. C. Kalempa, L. Piardi, M. Limeira, A. S. de Oliveira, Multi-robot task scheduling for consensus-based fault-resilient intelligent behavior in smart factories, Machines 11 (4) (2023) 431.

[3] Oxford English Dictionary, s.v. 'consensus (n.)', https://doi.org/10.1093/OED/7567688553, accessed: April 30, 2024 (Sep. 2023).

[4] J. H. Turner, A. Maryanski, Functionalism, Benjamin/Cummings Publishing Company Menlo Park, CA, 1979.

[5] C. Agius, et al., Social constructivism, Contemporary security studies 3 (2013) 87–103.

[6] R. Balasubramanian, D. Agarwal, Delphi technique–a review, International Journal of Public Health Dentistry 3 (2) (2012) 16–26.

[7] G. Zhang, F. Pan, Y. Mao, S. Tijanic, M. Dang'Ana, S. Motepalli, S. Zhang, H.-A. Jacobsen, Reaching consensus in the byzantine empire: A comprehensive review of bft consensus algorithms, ACM Computing Surveys 56 (5) (2024) 1–41.

[8] A. Hopkins, A fault-tolerant information processing concept for space vehicles, IEEE Transactions on Computers C-20 (11) (1971) 1394–1403. doi:10.1109/T-C.1971.223145.

[9] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, in: Concurrency: the works of leslie lamport, 2019, pp. 203–226.

[10] Y. Li, C. Tan, A survey of the consensus for multi-agent systems, Systems Science & Control Engineering 7 (1) (2019) 468–482.

[11] I. L. Horowitz, Consensus, conflict and cooperation: A sociological inventory, Social Forces 41 (2) (1962) 177–188.

[12] Y. Xiao, N. Zhang, W. Lou, Y. T. Hou, A survey of distributed consensus protocols for blockchain networks, IEEE Communications Surveys & Tutorials 22 (2) (2020) 1432–1465.

[13] H. Xu, Y. Fan, W. Li, L. Zhang, Wireless distributed consensus for connected autonomous systems, IEEE Internet of Things Journal 10 (9) (2023) 7786–7799. doi:10.1109/JIOT.2022.3229746.

[14] C. Feng, Z. Xu, X. Zhu, P. V. Klaine, L. Zhang, Wireless distributed consensus in vehicle to vehicle networks for autonomous driving, IEEE Transactions on Vehicular Technology (2023).

[15] H. Wu, C. Yue, L. Zhang, Y. Li, M. A. Imran, When distributed consensus meets wireless connected autonomous systems: A review and a dag-based approach, IEEE Network (2024).

[16] S. Nahavandi, Industry 5.0—a human-centric solution, Sustainability 11 (16) (2019) 4371.

[17] C. Zhang, Z. Wang, G. Zhou, F. Chang, D. Ma, Y. Jing, W. Cheng, K. Ding, D. Zhao, Towards new-generation human-centric smart manufacturing in industry 5.0: A systematic review, Advanced Engineering Informatics 57 (2023) 102121.

[18] M. Fukuyama, et al., Society 5.0: Aiming for a new human-centered society, Japan Spotlight 27 (5) (2018) 47–50.

[19] X. Xu, Y. Lu, B. Vogel-Heuser, L. Wang, Industry 4.0 and industry 5.0—inception, conception and perception, Journal of manufacturing systems 61 (2021) 530–535.

[20] K. Makovi, A. Sargsyan, W. Li, J.-F. Bonnefon, T. Rahwan, Trust within human-machine collectives depends on the perceived consensus about cooperative norms, Nature communications 14 (1) (2023) 3108.

[21] R. Guerraoui, A. Schiper, Consensus service: a modular approach for building agreement protocols in distributed systems, in: Proceedings of Annual Symposium on Fault Tolerant Computing, IEEE, 1996, pp. 168–177.

[22] C. Cachin, M. Vukolić, Blockchain consensus protocols in the wild, arXiv preprint arXiv:1707.01873 (2017).

[23] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, G. Danezis, Sok: Consensus in the age of blockchains, in: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, 2019, pp. 183–198.

[24] N. A. Lynch, Distributed algorithms, Elsevier, 1996.

[25] M. Pease, R. Shostak, L. Lamport, Reaching agreement in the presence of faults, Journal of the ACM (JACM) 27 (2) (1980) 228–234.

[26] M. J. Fischer, N. A. Lynch, M. S. Paterson, Impossibility of distributed consensus with one faulty process, Journal of the ACM (JACM) 32 (2) (1985) 374–382.

[27] C. Dwork, N. Lynch, L. Stockmeyer, Consensus in the presence of partial synchrony, Journal of the ACM (JACM) 35 (2) (1988) 288–323.

[28] F. B. Schneider, Implementing fault-tolerant services using the state machine approach: A tutorial, ACM Computing Surveys (CSUR) 22 (4) (1990) 299–319.

[29] M. Castro, B. Liskov, et al., Practical byzantine fault tolerance, in: OsDI, Vol. 99, 1999, pp. 173–186.

[30] L. Lamport, The part-time parliament, in: Concurrency: the works of Leslie Lamport, 2019, pp. 277–317.

[31] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 USENIX annual technical conference (USENIX ATC 14), 2014, pp. 305–319.

[32] G. Bracha, Asynchronous byzantine agreement protocols, Information and Computation 75 (2) (1987) 130–143.

[33] W. Vogels, Eventually consistent, Communications of the ACM 52 (1) (2009) 40–44.

[34] D. B. Terry, A. J. Demers, K. Petersen, M. J. Spreitzer, M. M. Theimer, B. B. Welch, Session guarantees for weakly consistent replicated data, in: Proceedings of 3rd International Conference on Parallel and Distributed Information Systems, IEEE, 1994, pp. 140–149.

[35] E. A. Brewer, Towards robust distributed systems, in: PODC, Vol. 7, Portland, OR, 2000, pp. 343–477.

[36] X. Wang, S. Duan, J. Clavin, H. Zhang, Bft in blockchains: From protocols to use cases, ACM Computing Surveys (CSUR) 54 (10s) (2022) 1–37.

[37] R. Baldoni, J.-M. Helary, M. Raynal, From crash fault-tolerance to arbitrary-fault tolerance: Towards a modular approach, in: Proceeding International Conference on Dependable Systems and Networks. DSN 2000, IEEE, 2000, pp. 273–282.

[38] C. Cachin, et al., Architecture of the hyperledger blockchain fabric, in: Workshop on distributed cryptocurrencies and consensus ledgers, Vol. 310, Chicago, IL, 2016, pp. 1–4.

[39] E. Buchman, Tendermint: Byzantine fault tolerance in the age of blockchains, Ph.D. thesis, University of Guelph (2016).

[40] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, M. A. Imran, A scalable multi-layer pbft consensus for blockchain, IEEE Transactions on Parallel and Distributed Systems 32 (5) (2021) 1146–1160. doi:10.1109/TPDS.2020.3042392.

[41] S. Liu, R. Zhang, C. Liu, D. Shi, P-pbft: An improved blockchain algorithm to support large-scale pharmaceutical traceability, Computers in biology and medicine 154 (2023) 106590.

[42] Y. Meshcheryakov, A. Melman, O. Evsutin, V. Morozov, Y. Koucheryavy, On performance of pbft blockchain consensus algorithm for iot-applications with constrained devices, IEEE Access 9 (2021) 80559–80570.

[43] S. Tang, Z. Wang, J. Jiang, S. Ge, G. Tan, Improved pbft algorithm for high-frequency trading scenarios of alliance blockchain, Scientific Reports 12 (1) (2022) 4426.

[44] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, P. Verissimo, Efficient byzantine fault-tolerance, IEEE Transactions on Computers 62 (1) (2011) 16–30.

[45] J. Zhang, J. Gao, K. Wang, Z. Wu, Y. Li, Z. Guan, Z. Chen, Tbft: efficient byzantine fault tolerance using trusted execution environment, in: ICC 2022-IEEE International Conference on Communications, IEEE, 2022, pp. 1004–1009.

[46] S. R. Tsaliki, Accelerating etcd insertion operations through advanced complexity reduction (2024).

[47] K. Xiong, S. Moon, J. Kang, B. Curto, J. Kim, J.-Y. Shin, Recraft: Self-contained split, merge, and membership change of raft protocol, arXiv preprint arXiv:2504.14802 (2025).

[48] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).

[49] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) (2014) 1–32.

[50] A. M. Antonopoulos, D. A. Harding, Mastering Bitcoin: Programming the open blockchain, " O'Reilly Media, Inc.", 2023.

[51] J. R. Douceur, The sybil attack, in: International workshop on peer-to-peer systems, Springer, 2002, pp. 251–260.

[52] P. Swathi, C. Modi, D. Patel, Preventing sybil attack in blockchain using distributed behavior monitoring of miners, in: 2019 10th international conference on computing, communication and networking technologies (ICCCNT), IEEE, 2019, pp. 1–6.

[53] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, Bitcoin and cryptocurrency technologies: a comprehensive introduction, Princeton University Press, 2016.

[54] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, C. Qijun, A review on consensus algorithm of blockchain, in: 2017 IEEE international conference on systems, man, and cybernetics (SMC), IEEE, 2017, pp. 2567–2572.

[55] J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol: Analysis and applications, Journal of the ACM 71 (4) (2024) 1–49.

[56] Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in: Financial Cryptography and Data Security: 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers 19, Springer, 2015, pp. 507–527.

[57] A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in: Annual international cryptology conference, Springer, 2017, pp. 357–388.

[58] K. J. O'Dwyer, D. Malone, Bitcoin mining and its energy footprint, in: 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), IET, 2014, pp. 280–285.

[59] J. Li, N. Li, J. Peng, H. Cui, Z. Wu, Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies, Energy 168 (2019) 160–168.

[60] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, et al., On scaling decentralized blockchains: (a position paper), in: International conference on financial cryptography and data security, Springer, 2016, pp. 106–125.

[61] S. King, S. Nadal, Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, self-published paper, August 19 (1) (2012).

[62] V. Buterin, Ethereum: platform review, Opportunities and Challenges for Private and Consortium Blockchains 45 (2016).

[63] M. A. Manolache, S. Manolache, N. Tapus, Decision making using the blockchain proof of authority consensus, Procedia Computer Science 199 (2022) 580–588.

[64] W. Li, S. Andreina, J.-M. Bohli, G. Karame, Securing proof-of-stake blockchain protocols, in: Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2017 International Workshops, DPM 2017 and CBT 2017, Oslo, Norway, September 14-15, 2017, Proceedings, Springer, 2017, pp. 297–315.

[65] P. Vasin, et al., Blackcoin's proof-of-stake protocol v2, URL: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper. pdf 71 (2014) 25.

[66] M. Vukolić, The quest for scalable blockchain fabric: Proof-of-work vs. bft replication, in: International workshop on open problems in network security, Springer, 2015, pp. 112–125.

[67] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, I. Abraham, Hotstuff: Bft consensus in the lens of blockchain, arXiv preprint arXiv:1803.05069 (2018).

[68] S. Gupta, J. Hellings, S. Rahnama, M. Sadoghi, An in-depth look of bft consensus in blockchain: Challenges and opportunities, in: Proceedings of the 20th international middleware conference tutorials, 2019, pp. 6–10.

[69] L. S. Sankar, M. Sindhu, M. Sethumadhavan, Survey of consensus protocols on blockchain applications, in: 2017 4th international conference on advanced computing and communication systems (ICACCS), IEEE, 2017, pp. 1–5.

[70] R. Pass, E. Shi, Hybrid consensus: Efficient consensus in the permissionless model, Cryptology ePrint Archive (2016).

[71] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, Algorand: Scaling byzantine agreements for cryptocurrencies, in: Proceedings of the 26th symposium on operating systems principles, 2017, pp. 51–68.

[72] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, I. Abraham, Hotstuff: Bft consensus with linearity and responsiveness, in: Proceedings of the 2019 ACM symposium on principles of distributed computing, 2019, pp. 347–356.

[73] M. Xu, X. Cheng, M. X. (Yife), Wireless Consensus, Springer, 2024.

[74] D. Yu, W. Li, H. Xu, L. Zhang, Low reliable and low latency communications for mission critical distributed industrial internet of things, IEEE Communications Letters 25 (1) (2021) 313–317. doi:10.1109/LCOMM.2020.3021367.

[75] M. Nischwitz, M. Esche, F. Tschorsch, Bernoulli meets pbft: Modeling bft protocols in the presence of dynamic failures, in: 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS), 2021, pp. 291–300. doi:10.15439/2021F38.

[76] H. Luo, X. Yang, H. Yu, G. Sun, B. Lei, M. Guizani, Performance analysis and comparison of nonideal wireless pbft and raft consensus networks in 6g communications, IEEE Internet of Things Journal 11 (6) (2024) 9752–9765. doi:10.1109/JIOT.2023.3323492.

[77] Y. Li, Y. Fan, L. Zhang, J. Crowcroft, Raft consensus reliability in wireless networks: Probabilistic analysis, IEEE Internet of Things Journal 10 (14) (2023) 12839–12853. doi:10.1109/JIOT.2023.3257402.

[78] L. Lundberg, M. Petrén, DApp Revolution: An Investigation into the Nature and Business Models of Web 3.0 Decentralized Applications (2022).

[79] A. K. Goel, R. Bakshi, K. K. Agrawal, Web 3.0 and Decentralized Applications, Materials Proceedings 10 (1) (2022) 8.

[80] L. Liu, S. Zhou, H. Huang, Z. Zheng, From technology to society: An overview of blockchain-based dao, IEEE Open Journal of the Computer Society 2 (2021) 204–215.

[81] Q. DuPont, Experiments in algorithmic governance: A history and ethnography of "the dao," a failed decentralized autonomous organization, in: Bitcoin and beyond, Routledge, 2017, pp. 157–177.

[82] DeepDAO, DeepDAO, accessed May. 19, 2024 (2022).
URL `https://deepdao.io/organizations`

[83] The LAO, The LAO: A For-Profit, Limited Liability Autonomous Organization, accessed May. 19, 2024 (2019).
URL https://medium.com/openlawofficial/the-lao-a-for-profit-limited-liability-autonomous-organization-9eae89c9669c

[84] S. Gabriel, P. Peter, S. Ameen, Metacartel Ventures, accessed May. 19, 2024 (2019).
URL https://github.com/metacartel/MCV/blob/master/Whitepaper.pdf

[85] V. Chávez, A. Luján, V. S. Dainton, A. Villa, M. Zamora, 2018-2019 environmental sustainability team.

[86] E. Ordano, A. Meilich, Y. Jardi, M. Araoz, Decentraland White paper, accessed May. 19, 2024.
URL https://decentraland.org/whitepaper.pdf

[87] S. Hassan, P. De Filippi, Decentralized autonomous organization, Internet Policy Review 10 (2) (2021) 1–10.

[88] C. Jentzsch, Decentralized autonomous organization to automate governance, White paper, November (2016).

[89] S. Wang, W. Ding, J. Li, Y. Yuan, L. Ouyang, F.-Y. Wang, Decentralized autonomous organizations: Concept, model, and applications, IEEE Transactions on Computational Social Systems 6 (5) (2019) 870–878.

[90] Y. Faqir-Rhazoui, J. Arroyo, S. Hassan, A comparative analysis of the platforms for decentralized autonomous organizations in the Ethereum blockchain, Journal of Internet Services and Applications 12 (1) (2021) 1–20.

[91] C. Sguanci, R. Spatafora, A. M. Vergani, Layer 2 blockchain scaling: A survey, arXiv preprint arXiv:2107.10881 (2021).

[92] C. Decker, R. Wattenhofer, A fast and scalable payment network with bitcoin duplex micropayment channels, in: Stabilization, Safety, and Security of Distributed Systems: 17th International Symposium, SSS 2015, Edmonton, AB, Canada, August 18-21, 2015, Proceedings 17, Springer, 2015, pp. 3–18.

[93] J. Poon, V. Buterin, Plasma: Scalable autonomous smart contracts, White paper (2017) 1–47.

[94] L. T. Thibault, T. Sarry, A. S. Hafid, Blockchain scaling using rollups: A comprehensive survey, IEEE Access 10 (2022) 93039–93054.

[95] O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, K. H. Duffy, E. Maler, D. Reed, M. Sporny, Decentralized identity: Where did it come from and where is it going?, IEEE Communications Standards Magazine 3 (4) (2019) 10–13.

[96] Y. Chen, H. Li, K. Li, J. Zhang, An improved p2p file system scheme based on ipfs and blockchain, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 2652–2657.

[97] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, A. Granzotto, Bigchaindb: a scalable blockchain database, white paper, BigChainDB (2016) 53–72.

[98] S. Williams, V. Diordiiev, L. Berman, I. Uemlianin, Arweave: A protocol for economically sustainable information permanence, Arweave Yellow Paper (2019).

[99] J. Han, M. Song, H. Eom, Y. Son, An efficient multi-signature wallet in blockchain using bloom filter, in: Proceedings of the 36th annual ACM symposium on applied computing, 2021, pp. 273–281.

[100] D. A. Zetzsche, D. W. Arner, R. P. Buckley, Decentralized finance, Journal of Financial Regulation 6 (2) (2020) 172–203.

[101] S. A. Amri, L. Aniello, V. Sassone, A review of upgradeable smart contract patterns based on openzeppelin technique, The Journal of The British Blockchain Association (2023).

[102] DAOstack, DAOstack Documentation, accessed May. 19, 2024 (2021).
URL `https://daostack-1.gitbook.io/v1/`

[103] Aragon, What is Aragon.
URL `https://aragon.org/about-aragon`

[104] L. Breidenbach, C. Cachin, B. Chan, A. Coventry, S. Ellis, A. Juels, F. Koushanfar, A. Miller, B. Magauran, D. Moroz, et al., Chainlink 2.0: Next steps in the evolution of decentralized oracle networks, Chainlink Labs 1 (2021) 1–136.

[105] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, F.-Y. Wang, An overview of smart contract: architecture, applications, and future trends, in: 2018 IEEE intelligent vehicles symposium (IV), IEEE, 2018, pp. 108–113.

[106] S. Bonnet, F. Teuteberg, Decentralized autonomous organizations: A systematic literature review and research agenda, International Journal of Innovation and Technology Management 21 (04) (2024) 2450026.

[107] A. Wright, The rise of decentralized autonomous organizations: Opportunities and challenges, Stan. J. Blockchain L. & Pol'y 4 (2020) 1.

[108] C. Santana, L. Albareda, Blockchain and the emergence of Decentralized Autonomous Organizations (DAOs): An integrative model and research agenda, Technological Forecasting and Social Change 182 (2022) 121806.

[109] Y. Fan, L. Zhang, R. Wang, M. A. Imran, Insight into voting in daos: Conceptual analysis and a proposal for evaluation framework, IEEE Network 38 (3) (2024) 92–99. doi:10.1109/MNET.137.2200561.

[110] R. Fritsch, M. Müller, R. Wattenhofer, Analyzing Voting Power in Decentralized Governance: Who Controls DAOs?, arXiv preprint arXiv:2204.01176 (2022).

[111] A. Lijphart, Comparative Perspectives on Fair Representation: The Plurality-majority Rule, Geographical Districting, and Alternate Electoral Arrangements, Policy Studies Journal 9 (6) (1981) 899.

[112] A. Vermeule, Absolute Majority Rules, British Journal of Political Science 37 (4) (2007) 643–658.

[113] Y. Fan, Z. Zhou, Z. Qiao, L. Zhang, Efficiency analysis of decentralized autonomous organization (dao) voting mechanisms, in: GLOBECOM 2024 - 2024 IEEE Global Communications Conference (Globecom Workshops), 2024.

[114] S. P. Lalley, E. G. Weyl, Quadratic voting: How mechanism design can radicalize democracy, in: AEA Papers and Proceedings, Vol. 108, 2018, pp. 33–37.

[115] Y. El Faqir, J. Arroyo, S. Hassan, A scalable voting system: Validation of holographic consensus in daostack., in: HICSS, 2021, pp. 1–10.

[116] X. Sun, Decentralization illusion, voting democracy and liquidity risk in decentralized finance, Ph.D. thesis, University of Glasgow (2024).

[117] X. Liu, The illusion of democracy? an empirical study of dao governance and voting behavior, in: The Illusion of Democracy? An Empirical Study of DAO Governance and Voting Behavior: Liu, Xuan, [Sl]: SSRN, 2023.

[118] B. A. Okutan, S. Schmid, Y.-A. Pignolet, Democracy for daos: An empirical study of decentralized governance and dynamics: Case study internet computer sns ecosystem, in: 2025 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, 2025, pp. 1–9.

[119] T. D. Monteiro, O. P. Sanchez, G. H. S. M. d. Moraes, Exploring off-chain voting and blockchain in decentralized autonomous organizations, RAUSP Management Journal 59 (4) (2024) 335–349.

[120] Y. Zheng, L. Jin, L. Gao, K. Li, Y. Wang, F. Wang, Development of a distributed cooperative vehicles control algorithm based on v2v communication, Procedia Engineering 137 (2016) 649–658.

[121] A. Eskandarian, C. Wu, C. Sun, Research advances and challenges of autonomous and connected ground vehicles, IEEE Transactions on Intelligent Transportation Systems 22 (2) (2019) 683–711.

[122] F. A. Alabdulwahhab, Web 3.0: the decentralized web blockchain networks and protocol innovation, in: 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), IEEE, 2018, pp. 1–4.

[123] H. Y. Wu, X. Yang, C. Yue, H.-Y. Paik, S. S. Kanhere, Chain or dag? underlying data structures, architectures, topologies and consensus in distributed ledger technology: A review, taxonomy and research issues, Journal of Systems Architecture 131 (2022) 102720.

[124] S. Munirathinam, Industry 4.0: Industrial internet of things (iiot), in: Advances in computers, Vol. 117, Elsevier, 2020, pp. 129–164.

[125] M. Asante, G. Epiphaniou, C. Maple, H. Al-Khateeb, M. Bottarelli, K. Z. Ghafoor, Distributed ledger technologies in supply chain security management: A comprehensive survey, IEEE Transactions on Engineering Management 70 (2) (2021) 713–739.

[126] S.-M. Cheng, P. Lin, D.-W. Huang, S.-R. Yang, A study on distributed/centralized scheduling for wireless mesh network, in: Proceedings of the 2006 international conference on Wireless communications and mobile computing, 2006, pp. 599–604.

[127] P. Bertin, S. Bonjour, J.-M. Bonnin, Distributed or centralized mobility?, in: GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, IEEE, 2009, pp. 1–6.

[128] Y. Xiao, N. Zhang, J. Li, W. Lou, Y. T. Hou, Distributed consensus protocols and algorithms, Blockchain for Distributed Systems Security 25 (2019) 40.

[129] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, G. Danezis, Consensus in the age of blockchains, arXiv preprint arXiv:1711.03936 (2017).

[130] S. Zhou, K. Li, L. Xiao, J. Cai, W. Liang, A. Castiglione, A systematic review of consensus mechanisms in blockchain, Mathematics 11 (10) (2023) 2248.

[131] D. Malkhi, M. Reiter, Byzantine quorum systems, Distributed computing 11 (4) (1998) 203–213.

[132] S. O'Dea, Failure rate of Apple iPhone worldwide by model 2017-2018 (June 2020). URL `https://www.statista.com/statistics/804359/iphone-failure-rate-by-model-worldwide/`

[133] H. Luo, Q. Zhang, G. Sun, H. Yu, D. Niyato, Symbiotic blockchain consensus: Cognitive backscatter communications-enabled wireless blockchain consensus, IEEE/ACM Transactions on Networking (2024).

[134] Y. Fan, L. Zhang, K. Li, Emi and iemi impacts on the radio communication network of electrified railway systems: A critical review, IEEE Transactions on Vehicular Technology 72 (8) (2023) 10409–10424.

[135] P. S. Narayanan, C. S. Joice, Vehicle-to-vehicle (v2v) communication using routing protocols: a review, in: 2019 International Conference on Smart Structures and Systems (ICSSS), IEEE, 2019, pp. 1–10.

[136] H. Ngo, H. Fang, H. Wang, Cooperative perception with v2v communication for autonomous vehicles, IEEE Transactions on Vehicular Technology 72 (9) (2023) 11122–11131.

[137] P. Kopelias, E. Demiridi, K. Vogiatzis, A. Skabardonis, V. Zafiropoulou, Connected & autonomous vehicles–environmental impacts–a review, Science of the total environment 712 (2020) 135237.

[138] A. A. Abbasi, M. F. Younis, U. A. Baroudi, Recovering from a node failure in wireless sensor-actor networks with minimal topology changes, IEEE Transactions on vehicular technology 62 (1) (2012) 256–271.

[139] M. M. Rana, K. Hossain, Connected and autonomous vehicles and infrastructures: A literature review, International Journal of Pavement Research and Technology 16 (2) (2023) 264–284.

[140] Y. Yanagisawa, Y. Yokote, A new approach to better consensus building and agreement implementation for trustworthy ai systems, in: International Conference on Computer Safety, Reliability, and Security, Springer, 2021, pp. 311–322.

[141] C. Feng, Z. Xu, X. Zhu, P. V. Klaine, L. Zhang, Wireless distributed consensus in vehicle to vehicle networks for autonomous driving, IEEE Transactions on Vehicular Technology 72 (6) (2023) 8061–8073. doi:10.1109/TVT.2023.3243995.

[142] H. Luo, Y. Wu, G. Sun, H. Yu, M. Guizani, Escm: An efficient and secure communication mechanism for uav networks, IEEE Transactions on Network and Service Management 21 (3) (2024) 3124–3139. doi:10.1109/TNSM.2024.3357824.

[143] J. Chen, J. Sun, G. Wang, From unmanned systems to autonomous intelligent systems, Engineering 12 (2022) 16–19.

[144] B. K. Mohanta, S. S. Panda, D. Jena, An overview of smart contract and use cases in blockchain technology, in: 2018 9th international conference on computing, communication and networking technologies (ICCCNT), IEEE, 2018, pp. 1–4.

[145] H. Luo, Uls-pbft: An ultra-low storage overhead pbft consensus for blockchain, Blockchain: Research and Applications 4 (4) (2023) 100155.

[146] Chainalysis, The Chainalysis State of Web3 Report: Your Guide to How Blockchains are Changing the Internet, accessed May. 19, 2024 (2022).
URL `https://go.chainalysis.com/2022-web3-report.html`

[147] K. Nabben, Is a" decentralized autonomous organization" a panopticon? algorithmic governance as creating and mitigating vulnerabilities in daos, in: Proceedings of the Interdisciplinary Workshop on (de) Centralization in the Internet, 2021, pp. 18–25.

[148] G. De La Torre, P. Rad, K.-K. R. Choo, Driverless vehicle security: Challenges and future research opportunities, Future Generation Computer Systems 108 (2020) 1092–1111.

[149] Q. Ding, W. Xu, Z. Wang, D. K. C. Lee, Voting schemes in dao governance, Forthcoming in Annual Review of Fintech (2023).

[150] L. Ceriani, P. Verme, The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini, The Journal of Economic Inequality 10 (2012) 421–443.

[151] A. Sen, On economic inequality, Oxford university press, 1997.

[152] M. Chang, Q. Min, Z. Li, Understanding members' active participation in a dao: an empirical study on steemit (2019).

[153] T. Dittmer, Diminishing marginal utility in economics textbooks, The Journal of Economic Education 36 (4) (2005) 391–399.

[154] M. C. Schouten, The Mechanisms of Voting Efficiency, Colum. Bus. L. Rev. (2010) 763.

[155] M. Singh, S. Kim, Blockchain Technology for Decentralized Autonomous Organizations, in: Advances in computers, Vol. 115, Elsevier, 2019, pp. 115–140.

[156] H. Axelsen, J. R. Jensen, O. Ross, When is a dao decentralized?, arXiv preprint arXiv:2304.08160 (2023).

[157] M. H. Tessler, M. A. Bakker, D. Jarrett, H. Sheahan, M. J. Chadwick, R. Koster, G. Evans, L. Campbell-Gillingham, T. Collins, D. C. Parkes, et al., Ai can help humans find common ground in democratic deliberation, Science 386 (6719) (2024) eadq2852.

[158] W. Kaal, Reputation as capital—how decentralized autonomous organizations address shortcomings in the venture capital market, Journal of Risk and Financial Management 16 (5) (2023) 263.

[159] Y. Zhong, J. G. Kuba, X. Feng, S. Hu, J. Ji, Y. Yang, Heterogeneous-agent reinforcement learning, Journal of Machine Learning Research 25 (32) (2024) 1–67.

[160] Z. H. Ismail, N. Sariff, E. G. Hurtado, A survey and analysis of cooperative multi-agent robot systems: challenges and directions, Applications of Mobile Robots 5 (2018) 8–14.

[161] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, S. Spanò, Multi-agent reinforcement learning: A review of challenges and applications, Applied Sciences 11 (11) (2021) 4948.

[162] D. S. Nunes, P. Zhang, J. S. Silva, A survey on human-in-the-loop applications towards an internet of all, IEEE Communications Surveys & Tutorials 17 (2) (2015) 944–965.

[163] F. Gou, J. Liu, C. Xiao, J. Wu, Research on artificial-intelligence-assisted medicine: a survey on medical artificial intelligence, Diagnostics 14 (14) (2024) 1472.

[164] J. Tian, H. Li, Y. Qi, X. Wang, Y. Feng, Intelligent medical detection and diagnosis assisted by deep learning, Applied and Computational Engineering 64 (2024) 120–125.

[165] Y. Zhang, Q. V. Liao, R. K. Bellamy, Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making, in: Proceedings of the 2020 conference on fairness, accountability, and transparency, 2020, pp. 295–305.

[166] C. Novelli, M. Taddeo, L. Floridi, Accountability in artificial intelligence: what it is and how it works, Ai & Society 39 (4) (2024) 1871–1882.

[167] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable ai: A brief survey on history, research areas, approaches and challenges, in: Natural language processing and Chinese computing: 8th cCF international conference, NLPCC 2019, dunhuang, China, October 9–14, 2019, proceedings, part II 8, Springer, 2019, pp. 563–574.

[168] C. O. Retzlaff, A. Angerschmid, A. Saranti, D. Schneeberger, R. Roettger, H. Mueller, A. Holzinger, Post-hoc vs ante-hoc explanations: xai design guidelines for data scientists, Cognitive Systems Research 86 (2024) 101243.

[169] V. Hassija, V. Chamola, A. Mahapatra, A. Singal, D. Goel, K. Huang, S. Scardapane, I. Spinelli, M. Mahmud, A. Hussain, Interpreting black-box models: a review on explainable artificial intelligence, Cognitive Computation 16 (1) (2024) 45–74.

[170] Y. Xing, D. Hou, J. Liu, H. Yuan, A. Verma, W. Shi, Deep learning and game theory for ai-enabled human-robot collaboration system design in industry 4.0, in: 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2024, pp. 0008–0013.

[171] X. Guo, X. Zhang, X. Zhang, Incentive-oriented power-carbon emissions trading-tradable green certificate integrated market mechanisms using multi-agent deep reinforcement learning, Applied Energy 357 (2024) 122458.

[172] L. Gudgeon, D. Perez, D. Harz, B. Livshits, A. Gervais, The decentralized financial crisis, in: 2020 crypto valley conference on blockchain technology (CVCBT), IEEE, 2020, pp. 1–15.

[173] Y. C. Lo, F. Medda, Uniswap and the emergence of the decentralized exchange, Journal of financial market infrastructures 10 (2) (2021) 1–25.

[174] C. Schinckus, C. P. Nguyen, F. H. L. Chong, Between financial and algorithmic dynamics of cryptocurrencies: An exploratory study, International Journal of Finance & Economics 28 (3) (2023) 3055–3070.

[175] D. Wang, S. Wu, Z. Lin, L. Wu, X. Yuan, Y. Zhou, H. Wang, K. Ren, Towards a first step to understand flash loan and its applications in defi ecosystem, in: Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Computing, 2021, pp. 23–28.

[176] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, et al., Blockchain technology: Beyond bitcoin, Applied innovation 2 (6-10) (2016) 71.

[177] V. Lemma, FinTech Regulation, Springer, 2020.

[178] R. Auer, J. Frost, L. Gambacorta, C. Monnet, T. Rice, H. S. Shin, Central bank digital currencies: motives, economic implications, and the research frontier, Annual review of economics 14 (1) (2022) 697–721.

[179] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, A. Gervais, Sok: Decentralized finance (defi) attacks, in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE, 2023, pp. 2444–2461.