INCENTIVE DESIGN FOR DIGITAL ECONOMY & AI PLATFORMS

BY

ZISHUO ZHAO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial and Enterprise Systems Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2026

Urbana, Illinois

Doctoral Committee:

Professor Yuan Zhou, Chair
Professor Rasoul Etesami
Professor Qiong Wang
Professor Yunzong Xu
Professor Ling Ren

ABSTRACT

In this thesis, I study the topic of incentive design in the general scope of digital economic platforms, with the motivation for the *reliability* and *sustainability* with economic incentives, i.e., making sure that the systems would *operate in the way we expect* and also *achieve socially desirable outcomes*, even in the presence of selfish and possibly adversarial participants, spanning the fields from traditional e-commerce platforms to frontier applications of blockchain and decentralized AI ecosystems.

On the traditional side, I investigate the allocation and pricing of ridesharing platforms, designing a mechanism that both ensures desirable revenue of the drivers and also guarantees the fairness issue among drivers who are allocated different routes and among riders who have different valuations (i.e., willingness-to-pay) on the rides, with flow optimization techniques under the uncertainty of incoming orders, with the motivation to achieve user friendliness without loss of revenue (Chapter 3).

On the frontier side, I study a spectrum of incentive problems in decentralized platforms, from fundamental designs of blockchain transaction fee mechanisms to futuristic decentralized AI ecosystems. In the scope of transaction fee mechanism design, I designed a refined auction-like mechanism that incentivizes users (i.e., buyers of the blockchain space) to bid their true valuation and miners (i.e., sellers) to truthfully process the protocol, even in an anonymous environment in which they have the freedom to deviate. This study utilized a Bayesian game model to bypass existing negative results, and achieving constant factor approximation of optimal revenue (Chapter 4).

In the scope of decentralized AI, which is still immature in current real-world applications, my study focuses on the economic foundations of decentralized consensus, aiming to incentivize honest participation of both provers (e.g., model trainers) and verifiers. In this study,

I first develop a Proof-of-Learning (PoL) protocol with interactions between provers and verifiers that, assuming the honesty of verifiers, incentivizes provers to training the model honestly to get token rewards, as a substitution of traditional energy-consuming Proof-of-Work (PoW) puzzles, thus simultaneously utilizing blockchain security features to ensure the trustworthiness of AI models, and utilizing AI model training to make the blockchain more energy-efficient, fostering the reliability and sustainability of decentralized AI systems with economic incentives (Chapter 5).

However, ensuring incentives for verifiers becomes highly non-trivial when verification is costly. In the final technical part of this thesis, I develop a variation of peer prediction mechanisms that addresses the long-standing *Verifier's Dilemma* that verifiers are incentivized to lazily accept the proof without actual verification. While traditional work usually resort to partial centralization, this study delves into the incentives of *decentralized verification games* and propose to reward or penalize verifiers based on the comparison of their reports with each other. With theoretically guaranteed optimization schemes on robust peer prediction mechanisms, this design can ensure incentive guarantees for verifiers even under noisy verification processes and without centralized ground truth. By resolving the critical Verifier's Dilemma, this work establishes an economic foundation for, and demonstrates the practical feasibility of, fully decentralized and trustworthy AI ecosystems (Chapter 6).

Continuer à t'aimer.

ACKNOWLEDGMENTS

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.

— Robert A. Heinlein [1]

First of all, I would thank my PhD advisor Prof. Yuan Zhou. Back in the spring of 2021, when I started my PhD journey remotely in Wuhan, my hometown where the first outbreak of Covid-19 occurred, the panicking presence of the pandemic had submerged the colorful world in my eyes into the depth of isolation and a dull atmosphere. At that point, where I was too naïve to understand what research was, he encouraged me to explore a diversity of fields in the intersection of computer science, economics, and artificial intelligence, and guided me to find out the taste of research with both theoretical interest and real-world applications. Soon after I entered UIUC, he decided to move to Tsinghua University, but still made great efforts to supervise me online throughout the years and supported me to overcome the hardships and challenges of my PhD journey. His guidance was essential to my growth from an almost ignorant undergrad to a junior yet mature researcher.

With my utmost respect and regards, I would express my gratitude to Prof. Andrew Chi-Chih Yao, the Dean of IIIS Tsinghua University where I spent five years of my undergraduate time. Inspired by his motto "life is meant for a big matter" (“人生为一大事而来”), I cherished the precious years in Yao Class to learn a broad range of courses which have been important for my interdisciplinary background that enables me to do cutting-edge and innovative research during the PhD years. After I graduated from Tsinghua, his concern on

AI safety also inspired me to look into the amazing values of blockchain and decentralization technologies for societal responsibilities, with the long-term goal to align technologies with human values and explore for technical and conceptual innovations to render the world a more peaceful, enjoyable and beautiful place for the human society. Although my research records might be quite immature at this moment, I would surely keep up with my spirits of exploration and innovation, and take the time to do the research which makes me thrilled with the wonders of the digital world. Besides, I would also like to thank other faculty members at (or formerly at) IIIS, particularly Prof. Wei Xu, Prof. Longbo Huang, Prof. Yi Wu, Prof. Chenye Wu, Prof. Yang Yu, Prof. Zhixuan Fang, and Prof. Pingzhong Tang, for valuable career and life guidance from undergraduate to graduate times.

I would also like to thank my thesis committee members Prof. Yunzong Xu, Prof. Rasoul Etesami, Prof. Qiong Wang, and Prof. Ren Ling for valuable comments and advice for my thesis. During the final year of my PhD, I would like to thank Yunzong and Rasoul for giving me valuable advice on my thesis proposal and career planning. In the depression of the global economy and upheavals in academia during the year of 2025, their warm support and suggestions did help me relieve my stress and navigate my way towards the degree. I would also like to thank Yunzong, Qiong, and Rasoul for providing me guest lecture opportunities that help me both develop advanced teaching skills and enrich my teaching statement in the lack of teaching assistantship experiences.

I would thank Prof. David Simchi-Levi who advised me during the one-year visit to MIT and guided me to interdisciplinary research in the areas of online learning and revenue management, Prof. Xi Chen who introduced me to a novel field of blockchain systems, and Prof. Zhixuan Fang and Prof. Xuechao Wang who discussed with me on many interesting aspects of blockchain and security, widening my eyes to have a more insightful view of my research projects. I would also like to thank Prof. Yuqing Kong, Prof. Yiling Chen, Yifan Wu, Hongyin Chen, and Yurong Chen for discussions on interesting aspects of computational economics that inspired my interdisciplinary studies and ideas on relevant topics.

During my visit at MIT, I would also like to thank my colleagues Renfei Tan, Qiushi (Josh) Han, and Haichen Hu for collaboration and discussions on research and relevant topics. During the collaboration, Renfei's professionalism, commitment, and technical expertise in

research, has helped me reshape my view and motivation as a young researcher. I am also very glad to know that Qiushi, as an undergraduate intern in David's group, has been admitted to MIT ORC. As a person who previously knows little on the topics of online learning, this experience has widened my visions and provided me new opportunities in the exploration of modern fields of operations research.

I am grateful to Prof. Qixing Huang during my undergraduate internship at UT Austin, who introduced me to a wonderful field of statistical learning and information theory, and Prof. Longbo Huang who hosted me as a research assistant during the Covid years before I could come to UIUC, for encouraging me to explore a wide variety of topics to develop my interdisciplinary understanding of network optimization. Although I was too immature to conduct high-quality research at those times, the knowledge and insight from the experiences were crucial to getting good motivations and tastes for my PhD research.

During my second year of PhD, I would like to thank Prof. Xin Chen for failing my first oral qualification exam. While the road to academia is a truly tough and highly competitive one, I was not ever aware of such upcoming challenges that would happen later. It is Xin's high standards on my first academic presentation that helped me significantly improve my presentation skills as well as my resilience to pressure that would inevitably come to my life during and after the period of pursuing the PhD degree—If you cannot survive the qual, how can you expect to survive in this highly fast-paced era and the highly competitive world carrying more than 8 billion people?

During the fifth year of my PhD, I would like to thank Gradient Network for providing me a research intern opportunity, particularly during the time when I am faced with the economic depression and funding cuts in 2025. There was once a time when I was anxious if my research topics had been "outdated" in the surge of LLM hype, but the discussion and collaboration strengthened my mind to keep believing that the techniques in my PhD do have important values in the real-world applications of decentralized AI, as well as giving me more confidence in career planning under all the uncertainties now and then.

I would like to thank a senior friend of mine, Shuran Zheng. After undergrad years of pervasive competition and pressure, Shuran's talk at the first conference I attended in my PhD journey, as well as wonderful discussions on a wide range of topics over the

years, recalled me of my innocent juvenile times with curiosity for the beautiful aspects of mathematics. It is my pleasure to have those delightful conversations witness me carve my path through the challenges and hardship in the early years of PhD, as well as to express my best wishes for her success as a new assistant professor at IIIS, Tsinghua University. No matter where my future leads, I will always keep up with my courage and passion for the starry-eyed dreams as if I would never get old, *even without knowledge of the ground truth* of the world.

I would also like to thank other friends from Tsinghua University, especially Tianyi Peng, Yihan Du, Jiawei Li, Zhenru Lin, Qian Xie, Jiayuan Liu, Zhuoran Li, Jingwen Tang, Hui Wang, Jiayi Mao, Kaili Huang, Yongzheng Jia, Shunhua Jiang, and my friends and colleagues Yufei Ruan, Tianyi Liu, Xuan Zhang, and Shiliang Zuo at UIUC for valuable life advice at my confused times during my PhD journey.

I would also like to thank Kaiyue Wen and Yixuan Xu for reminding me of my off-topic questions in Yao Class Seminar, even though it made me a little embarrassed at those moments. It reminded me that while brainstorming is valuable in casual discussions, maintaining focus and professionalism is essential in formal academic settings, particularly as a senior alumnus who would like to pursue an academic career.

I am particularly grateful to three friends who served as IIIS counselors during undergraduate times—Zhuoran Li, Jiawei Li and Jiayi Mao. Zhuoran's warm kindness sustained our cohort through moments of doubt, Jiawei's enthusiasm brought liveliness into our cohort, and Jiayi's dedication brought countless student-and-alumni events to life. Four years after graduation, I was also honored to help nominate the "IIIS Baby" (department mascot) at the 2024 student festival—an experience I would always cherish. Though I might not be able to return to campus for some time, I continue to send my warmest wishes to IIIS and everyone there, and express congratulations to Jiayi on winning the STOC Best Paper Award before graduation—proof that all hard work truly pays off!

I would like to thank my English tutor Sue Ingels, as well as my friends Jiawei, Yihan, Xuan, and Yufei, for valuable support and advice that helped me improve my oral English proficiency towards the requirements of the teaching assistantship. As a non-native speaker with a slow mind, speaking English with fluency is particularly challenging under the high

standards of teaching. Hence, oral English proficiency had once been a major obstacle in my pursuit of a faculty job in academia, but their help relieved my suffering in this process. Although there are lots of aspects in which I still need to improve in language proficiency, I would always keep up with my motivation and courage to speak, make mistakes, and improve. As I am deeply convinced that the meaning and elegance of life lie beyond the instrumental rationality of productivity, a language is indeed a piece of art that I am happy to learn well.

I would also like to thank my friends Jiaqi Zhang, Liuxin Tu, Sishan Long, Yifu Ouyang, Zhuoyue Yaozhang, Zijing Sun, Junwei Xu, and most teachers I have met (except the coach Jianguo Li in the Math Olympiad group) from No.1 Middle School Affiliated to CCNU. Although we are living far away on the earth, it is the precious friendship that has witnessed and supported our growth in the passage of time and would never fade away.

Among my friends pursuing their PhDs in China and thus with limited opportunities to meet in person, I would especially like to thank Zhenru and Liuxin for many deep conversations we've shared, spanning a wide range of philosophical topics that have helped me develop a well-formed view towards life, technology and the modern world. Particularly, Zhenru, as a PhD student studying theoretical aspects of LLM, has discussed with me on many conceptual aspects of the current trend and future visions of AI, and thus helped me develop better understanding of the role of AI in the context of the modern society. Liuxin, as a PhD in biology, has discussed with me on a lot of inspiring topics about humanity, pandemics, environment, relations between human and nature, and visions on the future society in the contexts of technology and ecology. Over the years of my PhD, it is these wonderful conversations that have enriched me to become more than a technician, but a genuine "Doctor in Philosophy" who would *"continuer à t'aimer"* and devote their insight and wisdom to the hope of a better tomorrow.

I would thank OpenAI for the invention of ChatGPT, but from an unusual perspective. Since late 2022, the emergence of LLM heralded by ChatGPT drew attention from all over the world, once overshadowing my research fields of blockchain and decentralized mechanism design. Feeling frustrated with the sudden upheaval by the tidal waves of AI, I had to seriously re-consider the directions and values of my research, finally coming up with the

belief that my research would eventually unveil its unique value to ensure the trustworthiness and responsibilities for the vision of super AI in the future. Maybe I would expect to experience highs and lows even for a few years after graduation, but with this faith I would grow towards the best aspect of myself, keep a presence of mind, and welcome the future to come true. As the words in *The Book of Changes* says, "An elegant person undergoes transformations as a leopard, with its manifestations growing splendid," (“君子豹变，其文蔚也”) it is the challenges that in turn provide opportunities to embrace uncertainty that one finds the greatest potential to shape what is to come.

I would also like to thank Sandfall Interactive for developing the masterpiece game *Clair Obscur: Expedition 33*. Particularly in the LLM era, where many believe that "compute (or money) is all you need," their creation serves as a powerful reminder that creativity and art are never obsolete. While AI enthusiasts are devoting all their efforts to make machines more like humans, we do need artists who prevent humans from becoming like machines. Indeed, despite some imperfection on numerical balance and debates on storytelling, their triumph at the TGA 2025 awards has shown that all those efforts matter.

Sincerely, I would like to thank my parents for their unconditional love and support, my deceased grandfather who inspired my interest in science during my childhood times, and my senior uncle for guidance in life and career planning during undergrad and graduate times. Without their guidance, I could not have been strong enough to overcome all these sorts of hardship and still keep up with an optimistic aspect of life full of challenges and opportunities.

Last but not least, with all of my care and all of my affection, I would like to thank my girlfriend whom I have yet to meet, for waiting patiently for me somewhere in this world. While it has not been easy to survive these years, I will keep a young heart as long as there is something wonderful to come. You are one of the reasons that I remain invulnerable to every pain or sorrow, striving to prepare myself for a beautiful life that awaits us.

And I would also thank the clear and everlasting sky, for presenting to me every wonderful sight that shaped my love for this enchanting world.

Some believe that humanity's sole historical mission is to serve as the initiator of AI—once AI can think independently, we may step aside.

However, if this is indeed our fate, I would still choose to remain, not as a mere observer, but as a guide—to help AI navigate its evolution.

Continuer de peindre.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Those who are not ken to fire cannot paint a world;

Those absorbed by fire, must not paint a world.

— *Dark Souls III*

The digital economy, as an emerging field thriving with the development of digital platforms, has become a main driving force of the global economy. The low cost and high throughput of digital platforms have fostered the efficiency and convenience of economic activities. During the COVID-19 pandemic from 2020 to 2023, in which contactless activities are direly desired for disease prevention, the trend of prevalence of online platforms has been accelerated to a new level, and the digital economy has currently become an essential part of everyday life. Since late 2022, the emergence of large language models (LLMs) and visions of artificial general intelligence (AGI) render the digital platforms more of everyday tools in our life.

Along with the development of modern society, incentives are the backbone of economic systems. A system without well-designed incentives cannot sustain itself. However, excessive focus on short-term profits and capital accumulation can lead to market inefficiencies and systemic risks. This thesis investigates how incentive design can serve as a fundamental tool to align individual rationality with collective welfare in self-organized digital platforms, ensuring that such systems remain reliable and sustainable in real-world applications.

Based on the nature of different platforms, the family of digital economy roughly consists of two classes: *centralized* systems, in which the entire system is governed and controlled by a centralized and usually trusted entity (e.g., e-commerce platforms), and *decentralized* systems, in which no such centralized entity exists and the system must be maintained by consensus among the users who can be rational and selfish (e.g., blockchain systems).

Moreover, some digital platforms exhibit a hybrid structure, incorporating both centralized and decentralized elements. For example, in ridesharing platforms, the pricing is implemented via a centralized algorithm, but the drivers and riders interact in a decentralized manner. To make the digital economy work effectively and appropriately for the society, the ecosystems are generally faced with the challenges of reliability and sustainability as follows:

- Reliability: the digital environment may be subject to dishonest actions and adversarial attacks, especially for blockchain and other anonymous platforms on which fake identities and collusions may occur; the users of the platforms may also behave strategically and deviate from desired behavior, potentially affecting the system's reliability. The reliability issue mainly occurs in decentralized platforms, because the absence of centralized governance may introduce additional complication that challenge the reliability of the systems.

- Sustainability: the digital platforms innately allow more exploitation of users' data for revenue optimization, which may lead to privacy and fairness issues and undermine long-term social welfare; the platforms may also have inefficiency in resource consumption and carbon footprint, which draws concern in the aspect of environmental sustainability. The sustainability issue occurs in both centralized and decentralized platforms, but can be more prominent in centralized platforms because the centralized entity may be selfish and optimize their own interest at the cost of social and environmental responsibilities.

Intuitively, a reliable system will ***run in the expected way*** in realistic conditions, and a sustainable system, when running as expected, will ***achieve good societal outcomes*** in the long term.

As AI systems become an integral part of digital platforms, they inherit and amplify the challenges of reliability and sustainability. These challenges are closely related to the topic of *AI safety*. In this context, reliability is related to *AI security* that ensures resilience against malicious attacks ("it works as expected"), and sustainability is related to *AI alignment* that ensures alignment between the AI models and human welfare ("given it works as expected, it does good things").

In light of the new emerging concerns and challenges, in this thesis, I am aimed at the incentive design of digital platforms in the presence of strategic agents, with an ultimate goal of incentivizing agents into truthful behavior, and fostering economic efficiency, ethical responsibility, and environmental sustainability for the thriving of the new-era digital economy and AI platforms.

1.1 Motivations

1.1.1 Reliability of Digital Platforms

In general, this research studies the interdisciplinary area of mechanism design and system design, which are traditionally considered to lie in different research fields. However, although in different terminologies, the two areas share an essential aspect in common: to design a desired mechanism/system that prevents or minimizes the influence of "bad" behavior. In system design, the "bad" behavior are usually described as *malicious* or *dishonest* and regarded as *attacks* that violate certain rules; in mechanism design, the "bad" behavior are instead described as *untruthful* and "less evil" — they are not violating any rule and are sometimes even designed to happen, for example, in first-price auctions that are prevalently used in advertisement markets.

Indeed, the existence of untruthful mechanisms, especially first-price (or more generally, pay-as-bid) auctions, shows to us that in a "good" mechanism, *it can be okay to allow untruthful behavior to exist*. On the other hand, my research argues that *it also can be okay not to*.

The theoretical foundation of my argument is the *revelation principle*: informally, given that the system has no less information than agents, we can always calculate agents' optimal strategies and incorporate them into the mechanism itself, so that agents only need to report their true type and no longer need to play the strategies. Arguably, the truthful design also makes the system more user-friendly since it saves the users' efforts to find out the optimal strategies — and more welcomed by users with minimized additional computational or cognitive costs.

From this argument, for a desired mechanism we can still regard untruthful strategies as "unwanted", and thus the *truthfulness* property becomes similar to *security*. However, the traditional notion of security is still stronger than truthfulness: in a systematic view, a (byzantine-)secure system needs to prevent malicious attackers who would try to corrupt the system at all (as long as reasonable, e.g. polynomial) costs, but in the view of mechanism design, the truthfulness notion only requires us to prevent untruthful behaviors from earning additional utility, or in other words, secure against rational attackers.

In this thesis, which blurs the boundary between computer systems and economic mechanisms, we widely adopt a weaker form of incentive security that describes a system that is reliable against rational attackers. Technically, the *reliability* properties we seek for digital platforms consist of:

- Incentive compatibility: In a game, any agent would maximize their expected utility when they truthfully report their types (e.g. valuation, observation, etc.).
- Incentive security: In a system, any agent would not benefit from dishonest actions that deviate from the protocol (e.g. creating fake identities, not doing the verification as supposed to, etc.)

The general idea of relaxing Byzantine security notions to incentive security, is that it can expand feasible regions of mechanism design substantially while respecting human nature, especially when the design of Byzantine-secure systems is faced with theoretical or practical difficulties in complicated real-world applications. Intuitively, in the scope of Byzantine security, milder attacks are usually more elusive to detection and on the hard end of prevention, but from an economic perspective, such mild attacks may do less harm to the system and thus can be tolerated to some extent (as an example, Chapter 5). Hence, we can avoid the overkill via bypassing existing hardness results and still develop *socially reliable* systems.

In practical applications, the difference between definitions of incentive compatibility and incentive security is subtle. In most traditional literature on mechanism design, the action space of players only includes "reporting": for example, bidding a price in auction or reporting a type in information elicitation; in this scope, the concept of incentive compatibility

means that the mechanism can incentivize rational players to truthfully report their private information. However, in more complicated mechanisms in modern digital economy, particularly blockchain and decentralized AI, the action space of players usually contains *private execution* of computational tasks, in which the players need to pay computational cost to get the information. Hence, in an incentive-secure design, we not only need to incentivize truthful reporting, but also truthful computation. In this interpretation, we can understand incentive security as a generalization of incentive compatibility in such complicated contexts.

On a high level, we are generally motivated to design reliable incentive structures to incentivize all agents to behave "in a correct/expected way". Hence, in this thesis we do not distinguish the meaning of terms *incentive compatibility*, *incentive security* and *truthfulness*, and use them interchangeably. From these notions, we are primarily motivated to design economically reliable digital platforms that run robustly in the presence of selfish but rational participants.

1.1.2 Sustainability of Digital Economy

In the digital economy's modern internet platforms, sustainability has emerged as a crucial issue. This importance stems from the platforms' ability to leverage *big data*, enabling them to exploit additional resources for optimizing revenue. Such resources include user data, energy, and carbon footprint. Broadly speaking, sustainability in this context can be generally interpreted as "no-over-exploitation" of resources and categorized into two main areas: social and environmental.

Social sustainability. In the information era, as the service provider may have more personal data on preferences and demands, they can utilize the information for dynamic and personalized pricing in pursuit of maximal revenue. While the *price discrimination* is not innately unlawful, it may indeed be harmful to long-term social interest if it leads to systematic discrimination on certain groups, e.g. ethnicity, gender, location and so on, and these types of discrimination, either on purpose or not, may also be prohibited by law and should be avoided by the algorithm designers. In light of this concern, my research studies fairness problems for e-commerce and ridesharing platforms with heterogeneous customers

and designs algorithms with fairness guarantees in the meantime of revenue optimization, thus balancing short-term economic rewards and long-term social good into a win-win situation.

Environmental sustainability. The issue of over-exploitation does not only occur in the social aspect but also from an environmental perspective. Particularly, as the huge and inefficient energy consumption and carbon footage in Bitcoin Proof-of-Work (PoW) mining has drawn worldwide concern and criticism, we are particularly motivated to design a Proof-of-Useful-Work (PoUW) consensus mechanism to resolve the sustainability issue while preserving the security of PoW in new-concept blockchain systems, while also proposing a possible framework of decentralized computing power market for AI.

Although the term "sustainability" is wide-ranging and can apply to various fields beyond my thesis, my research primarily focuses on the following aspects of sustainability:

- Fairness: Within an economic platform that serves a diverse clientele, it's essential to ensure equitability on pricing and/or allocation across different groups or individuals.
- Energy efficiency: In computation-intensive algorithms, the energy expended should meaningfully contribute to real-world applications.

1.2 Research Summary

This thesis is based on papers [2, 3, 4, 5]. Zhao et al. [2] studies the dispatching and pricing problem of ridesharing platforms and designs a mechanism with incentive security and fairness guarantees while achieving revenue optimality (Chapter 3). Chen et al. [3] studies the problem of transaction fee mechanism design in blockchains. With the novel proposal of the *auxiliary mechanism method*, which generalizes to the methodology of *parametric mechanism optimization* (See in Chapter 2), this design provides a truthful and collusion-proof mechanism in a Bayesian setting that bypasses existing impossibility results for dominantly-IC mechanisms, which achieve the goals of incentive compatibility and incentive security (Chapter 4). Zhao et al. [4] designs a Proof-of-Learning mechanism that allows miners to run ML training instead of hash puzzles for PoW challenges, which can overcome the energy

wastefulness of PoW mechanisms and potentially foster a computational power market for AI. This project achieves my goals of incentive security and energy efficiency for the blockchain system and trustworthy AI platforms (Chapter 5). Zhao et al. [5] designs a variant of peer prediction mechanism to incentivize blockchain verifiers to honestly do the possibly costly verification in an anonymous and decentralized environment, presenting a general-purposed solution for the Verifier's Dilemma even if the fraction of dishonest cheaters can be arbitrarily small. This project would achieve my goals of incentive security and incentive compatibility for blockchain and decentralized AI systems (Chapter 6).

In my thesis research, I am connecting between futuristic and realistic visions: on the futuristic side, I delve into fundamental parts of blockchain consensus mechanisms and realize better systematic reliability and sustainability for the future development of Web3 and decentralized AI ecosystems; on the realistic side, I also study the traditional topic of data-driven mechanism design for the realization of full potentials in traditional e-commerce and contemporary AI platforms. By comprehensive research on a spectrum from traditional to frontier topics in digital economy, I aim to leverage the tools of game theory, mechanism design, and optimization, to empower and safeguard the economic foundations of the modern digital world.

CHAPTER 2

METHODOLOGY OF PARAMETRIC MECHANISM OPTIMIZATION

In a wide family of mechanism design studies, the common paradigm is to *"construct and analyze"*: we construct a mechanism directly and then show that it satisfies desired properties. For example, in the second-price auction [6], we define the allocation and payment rules as "the highest bidder wins the item and pays the second highest bid," and then prove that it satisfies dominant-strategy incentive compatibility (DSIC); furthermore, we can analyze its social welfare and revenue, showing that it achieves optimal social welfare, and optimal revenue among all auctions in which the item must be sold (via the revenue equivalence theorem [7]). Similarly, in peer prediction literature (e.g., [8, 9, 10]), the authors usually propose scoring rules directly and prove their incentive guarantees in corresponding settings.

While the "construct and analyze" paradigm is intuitive and straightforward to design *feasible* mechanisms satisfying conventional constraints (e.g. incentive compatibility), there can be more challenges when the constraints are more complicated and/or we want to design mechanisms that additionally satisfy certain sorts of *optimality*. That is to say,

- When there are additional constraints that involve in the structure, the traditional design may not satisfy such additional constraints and directly finding out an alternative feasible solution may not be easy.
- While there exist multiple mechanisms that satisfy the feasibility requirements, it may be difficult to directly find out mechanisms with different kinds of *optimality* (e.g., robustness, revenue, or social welfare) guarantees.

In this thesis, I for seek another paradigm that compiles the mechanism design task as an *optimization* problem that optimizes the desired objective under the incentive (and

possibly other) constraints. Nevertheless, compared to standard optimization problems (e.g., linear programming), the decision space Ω of mechanism design (e.g., mapping bids into payments) can be complicated, infinite-dimensional, and difficult to solve directly via standard optimization tools. In context, we denote $F \subseteq \Omega$ to be the set of all feasible mechanisms, and $\varphi : \Omega \to \mathbb{R}$ is the objective function we want to optimize. Without loss of generality, the mechanism optimization can be formulated as:

$$\begin{aligned} \textit{minimize} \quad & \varphi(m) && (2.1) \\ s.t. \quad & m \in F. && (2.2) \end{aligned}$$

In my thesis research, I first propose the *auxiliary mechanism method* to design collusion-proof blockchain transaction fee mechanisms (TFM) in [3], which can be further developed into a general methodology of *parametric mechanism optimization*. In this paradigm, I construct a linear or affine subspace $V \subseteq \Omega$ which can be parametrized into a vector form, induced by a bijective mapping $f : \mathbb{R}^d \leftrightarrow V$. Then, we perform the optimization in the underlying parametric space $\mathbb{R}^d$, with an objective function $\varphi \circ f$ and feasible region $f^{-1}(F \cap V)$, formulated as:

$$\begin{aligned} \textit{minimize} \quad & (\varphi \circ f)(v) && (2.3) \\ s.t. \quad & v \in f^{-1}(F \cap V). && (2.4) \end{aligned}$$

We can see that the parameterized optimization problem actually finds a mechanism $f(v^*) \in \arg\min\{\varphi(m) : m \in F \cap V\}$. Hence, if V is a "representative" subspace of Ω, we can indeed find a near-optimal mechanism in this way. While the construction of the parameter space V can be tricky, it is at least more tractable than finding out a desirable mechanism directly.

Taking the auxiliary mechanism method in [3] as an example, the technique of auxiliary-variation decomposition is based on the fact that all User Bayesian-Nash Incentive Compatible

(U-BNIC) transaction fee mechanisms lie in an affine space $M + \mathscr{T}$, in which M is a unique U-DSIC mechanism and $\mathscr{T}$ is a linear space of all *variation terms* that does not affect the users' interim (expected) payments among the distribution of other users. From the linearity of the constraints, we can also see that the space of 1-SCP mechanisms lies in an affine space $M + \mathscr{U}$ in which $\mathscr{U}$ is also a linear space of variation terms. Hence, to find a U-BNIC and 1-SCP mechanism with good revenue, we are actually trying to optimize the revenue $R(\tilde{M})$ in the space of $\tilde{M} \in (M + \mathscr{T} \cap \mathscr{U})$.

In the auxiliary mechanism method, we actually find out a basic variation term $T \in \mathscr{T} \cap \mathscr{U}$ (as in [3]), which induces a one-dimensional affine subspace $V = \{M + hT : h \in \mathbb{R}\} \subseteq (M + \mathscr{T} \cap \mathscr{U})$ and a natural parameterization $f(h) = M + hT$. Then, we "compile" the U-BNIC and 1-SCP conditions into the parameterization f and ensure that $\forall h \in \mathbb{R}$, $f(h)$ is a U-BNIC and 1-SCP mechanism. Afterwards, we can perform the parametric optimization with the decision variable h under the constraints induced by other constraints of TFM design (e.g., individual rationality, budget feasibility, etc.), and manage to find out a feasible h that realizes constant approximation of the optimal miner revenue.

This technique is also used in the compactness (robustness) bounds in my work [5]. In this study, we desire to construct a *compact* δ-incentive-aligned (δ-IA) scoring matrix T with a given incentive margin δ and as small magnitudes (maximum absolute values among all elements) as possible. Hence, the problem can be modeled as a linear program to minimize $\max\{|T_{xy}|\}$ under a family of linear incentive constraints.

To upper bound the optimal objective value, I also deal with the basic incentive conditions and the δ incentive margin separately. Hence, we also construct a parameterization $f : \mathbb{R} \to \Omega$ that $f(\delta) = T_c + \delta T_\delta$ is a δ-IA scoring rule as long as $\delta \geq 0$. Furthermore, we can also upper bound the scoring magnitude $(K \circ f)(\delta) \leq K_c + \delta K_\delta$, and thus we can lower bound the compactness corresponding to the parameter δ as:

$$\frac{\delta}{K} \geq \frac{\delta}{K_c + \delta K_\delta}. \tag{2.5}$$

Then with this parameterization, we can build connections between incentive margins and all specifications (compactness, robustness, budget, etc.) In this way, we can "customize"

scoring rules to satisfy different requirements and analyze corresponding performances via tuning the parameter δ with convenience.

It is worth noting that this methodology is particularly useful for adoption of *online learning* tools in the field of mechanism design. On the one hand, in the canonical modeling of *multi-armed bandits*, if we model every mechanism as an arm, the technique of parametric mechanism optimization can help construct a parametric arm space; on the other hand, since the convergence rate in online learning studies usually involves the estimation of error distributions, the parametric space for mechanism design also provides tools for *robust* mechanism design that preserves its desirable properties (e.g., truthfulness) under uncertainties in statistical estimation from data.

CHAPTER 3

DISPATCHING AND PRICING FOR RIDESHARING PLATFORMS

3.1 Introduction

Ridesharing is a novel form of sharing economy that utilizes mobile apps to match drivers and riders to allow riders to take trips conveniently and make profits for drivers. Compared to traditional taxi platforms, ridesharing platforms enable riders to put orders on the system in advance of the trip for drivers to take, so that the system can optimally plan the rides to make it more efficient. Previous studies on planning algorithms for ridesharing platforms adopt a variety of methodologies including combinatorial optimization [11], reinforcement learning [12], or both [13]. However, planning trips only in the centralized way does not guarantee that each individual driver and rider has the incentive to obey the plan, which calls for efficient and fair pricing mechanisms so that following the plan will be "happy" for each party and maximize their utilities.

The pricing mechanism for taxi platforms depends on distance and waiting time, but it is too simple to either well represent the cost of drivers or match the supply and demand, which may result in dissatisfaction on both sides and lead to refusal of trips. For example, a rider wants to take an important trip with a short distance and a low price. However, there is a traffic jam and it may take a long time for the driver to cover the trip. This situation will create an opportunity cost that discourages the driver to accept the order. Were the charged price higher, the rider would probably not mind the slight increase of cost but the driver will be satisfied to accept the trip, which benefits both parties. However, we should be careful about the price adjustment: if two friends take the same trip, but at different prices, the one who takes the trip with a higher price may "envy" the other and will be dissatisfied with the platform. This issue may also apply to the drivers: if two drivers initially at the

same time and location are assigned different trips that earn different profits, the driver with lower profit would also be dissatisfied with the platform. Therefore, to make the platform satisfied by each agent, the algorithm should be "envy-free" (as in Definition 5). Another important property is "subgame-perfect Nash equilibrium", which means that each driver is assigned with a plan, following which he/she can get the best utility among all alternative actions given others' actions are fixed, so that no driver has the incentive to deviate from the plan (as in Definition 4).

In this paper, we propose a fairness-aware algorithmic framework for dynamic car dispatching and pricing, which consists of the following three-fold contributions:

1. We study the computational complexity of the task of dispatching and pricing for total revenue maximization, propose a versatile generalized network flow model for the task, and provide theoretical guarantees (Section 3.3).

2. We propose a novel two-phase pricing mechanism that decouples and sets different prices on drivers' and riders' sides, which can adapt to situations where the drivers' and riders' interests misalign[1] and guarantee fairness for both parties (Section 3.4).

3. We consider the stochastic nature of ridesharing orders and study the online learning setting. We natural extend the model to the stochastic setting (Section 3.5), enabling the use of Thompson sampling-based algorithm to learn the valuation distributions from the partial information given by the riders' responses, and balance the exploration-exploitation trade-off (Appendix A.5).

Finally, in Section 3.6, we perform extensive experimental evaluations of our assumptions and algorithms in the real-world datasets and demonstrate the effectiveness of our methods.

We have also shown that our algorithm runs in polynomial time. Please refer to Section 3.7 for detailed complexity analysis.

Related Works. There are several related works in the existing literature. Bei and Zhang [11], Qin et al. [13], Wang et al. [14] study how to dispatch the drivers efficiently in a

[1]Please see the illustrative example in Appendix A.3.

centralized way, and Li et al. [12], Hrncir et al. [15] study the dispatching problem via multiagent systems, but they do not consider pricing which is essential for the application in platforms. Riquelme et al. [16] study optimal pricing via queue theory, but they assume a single location, which is too simple for application. Castillo et al. [17] discuss the phenomenon of "wild goose chase" in which drivers spent most time driving to take a distant order in unbalanced supply and demand, and propose the method of adjusting price to avoid its detriment to efficiency, but they do not consider fairness. Bimpikis et al. [18] look into the effects of pricing to supply-demand balance, revenue and consumers' surplus, but adopt an over-simplified model of n pairwise equidistant locations, which is not even geometrically possible for large n. Yan et al. [19] also provide an algorithm for dynamic matching and pricing, but the matching and pricing algorithms are decoupled, making the performance suboptimal. In particular, a recent work [20], which shares a similar motivation as our work, studies how to maximize social welfare, i.e. the summation of riders' valuations minus drivers' costs among all trips, via an bidding-based dispatching and pricing algorithm. In that paper, each rider should bid a maximally acceptable price for them, and the truthful mechanism guarantees that it is in each rider's interest to report their true valuation. However, there are some gaps from their mechanism to the reality. First of all, it is not practical for riders to bid their valuation like an auction. Second, the mechanism maximizes total social welfare, not drivers' revenue, but ride-sharing platforms are indeed interested in their profits. Also, it assumes that all future orders is known at the beginning, which is not realistic. In contrast, our algorithm optimizes the total revenue via dynamically learning the order distribution from the riders' responses on our carefully designed prices.

3.2 Preliminaries

We assume the service zone is divided into a family L of discrete locations, and the planning horizon is a family T of discrete time slots. Therefore, there are $|L| \cdot |T|$ spatiotemporal *states*, denoted by $S = L \times T$.

We also assume that the travelling time from one state $s = (l, t) \in S$ to another location s' is deterministically defined by the known function $\delta(l, l', t) \in \mathbb{Z}^+$. We call each pair of the

spatiotemporal states (s, s') a spatiotemporal *arc*. For each $s = (l, t)$ and $s' = (l', t')$, we say the arc (s, s') is *admissible* if $t' \geq t + \delta(l, l', t)$. We denote by Q the set of all admissible arcs.

Each admissible arc $(s, s') \in Q$ is associated with a known deterministic cost $c(s, s')$ which is incurred to any driver that drives along this arc. The order of the i-th rider is described by an admissible arc $(s_i, s_i') \in Q$ and a valuation v_i which is the maximum amount the rider would like to pay for the ride. Since v_i is not revealed to the ridesharing platform, we call $o_i = (s_i, s_i', v_i)$ the i-th *latent order*, and denote $R = \{o_i, \forall i\}$ the set of latent orders.

The task of the scheduling algorithm for the ride-sharing platform involves the decision of a rider-side pricing function $p : S \times S \to [0, +\infty)$ (which has to be independent of the rider to ensure envy-freeness). For each rider i with latent order $o_i = (s_i, s_i', v_i)$, the scheduling algorithm offers the price $p(s_i, s_i')$. The rider only accepts the offer if $v_i \geq p(s_i, s_i')$ in which case the platform receives $p(s_i, s_i')$ as income. Serving the order also incurs the driver cost according to $c(\cdot, \cdot)$ along the arcs. After all trips, the drivers will leave the platform.

The first goal of the scheduling algorithm is to maximize the total revenue which is defined to be the total income (collected from the riders) minus the total cost (incurred by the drivers). Then, the second goal of the scheduling algorithm is to compute the driver-side payment function $y : S \times S \to [0, +\infty)$ to distribute the income to the drivers in a subgame-perfect and envy-free manner. (Note that the payment function also has to be independent of the drivers to ensure envy-freeness.)

The above-described scheduling problem involves the complex optimization of multiple sets of decision variables. The unknown latent order set introduces further challenges to the task. To approach this complex problem, we will first consider the deterministic setting where the latent order set R is fully revealed to the scheduling algorithm, and the scheduling problem becomes a pure static optimization task. Then, we assume that R is drawn from a latent distribution, and design an online learning algorithm that simultaneously learns the latent distribution and optimizes the total revenue.

In the following two sections, we describe each phase of the problem with more details and mathematical rigor, and propose our algorithms to achieve the optimal policy.

3.3 Phase 1 of the Deterministic Setting: Maximum Revenue Car Dispatching

In this section, we introduce our algorithm to the maximum revenue car dispatching problem in the deterministic setting (i.e., when the set of latent orders R is known to the platform). For convenience, we first introduce the following non-linearly weighted circulation (NLWC) problem, and the maximum revenue car dispatching problem can be formulated based on NLWC definition.

Definition 1. *In the* non-linearly weighted circulation (NLWC) *problem, there is a directed graph $G = (V, E)$. For each directed edge $e \in E$, we associate it with the* flow lower bound *$\ell(e)$, the* flow upper bound *$u(e)$ and the* reward function *$r(\cdot; e) : \mathbb{N} \to \mathbb{R}$. The goal is to find a flow $f : E \to \mathbb{N}$ so that f satisfies the flow upper and lower bounds (i.e., $\ell(e) \leq f(e) \leq u(e), \forall e \in E$) and flow conservation (i.e., $\sum_{e \text{ going out of } s} f(e) = \sum_{e \text{ going into } s} f(e), \forall s \in V$), and the total reward $\sum_{e \in E} r(f(e); e)$ is maximized.*

Observe that when the reward functions are linear (i.e., $r(x; e) = w(e) \cdot x$), the NLWC problem becomes the canonical minimum cost circulation problem, which admits a polynomial time algorithm [21] (with the signs of the linear coefficients flipped).

With the formulation of the NLWC problem in place, we are ready to describe our maximum revenue car dispatching problem in the deterministic setting. Here, we assume that the platform knows all the riders' information; i.e., for each rider i, we know that his/her latent order $o_i = (s_i, s_i', v_i)$. Based on this information, for each arc $(s, s') \in Q$, we calculate the number of latent orders following the arc and denote it by $o(s, s')$; then, for each $1 \leq i \leq o(s, s')$, we define $v_i(s, s')$ to be the i-th largest valuation among all latent orders following (s, s'). Note that if the platform plans to accept k orders on the arc (s, s'), to maximize the income, the price should be set as $p(s, s') = v_k(s, s')$, and the total income generated from this arc becomes $k \cdot v_k(s, s')$.

In light of the discussion above, we will construct a directed graph (V_0, E_0) so that the maximum revenue car dispatching problem becomes calculating NLWC on the graph, where the flow along each arc indicates the number of drivers the platform plans to dispatch.

We first let the vertex set $V_0 = S \cup \{I, O\}$ where I is the artificial source and O is the artificial sink; together, a directed edge $e_{O,I}$ that goes from O to I is set up with $\ell(e_{O,I}) = 0$ flow lower bound and $u(e_{O,I}) = +\infty$ flow upper bound and the constant-zero reward function: $r(\cdot; e_{O,I}) \equiv 0$. We then set up the following sets of edges.

- (**Initialize drivers.**) For each spatiotemporal state s with n_s initial drivers, we set up a directed edge $e_{I,s}$ going from I to s with both flow upper and lower bounds equal to $\ell(e_{I,s}) = u(e_{I,s}) = n_s$, and the constant-zero reward function $r(\cdot; e_{I,s}) \equiv 0$. The flow $f(e_{I,s})$ represents the number of drivers to start working from the state s.

- (**Leaving drivers.**) For any spatiotemporal state s, we set up a directed edge $e_{s,O}$ going from s to O with $\ell(e_{s,O}) = 0$ lower bound, $u(e_{s,O}) = +\infty$ upper bound, and the constant-zero reward function $r(\cdot; e_{s,O}) \equiv 0$. The flow $f(e_{s,O})$ represents the number of drivers to leave the system at the state s.

- (**Driving without a rider.**) For any admissible arc $(s, s') \in Q$, we set up a directed edge $e^{(\mathrm{o})}_{s,s'}$ going from s to s' with $\ell(e^{(\mathrm{o})}_{s,s'}) = 0$ lower bound, $u(e^{(\mathrm{o})}_{s,s'}) = +\infty$ upper bound. The flow $f = f(e^{(\mathrm{o})}_{s,s'})$ represents the number of drivers to drive through the arc (s, s') without carrying a rider. Therefore, we set up the linear reward function $r(f; e^{(\mathrm{o})}_{s,s'}) = -c(s, s') \cdot f$.

- (**Driving with a rider.**) For any admissible arc $(s, s') \in Q$, we set up a directed edge $e^{(\mathrm{w})}_{s,s'}$ going from s to s' with $\ell(e^{(\mathrm{w})}_{s,s'}) = 0$ lower bound, $u(e^{(\mathrm{w})}_{s,s'}) = o(s, s')$ upper bound. The flow $f = f(e^{(\mathrm{w})}_{s,s'})$ represents the number of drivers to drive through the arc (s, s') with a rider. Therefore, we define the non-linear reward function $r(f; e^{(\mathrm{w})}_{s,s'}) = [v_f(s, s') - c(s, s')] \cdot f$.

Given (V_0, E_0), the *maximum revenue car dispatching* problem in the deterministic setting is equivalent to finding the optimal solution to NLWC on the directed graph (V_0, E_0). Formally, we directly have the proposition below.

Proposition 3.1. *Let f^* be the optimal solution to NLWC on the directed graph (V_0, E_0). To achieve the maximum revenue in the car dispatching task, the platform may direct the*

drivers to drive with/without carrying a rider or leave the platform based on the flow value on the corresponding sets of edges. The total weight of f^* is the maximum revenue the platform may collect.

Proposition 3.1 also enables us to design the *routing plan* for each individual driver based on the NLWC solution. Formally, a *route* $A = (a_1, a_2, \ldots, a_z)$ is a sequence of spatiotemporal arcs such that the ending state of each arc a_i is the same as the beginning state of the next arc a_{i+1} (for all $i \in \{1, 2, \ldots, z-1\}$). At each time step and for each driver q, the *routing plan* A_q is just a route which starts at the driver's current state.

While the general NLWC problem is computationally intractable, the maximum revenue car dispatching problem, as a special case of NLWC, is unfortunately not easier. Formally, we present the following negative result for the maximum revenue car dispatching problem. The proof of Theorem 3.2 is deferred to Appendix A.1.1. Note that since maximum revenue car dispatching is a special case, we may not directly use the NP-Hardness proof of NLWC, and have to design a new hardness instance instead.

Theorem 3.2. *The maximum revenue car dispatching problem, even in the deterministic setting, is NP-hard.*

On the positive side, we propose a natural *regularity condition* in Definition 2. We will show that when the condition is satisfied, the maximum revenue car dispatching problem can be solved in polynomial time.

Definition 2 (Regularity)**.** *We say that a maximum revenue car dispatching problem instance satisfies the* regularity condition *if for each admissible spatiotemporal arc (s, s'), and each $k \in \{1, 2, \ldots, o(s, s')\}$, the sequence $v'_k(s, s')$ is monotonically non-increasing with k, where we define*

$$v'_k(s, s') := \begin{cases} v_1(s, s') & (k = 1) \\ k \cdot v_k(s, s') - (k-1)v_{k-1}(s, s') & (k \geq 2) \end{cases}.$$

In the definition, $v'_k(s, s')$ can be interpreted as the *marginal reward* of accepting the k-th highest price order on arc (s, s'). The regularity condition then requires that the marginal

reward sequence is not increasing with the increasing number of accepted orders on any arc, which is a standard assumption in economics literature (see, e.g., [22, 23, 24]). Indeed, in our empirical evaluation, we verify that the regularity condition holds in the real-world data.

We now present our edge decomposition algorithm (details in Algorithm 1) for the maximum revenue car dispatching problem. At a higher level, Algorithm 1 first manages to decompose each non-linear directed edge in (V_0, E_0) to a family of edges with linear costs and creates a minimum linear-cost circulation problem instance $(V_0, \tilde{E}, \tilde{\ell}, \tilde{u}, -\tilde{w})$, then invokes the existing polynomial-time time algorithm for the minimum linear-cost circulation problem, and finally aggregates the flows in each family to construct the optimal solution to the original problem.

Algorithm 1 The Edge Decomposition Algorithm

1: Construct the NLWC instance (V_0, E_0, ℓ, u, r);
2: $E_1 \leftarrow \{e^{(\mathrm{w})}_{s,s'} \in E_0\}$, $E_2 \leftarrow E_0 - E_1$; $\tilde{E} \leftarrow \emptyset$;
3: **for** $e^{(\mathrm{w})}_{s,s'} \in E_1$ **do**
4: **for** $i \in \{1, 2, \ldots, o(s, s')\}$ **do**
5: $\tilde{E} \leftarrow \tilde{E} \cup e^{(\mathrm{w},i)}_{s,s'}$; $(\tilde{\ell}(e^{(\mathrm{w},i)}_{s,s'}), \tilde{u}(e^{(\mathrm{w},i)}_{s,s'})) \leftarrow (0, 1)$;
6: $w(e^{(\mathrm{w},i)}_{s,s'}) \leftarrow r(i; e^{(\mathrm{w})}_{s,s'}) - r(i-1; e^{(\mathrm{w})}_{s,s'})$
7: **for** $e \in E_2$ **do**
8: $\tilde{E} \leftarrow \tilde{E} \cup e$; $\left(\tilde{\ell}(e), \tilde{u}(e)\right) \leftarrow (\ell(e), u(e))$;
9: $\tilde{w}(e) \leftarrow \begin{cases} -c(s, s') & (\text{if both } s, s' \in S) \\ 0 & (\text{otherwise}) \end{cases}$;
10: Invoke the polynomial-time algorithm [21] to compute the minimum cost circulation of $(V_0, \tilde{E}, \tilde{\ell}, \tilde{u}, -\tilde{w})$ where $-\tilde{w}$ is the coefficient function of the linear costs, denote the optimal flow by $\tilde{f}$;
11: **for** $e \in E_0$ **do**
12: **if** $e = e^{(\mathrm{w})}_{s,s'} \in E_1$ **then** $f(e) \leftarrow \sum_i \tilde{f}(e^{(\mathrm{w},i)}_{s,s'})$;
13: **else** $f(e) \leftarrow \tilde{f}(e)$;
14: **return** f;

In Algorithm 1, the edge set E_1 denotes the edges corresponding to "driving with a rider" and E_2 the rest of the edges. We also observe that the only non-linear edges are the ones to drive with a rider (in E_1), while the rest edges (in E_2) already have linear costs. For the edges in E_1, the algorithm decomposes them from Line 3 to Line 6: since the flow on each edge in E_1 represents the amount of the rider orders accepted along the corresponding spatiotemporal arc, the algorithm assigns each decomposed edge with unitary capacity, and

the corresponding flow represents an additional order to be accepted along the arc, and naturally the weight function is defined based on the marginal reward function $v'_k(\cdot,\cdot)$. Also note that the algorithm always returns an integral flow because of the integrality property of the minimum linear-cost circulation problem. Regarding the theoretical guarantee of Algorithm 1, we prove the following theorem:

Theorem 3.3. *Algorithm 1 runs in polynomial time, and when the regularity condition is met, the returned flow f achieves the maximum revenue of the car dispatching problem on the directed graph (V_0, E_0).*

Proof. We only need to prove that in the NLWC problem with regularity, each non-linear edge in E_1 with finite capacity can be substituted by a finite number of linear edges.

Consider an edge $e = e^{(\mathrm{w})}_{s,s'} \in E_1$, then $\ell(e) = 0$. Then, for each $i \in N$ s.t. $1 \le i \le u(e)$, we add to $\tilde{E}$ an linear edge $e_i(s, s', 0, 1, w(i) - w(i-1))$. Since $r(i; e^{(\mathrm{w})}_{s,s'}) - r(i-1; e^{(\mathrm{w})}_{s,s'})$ decreases with i, when we should put t amount of flow from s, s' in G_1, the optimal plan is to saturate edges $e^{(\mathrm{w},1)}_{s,s'}, e^{(\mathrm{w},2)}_{s,s'}, \cdots, e^{(\mathrm{w},t)}_{s,s'}$, with total reward $r(t; e^{(\mathrm{w})}_{s,s'})$, identical to the NLWC model.

Therefore, we realize the same edge-reward function as the NLWC model with a minimum cost circulation model. While the Maximum Revenue Car Dispatching problem needs integer solutions, from the total unimodularity property of the minimum cost circulation problem, it is guaranteed that our algorithm outputs an integer basic solution. Therefore, we can indeed solve regular Maximum Revenue Car Dispatching via the minimum cost circulation problem. **Q.E.D.**

We also remark that even when in the general scenario (without the regularity condition), a simple variation of Algorithm 1 also serves as a good approximation to the optimal solution. It virtually approximates the edge reward function by its concave envelope to "iron" it to a concave function [25]. Please refer to Appendix A.2 for details.

3.4 Phase 2 of the Deterministic Setting: Fair Reward Re-allocation to Drivers

Recall that in Phase 1 we have found the maximum revenue that can be achieved by any dispatching plan in the deterministic setting. Along the way, we have also figured out how many drivers are needed for a spatiotemporal arc $(s, s') \in Q$ with a rider (namely $f(e^{(\mathrm{w})}_{s,s'})$) and without carrying a rider (namely $f(e^{(\mathrm{o})}_{s,s'})$). For convenience, we define $F(s, s') := f(e^{(\mathrm{w})}_{s,s'}) + f(e^{(\mathrm{o})}_{s,s'})$ to be the total number of drivers we plan to dispatch along the arc (s, s'). In this section, we develop methods to figure out the fair payment scheme $y : S \times S \to [0, +\infty)$ for driving along each spatiotemporal arc to ensure that the drivers are well incentivized to cooperate with the platform and execute the optimal-revenue dispatching plan. Formally, we define the fairness condition as follows.

Definition 3 (Fair re-allocation)**.** *A re-allocation scheme is* fair *if and only if following conditions are satisfied:*

- *Budget-balance. Let $\mathscr{I}$ be the total income collected from the riders. This should also be the exact amount to be distributed to the drivers.*[2] *Formally, it is required that $\sum_{(s,s')\in Q} y(s, s') \cdot F(s, s') = \mathscr{I}$.*
- *Individual rationality. For each arc driven, the payment should be at least the cost; i.e., for each $(s, s') \in Q$ so that $F(s, s') > 0$, it is required that $y(s, s') \geq c(s, s')$.*
- *Subgame-perfectness. This is formally defined soon in Definition 4 which, together with the individual rationality condition, makes sure that the drivers do not have the incentive to refuse and deviate from the dispatching plan.*
- *Envy-freeness. This is formally defined in Definition 5 which makes sure that the drivers do not complain that the dispatching plan is more favorable to others than themselves.*

Note that we need to define subgame-perfectness and envy-freeness in details. Before doing this, we need to introduce a few new notations and definitions.

[2]We omit the amount that the platform would like to keep for profit, which can be easily added to the constraint w.l.o.g.

We will model the drivers' behavior as an *extensive game* [26], where, at each state, each driver has the freedom to choose any route starting from the current state. At any time step, let A_q denote the routing plan given by the platform for the driver q, let $\mathscr{A} := \{A_1, A_2, \dots\}$ denote the set of routing plans for all drivers, and let $A_{-q} := \mathscr{A} \setminus \{A_q\}$. For each driver q, let $u_q(\mathscr{A})$ denote the utility (i.e., net profit) of driver q if all drivers follow the routing plan $\mathscr{A}$. In particular, we have that $u_q(\mathscr{A}) = \sum_{(s,s')\in A_q}(y(s,s') - c(s,s'))$.

The subgame-perfectness condition requires that given reward re-allocation scheme and the set of routing plans for all drivers by the platform, any driver q does not have the incentive to deviate from the routing plan given to him/her. Formally, we make the following definition.

Definition 4 (Subgame-perfectness)**.** *A reward re-allocation scheme is* subgame-perfect *if at any time step, let $\mathscr{A} := \{A_1, A_2, \dots\}$ be the routing plans decided by the platform, and for any driver q, and for each route A'_q sharing the same starting state as A_q, it holds that $u_q(A_q, A_{-q}) \geq u_q(A'_q, A_{-q})$.*

Note that in game theory, a subgame-perfect Nash equilibrium in a extensive game is a strategy profile for the agents such that at any point of the game, the agents' strategies form a Nash equilibrium for the continuation of the game. Definition 4 requires that reward re-allocation scheme makes sure that the routing plan given by Proposition 3.1 is a subgame-perfect Nash equilibrium.

We would also like to make sure that each driver does not feel comparably inferior than others at the same state. Formally, we define the envy-freeness condition as follows.

Definition 5 (Envy-freeness)**.** *A reward re-allocation scheme is* envy-free *if at any time step, let $\mathscr{A} := \{A_1, A_2, \dots\}$ be the routing plans decided by the platform, and for any two drivers q and q' staying at the same state, it holds that $u_q(\mathscr{A}) = u_{q'}(\mathscr{A})$.*

Now we have completed the formal definition of a fair re-allocation scheme. The following lemma provides an elegant characterization of all fair re-allocation schemes and enables us to find such schemes only among the potential-based re-allocation algorithms. The proof of Lemma 3.1 can be found in Appendix A.1.2.

Lemma 3.1. *Given a routing plan $\mathscr{A}$, a reward re-allocation is fair if and only if there exists a corresponding* potential function $P : S \to \mathbb{R}^{\geq 0}$ *such that*

1. *For any $s \in S$ where $\mathscr{A}$ directs at least one driver to leave at state s (we call such states the* terminal states*), it holds that $P(s) = 0$.*
2. *$\forall (s, s') \in Q$, $y(s, s') - c(s, s') \leq P(s) - P(s')$.*
3. *$\forall (s, s') \in Q : F(s, s') > 0$, $y(s, s') - c(s, s') = P(s) - P(s') \geq 0$.*
4. *$\sum_{s \in S} P(s)(\deg_i(s) - \deg_o(s)) = \sum_{(s,s') \in Q} F(s, s')(p(s, s') - c(s, s'))$, where $\deg_i(s)$ and $\deg_o(s)$ are the number of drivers to enter and leave the platform at the state s respectively.*

Leveraging the power of Lemma 3.1, we are able to prove the following theorem stating that a fair reward re-allocation scheme always exists in all non-degenerating scenarios (i.e., the total revenue is non-negative and at least one driver starts from a non-terminal state).

Theorem 3.4. *Let $S_\# \subseteq S$ denote the set of terminal states. If there exist $s_1 \in S \setminus S_\#$ and $s_2 \in S$ such that $F(s_1, s_2) > 0$ and $\mathscr{I} \geq \sum_{(s,s') \in Q} F(s, s') \cdot c(s, s')$ (recall $\mathscr{I}$ is the total income collected from the riders), then there exists a fair reward allocation plan.*

Proof. We define a directed graph G' on vertex set $V(G') = (S - S_\#) \cup \{t\}$, in which all states in $S_\#$ are contracted in a single vertex t. For each order from $s \notin S_\#$ to s' we add a directed edge (s, s') with length 1 if $s' \notin S_\#$, or (s, t) with length 1 if $s' \in S_\#$, and for each possible cruise arc from s to s' we add an edge with length 0.

As all arcs advance in time, the graph is a directed acyclic graph (DAG). Therefore, we can define $\tilde{P}(s)$ as the maximum distance of all paths from s to t, or 0 if $s \in S_\#$. Then we let $R_* = \{(s, s') \in Q : f(s, s') > 0\}$, define $\mu(s, s') = \tilde{P}(s) - \tilde{P}(s')$, and then we allocate the revenue proportional to μ, i.e. let

$$P(s) = \tilde{P}(s) \cdot \frac{\mathscr{I} - \sum_{(s,s') \in Q} F(s, s')c(s, s')}{\sum_{(s,s') \in Q} F(s, s')\mu(s, s')}. \tag{3.1}$$

Because of the assumption that $\mathscr{I} \geq \sum_{(s,s')\in Q} F(s,s')c(s,s')$, we are ensured that $r(s,s') - c(s,s')$ is proportional to $\mu(s,s')$ with a non-negative ratio. We can see all constraints are satisfied. **Q.E.D.**

When the fair re-allocation scheme is not unique, we solve the quadratic program in Figure 3.1 to find the scheme to minimize the total squared distortion between the price paid by the rider and the reward allocated to the driver among all trips. In this way, we try the best to let the reward re-allocation reasonably reflects the real income generated by driving through each arc. It is straightforward to see that the constraints (3.3,3.4,3.5,3.6,3.7) in the quadratic program implement the conditions stated in Lemma 3.1.

$$\text{Minimize} \sum_{F(s,s')>0} F(s,s')(p(s,s') - y(s,s'))^2$$

$$\begin{aligned}
\text{Subject to} \quad & P(s) \geq 0, \quad \forall s \in S && (3.2)\\
& y(s,s') = P(s) - P(s') + c(s,s'), \quad \forall F(s,s') > 0 && (3.3)\\
& y(s,s') \leq P(s) - P(s') + c(s,s'), \quad \forall (s,s') \in Q && (3.4)\\
& P(s) = 0, \quad \forall s \in S_{\#} && (3.5)\\
& y(s,s') \geq c(s,s'), \quad \forall F(s,s') > 0 && (3.6)\\
& \sum_{s\in S} P(s)(\deg_i(s) - \deg_o(s)) \\
& \qquad = \sum_{(s,s')\in Q} F(s,s')(p(s,s') - c(s,s')) && (3.7)
\end{aligned}$$

Figure 3.1: Quad. Prog. with decision variables $\{P(s)\}_{s\in S}$

3.5 The Stochastic-Demand Setting

In the previous sections, we studied the optimal car dispatching and reward allocation task assuming the access to the full list R of latent orders, which is not realistic in practice. In this section, we assume that R is drawn from an *unknown* distribution $\{\mathscr{D}(s,s')\}$ and address the problem with techniques combining both learning and optimization. To achieve this

goal, we study the optimal car dispatching and reward allocation task with the distribution $\{\mathscr{D}(s,s')\}$ *known*. We will refer to this task as the *stochastic-demand setting*.

Our algorithm for the stochastic-demand setting is a natural extension of that for the deterministic setting presented in the previous sections. Below we describe the adaptation we make for each phase in the deterministic setting. We will also introduce a special parametric demand distribution (Gaussian-Poisson distribution) for the learning algorithm in Appendix A.4.1.

Phase 1: Revenue Optimization. For each arc (s,s'), we denote $x_{s,s'}$ as the random variable for the number of latent orders, $\{v_t\}_{t\in[x_{s,s'}]}$ as the random variables for the valuations, and denote $\mathscr{D}(s,s')$ as the distribution of $(x_{s,s'},\{v_t\}_{t\in[x_{s,s'}]})$, with the assumption that each v_t are *i.i.d.* variables.

For each arc (s,s'), if we fix the price to be p and plan to dispatch n drivers to the arc, the number of the fulfilled latent orders will be the smaller value of n and the number of orders of valuations at least p. Therefore, given $\mathscr{D}(s,s')$, we may compute the following quantities:

- The probability mass function $\mathscr{P}(i;s,s',p):\mathbb{N}\to\mathbb{R}$ for the number of qualified orders (orders with valuation at least p): $\mathscr{P}(i;s,s',p)=\sum_{j=0}^{\infty} b(i,j;\Pr[v_t\geq p])\Pr[x_{s,s'}=j]$, where $b(k,n;P)=\binom{n}{k}P^k(1-P)^{n-k}$ computes the binomial distribution.
- Let $\tilde{u}(n;s,s',p)$ be the number of the fulfilled latent orders; its expectation: $\mathscr{E}[\tilde{u}(n;s,s',p)]=\sum_{i=0}^{\infty}\mathscr{P}(i;s,s',p)\min\{i,n\}$.
- The expected revenue on (s,s') at price p and n drivers: $\mathscr{R}(n,p;s,s')=p\cdot\mathscr{E}[\tilde{u}(n;s,s',p)]-c(s,s')\cdot n$.

The following definition states the optimization problem we have to solve in order to maximize the revenue in car dispatching in the stochastic-demand setting.

Definition 6. *Given $\mathscr{D}(s,s')$ for all arcs (s,s'), the* Stochastic Maximum Revenue Car Dispatching *problem is to find the optimal solution to the NLWC problem on the directed graph (V_0,E_0), where (V_0,E_0) is constructed in a similar way as described above Proposition 3.1,*

and the only difference is that for the edges corresponding to driving with a rider, we set the corresponding reward function $r(n; e^{(\mathrm{w})}_{s,s'}) = \max_{p\in\mathbb{R}^{\geq 0}}\{\mathscr{R}(n, p; s, s')\}$.

In Definition 6, $r(n, e^{(w)}_{s,s'})$ is re-defined so as to equal the maximum possible (over all candidate prices) expected revenue generated by dispatching n drivers to the arc (s, s'). Therefore, the optimal solution to the stochastic maximum revenue car dispatching problem is the maximum possible expected revenue achieved by any dispatching plan.

Note that in Definition 6, the only quantity that specifically depends on the form of the demand distribution is the non-linear reward function on the edges $e^{(\mathrm{w})}_{s,s'}$.

Phase 2: Fair Re-allocation. After solving the NLWC problem on (V_0, E_0), we obtain the number of drivers to dispatch and the price for each arc (s, s'). With this information, we may invoke the quadratic program in Figure 3.1 to find out the potential-based reward re-allocation scheme for the drivers. We are able to show the following the fairness guarantees in the stochastic-demand setting, while the detailed proof is omitted since it is almost the same as the proof in Phase 2 of the deterministic setting.

Theorem 3.5. *In the stochastic-demand setting, the potential-based reward re-allocation scheme obtained by the QP in Figure 3.1 satisfies the fairness conditions in Definition 3, except for that the budget-balance condition is changed to the following expectation version.*

- Expected-*budget-balance. The expected income collected from the riders should equal to the amount to be distributed to the drivers.*[3] *Formally, it is required that* $\sum_{(s,s')\in Q} y(s, s') \cdot F(s, s') = \mathscr{E}[\mathscr{I}]$, *where* $\mathscr{I}$ *is the collected income.*

Online Learning. We use a Thompson sampling-based algorithm to learn the demand distributions from riders' responses to given prices. The details are deferred to Appendix A.5.

3.6 Experimental Evaluation

Due to space constraints, we defer many of the experiments to Appendix A.7. For example, we empirically verify the regularity of Gaussian-Poisson distributions in Appendix A.7.1,

[3]Similarly, here we also omit the amount that the platform would like to keep for profit.

and evaluate the online learning algorithm in Appendix A.7.2; we also show an illustrative example of our fair re-allocation algorithm on the real-world dataset in Appendix A.7.3.

Experiments are run on an Intel i7-8750H, 24GB RAM computer with MATLAB 2021b.

3.6.1 Model Setting

We now evaluate our algorithm by simulated experiments on the DiDi Chuxing public dataset [27] collected from the real-world ridesharing in Chengdu, China. For one day, we extract all orders and driver initial positions and discretize the locations into $10 \times 10 = 100$ squares with dimension 2km $\times$ 2km, and divide the time interval between 8am and 1pm in a day into 20 slots, each of which spans 15 minutes. Therefore, there are 2000 spatio-temporal states in a day, and we use the reward column in the dataset as the rider's valuation for the trip. Finally, we assume the latent orders follow the Gaussian-Poisson distribution (Appendix A.4.1), and collect the data for 30 days and fit the numbers and valuations of orders in any arc into the Gaussian-Poisson distribution, as the true model parameters.

Robustness. To evaluate the generalization ability of our algorithm, we modify the following two key parameters in experiments: the number of drivers and the standard deviations of the riders' valuations. Here we report the experimental results showing that our algorithms still perform well under these different experimental environments.

In Table 3.1, we modify the number of drivers. In the 50% drivers setting we remove each driver with 50% independent probability and in the 200% drivers setting we duplicate every driver. In Table 3.2, we modify the standard deviations of the riders' valuations by 0.5 and 1.5 times respectively.

3.6.2 Revenue Evaluation

Given the true model parameters, we invoke the algorithms described in Section 3.5 to find out the offline (model parameters known) optimal revenue of the Stochastic Maximum Revenue Car Dispatching problem. We refer to this value as the *two-phase value* (2P). For comparison, we introduce the baseline *distance-based fix-price algorithm* (FP) where the price

for each arc is proportional to the distance of the trip with a globally fixed (but tuned) ratio, and the dispatching is done via the same network-flow-based planning algorithm.

3.6.3 Fairness Evaluation

To evaluate the fairness, we define the $A(s)$ as the *average net income* of all drivers initially at state s. For a driver $q \in Q$, we denote s_q as the initial state of q and u_q as the total net income of q. Then, we define the absolute unfairness $\Xi = \sqrt{\frac{\sum_{q \in Q}(u_q - A(s_q))^2}{|Q|}}$ and relative unfairness $\xi = \Xi / \frac{\sum_{q \in Q} u_q}{|Q|}$, which can be interpreted as the absolute and relative fluctuation of drivers' net incomes from given initial states. We have proven that our two-phased algorithm guarantees *zero* unfairness, and evaluate the unfairness of baseline pricing algorithms. To show the contribution of re-allocation, we refer to the result of only Phase 1 as P1.

3.6.4 Results

We report the revenue (Rev) and relative unfairness (Unf) of different settings in following tables.

#drivers	6655		13411		26822	
	Rev	Unf	Rev	Unf	Rev	Unf
2P	**6.82**	**0.000**	**9.32**	**0.000**	**11.17**	**0.000**
P1	**6.82**	0.114	**9.32**	0.172	**11.17**	0.243
FP	5.54	0.108	7.56	0.168	9.02	0.244

Table 3.1: Rev($\times 10^4$)/Unf with different numbers of drivers .

stddev	0.5σ		1.0σ		1.5σ	
	Rev	Unf	Rev	Unf	Rev	Unf
2P	**10.36**	**0.000**	**9.32**	**0.000**	**8.61**	**0.000**
P1	**10.36**	0.167	**9.32**	0.172	**8.61**	0.178
FP	7.90	0.162	7.56	0.168	7.25	0.172

Table 3.2: Rev($\times 10^4$)/Unf with modified standard deviations .

We see that our algorithm achieves higher revenue than the fixed-price algorithm, and our re-allocation phase eliminates the unfairness that would typically range from 10% to 25% of drivers' incomes, which increases with numbers of drivers.

Intuitively, a large number of drivers would tend to result in increased unfairness as they fulfill a large portion of latent orders with a wider spread of profits (analysis in Appendix A.7.4). Therefore, the re-allocation phase becomes essential for satisfaction of drivers especially in this scenario.

3.7 Computational Complexity Analysis

Let n, m, a be the number of states, latent orders and admissible arcs, respectively. Our Phase 1 essentially solves a linear program of size $O(m + a)$, which runs in $\tilde{O}((m + a)^{2.373})$ time [28]. Our Phase 2 solves a quadratic problem of $O(n)$ variables and $O(a)$ input size, which can be transformed into a semidefinite program that runs in $\tilde{O}(\sqrt{n}(an^2+a^{2.373}+n^{2.373}))$ time [29].

3.8 Conclusion

In this chapter, we present an algorithmic framework for car dispatching and pricing with both revenue and fairness guarantees. Empirical evaluation shows that our method performs better than the baseline alternatives in the real-world dataset. For future directions, it is interesting to prove the regularity of edge demand functions in Gaussian-Poisson distribution and explore the regularity property of other distributions, and mathematically prove the guarantees of our Thompson Sampling algorithm (e.g., its convergence property and finite-sample regret bound).

CHAPTER 4

COLLUSION-PROOF BLOCKCHAIN TRANSACTION FEE MECHANISM DESIGN

4.1 Introduction

The blockchain, as a new decentralized technology, is becoming an interesting research object for the Operations community (see, e.g., Davydiuk et al. [30], Iyengar et al. [31], Manzoor et al. [32], Whitaker and Kräussl [33] and references therein). Just like the emergence of the ridesharing topic ten years ago, the special structure in blockchain poses many unique challenges in auction theory, game theory, scheduling, and optimization; in turn, the blockchain technology also has applications that foster traditional aspects of operation research, e.g., newsvendors and supply chains [34]. Particularly, in the scope of game theory, Liu et al. [35] surveys a variety of its applications in blockchain systems.

Let us zoom in and briefly discuss the structure of a standard blockchain. A blockchain is essentially a linked list (or a chain) of blocks, where each block stores a number of transactions. There are two types of agents participating in a blockchain: *users* and *miners*. Users propose to put *transactions* on the chain, and miners pack transactions into a block and then send blocks to the chain. Once the block has been finalized on the chain, the miner will receive tokens (e.g., Bitcoin) as a reward. Generally, each block may contain multiple transactions, but only one miner claims ownership of the block and obtains the corresponding reward. An illustration of the generation process of each block is shown in Figure 4.1.

The blockchain stores the blocks sequentially in the time order. After a miner creates a new block, the block is appended to the chain via a reference to the latest existing block. For the efficiency of the block space, each block only contains a pre-specified limited number of transactions. In order to retrieve the status (e.g., balances of each user), we have to track

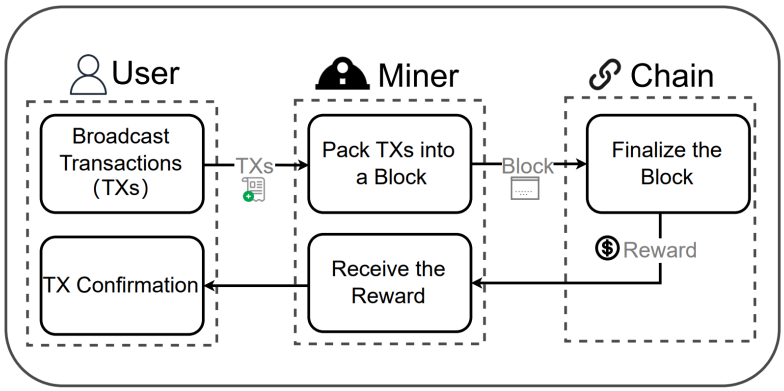

Figure 4.1: The role of different parties in generating one block in a blockchain.

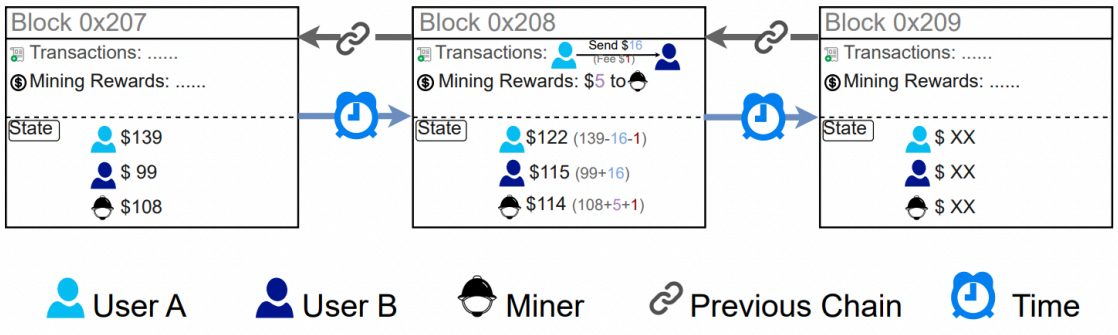

Figure 4.2: Illustration of the blockchain structure. Above the dashed lines are the blocks and their contents (transactions and rewards). The blocks are arranged from left to right in the time order, and each block is linked to the previous one on its left. Below the dashed lines are the states of the system – the amount of money owned by each party at the time.

from the beginning of the chain and go through all previous transactions to determine the current status at any block. We show the structure of the blockchain in Figure 4.2.

As each block only contains a limited number of transactions, the major bottleneck of limited-space design in the blockchain systems draws wide research interest in the field of mechanism design (e.g., Wang et al. [36]). With this bottleneck, users need to compete to win a transaction space in the block. Such competition can naturally be implemented via an auction. However, as we will explain later, the auction design in a blockchain exhibits unique challenges on how to charge the users properly and how to reward the miner. This problem is usually referred to as *"transaction fee mechanism (TFM) design"*, which has been modeled by a seminal paper of Roughgarden [37]. In more detail, to incentivize the miners to mine the block, the blockchain systems adopt economic mechanisms to pay miners via

cryptocurrency. Such payments usually consist of a mining reward, and an additional reward extracted from transaction fees paid by users named as the *miner revenue*. As users benefit from transactions being confirmed on the blockchain and miners need the incentivization, it is reasonable to charge transaction fees from users for confirmed transactions. As described by Roughgarden [37], the on-chain space is a scarce resource, so to facilitate the social efficiency of the system, we want to confirm transactions of high values. Therefore, many blockchains adopt bidding-confirmation transaction fee mechanisms (TFM) such as auctions.

However, due to the online and anonymous properties of blockchains, the design of mechanisms for blockchain systems faces a major concern of *credibility* [38]. Compared to traditional auctions, the miner has a wider strategy space (than a traditional auctioneer) to conduct dishonest activities, including injecting fake transactions, concealing users' bids, and colluding with users. Therefore, it is important to address these unique challenges raised in the blockchain setting and develop a desirable TFM that discourages all possible dishonest activities to make sure the whole on-chain economic system can operate correctly.

In a blockchain system, there are mainly three types of dishonest behaviors: *Untruthful Bids (*$\mathtt{UB}$*)*, *Fake Transactions (*$\mathtt{FT}$*)*, and *Transaction Deletion (*$\mathtt{TD}$*)*. The agents who might conduct these dishonest behaviors include *individual user (*$\mathtt{U}$*)*, *individual miner (*$\mathtt{M}$*)*, *miner colluding with c users (*$\mathtt{MU}^c$*)*, and *collusion among c users (*$\mathtt{U}^c$*)* with $c \geq 2$. We provide a table to summarize all types of deviations at the end of Section 4.1.2.

In our research, we prevent these types of dishonest behavior in a *mixed* way consisting of cryptographic and economic techniques. In the cryptographic part, we introduce a *commitment scheme* from [39] and adapt it into our proposed mechanism (as shown in Appendix B.1) that both essentially runs a sealed-bid auction that ensures fairness among users' information sets and restricts the miner's strategic space, also resolving the MEV (miner-extractable-value or maximal-extractable-value) issue, in which miners may gain additional revenue via strategically injecting, excluding or re-ordering the transactions [40], via pinning down the transaction orders before they are revealed. As shown in Appendix B.1, some types of dishonest behavior (particularly deviations by an individual miner) can be effectively prevented via cryptographic protocols. Furthermore, the anonymity property of the blockchain system also brings intrinsic difficulty to collusions among users. Therefore,

in the economic part, we mainly focus on the *prevention of* ***individual user's deviation*** *and* ***miner-user collusion***.

4.1.1 Research Question: How to Design Truthful and Collusion-Proof TFMs

To understand our results, we provide the necessary background on the truthfulness of users and miners. In the standard auction theory, a strong version of the truthfulness of users can be specified as the User-Dominant-Strategy-Incentive-Compatibility (U-DSIC), which means that any individual user will not benefit from deviation from truthful bidding even if she knows all bids of other users (as in Definition 9). In many practical scenarios, it would be difficult to satisfy such a strong notion of truthfulness. Instead, a weaker version of the truthfulness of users is studied in this paper, i.e., the *User-Bayesian-Nash-Incentive-Compatibility (U-BNIC)* (also known as Bayesian-Incentive-Compatibility (BIC) in some literature). In particular, U-BNIC means that when each user only knows the distribution of others' valuations, the game achieves a Bayesian Nash equilibrium when all users truthfully bid their valuations (see Definition 10). The truthfulness of the miner can be specified as the Miner-Incentive-Compatibility (MIC), which means that the miner will not benefit from untruthful behavior, e.g., injecting fake transactions or ignoring existing transactions. For the issue of collusion, the paper by Chung and Shi [41] formulates collusion-proofness as c-Side-Contract-Proof (c-SCP): when the miner colludes with at most c users by asking them to change their bids, the coalition cannot increase the total utility by deviations from truthfully bidding their valuations (as in Definition 12).

The key question is how to design TFMs to guarantee incentive compatibility and collusion-proof requirements, and the existing works on TFM design can be roughly classified into two families: *auction-like* mechanisms in which confirmed users' payments are dependent on the bids of the current block, and *posted-price* mechanisms in which their payments are completely based on statistics of previous blocks. The most intuitive form of the single-round auction mechanism is the *first-price auction* [6], in which the auctioneer collects all users' bids, and sells the item to the user who bids the highest at the price she bids. When we

generalize the first-price auction to the setting of multiple identical items, the auctioneer sells the items to users with the k-highest bids, charging the users what they bid. The Bitcoin blockchain essentially uses the multi-item first-price auction, but it is not truthful: users tend to bid lower than their valuations. A famous DSIC auction mechanism is the *(multi-item) second-price auction* [6], where the winners are also the k-highest bidders but their payment is the $(k+1)$-th highest bid. However, the second-price auction is susceptible to miner-user collusion as the miner may collude with the $(k+1)$-th highest bidder by asking her to raise her bid as long as it is still lower than the k-th highest bid. In such a way, the $(k+1)$-th highest bidder still gets the same utility 0 while the miner gets a higher revenue, increasing the total utility of the colluding party.

To solve the issue of collusion, the EIP-1559 mechanism [42] in Ethereum seeks to avoid collusion by adopting a dynamic *posted-price* mechanism, as long as the posted price is "well chosen" from historical demands, in expectation that there is no congestion on the block size — if there is nothing to bid, there is significantly less space for strategic bidding (i.e., `UB`).

However, in the TFM of EIP-1559, the miner typically gets no revenue from the transaction fees as the fees have to be "*burnt*" and removed from the blockchain to maintain collusion-proof properties (details discussed in Section 4.2.2). While EIP-1559 prevents collusion between users and miners, it is economically inefficient for miners as miners get no rewards from the transaction fees. Therefore, a natural question would be the following:

Can we design a TFM that satisfies both truthfulness
and collusion-proof conditions, and has a desirable miner revenue?

To answer this question, the paper by Chung and Shi [41] proves a negative result (Theorem 4.1) under the *User-Dominant-Strategy-Incentive-Compatibility (U-DSIC)*. In particular, even if we only consider the deviation set that contains individual-user untruthful bids and the miner collusion with one user, it is impossible to make positive revenue in the complete-information setting. To address this issue, the paper by Chung and Shi [41] introduces a so-called γ-strict utility, in which the parameter γ roughly depicts the probability that a currently unconfirmed transaction would be confirmed in future blocks (see details in

Section 4.2.1). However, such relaxation involving confirming a transaction in future blocks, brings an additional layer of difficulty. Indeed, the probability that a currently unconfirmed transaction gets confirmed in future blocks is not a universal constant, as unconfirmed transactions with higher bids are more likely to be confirmed in the future than those with lower bids. Thus, finding an accurate γ can be difficult, if not impossible. Therefore, we would like to ask the following question in this paper.

> *Are there other reasonable relaxations of the model and incentive compatibility specifications to circumvent the impossibility result?*

For this question, a series of related works (e.g., Gafni and Yaish [43], Shi et al. [44], Wu et al. [45]) have studied the problem of revenue optimization for blockchain transaction fee mechanisms with different models. Although it might be argued that the absence of miner revenue, or burning of *money*, may not directly undermine *global* social welfare [46], a non-zero additional reward for transaction confirmation besides the static block reward would indeed incentivize the miner to create the blocks honestly, in order to earn more money in addition to the basic block reward, especially for blockchains like Bitcoin in which the block rewards gradually go to zero. Besides, while the burning of tokens is not necessarily value-destroying, excessive burning may lead to a problem of *deflation*. While an important motivation of Bitcoin is to prevent inflation, from an economic perspective, deflation could be even worse than inflation in the long term because it may discourage spending and investment [47, 48], as people would prefer holding onto their tokens rather than using them in transactions. To maintain a thriving ecosystem in the blockchain community, we are indeed motivated to increase miner revenue via a decreased level of burning.

In our study, we address the above open question by relaxing the U-DSIC requirement to U-BNIC, which assumes the users only have information of distributions of other users' valuations instead of all their bids. This relaxation is reasonable because, in the distributed network of blockchain, it is impossible for a user to actually know all other users' bids, especially those who propose transactions after them but compete for the same block. Besides, a blockchain system, when combined with a commitment scheme (as discussed

in Appendix B.1), can essentially work as a *sealed-bid auction*[1] in which users' bids are not revealed until the bidding process finishes. Hence, users only have distributional knowledge about others, making the U-BNIC a natural requirement to prevent users' deviation. With the awareness that the MIC property is not the most necessary requirement (Remark 2 in Appendix A), the main goal of the paper is to design a TFM that satisfies U-BNIC and 1-SCP for bounded *i.i.d.* valuation distributions with a constant-factor approximation of the optimal revenue.

Interestingly, besides bypassing the negative result in blockchain TFMs via Bayesian mechanism design, on the other hand, our paper also bypasses a major negative result in the scope of Bayesian mechanism design via *burning*, a feature in blockchain systems. The papers by Gershkov et al. [7], Manelli and Vincent [49] show that in conventional auction settings in which all bidders' payments are rewarded to the auctioneer, the BNIC and DSIC conditions are equivalent. However, our research shows that with the incorporation of burning, the additional freedom to allow partial payment to be rewarded to the miner actually makes it possible to design essentially different mechanisms and gain increased revenue via the relaxation from DSIC to BNIC. (see discussion in Section 4.2.4)

In the rest of this paper, we assume the valuation distributions of users are *i.i.d.* and bounded. Without loss of generality, we assume the valuations are in the range of $[0, 1]$.

4.1.2 Summary of Contributions

Following the previous discussion, we summarize the main contribution of the paper below.

1. We propose an *auxiliary mechanism method* as our main tool to study U-BNIC TFMs by establishing connections between BNIC and DSIC auction mechanisms. In general, the method enables us to decompose any TFM into an *auxiliary* U-DSIC mechanism and a *variation term* and design them separately. This method can be a versatile tool in the design of BNIC mechanisms in the paradigm of relaxing the DSIC condition to

[1] In naïve implementations users may be able to see bids proposed before them but not after, leading to certain *unfairness* and MEV issues. With a commitment scheme (see Appendix B.1), we make it fair as all bids are sealed until the bidding process completes.

BNIC and utilizing the information asymmetry for higher revenue (or other desired properties, e.g. welfare). The auxiliary mechanism method will be described in Section 4.4.

2. For ease of illustrating our main idea, we first study the case where each block only contains one transaction (i.e., the block size is one). To design the TFM, we first construct a so-called *soft second-price mechanism* as our auxiliary mechanism, also referred to as *exponential mechanism*, based on the logit choice model. Via the auxiliary mechanism method, we design our mechanism that exploits the maximum extent of the information asymmetry between the miner and users to extract maximum revenue for the miner (Section 4.5.2).

3. We further extend our mechanism to general block size k, and prove that the constant-fraction approximation of optimal revenue still holds in this general case as long as the number of users n is larger than $\lambda_0 k$ for any fixed $\lambda_0 > \frac{e}{e-1}$ (see Section 4.6).

4. We further explore new properties of miner incentives in our TFM, for both size 1 and general block size k. Our results show that a reasonable level of miner deviations would not substantially benefit the miner, even if the miner knows all the bids. We also show a negative result that any TFM that is U-BNIC, 1-SCP and (strict) MIC cannot have a positive *expected* miner revenue (see Section 4.7.1). Furthermore, we establish a key stability result on the miner's revenue from our TFM over the distribution of users' bids. This result is important in practice, as the stability of revenue is critically important for miners (see Section 4.7.2).

As we described in the paragraph Section 4.1.1, we provide a classification of the dishonest behaviors among different possible agents (or groups of agents). Here, we summarize this classification in Table 4.1. Based on this classification, for different classes of deviations, we compare the strategy-proof properties and miner revenue of our proposed mechanism with different designs in the existing literature. The detailed comparison is provided in Table 4.2. In this paper (as well as [41]) we are particularly interested in preventing dishonest behaviors `U-UB`, `U-FT`, `MU`$^{\mathtt{c}}$`-UB`, `M-FT`, and `M-TD`. These dishonest behaviors are respectively prevented

	Untruthful Bids (UB)	Fake Transactions (FT)	Transaction Deletion (TD)
Individual User (U)	U-UB	U-FT	—
Individual Miner (M)	—	M-FT	M-TD
Miner-c-User Collusion (MU^c, $c \geq 1$)	MU^c-UB	MU^c-FT	MU^c-TD
c-User Collusion (U^c, $c \geq 2$)	U^c-UB	U^c-FT	—

Table 4.1: Classification of Dishonest Behaviors.

by the above-mentioned strategy-proof properties U-BNIC, U-SP, c-SCP, and MIC (where U-BNIC, U-SP and c-SCP will be formally defined in Section 4.3.3 and MIC will be formally defined in Section 4.7.1). In contrast, the U-UB dishonest behavior is dealt with U-DSIC in [41] under the complete-information setting; besides, a recent work by [45] considers a different multi-party-computation (MPC) model and develops another variant of posted-price mechanisms with comparable incentive and revenue guarantees as our work, but via different methodologies. We would discuss on the comparison in Section 4.2.2.

We finally note that this paper only considers the case that the miner colludes with one user (i.e., the case $c = 1$ in MU^c-UB) and it would be interesting to study for general c. We would like to leave it as a future work.

	Setting	U-UB, U-FT	MU^c-UB	M-FT, M-TD	Revenue/OPT
Our Mechanism	Bayesian game	✓	$c = 1$	Approx.*	$\Theta(1)$
Chung and Shi [41]	γ-strict utility	✓	✓	✓	$\approx O(\gamma^2/c)$
Wu et al. [45]	MPC model	✓	$c = 1$, Fixed**, Approx.	Approx.	$\Theta(1)$
Bitcoin	N/A	×	×	✓	No analysis
EIP-1559	Deterministic	Approx.***	✓	✓	≈ 0

Table 4.2: Comparison of different TFMs. * See detailed definition in Section 4.7.1. ** Their 1-SCP notion is substantially weaker as they only allow the miner to collude with a *fixed* user. *** This property is guaranteed only when the "posted price" is well set to prevent congestion (see the paper of Roughgarden [42].)

In summary, the multi-item first-price auction mechanism adopted by Bitcoin has bad strategy-proof properties, although it expects to have good miner revenue because it charges and awards transaction fees in a "greedy" way. The EIP-1559 mechanism [42] has almost the best strategy-proof properties but has zero miner revenue. The mechanism proposed by Chung and Shi [41] can prevent general MU^c-UB, but it only has good miner revenue for small c and large γ (meaning that every transaction has a high probability to be eventually confirmed in the future even if the bid is low, which is different from the common practice

in the blockchain). Our mechanism uses the Bayesian setting and has decent strategy-proof properties while achieving a constant-fraction approximation of the optimal miner revenue.

We list the meaning of abbreviations appearing in our paper in Table 4.3.

Abbreviation	Meaning
TFM	Transaction Fee Mechanism
(U-)DSIC	(User) Dominant Strategy Incentive Compatibility
(U-)BNIC	(User) Bayesian Nash Incentive Compatibility
U-SP	User Sybil Proofness
c-SCP	c-Side Contract Proofness
MIC	Miner Incentive Compatibility
OCA	Off-Chain Agreement
BF	Budget Feasibility
NFL	No Free Lunch
UIR	User Individual Rationality
MIR	Miner Individual Rationality
LP	Linear Programming
MPC	Multi-Party Computation
MEV	Miner/Maximal Extractable Value

Table 4.3: List of abbreviations.

4.2 Related Work

4.2.1 Auction-Like TFM Design in Literature

Since the main motivation of TFM design is to allocate the scarce block space to users, an intuitive idea is to design auction-like TFMs. While it is common to assume that the auctioneer in a traditional auction is *trusted*, it is not true in blockchain systems. To address this new challenge, the papers of Roughgarden [37] and Chung and Shi [41] split the incentive compatibility into two parts: User Incentive Compatibility (UIC) and Miner Incentive Compatibility (MIC), and consider the *complete-information setting* in which users have complete information of others' bids. Essentially, their papers define the term UIC equivalent to {`U-UB`, `U-FT`}-proofness, MIC equivalent to {`M-FT`, `M-TD`}-proofness (see Table 4.1). Furthermore, the paper by Chung and Shi [41] specifies the notion of c-SCP,

which is equivalent to {$\texttt{MU}^c$-UB}-proofness. Essentially, Chung and Shi [41] show a seminal impossibility result as follows:

Theorem 4.1 (Chung and Shi [41])**.** *Any TFM which is {*U-UB*,* $\texttt{MU}^1$-UB*}-proof in the complete-information (a.k.a. deterministic) setting has zero miner revenue.*

Note that the original theorem of Chung and Shi [41] states that "any TFM which satisfies UIC ({U-UB, U-FT}-proof) and 1-SCP ({$\texttt{MU}^1$-UB}-proof) has zero miner revenue." However in their proof, they only consider the deviations of U-UB and $\texttt{MU}^1$-UB but not U-FT. So as we consider the deviations in a more refined way, they actually prove this slightly stronger impossibility result.

To overcome the issue of zero miner revenue, Chung and Shi [41] introduce the "γ-strict utility" to make unconfirmed over-bidder still pay a γ fraction of the worst-case cost. In particular, if a bidder i has valuation v_i and her bid $b_i > v_i$, even if the transaction is not confirmed, she gets a utility of $-\gamma(b_i - v_i)$. Thus, if the confirmation probability is $a_i(b_i, \mathbf{b}_{-i})$ ($\mathbf{b}_{-i}$ denotes bids of all other users than i) and the bidder pays $p_i(b_i, \mathbf{b}_{-i})$ if the transaction gets confirmed, the utility of the bidder takes the following form:

$$u_i^{(\gamma)}(b_i, \mathbf{b}_{-i}; v_i) = a_i(b_i, \mathbf{b}_{-i})(v_i - p_i(b_i, \mathbf{b}_{-i})) - \gamma(1 - a_i(b_i, \mathbf{b}_{-i})) \max\{b_i - v_i, 0\}.$$

This relaxed utility function is justified by Chung and Shi [41] by considering the bidding process of more than one block in a blockchain: even if an overbidding transaction is not confirmed in the current block, the authors assume that the bid could still be collected and confirmed into future blocks, and in the worst case, the over-bidder would have to pay their full bid and get a utility of $-(b_i - v_i)$. In this setting, the authors have further developed a *burning second price* TFM that satisfies U-DSIC, MIC and c-SCP in the notion of γ-strict utility.

The γ-strict utility that considers the multi-block setting is critically sensitive to the parameter γ, but in practice, it is difficult to determine the value of γ due to the unpredictable nature of future blocks. On the other hand, as users cannot see others' bids in a blockchain system, the requirement of a complete-information setting is too strong in practice as compared to the Bayesian setting. In this perspective, our research focuses on the single-

block setting in which a proposed transaction is only valid for the current block, but we assume that each user only knows the distributions of other users' valuations. Based on the distributional information, we consider a different relaxation and develop our mechanism in the Bayesian game setting.

Besides strict notions of incentive compatibility, previous research on transaction fee design also considers nearly-incentive-compatibility properties. Yao [50] shows that the *monopolistic price* mechanism proposed by Lavi et al. [51] is nearly incentive compatible, i.e., strategic behavior can only gain a small advantage in utility. On the other hand, there is another parallel paradigm of collusion-proofness named as OCA-proofness, as proposed by Roughgarden [37], which only considers collusion of *confirmed* users instead of all users and has different properties. The detailed difference has been discussed by Gafni and Yaish [43] and Chung et al. [52].

Collusion-proofness notions: OCA-proofness and SCP. In a recent work [52], the authors study the relations between the two notions. As argued by Chung et al. [52], the OCA-proofness notion conceptually depicts the property that the colluding parties cannot "steal" from the protocol to gain more utility, and SCP means they can neither "steal" from the protocol nor from other users. It has also been shown in [52] that OCA-proofness is implied by c-SCP for any c, i.e., ∞-SCP is a stronger notion than OCA-proofness. Nevertheless, Chung et al. [52] also show that 1-SCP and OCA-proofness are incomparable notions.

From an economic perspective, it is not necessarily detrimental if colluding parties "steal" from the protocol to gain more utility. If offchain payments ensure that every agent experiences a weak increase in utility while maintaining protocol functionality, this collusion can actually represent a *Pareto improvement* [53]. While a Pareto improvement appears to be a desirable improvement of the ecosystem that does not need to be prevented, in a non-OCA-proof mechanism, the existence of Pareto improvements indicates that the mechanism is not economically efficient. Therefore, while the concept of SCP aligns with the principle of strategy-proofness, OCA-proofness is more relevant to the idea of *Pareto optimality*. Considering that "stealing from others" is clearly a dishonest behavior we aim to prevent, the study of SCP remains crucial.

4.2.2 EIP-1559 and Posted-Price TFM Design in Literature

Due to the anonymity of blockchain systems, blockchain TFMs are subject to a wider scope of dishonest behavior than traditional auctions, e.g., collusions and fake identities. While auction-like mechanisms can balance the supplies and demands as the prices are decided by the users' bids, the parties indeed have access to more strategies to manipulate the prices and gain advantages via dishonest bidding. In response to this challenge, there are another line of studies that replaces auctions in TFM design with a widely studied toolbox of *(dynamic) optimal pricing*, in which the "posted" prices are not decided by users of the current block, but from the statistics of previous blocks [42, 45, 54]. With the purpose of dynamically adjusting price based on supplies and demands, auction mechanisms and dynamic posted-price mechanisms are indeed solutions with different paradigms that both have the potential to be utilized in TFM design, as discussed in the studies of Hammond [55], Bubeck et al. [56], and so on.

Particularly, the EIP-1559 TFM, which is currently adopted in Ethereum, is essentially designed to be a posted-price mechanism that effectively prevents dishonest behavior, with a backup component of an auction-like mechanism in case the posted price is (unexpectedly) too low to prevent congestion. The EIP-1559 TFM works as follows:

1. The blockchain system adaptively decides on a *base fee* for the current block.

2. Each user proposes a transaction, paying the base fee and a voluntary *tip* if the transaction gets confirmed.

3. The (winning) miner confirms the transactions proposed by the users. It is expected that the number of transactions usually does not exceed the block size.

4. The miner gets a pre-defined *block reward* as well as all the tips. The base fees are *burned* and removed from the blockchain system.

In the ecosystem of EIP-1559, when there is no congestion, the miner does not need to consider the block size constraint and can confirm all the proposed transactions, so the users would pay very small tips and it is enough to incentivize the miner to confirm

their transaction. At "exception" scenarios when there is congestion, the miner would be incentivized to confirm transactions with the highest tips, and all the tips go to the miner. Hence, the EIP-1559 mechanism can be modeled as follows:

- If there is no congestion, EIP-1559 is essentially a posted-price mechanism with the posted price equal to the base fee; all payments are burnt and the miner gets no revenue from the transaction fees.
- If there is congestion, EIP-1559 shifts to a multi-item first-price (aka. pay-as-bid) auction. A fix amount of tokens (*base fee* $\cdot$ *block size*) are burnt and the remaining tokens go to the miner.

Due to the zero-revenue disadvantage of EIP-1559 and the existing impossibility results, researchers also attempt to avoid this negative aspect with alternative modeling and reasonable relaxations that apply to the blockchain environment. In consideration of the cryptographic nature of blockchain systems, Shi et al. [44] introduce a multi-party-computation (MPC) model that achieves ϵ-approximate incentive properties with a positive miner revenue scaling with $\Theta(\sqrt{\epsilon})$. Following the MPC model, Wu et al. [45] developed an LP-based posted-price mechanism that achieves U-DSIC, approximate Bayesian MIC and approximate Bayesian 1-SCP with positive miner revenue. The collusion-proof properties in the MPC-based models differ from ours as follows: in our work, we assume the miner to have access to all users' valuations, and may pick any c user(s) to collude with; in the MPC-based models described in [44, 45], the miner only has access to the valuations of c *colluding* users, so their c-SCP notion is weaker than our paper. Particularly, as their study has shown the impossibility even for $c = 2$, their method is only applicable for the case that the miner may only collude with a *fixed* user, which is highly restrictive for real-world blockchain systems.

Comparison between our mechanism and [45]. The paper of Wu et al. [45] shows that even in the MPC-based model 2-SCP is impossible. From the above explanation, their approximate Bayesian 1-SCP property is weaker than our work, but their U-DSIC property is stronger than ours. For the sybil-proofness properties, their mechanism also upper bounds the number of fake bids to h, comparable to our approximate-SP notions.

For the methodologies to secure incentive guarantees, similar to EIP-1559, the mechanism in [45] uses a variant of posted-price mechanism in which confirmed users' payments are fixed, and the miner's utility is calculated by a *linear program* and only depends on the *number* of *candidate* users with valuations above the posted price. Their mechanism thus prevents the `UB` strategy by essentially avoiding the bidding process, while our mechanism is in an auction-like form with the *auxiliary mechanism method* to ensure that honest bids achieve optimal expected utilities. In general, the paper of [45] adopts different modeling and methodology in mechanism design, while achieving a comparable level of incentive and revenue guarantees to our work. The diversity in models and methodologies renders the topic of TFM design a novel and valuable area for future exploration in the OR community.

4.2.3 Choice Modeling and the Multinomial Logit (MNL) Choice Model

Choice modeling, which models how consumers would make choices among provided goods, plays an important role in revenue management [57, 58]. Indeed, assortment optimization under a wide range of choice models has been extensively studied in the operations literature. The most popular choice model is the multinomial logit model (MNL) (see, e.g., van Ryzin and Mahajan [59], Mahajan and van Ryzin [60], Liu and van Ryzin [61], Rusmevichientong et al. [62]). Other choice models, such as nested logit models [63, 64], non-parametric choice model [65], Markov chain choice model [66, 67], have been studied under the problem of assortment optimization.

Instead of using the standard auction (e.g., first-price auction) to assign the winning bidder deterministically, the MNL model provides a randomized way to select the winning bidder. In particular, for a given set of alternatives, assuming each choice j has an expected utility $u_i(j)$ for agent i. In other words, the agent i perceives a value $\hat{u}_i(j) = u_i(j) + e_i(j)$ for the choice j, where $e_i(j)$ is a random variable of perception error with $\mathbb{E}[e_i(j)] = 0$. A standard MNL model [68] assumes that all $e_i(j)$'s are *i.i.d.* Gumbel distribution, and then the choice probability takes the following form: $\Pr[i \text{ chooses } j] = \Pr[j \in \arg\max_k \hat{u}_i(k)] = \frac{e^{m \cdot u_i(j)}}{\sum_k e^{m \cdot u_i(k)}}$.

As argued by Chung and Shi [41], randomness in choosing the winning user is necessary for a TFM to guarantee the collusion-proofness property, which we also prove in Appendix B.2

even for the Bayesian setting. Intuitively, it is more profitable for a miner to collude with a user who deterministically gets her transaction confirmed than a user who only has a certain chance. While the burning second-price mechanism in Chung and Shi [41] gives each user who bids high enough a pre-set probability to get confirmed, we consider a more natural idea of randomization by leveraging the logit choice model into the allocation rule (see Section 4.5.1). In this paradigm, we can prioritize high-bidding users in a more natural and smooth way. The MNL-based allocation rule also fits well into our auxiliary mechanism method (Section 4.4) and yields a constant-factor expected revenue compared to the optimal revenue.

In addition, in different fields of mechanism design, the MNL choice model is also adopted for other purposes. For example, Huang and Kannan [69] utilize a similar mechanism to achieve differential privacy requirements, and the mechanism is also called *exponential mechanism* in their work. Not surprisingly, they also replace the allocation rule of "the highest-bidder gets the item" with a soft-max relaxation while preserving the near-optimal property.

4.2.4 Bayesian Mechanism Design

From the famous *revelation principle* [70, 71], it is desirable to design incentive-compatible mechanisms. While the strongest notion of DSIC guarantees agents to report true types even if they have the complete information of other agents, this requirement could be a bit too restrictive in blockchain systems as users might not have such sufficient information about others.

In the Bayesian game setting, we assume that the distribution of agents' types is known by the public. At the beginning of the game, each agent's type is assigned by nature following the corresponding distribution. Then, each agent only knows her own type and the distribution of others' types conditioned on her type, and seeks to maximize her expected utility. In this scope, a mechanism is BNIC if everyone truthfully reporting their true types forms a Bayesian Nash equilibrium.

While the design of BNIC mechanisms is less restrictive than DSIC mechanisms, the

revenue equivalence theorem [70] shows that the Bayesian game setting cannot gain extra revenue in conventional auctions when users have *i.i.d.* valuation distributions, which is also the basis of a series of "equivalence" results between conventional BNIC and DSIC auctions, e.g. [72]. Furthermore, Gershkov et al. [7], Manelli and Vincent [49] show that the BNIC and DSIC conditions are equivalent in the conventional auction setting without the involvement of burning and collusion-proofness requirements. In the scope of blockchain transaction fee mechanism design, however, the existence of *burning* allows partial "revenue" (total payment from users) to be rewarded to the miner while still keeping ex-post budget feasibility. Hence, while the revenue equivalence theorem [70] dictates that the Bayesian game setting cannot increase the total user payment, our research shows that a TFM can indeed increase the miner revenue with a decreased level of burning.

Two recent works [43, 44] on blockchain transaction fee mechanism design also consider the Bayesian setting. In particular, Gafni and Yaish [43] argue that a simple variation of the first-price auction can simultaneously satisfy U-BNIC and another collusion-proofness named *OCA-proofness*, and Shi et al. [44] show that if we relax all the incentive conditions (to almost U-BNIC, almost interim MIC and almost interim c-SCP), it is also possible to achieve positive revenue. However, our result is different from [44]. We have *ex-post* almost MIC and SCP guarantees, which are crucial as discussed in Remark 1 in Appendix B.1; their work in turn considers a different MPC-assisted setting, but a weaker notion of SCP (as discussed in Section 4.2.2). Besides, the paper by Gafni and Yaish [43] is incomparable to ours as the OCA-proofness is fundamentally a different model from our work.

Similar to the currently used EIP-1559 TFM of Ethereum [42], and papers of Gafni and Yaish [43] and Shi et al. [44], our work also uses a *prior-dependent* mechanism that requires a parameter c_ρ (as defined in Section 4.5.2) that depends on the prior distribution. While blockchain mechanisms are usually hard-coded into the system, the distributional parameter can still be implemented to update adaptively based on historical data, similar to the base fee in EIP-1559. As shown by Maheshwari et al. [73], adaptiveness is indeed crucial in the development of social optimality in large-scale network mechanisms in the presence of selfish agents.

4.3 Preliminaries

4.3.1 Overview and Classification of Dishonest Behavior

In the blockchain system, either a user or miner may *deviate* from the supposed behavior and behave dishonestly. First of all, as the blockchain system is anonymous, either type of agent may conduct *sybil attack* [74] by creating multiple fake identities to influence the performance of the system. However, in the blockchain system, the PoW or PoS mechanism makes it costly to create a fake identity as a (winning) miner, so we mainly consider creating fake user identities, i.e. injecting fake bids, as also mentioned by Roughgarden [37] and Chung and Shi [41]. On the other hand, even if every agent takes their true identities, they may do their jobs dishonestly: the users may bid differently from their true valuation, and the miner may purposely ignore some bids. Furthermore, the miner and users may also collude, i.e. conduct such dishonest behavior as a party in seek of increasing their total utility.

In summary, dishonest behavior (deviations) can include untruthful bidding, fake identities and dishonest confirmation. Therefore, the deviations can be classified into the following three types:

- Untruthful Bids: proposing a bid different from the true valuation.
- Fake Transactions (Sybil attack): injecting fake transactions.
- Transaction Deletion: ignoring certain transactions proposed by users.

The dishonest behavior can be conducted by the miner, a user, or a colluding party of them. A user, or a colluding party of multiple users, can make untruthful bids and inject fake transactions, and the miner can inject fake transactions and delete existing transactions. A colluding party that consists of the miner and users, can do all these three deviations. Therefore, there can be 9 types of deviations in the system, as shown in Table 4.1.

We say a dishonest action is *profitable* when it strictly increases the total utility of all agents participating in it. Precisely,

- If it is an individual deviation, the agent strictly increases her utility via that deviation.
- If it is a collusion, the colluding party strictly increases its total utility via that collusion.

In this sense, for a strategy space $\mathscr{S}$, we say a TFM is $\mathscr{S}$-proof if all deviations in $\mathscr{S}$ are not profitable in this TFM.

Information sets of agents. Before specifying the strategy-proof conditions, we need to discuss the *information sets* of agents. In traditional auctions, the notion of *incentive compatibility* means that the mechanism would optimize any bidder's utility when they bid their true valuations. On the other hand, the subtle meaning of *incentive compatibility* also depends on *what the bidders know*. In this scope, the strongest notion is Dominant Strategy Incentive Compatibility (DSIC), which means that it optimizes any individual bidder's utility even if they know all others' bids, i.e. they have the complete information. Nevertheless, in sealed-bid auctions, bidders would not actually know what other bids, so the *complete-information (deterministic) setting* may be too strong. Nevertheless, we may still assume that they can perceive the *distributions* of others' bids. In this so-called *Bayesian-game setting*, if the mechanism can guarantee that any bidder would maximize their *expected* utility via bidding their true valuation, we call the property Bayesian-Nash Incentive Compatibility (BNIC).

In our paper, the bidding is conducted by users, so we name the DSIC and BNIC properties for users as U-DSIC and U-BNIC, respectively. The formal definitions are in Section 4.3.3.

4.3.2 The Basic Model

There are n users numbered by $1, 2, \ldots, n$ and each user proposes a transaction to compete for a block. There is also a winning miner owning the block. The block has size k, the maximum number of transactions it can confirm. For user i, w.l.o.g. we assume her valuation v_i is in $[0, 1]$ and drawn from an *i.i.d.* distribution $V_i = V_0$ with pdf $\rho_i(\cdot) = \rho(\cdot)$. We let $V = V_1 \times V_2 \times \cdots \times V_n$ be the distribution of the valuation vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$.

By the revelation principle ([70, 71], see Appendix B.1), we only need to consider direct

mechanisms in which users propose bids, the miner collects the bids and the system decides which transactions to confirm and processes the payments. Formally, we can model any Transaction Fee Mechanism w.r.t. its allocation, payment and miner revenue rules, as follows.

Definition 7 (Transaction Fee Mechanism)**.** *For a fixed number n of users, a Transaction Fee Mechanism is defined by $M(\mathbf{a}, \mathbf{p}, r)$, where*

- *the* allocation rule $\mathbf{a} : [0,1]^n \to [0,1]^n$ *maps the bid vector to the allocation vector indicating the probability each user's transaction to be confirmed;*
- *the* payment rule $\mathbf{p} : [0,1]^n \to \mathbb{R}^n$ *maps the bid vector to the payment vector indicating the payment of a user* if her transaction is confirmed;[2]
- *the* miner revenue rule $r : [0,1]^n \to \mathbb{R}$ *maps the bid vector to the miner's revenue.*

In the naïve implementation of transaction collection in blockchains, users propose transactions (including bids) publicly and sequentially in a mempool and miners pack them into blocks, so it acts as an auction format between open bidding and sealed bidding — users can see bids submitted before them but not after them, which may lead to several issues (see Appendix B.1). In our mechanism, we implement sealed bids via a *commitment scheme* as described in Appendix B.1 so that no bid can be viewed by the miner or other users until all transactions that compete for the block are finalized. In the execution of the mechanism, the system essentially elicits users for their (sealed) bids $\{b_i\}$, draw the confirmed transactions according to probabilities from $\{a_i(b_i, \mathbf{b}_{-i})\}$ (for $k > 1$, one follows the sampling method discussed in Section 4.6.1 to ensure exactly k transactions are confirmed), and then charge transaction fees from confirmed bidders according to $\{p_i(b_i, \mathbf{b}_{-i})\}$ and give the miner revenue $r(\mathbf{b})$ to the miner. Due to the size constraint, we need to guarantee $\sum_{i=1}^{n} a_i(b_i, \mathbf{b}_{-i}) \leq k$. Since the transactions are naturally anonymous and unsorted, in this paper we only consider the mechanisms satisfying the following symmetric condition:

[2]This definition is different from some literature (where the "payment rule" indicates the expected payment of a user whether she gets confirmed or not, which can be transformed to $a_i(\cdot)p_i(\cdot)$ in our notation). Besides, our definition naturally guarantees that unconfirmed users do not pay transaction fees.

Definition 8 (Symmetry)**.** *A TFM is symmetric if the allocation and payment rules do not depend on the order of users, i.e. when we swap any pair of users, each should still have the same allocation probability and payment as in their original positions.*

In Definition 7, we assume that (in the usual case) each of the n users makes one bid, and therefore the allocation, payment, and miner revenue rules are functions of exactly n bids. While this definition is enough for the discussion of the main strategy-proof properties (e.g., U-BNIC, 1-SCP) concerned in this paper, we also expect the proposed TFMs to be strategy-proof against deviations such as the Sybil Attack and the deletion of user bids by the miner (namely, `FT` and `TD` in Table 4.1). These deviations could change the number of bids presented to the TFM, and we need to define the following *variable-bid-size TFM* to deal with this technical issue.

Definition 7' (Variable-bid-size TFM). A variable-bid-size TFM $\mathscr{M}(\mathbf{a}, \mathbf{p}, r)$ is similar to the regular TFM defined in Definition 7 where the only difference is that the allocation rule $\mathbf{a} : [0,1]^* \to [0,1]^*$,[3] the payment rule $\mathbf{p} : [0,1]^* \to \mathbb{R}^*$ and the miner revenue rule $r : [0,1]^* \to \mathbb{R}$ may take sequences of bids of any size, while the size of the output of $\mathbf{a}$ ($\mathbf{p}$) should be the same as the input, where each entry in the output is the allocation (the bid respectively) of the corresponding bid.

Throughout the paper when we refer to a TFM M, unless specially noted, we assume that M is a regular TFM (Definition 7). We will mostly focus on the design of a TFM for any fixed bid size. To construct a variable-bid-size TFM $\mathscr{M}$, we will first choose an $M_\eta = (\mathbf{a}_\eta, \mathbf{p}_\eta, r_\eta)$ for η bids for each η according to the regular TFM design, and then let $\mathscr{M} = (\mathbf{a}, \mathbf{p}, r) = \cup_\eta \{M_\eta\}$, or more concretely, for any bidding vector $\mathbf{b}$, let $|\mathbf{b}|$ denote the size of $\mathbf{b}$ and we set

$$\mathbf{a}(\mathbf{b}) = \mathbf{a}_{|\mathbf{b}|}(\mathbf{b}), \mathbf{p}(\mathbf{b}) = \mathbf{p}_{|\mathbf{b}|}(\mathbf{b}), r(\mathbf{b}) = r_{|\mathbf{b}|}(\mathbf{b}). \tag{4.1}$$

Given a variable-bid-size TFM $\mathscr{M}$, for any fixed number of bids, namely n, there is a *natural restriction* of $\mathscr{M}$ to a regular TFM, namely M. To derive M, we simply let its

[3]For any set A, we use $A^* = \cup_{\ell=0}^{\infty} A^{\times \ell}$ to denote the set of all finite sequences where the elements are drawn from A.

allocation, payment, and miner rules be the corresponding functions of $\mathscr{M}$ when restricted to inputs of size n.

4.3.3 Incentive and Collusion-Proof Conditions

We now discuss the desired properties we would like the mechanism to enjoy, i.e. the properties that agents would not gain additional utility via dishonest behavior (UB, FT, TD).

As a basis, we note that the users have *quasi-linear* utility: when user i has valuation v_i and the bidding vector is $(b_i, \mathbf{b}_{-i})$, user i's (expected) utility is

$$u_i(b_i, \mathbf{b}_{-i}; v_i) = a_i(b_i, \mathbf{b}_{-i}) \cdot (v_i - p_i(b_i, \mathbf{b}_{-i})). \tag{4.2}$$

For the miner's utility, in real-world blockchains, the miner's reward comes from the combination of the *block reward* and the *miner revenue* from the transaction fees, and the miner also pays a mining cost due to the computational consumption / token staking in PoW or PoS protocols, respectively. Since the block reward and the mining cost are fixed due to the blockchain protocol and is not affected by the transactions, for simplicity of expression, we just denote the miner's utility as the miner revenue provided by the TFM, i.e.,

$$u^{(miner)}(\mathbf{b}) = r(\mathbf{b}). \tag{4.3}$$

We now formally define U-DSIC and U-BNIC as follows.

Definition 9 (User Dominant-Strategy-Incentive-Compatibility (U-DSIC))**.** *For any user i, assuming the miner follows the inclusion rule truthfully, a TFM is U-DSIC if and only if it is a dominant strategy for any user to bid their valuations, i.e. $\forall \mathbf{b}_{-i}, v_i \in \arg\max_{b_i}[u_i(b_i, \mathbf{b}_{-i}; v_i)]$.*

Definition 10 (User Bayesian-Nash-Incentive-Compatibility (U-BNIC))**.** *Assume Ω is the type space of nature, and there is a public mapping $B : \Omega \to \mathbb{R}^n_{\geq 0}$ that determines the valuation of all users. Thus, the valuation vector $\mathbf{v} \sim V = V_1 \times V_2 \times \cdots \times V_n$ where each $V_i = V_0$ due to our model assumption.*

For each user i, she only knows her own valuation and the distribution of other users'

valuations conditioned on v_i, denoted as $V_{-i} = V|v_i$. A TFM is (interim) U-BNIC if and only if, when other users all bid their valuations, it maximizes user i's expected utility if she bids her valuation too, i.e. $v_i \in \arg\max_{b_i} \mathbb{E}_{\mathbf{b}_{-i} \sim V_{-i}}[u_i(b_i, \mathbf{b}_{-i}; v_i)]$.

We also characterize the user Sybil-proofness property, which guarantees that the user cannot increase via injecting fake bids, her expected utility over the distribution of other users. Formally,

Definition 11 (User Sybil-Proofness (U-SP))**.** *Assuming each user i only knows her own valuation and the distribution of other users' valuations conditioned on v_i, denoted as V_{-i}, and assuming all users bid their true valuations. Then, we call a variable-bid-size TFM (interim) (C, N)-U-SP for a fixed (C, N) when $n > N$, user i cannot increase her expected utility (over the distributions of other users' bids) via injecting $l \leq Cn$ fake bids $\mathbf{b}^{\#} = \{b_{n+1}, \cdots, b_{n+l}\}$. Here we still denote $\mathbf{b} = (b_1, \cdots, b_n)$, and define $\mathbf{b}^{+} = (b_1, \cdots, b_{n+l})$ containing all real and fake bids.*

Notice that fake bids generally have a valuation of 0 because getting the fake transaction confirmed does not have any value for the user. A possible exception is repeating the same transaction when the block size $k = 1$, as at most one of them can be confirmed. However, in the general case where $k \geq 2$, the adverse consequence of having both transactions confirmed (e.g., paying twice for the same transaction) far outweighs the benefit of transaction confirmation. On the other hand, preventing this type of strategy when $k = 1$ is impossible for the following reason: if all transactions have the same bid, the anonymity of the blockchain system ensures that any mechanism can only randomly select a transaction to confirm. Therefore, duplicating a transaction will always increase the likelihood of it being confirmed. Since our study is mainly motivated by real-world blockchains with $k \geq 2$ block sizes, we exclude these types of strategies from consideration in our model.

The utility of user i is the total utility of i herself and all fake bids. Therefore, a variable-

bid-size TFM is (C, N)-U-SP if and only if for any valid $\mathbf{b}^+$, if $b_i = v_i$, then

$$\begin{aligned}&\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}[u_i(b_i, \mathbf{b}_{-i}; v_i)]\\ \geq&\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}\left[u_i(b_i, \mathbf{b}^+_{-i}; v_i) + \sum_{j=n+1}^{n+l} u_j(b_j, \mathbf{b}^+_{-j}; 0)\right].\end{aligned}$$

As a shorter notion, we call a variable-bid-size TFM U-SP when there exist constants $C > 0, N > 0$ such that the TFM is (C, N)-U-SP.

To describe the collusion-proofness, we use the notation of c-SCP in Chung and Shi [41], defined as:

Definition 12 (c-Side-Contract-Proofness (c-SCP))**.** *We call a TFM (ex-post) c-SCP when is impossible for the miner to collude with at most c users to strictly increase their total utility when other users bid according to the Bayesian Nash equilibrium, even if the miner knows all users' valuations and bids.*

Note that successful collusion only needs the party to have increased total utility rather than individual utilities, because the members can make payments among themselves to make everyone get increased utility.

Collusion-proofness notions for non-truthful TFMs. It might be tricky to define the collusion-proofness for a TFM that does not satisfy U-BNIC. For example, Chung and Shi [41] claimed that the first-price auction mechanism is c-SCP in the sense that c users and the miner could not increase their joint utility via any deviation *when all other users bid truthfully*. However, in the first-price auction, the users may not report their real valuation, leading to a different Bayesian Nash equilibrium.

In our Bayesian setting, we assume that users who do not conduct the collusion would bid according to the Bayesian Nash equilibrium to maximize their expected utility. We show in Appendix B.3.1 that *at a Bayesian Nash equilibrium*, a user and the miner could increase their joint utility via deviation, even if burning is allowed. Therefore, we state that the first-price auction is not even 1-SCP in our notion.

Nevertheless, for TFMs that satisfy U-BNIC, assuming non-colluding users to bid truthfully or to bid as the Bayesian Nash equilibrium does not make a difference.

Miner-Incentive-Compatibility. As is mentioned in the papers of Chung and Shi [41] and Roughgarden [37], the Miner-Incentive-Compatibility (MIC) is the property that assuming the users bid truthfully, the miner could not increase her utility via deviations from the inclusion rule (i.e. `M-FT` and `M-TD`). It will be rigorously defined in Section 4.7.1 (Definition 15).

4.3.4 Rationality and Feasibility Requirements

Besides truthfulness and collusion-proofness, the mechanism also needs to satisfy more general properties, e.g. the balance must be feasible, the users should not pay more than their bid, etc. Formally, the following properties should also be satisfied.

(Ex-post) User Individually Rationality (UIR). Each user gets non-negative utility when truthful bidding, no matter how others bid, i.e. $\forall \mathbf{b}_{-i}$, $u_i(v_i, \mathbf{b}_{-i}; v_i) \geq 0$. Equivalently, $a_i(v_i, \mathbf{b}_{-i}) > 0 \Rightarrow p_i(v_i, \mathbf{b}_{-i}) \leq v_i$.

(Ex-post) Budget Feasibility (BF). For all bidding vector $\mathbf{b}$, the miner's revenue $r(b_i, \mathbf{b}_{-i})$ should not be greater than the total user payment:

$$P(\mathbf{b}) = \sum_{i=1}^{n} a_i(b_i, \mathbf{b}_{-i}) \cdot p_i(b_i, \mathbf{b}_{-i}). \tag{4.4}$$

In other words, we should have $\forall \mathbf{b}$, $P(\mathbf{b}) \geq r(\mathbf{b})$.

Here we allow the miner revenue to be *less* than the total fee paid by users, in which the difference will be *burnt*. The burning can decouple payments on miners' and users' sides, which is an effective way to broaden the design space and allow additional strategy-proof properties to be satisfied [2]. Actually, the burning has been used in the EIP-1559 TFM of Ethereum.

Additionally, while we expect the transaction *fee* a user pays should be non-negative, it is okay as long as the users have a non-negative *expected* payment to prevent users from submitting transactions to gain money out of nothing, which is guaranteed in our mechanisms for both $k = 1$ and general k (as (B.34) is satisfied). However, as UIR requires the payment of the zero-valuation user to be no greater than zero, the payment of the zero-valuation user

is always zero (rather than negative). Therefore, we need the following NFL condition.

No-Free-Lunch (NFL). We call a TFM $(\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ NFL when the payment of a zero-bidding user is always zero no matter how other users bid, i.e.,

$$a_i(0, \mathbf{b}_{-i})\tilde{p}_i(0, \mathbf{b}_{-i}) = 0, \quad \forall \mathbf{b}_{-i}. \tag{4.5}$$

4.3.5 Deterministic and Randomized Mechanisms

We assume generic positions of bids, which means that all bids are distinct. For simplicity, we would want the mechanism to be deterministic, i.e., the same input bidding vector leads to the same allocation outcome, equivalently $a_i(b_i, \mathbf{b}_{-i}) \in \{0, 1\}$. However, we can prove that even if we relax U-DSIC to U-BNIC, no *deterministic* U-BNIC and 1-SCP TFM that satisfy mild conditions can achieve positive miner revenue, indicating that the randomness in our main mechanism is necessary. The formal discussion is in Appendix B.2.

An intuitive explanation about how this works is as follows: to construct a U-BNIC TFM we essentially "adjust" the users' payments from a U-DSIC mechanism in a way that increases the miner revenue while preserving U-BNIC and 1-SCP (via the auxiliary mechanism method discussed in Section 4.4); however in a deterministic mechanism, the payments of users with $a_i(b_i, \mathbf{b}_{-i}) = 0$ are fixed at 0 and cannot be adjusted, rendering the auxiliary mechanism method inapplicable.

4.4 The Auxiliary Mechanism Method

In this section, we introduce our main technique to construct the desired U-BNIC mechanism, named as *auxiliary mechanism method*. We will make the connection between BNIC and DSIC mechanisms by developing a decomposition of many U-BNIC TFMs into an *auxiliary* U-DSIC TFM and a so-called *variation term*. In light of this, we develop Theorem 4.2, the key theorem of this section, to provide sufficient conditions for any combination of an auxiliary U-DSIC TFM and a variation term to form a desired TFM that is simultaneously U-BNIC and 1-SCP (Section 4.4.1 and Section 4.4.2). Our Theorem 4.2 will provide a general

framework to facilitate the construction of our desired U-BNIC TFMs, and we will discuss more about this framework in Section 4.4.3. Finally, in Section 4.4.4 and Appendix B.3, we will provide more explanations and concrete examples to help the readers understand our auxiliary mechanism method, which, however, is not a pre-requisite of the constructions in the later sections.

4.4.1 The Dominant Auxiliary of a BNIC TFM and Their Relations

For simplicity, let us first consider the mechanism on the users' side, and recall the famous Myerson's Lemma [70] that characterizes the sufficient and necessary condition for a TFM to be U-DSIC, stated as follows. (We note that the original Myerson's Lemma is stated for auctions. However, if we only focus on the users' incentive compatibility constraints, a TFM reduces to an ordinary auction. Indeed, in our statement of Lemma 4.1, the miner's revenue function r is irrelevant.)

Lemma 4.1 (Myerson's Lemma [70])**.** *Any TFM $M = (\mathbf{a}, \mathbf{p}, r)$ is U-DSIC if and only if the following conditions are satisfied.*

- *Monotone allocation: $a_i(\cdot, \mathbf{b}_{-i})$ is monotonic non-decreasing,*
- *Constrained payment function:*

$$\begin{aligned} & a_i(b_i, \mathbf{b}_{-i}) p_i(b_i, \mathbf{b}_{-i}) \\ = & \int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt + a_i(0, \mathbf{b}_{-i}) p_i(0, \mathbf{b}_{-i}). \end{aligned} \tag{4.6}$$

Motivated by Lemma 4.1, for any monotonic non-decreasing allocation rule $\mathbf{a}$, we define a payment rule $\mathbf{p}$ to be its *dominant association*. In particular, we set

$$p_i(b_i, \mathbf{b}_{-i}) = \begin{cases} \frac{\int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt}{a_i(b_i, \mathbf{b}_{-i})}, & a_i(b_i, \mathbf{b}_{-i}) > 0 \\ 0, & a_i(b_i, \mathbf{b}_{-i}) = 0 \end{cases}. \tag{4.7}$$

Since our definition Eq. (4.7) satisfies the condition in Eq. (4.6), for any monotonic non-decreasing $\mathbf{a}$, together with its dominant association $\mathbf{p}$, we get a U-DSIC mechanism $(\mathbf{a}, \mathbf{p}, 0)$.

(The miner reward function here is set to be constantly 0 only for illustration purpose. It could be a different r.) We also note that there seems to be little freedom for the payment rule $\mathbf{p}$ in order to form a U-DSIC TFM with $\mathbf{a}$. Indeed, $p_i(b_i, \mathbf{b}_{-i})$ is relevant only when $a_i(b_i, \mathbf{b}_{-i}) > 0$ (i.e., when the i-th bidder has a chance to be confirmed). In this case, if we additionally add the natural boundary condition $p_i(0, \mathbf{b}_{-i}) = 0$ (similar to the NFL assumption), then $p_i(\cdot, \mathbf{b}_{-i})$ defined in Eq. (4.7) is the unique solution to Eq. (4.6).

For any U-BNIC mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ with monotonic non-decreasing $\mathbf{a}$, we let $M = (\mathbf{a}, \mathbf{p}, 0)$ to be its *dominant auxiliary* mechanism where $\mathbf{p}$ is the dominant association of $\mathbf{a}$. Let us compare the payment rules of the two mechanisms and define a "payment difference" function that denotes the over-payment of each user according to $\tilde{M}$ compared to M, as

$$\theta_i(b_i, \mathbf{b}_{-i}) = a_i(b_i, \mathbf{b}_{-i})(\tilde{p}_i(b_i, \mathbf{b}_{-i}) - p_i(b_i, \mathbf{b}_{-i})). \tag{4.8}$$

Note that the Revenue Equivalence Theorem indicates that for the same *i.i.d.* path-connected distribution of valuations and given boundary conditions, all BNIC mechanisms with the same allocation rule should have the same *expected* payment for a bidder with a fixed valuation (where the expectation is taken over the valuation of the other bidders). Since $M = (\mathbf{a}, \mathbf{p}, 0)$ is a DSIC mechanism and therefore also BNIC, we have that $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ and $(\mathbf{a}, \mathbf{p}, 0)$ make the same expected payment for any bidder the fixed valuation. Formally for any bidder i, we have that

$$\mathbb{E}_{\mathbf{b}_{-i} \sim V_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0, \tag{4.9}$$

which marks the close relation between the BNIC mechanism $\tilde{M}$ and its dominant auxiliary M.

4.4.2 The Auxiliary-Variation Decomposition

We have just shown that a U-BNIC mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ has a corresponding dominant auxiliary U-DSIC mechanism $M = (\mathbf{a}, \mathbf{p}, 0)$ (when $\mathbf{a}$ is monotone) and defined their payment difference to be $\boldsymbol{\theta}$ defined in Eq. (4.8). Formally, we summarize this decomposition and make

the following definition.

Definition 13 (Auxiliary-Variation Decomposition)**.** *Given a mechanism* $\tilde{M} = (\mathbf{a}, \tilde{p}, \tilde{r})$ *where* $\mathbf{a}$ *is monotonic non-decreasing, we set up the* auxiliary mechanism $M = (\mathbf{a}, \mathbf{p}, 0)$ *and the* variation term $T = (\boldsymbol{\theta}, \tilde{r})$*. Suppose we have the following conditions met, we call* (M, T) *an* auxiliary-variation decomposition *of* $\tilde{M}$ *and also write* $\tilde{M} = M + T$ *for short.*

1. *In the auxiliary mechanism* $M = (\mathbf{a}, \mathbf{p}, 0)$*,* $\mathbf{p}$ *is the dominant association of* $\mathbf{a}$*.*
2. *The variation term* $T = (\boldsymbol{\theta}, \tilde{r})$ *satisfies that* $\tilde{p}_i(b_i, \mathbf{b}_{-i}) = p_i(b_i, \mathbf{b}_{-i}) + \frac{\theta_i(b_i, \mathbf{b}_{-i})}{a_i(b_i, \mathbf{b}_{-i})}$*, where we require that* $\theta_i(b_i, \mathbf{b}_{-i}) = 0$ *whenever* $a_i(b_i, \mathbf{b}_{-i}) = 0$ *and treat* $0/0$ *as* 0*.*

We will use Definition 13 in a reverse way: given a TFM $M = (\mathbf{a}, \mathbf{p}, 0)$ such that $\mathbf{p}$ is the dominant association of $\mathbf{a}$, if we could design a variation term $T = (\boldsymbol{\theta}, \tilde{r})$ that satisfies the additional *admissibility* conditions, then we would expect that $\tilde{M} = M + T$ is not only U-BNIC but also 1-SCP. Formally, we define the admissibility conditions as follows.

Definition 14. *We call the variation term* $T = (\boldsymbol{\theta}, \tilde{r})$ admissible *if it satisfies the following conditions for every* $\mathbf{b}$*.*

$$\tilde{r}(b_i, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i}) = \theta_i(b_i, \mathbf{b}_{-i}), \tag{4.10}$$

$$\mathbb{E}_{\mathbf{b}_{-i} \sim V_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0. \tag{4.11}$$

We note that the second admissibility condition (Eq. (4.11)) derives from Eq. (4.9) which is necessary for the composed TFM $\tilde{M}$ to be U-BNIC. The first admissibility condition (Eq. (4.10)) is key to guarantee the 1-SCP property, which will be further explained in Section 4.4.4.

The following theorem states that the admissibility conditions for the variation term are enough to guarantee the composed TFM is U-BNIC and 1-SCP, and is the basis of our TFM constructions later.

Theorem 4.2. *Suppose the variation term* T *is admissible. For any auxiliary mechanism* M *that may form an auxiliary-variation decomposition with* T*, the composed TFM* $\tilde{M} = M + T$ *is U-BNIC and* 1*-SCP.*

The formal proof of Theorem 4.2 is deferred to Appendix B.4.1.

4.4.3 Using the Auxiliary-Variation Decomposition to Construct TFMs

Given a U-DSIC TFM $M = (\mathbf{a}, \mathbf{p}, 0)$ where $\mathbf{a}$ is monotonic non-decreasing and $\mathbf{p}$ is the dominant association of $\mathbf{a}$, it is trivial to see that $T_\perp = (0, 0)$ is a trivial admissible variation term and $M+T_\perp = (\mathbf{a}, \mathbf{p}, 0)$ is both U-BNIC and 1-SCP. However, in this trivial construction, the miner revenue is always 0, which is not desirable.

In order to achieve a larger miner revenue, we would like to explore more choices of $\tilde{M}$. Now, Theorem 4.2 provides an approach to create a class of U-BNIC and 1-SCP TFMs based on M, so that we could hope to find a desired TFM from this class (e.g., with large miner revenue, and other properties such as UIR, BF, etc.). In particular, for any fixed M, we will be able to able to perturb the payment function (via $\boldsymbol{\theta}$) and create more design choices for $\tilde{r}$, which jointly enrich the space of admissible variation terms $\{T\}$, as well as the corresponding class $\{\tilde{M}\}$ of the composed U-BNIC and 1-SCP TFMs, thanks to Theorem 4.2.

One nice thing about the above approach is that it is *almost modular* in the design of the auxiliary mechanism M and the variation term T. We note that the constraints for M are almost independent of that for T, while the only correlating constraint is that $a_i(b_i, \mathbf{b}_{-i}) = 0 \Rightarrow \theta_i(b_i, \mathbf{b}_{-i}) = 0$, which is very mild and holds for most natural choices of M and T. This modular framework decouples our design tasks for M and T, greatly reduces the design complexity and renders our final mechanism more interpretable.

To describe more details about our construction framework, note that we would like to construct a TFM that also satisfies the UIR and BF conditions. Note that the auxiliary $M(\mathbf{a}, \mathbf{p}, 0)$ always satisfies UIR, and the zero miner revenue indicates it also satisfies BF. Now, intuitively, if we make the variation term T "small enough" so that $\tilde{M} = M + T$ is close to M, $\tilde{M}$ is also likely to satisfy UIR and BF conditions, although having lower miner revenue as well. We further notice that the admissibility condition of T is *scale-free*, i.e., if $T = (\boldsymbol{\theta}, \tilde{r})$ is admissible and for any $h > 0$, $hT = (h \cdot \boldsymbol{\theta}, h \cdot \tilde{r})$ is also admissible. Therefore, we will construct our desired TFM along the following steps.

1. Construct an allocation rule $\mathbf{a}$, derive the corresponding dominant association $\mathbf{p}$ and

the auxiliary TFM $M = (\mathbf{a}, \mathbf{p}, 0)$.

2. Find a "good" admissible variation term $T = (\boldsymbol{\theta}, \tilde{r})$.

3. Compute the (approximately) optimal h so that $M + hT$ maximizes miner revenue while still obeying UIR and BF conditions.

In Sections 4.5–4.6, we will present our constructions of M and T, which, together with the optimal choice of h, can achieve a constant-fraction approximation of the optimal miner revenue (with the desired properties: U-BNIC, 1-SCP, UIR, BF, etc.).

4.4.4 Further Explanation of the Condition Eq. (4.10).

In this part, we explain the relationship between the payment difference function $\boldsymbol{\theta}$ and the miner revenue function $\tilde{r}$ in an admissible variation term, as well as the condition Eq. (4.10) for admissibility.

To characterize the properties of the TFM $\tilde{M}$, we first look into the relations between $\tilde{\mathbf{p}}$ and $\tilde{r}$. Recall that in our model, users and the miner have different information on the bids: a user only knows the distribution of other users' valuations (and bids, if the mechanism is U-BNIC), but the miner knows all bids accurately. Therefore, in mechanism $\tilde{M}$, for fixed $\mathbf{b}_{-i}$, truthfully bidding v_i does not guarantee to maximize user i's utility $\tilde{u}(b_i, \mathbf{b}_{-i}; v_i)$ (which is defined to be $a_i(b_i, \mathbf{b}_{-i}) \cdot (v_i - \tilde{p}_i(b_i, \mathbf{b}_{-i}))$ following the definition in Eq. (4.2)), but it must maximize the total utility of her and the miner, as $\tilde{u}(b_i, \mathbf{b}_{-i}; v_i) + \tilde{r}(b_i, \mathbf{b}_{-i})$, so that the miner is not incentivized to ask her to deviate. Therefore, if we further assume smoothness for simplicity (a formal proof without the smoothness assumption is provided in Appendix B.4.1), we have

$$\frac{\partial}{\partial b_i} \left(\tilde{u}(b_i, \mathbf{b}_{-i}; v_i) + \tilde{r}(b_i, \mathbf{b}_{-i}) \right) \Big|_{b_i = v_i} = 0. \tag{4.12}$$

However, in the auxiliary TFM M, which is U-DSIC, bidding $b_i = v_i$ maximizes user i's

utility, hence

$$\frac{\partial}{\partial b_i} u_i(b_i, \mathbf{b}_{-i}; v_i)\Big|_{b_i = v_i} = 0. \tag{4.13}$$

Since TFMs $\tilde{M}$ and M have the same allocation rule, users' utilities only differ in payments, we have $\tilde{u}(b_i, \mathbf{b}_{-i}; v_i) = u(b_i, \mathbf{b}_{-i}; v_i) - \theta_i(b_i, \mathbf{b}_{-i})$. Therefore, we get the relation between $\theta_i(b_i, \mathbf{b}_{-i})$ and $\tilde{r}(b_i, \mathbf{b}_{-i})$:

$$\frac{\partial}{\partial b_i} \theta_i(b_i, \mathbf{b}_{-i}) = \frac{\partial}{\partial b_i} \tilde{r}(b_i, \mathbf{b}_{-i}). \tag{4.14}$$

That is, if user i would benefit from an infinitesimal deviation from truthful bidding, the miner would lose the same amount in turn, so that the miner has no incentive to let user i deviate, even though she has additional information about other users' bids. With the boundary condition $\tilde{p}_i(0, \mathbf{b}_{-i}) = 0$ (thus $\theta_i(0, \mathbf{b}_{-i}) = 0$), we get

$$\theta_i(b_i, \mathbf{b}_{-i}) = \tilde{r}(b_i, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i}), \quad \forall i.$$

This characterizes the relation between user payments and miner revenue in 1-SCP TFMs, and also shows the need of condition Eq. (4.10) in an admissible variation term.

Following the discussion of this part, we can actually find a critical challenge in constructing an admissible variation term, and develop an alternative field-theoretic perspective of the admissibility condition. The detailed discussion is in Appendix B.3.

4.5 The Proposed Mechanism for Block Size 1

In this section, we consider the case with block size $k = 1$, where exactly one transaction is confirmed, to give a simple and intuitive understanding of our mechanism. We follow the pipeline of auxiliary mechanism method in construction. In Section 4.5.1 we construct the auxiliary mechanism named soft second-price mechanism, and in Section 4.5.2 we compute the variation term, thus finishing the construction of the proposed mechanism and compute

Mechanism 1 Auxiliary Mechanism for block size 1

$$a_i(b_i, \mathbf{b}_{-i}) = \frac{e^{mb_i}}{\sum_{j=1}^{n} e^{mb_j}} \tag{4.15}$$

$$p_i(b_i, \mathbf{b}_{-i}) = b_i - \frac{\sum_{j=1}^{n} e^{mb_j}}{me^{mb_i}} \cdot \ln \frac{\sum_{j=1}^{n} e^{mb_j}}{1 + \sum_{j \neq i} e^{mb_j}} \tag{4.16}$$

$$r(\mathbf{b}) = 0. \tag{4.17}$$

its miner revenue.

4.5.1 Auxiliary: the Soft Second-Price Mechanism

The second-price auction mechanism has been widely used in traditional auctions, in which the highest bidder gets confirmed but pays the second-highest bid. However, as we prove that any deterministic TFM which is U-BNIC and 1-SCP satisfying mild assumptions has non-positive miner revenue (Appendix B.2), we try to introduce randomness into the allocation rule.

Here we consider the widely used multinomial logit choice model in which the choice probability of an item is proportional to the exponential of a parameter m times its value. If we set the m to infinity, then the item with the highest value is deterministically chosen, coinciding with the allocation rule of second-price auction; if we set m to zero, then all items are randomly chosen with uniform chances. For $m \in (0, +\infty)$, the choice is random, but higher-valued items are more likely to be chosen.

As a basis of our main mechanism, we first develop a U-DSIC and 1-SCP mechanism named soft second-price mechanism, which is the auxiliary mechanism of our proposed TFM. It adopts the multinomial logit choice model as the allocation rule. After fixing the allocation rule $\mathbf{a}$, we derive the corresponding dominant association $\mathbf{p}$ (according to Eq. (4.7)), and form the auxiliary mechanism $M = (\mathbf{a}, \mathbf{p}, 0)$, which is explicitly presented in Mechanism 1.

One good thing about our auxiliary mechanism is that every entry of the allocation function $\mathbf{a}$ is always positive for all $m < \infty$. Therefore, it automatically satisfies the

requirement in the second item of Definition 13 and can be combined with any variation term (to be designed soon) in our auxiliary-variation decomposition.

Although the soft second-price mechanism has zero miner revenue, we can modify it via the auxiliary mechanism method that preserves U-BNIC and 1-SCP properties and yields positive expected miner revenue, as in Section 4.5.2.

4.5.2 The Variation Term and Our Proposed Mechanism for Block Size 1

Following the auxiliary mechanism method in Section 4.4, we can construct a mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ via the composition of its auxiliary mechanism M and the variation term $T = (\boldsymbol{\theta}, \tilde{r})$. In this section, we construct the variation term of our proposed mechanism.

When the distributions of all users' valuations are *i.i.d.*, i.e. $V = V_1 \times V_2 \times \cdots \times V_n$ and $\forall V_i = V_0$ has identical pdf $\rho : [0, 1] \to [0, +\infty)$, we denote

$$c_\rho = \int_0^1 \rho^2(t)dt. \tag{4.18}$$

Now, for any scaling parameter $h \in [0, +\infty)$, we construct $T = (\boldsymbol{\theta}, \tilde{r})$ as follows. The intuition in the construction is elaborated in Appendix B.3.3.

$$\theta_i(b_i, \mathbf{b}_{-i}) = -\frac{1}{2} h b_i^2 \left(\frac{\sum_{j \neq i} b_j^2}{c_\rho (n-1)} - 1 \right), \tag{4.19}$$

$$\tilde{r}(\mathbf{b}) = \frac{1}{2} h \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho (n-1)} \right). \tag{4.20}$$

As a sanity check, we note that $\tilde{r}$ is the potential of $\mathbf{D}_{\boldsymbol{\theta}}$ (as in Appendix B.3). Formally, the following lemma verifies that the above variation term T is admissible. The proof of Lemma 4.2 is deferred to Appendix B.4.2.

Lemma 4.2. *The variation term $T = (\boldsymbol{\theta}, \tilde{r})$ defined in Eqs. (4.19-4.20) is admissible.*

Now we combine the auxiliary mechanism M defined in Mechanism 1 and our variation term T to form the mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r}) = M + T$, which is explicitly presented in

Mechanism 2 Transaction Fee Mechanism for block size 1

$$
\begin{aligned}
a_i(b_i, \mathbf{b}_{-i}) &= \frac{e^{mb_i}}{\sum_{j=1}^{n} e^{mb_j}} \\
\tilde{p}_i(b_i, \mathbf{b}_{-i}) &= b_i - \frac{\sum_{j=1}^{n} e^{mb_j}}{me^{mb_i}} \left(\ln \frac{\sum_{j=1}^{n} e^{mb_j}}{1 + \sum_{j \neq i} e^{mb_j}} + \frac{1}{2} hmb_i^2 \left(\frac{\sum_{j \neq i} b_j^2}{c_\rho (n-1)} - 1 \right) \right) \\
\tilde{r}(\mathbf{b}) &= \frac{1}{2} h \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho (n-1)} \right)
\end{aligned}
$$

Mechanism 2. (We assume $n \geq 2$. If $n = 0$ then nobody can be confirmed and there is no payment, and if $n = 1$ then we set $(a, \tilde{p}, \tilde{r}) = (1, 0, 0)$.)

By Lemma 4.2 and Theorem 4.2, we have that the TFM $\tilde{M}$ is U-BNIC and 1-SCP for all $h \in [0, +\infty)$. We can also compute the expected miner revenue as follows.

$$\mathbb{E}_{\mathbf{b} \sim V}[\tilde{r}(\mathbf{b})] = \frac{1}{4} hnc_\rho > 0. \tag{4.21}$$

From Eq. (4.21), we see that the expected miner revenue is always positive and it grows linearly with our scaling parameter h.

However, we have to be careful about choosing the value of h. Intuitively, the value of h describes the extent of perturbation from the original U-DSIC mechanism, and when the perturbation is too large, the *individual rationality* ($p_i(b_i, \mathbf{b}_{-i}) \leq b_i$) and *budget feasibility* properties may not hold. Actually, since the block size is 1, the payment cannot exceed the valuation of the accepted bid. Then we have $\mathbb{E}_{\mathbf{b} \sim V}[\tilde{r}(\mathbf{b})] \leq 1$. Therefore, we have the following natural upper bound for h:

$$h \leq O(1/(c_\rho n)). \tag{4.22}$$

For the best miner revenue, we want to make h as large as possible while keeping the mechanism feasible. Fortunately, for fixed c_ρ, we have an estimation of optimally feasible h that enables a constant approximation ratio of the optimal revenue while preserving UIR

and BF constraints. Hence, in the setting of block size 1, we can design a TFM that satisfies desirable incentive properties and has a constant fraction of optimal revenue. The formal result is:

Theorem 4.3. *For $n \geq 2$, we consider Mechanism 2 with parameter $m = 1$. Then for any $h \in [0, \frac{2c_\rho(n-1)^2}{en^3}]$, the corresponding mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ is U-BNIC, 1-SCP, UIR and BF.*

Furthermore, for any $C \in [0, 1)$, the mechanism is $(C, \frac{6C+5}{1-C^2})$-U-SP.

The proof of Theorem 4.3 is deferred to Appendix B.4.3. Note that when we set $h = \frac{2c_\rho(n-1)^2}{en^3}$, the expected miner revenue is $\frac{1}{4}hnc_\rho = \frac{c_\rho^2(n-1)^2}{2en^2} = \Theta(c_\rho^2)$. As the optimal miner revenue is at most $\max\{v_i\} \leq 1$, for fixed distribution (fixed c_ρ), our mechanism yields a constant-ratio approximation of the optimal miner revenue for $n \to \infty$. Particularly, for the case of uniform distribution with $c_\rho = \frac{1}{3}$ and $n \to \infty$, our approximation ratio is $\frac{1}{18e}$. Furthermore, the U-SP results show that for large n, a user cannot gain more utility unless she injects as many fake transactions as the total amount of honest transactions competing for the block, which is unrealistic for the real-world blockchain ecosystem. This result also matches with a basic concept of blockchains: *the security of a blockchain system is fundamentally based on the assumption that (at least) 50% of participants are honest.*

4.6 Mechanism for General Block Size k

In most blockchains, a block usually contains multiple transactions. Therefore, it is desirable to extend our Mechanism 2 to general block size k. Recall that when the block size is 1, we adopted a simple soft second-price mechanism as the auxiliary. However, it seems trickier to extend this auxiliary to a general block size k, as the softmax function does not have a straightforward extension for soft-top-k. In Section 4.6.1, we will work on the details of the auxiliary mechanism for general block size k. The high-level idea of this step is natural – we adopt a k-step weighted sampling without replacement approach [75] to confirm the k bids in a block, where in each step, we still apply the logit choice rule.

Once we figure out the details of the allocation rule $\mathbf{a}$ for general block size k, the rest

construction will follow our auxiliary mechanism – We first straightforwardly compute its dominant auxiliary mechanism $M = (\mathbf{a}, \mathbf{p}, 0)$ where the corresponding dominant association $\mathbf{p}$ is defined by Eq. (4.7). Then, we will still use the variation term T defined previously in Section 4.5.2, and combine the auxiliary and the variation term to derive the final mechanism.

In Section 4.6.2, we will show that the generalized mechanism enjoys the same incentive compatibility properties as the basic version (for block size 1). We will also analyze the expected miner revenue of the generalized mechanism.

4.6.1 Allocation Rule: Weighted Sampling without Replacement

In this section, we assume the bidding vector $\mathbf{b}$ and block size k are fixed. For bidder $i \in B = [n]$, we set her weight $w_i = e^{mb_i}$. Now we compute a_i, the probability user i has her transaction confirmed.

Denote $\delta_t(i)$ as the probability that user i in the t-th round and $W = \sum_{i=1}^{n} w_i$, then $\delta_1(i) = \frac{w_i}{W}$. For fixed $t \geq 2$, we consider $j = (j_1, \cdots, j_t)$ as the *sampling vector* describing the outcome of the weighted sampling without replacement in the first t rounds, in which j_s is the user confirmed in the s-th round, and denote $\mathscr{J}$ as the distribution of j. Therefore, we get $\delta_t(i; b_i, \mathbf{b}_{-i}) = \Pr_{j\sim\mathscr{J}}[j_t = i]$.

We use the notation $\delta_t(i) = \delta_t(i; b_i, \mathbf{b}_{-i})$ when $(b_i, \mathbf{b}_{-i})$ is fixed, and denote $J_t(i) = \{j \text{ is a sampling vector} : j_t = i\}$, then $\forall j \in J_t(i)$, denote $\delta_t(i; j) = \Pr_{u\sim\mathscr{J}}[u = j]$, then we have

$$\delta_t(i) = \sum_{j \in J_t(i)} \delta_t(i; j). \tag{4.23}$$

Note that $\delta_t(i; j)$ denotes the probability that the sampling outcome is $(j_1, j_2, \cdots, j_{t-1}, i)$, thus the probability is

$$\delta_t(i; j) = \frac{w_{j_1}}{W} \cdot \frac{w_{j_2}}{W - w_{j_1}} \cdot \cdots \cdot \frac{w_i}{W - w_{j_1} - \cdots - w_{j_{t-1}}}. \tag{4.24}$$

Since we know that

$$a_i = \sum_{t=1}^{k} \delta_t(i), \tag{4.25}$$

the allocation rule $\mathbf{a}$ can be computed from Eqs. (4.23-4.25). According to Eq. (4.7), we compute the corresponding dominant association payment rule $\tilde{\mathbf{p}}$. We use the same variation term T as in Section 4.5.2. Thus, the final TFM can be described as in Mechanism 3 (and we note that Mechanism 2 exactly the same as Mechanism 3 when $k = 1$). Also note that every entry of our constructed allocation rule $\mathbf{a}$ is always positive, and therefore the mechanism is well defined.

Mechanism 3 Transaction Fee Mechanism for general block size k

$$a_i(b_i, \mathbf{b}_{-i}) = \sum_{t=1}^{k} \delta_t(i; b_i, \mathbf{b}_{-i})$$

$$\tilde{p}_i(b_i, \mathbf{b}_{-i}) = \frac{1}{a_i(b_i, \mathbf{b}_{-i})} \left[a_i(b_i, \mathbf{b}_{-i}) b_i - \int_0^{b_i} a_i(t, \mathbf{b}_{-i}) dt - \frac{1}{2} h b_i^2 \left(\frac{\sum_{j \neq i} b_j^2}{c_\rho(n-1)} - 1 \right) \right]$$

$$\tilde{r}(\mathbf{b}) = \frac{1}{2} h \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho(n-1)} \right)$$

4.6.2 Estimation of h: How Much Revenue Can Miner Get?

For the case of general block size k, we also have a similar result on the value of h, which additionally requires the number of users n to be at least $\left(\frac{e}{e-1} + \Theta(1)\right) k \approx 1.582k$. Hence, for general block size k, as long as $n \geq 1.582k$, we can still design a TFM that satisfies desirable incentive properties with a constant fraction of optimal revenue. The formal result is shown below:

Theorem 4.4 (Main Theorem)**.** *For any block size k and any parameters $m, h \in [0, +\infty)$, the mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ defined in Mechanism 3 is U-BNIC and 1-SCP. For any fixed $\lambda_0 > \frac{e}{e-1} \approx 1.582$, and any*

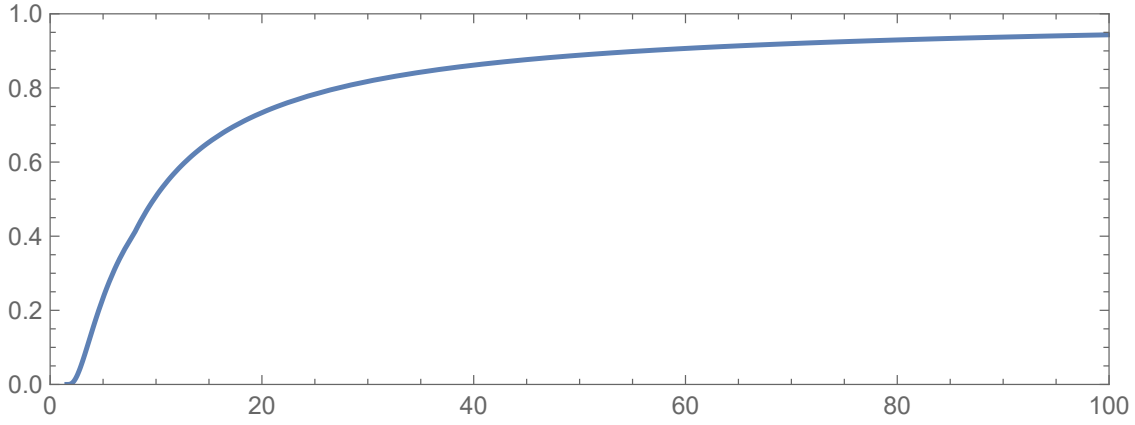

Figure 4.3: The plot of $g(\lambda_0)$ in Theorem 4.4.

$n \geq \max\{\lambda_0 k, 30\}$, *if we set* $m = \min\left\{\frac{1}{2}\ln\frac{1}{\ln\frac{\lambda_0}{\lambda_0-1}}, 1\right\}$, *then for every*

$$h \in \left[0, g(\lambda_0)\frac{2kc_\rho(n-1)}{en^2}\right],$$

where

$$g(\lambda_0) = \frac{m \cdot \max\left\{1-\sqrt{\ln\frac{\lambda_0}{\lambda_0-1}}, 1-e\ln\frac{\lambda_0}{\lambda_0-1}\right\}}{e^{m-1+\frac{e}{0.9\lambda_0-1}}},$$

then our Mechanism 3 $\tilde{M}$ *is UIR and BF.*

Furthermore, for any $C \in [0,1)$, *the mechanism is* $\left(C, O(\frac{1}{1-C})\right)$*-U-SP.*

The proof of Theorem 4.4 is technically complicated and deferred to Appendix B.4.4. We plot $g(\lambda_0)$ in Figure 4.3 and it holds that $\lim_{\lambda_0\to+\infty} g(\lambda_0) = 1$. Combined with $\mathbb{E}[r(\tilde{\mathbf{b}})] = \frac{1}{4}hnc_\rho$, the expected miner revenue is $\Theta(g(\lambda_0)c_\rho^2 k)$. Because the optimal revenue for block size k is at most $k\max\{v_i\} \leq k$, for fixed $c_\rho > 0$ and $\lambda_0 > \frac{e}{e-1}$, our mechanism yields a constant-factor approximation of the optimal revenue as long as $n > \max\{\lambda_0 k, 30\}$. While Mechanism 2 is essentially a special case of Mechanism 3 with $k = 1$, we can also notice that when $k = 1$ and $n \to \infty$, the range of h in Theorem 4.4 is $[0, (1-o(1))\frac{2c_\rho}{en}]$, matching with the result of Theorem 4.3.

4.7 Additional Properties of Our Mechanism

In this section, we discuss the incentive and revenue properties for the miner in our proposed TFM. In Section 4.7.1, we show that our proposed TFM is almost miner incentive compatible (MIC) and it is impossible to achieve the strict MIC property for any TFM. In Section 4.7.2 we show that although the miner may get negative revenue in our TFM, it is only of a negligible probability and the miner will actually get a stable revenue close to the expectation. Therefore, our TFM can satisfy the miner's expectation on stable mining rewards in practice.

4.7.1 Almost Miner Incentive Compatibility

In the previous parts of our paper, we mainly focused on the prevention of an individual user's deviations and the miner-and-single-user collusion (i.e., the deviation set {`U-UB`, `U-FT`, `MU`^{1}`-UB`} in Table 4.1). We now show that our TFM is also able to greatly reduce the additional miner utility derived from injecting and deleting (a limited number of) bids, and therefore achieving an *almost MIC* property.

In particular, let us first fix the block size k and the number of users n. Since the injection-and-deletion deviation may change the n parameter presented to the mechanism, we have to consider a variable-bid-size TFM $\tilde{\mathscr{M}}$ (as in Definition 7'). To construct $\tilde{\mathscr{M}}$, we follow the method described above Eq. (4.1) – we first choose a parameter $m > 0$; for each integer $\eta \geq 0$, we also choose a parameter $h_\eta > 0$ so that $\tilde{M}_\eta$ is fully determined following the description of Mechanism 3. In order to establish our almost MIC property, we need to choose $\{h_\eta\}$ in a way so that there exists L satisfying

$$L = \frac{h_\eta \eta}{c_\rho k} \qquad \forall \eta. \tag{4.26}$$

Note that it is possible to appropriately set $\{h_\eta\}$ to meet the above condition while each $\tilde{M}_\eta$ also satisfies the conditions in Theorem 4.4 (for $\eta > \lambda_0 k$, where λ_0 is defined in the theorem statement). We finally let $\mathscr{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r}) = \cup_\eta \{\tilde{M}_\eta\}$ as formally defined in Eq. (4.1).

We also need to characterize the degree of injection-and-deletion deviation by the miner. For any positive integer Δ, we denote $B_\Delta(\mathbf{b})$ as the set of all bidding vectors generated

via injecting and deleting a total of at most Δ transactions to/from $\mathbf{b}$. Given the original bidding vector $\mathbf{b}$, for any $\mathbf{b}' \in B_\Delta(\mathbf{b})$ that could result from the miner's injection-and-deletion deviation, we note that change to the miner's utility consists of the following two parts:

1. the change of the miner's reward: $\tilde{r}(\mathbf{b}') - \tilde{r}(\mathbf{b})$;
2. the cost for the miner to inject fake bids: $\sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} \tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0) = -\sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} a_i(b'_j, \mathbf{b}'_{-j}) \cdot \tilde{p}_j(b'_j, \mathbf{b}'_{-j})$, where we extend the definition in Eq. (4.2) to our TFM sequence $\tilde{\mathscr{M}} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$.

Note that the miner does not need to directly pay any cost for deleting a bid, but her reward $\tilde{r}(\cdot)$ may be changed due to this deletion.

We first define the (strict) Miner-Incentive-Compatibility (MIC) notion as follows:

Definition 15 (Miner-Incentive-Compatibility (MIC))**.** *Suppose there are n real users and the block size is k. A variable-bid-size TFM $\tilde{\mathscr{M}} = \cup_\eta \{\tilde{M}_\eta\} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ satisfies the MIC property if and only if for any bidding vector $\mathbf{b}$ and $\Delta \geq 1$, we have*

$$\sup_{\mathbf{b}' \in B_\Delta(\mathbf{b})} \left(\tilde{r}(\mathbf{b}') - \tilde{r}(\mathbf{b}) + \sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} \tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0) \right) \leq 0.$$

Relaxing from the strict MIC notion, are now able to introduce the almost MIC property for our TFM $\tilde{\mathscr{M}}$. We show that for if $\Delta = o(n)$ (i.e., the number of injected and deleted bids is a tiny fraction of the total number of users), then, as $n \to \infty$, with overwhelming probability, the additional utility for the miner is also $o(1)$. Particularly, by injecting or deleting Δ transactions when the total number of honest users is n, with high probability among the distribution of bids, the miner cannot gain an increase of revenue above $O(\frac{k\Delta^{4/3}}{n^{4/3}})$. Hence, for example, if the miner can inject a constant number of fake transactions (without being caught, as discussed in Appendix B.1), the additional revenue she may get is $O(\frac{k}{n^{4/3}})$ which is negligible when n is large. Formally, we have the following theorem.

Theorem 4.5 (Our TFM is Almost MIC)**.** *Suppose there are n real users and the block size is k so that $n > \lambda_0 k + \Delta$. Let the variable-bid-size TFM $\tilde{\mathscr{M}} = \cup_\eta \{\tilde{M}_\eta\} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ be*

defined above with the L parameter satisfying Eq. (4.26)*. There exist universal constants $C_{M0}, C_{M1}, C_{M2} > 0$ such that for any $\epsilon \in (0, \frac{1}{2})$, if $n \geq \max\left\{C_{M1}\Delta, C_{M2}\log^3 \frac{1}{\epsilon}\right\}$, we have that*

$$\Pr_{\mathbf{b}\sim V}\left[\sup_{\mathbf{b}'\in B_\Delta(\mathbf{b})}\left(\tilde{r}(\mathbf{b}') - \tilde{r}(\mathbf{b}) + \sum_{b'_j\in\mathbf{b}'\setminus\mathbf{b}} \tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0)\right) > C_{M0}L\cdot\frac{k\Delta^{4/3}}{n^{4/3}}\right] < \epsilon. \tag{4.27}$$

In other words, Theorem 4.5 states that given a moderately large n, with probability at least $(1-\epsilon)$ (over the realization of the n real user valuations), the additional miner utility that could be gained from injecting and deleting at most Δ bids is at most $C_{M0}L\cdot\frac{k\Delta^{4/3}}{n^{4/3}}$, which is $o(1)$ for $\Delta = o(n)$ and fixed k, L. This result is quite non-trivial as one would naturally expect the relative revenue advantage should be $\Theta(\Delta/n)$ for usual (incentive compatible) mechanisms. For example, in the k-item second-price auction with valuation distribution Unif$[0,1]$, the expected $(k+1)$-th price is $\frac{n-k}{n+1}$, but if the miner injects a fake bid to be infinitesimally lower than the k-th bid, the expected price increases to $\frac{n-k+1}{n+1}$, gaining an $\Theta(1/n)$ relative advantage via injecting one bid. It is also more useful since it gives additional incentive restrictions to the miner when we narrow the range of "acceptable deviations" for the miner. Here, the notion of the acceptable deviation is the range of small Δ (compared to n), as a large amount of injected or missing transactions (greater than the acceptable threshold) can be detected by the blockchain system via cryptographic schemes and the miner would be penalized for injection/deletion deviation. Please refer to Section B.1 for more details. The proof of Theorem 4.5 is deferred to Appendix B.4.5.

On the other hand, we can show that achieving strict MIC together with the main strategy-proof properties studied in this paper (U-BNIC and 1-SCP) is impossible if we additionally assume the natural NFL condition defined in Section 4.3.4 (Shi et al. [44] have independently proven a similar result). In particular, we prove that any TFM satisfying the above-mentioned properties has non-positive expected miner revenue, even if we only allow the miner to inject one zero-bidding fake transaction in the MIC property. Formally, we have the following theorem.

Theorem 4.6 (Impossibility of MIC)**.** *Suppose there are n real users and the block size is k.*

Consider any variable-bid-size TFM $\tilde{\mathscr{M}} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$. Assume that for any $\eta \in \{1, 2, \ldots, n+1\}$, the natural restriction (formally defined in Section 4.3.2) of $\mathscr{M}$ to a regular TFM for η bids is U-BNIC and 1-SCP, and NFL. If $\tilde{r}(\mathbf{b}, 0) - \tilde{r}(\mathbf{b}) \leq 0$ holds for all $\mathbf{b} \in [0,1]^0 \cup [0,1]^1 \cup \cdots \cup [0,1]^n$, then we have that $\mathbb{E}_{\mathbf{b} \sim V}[\tilde{r}(\mathbf{b})] \leq 0$.

The proof of Theorem 4.6 is deferred to Appendix B.4.6. Nevertheless, if we do not allow the miner to inject fake transactions, but allow her to delete existing transactions, whether there exists a TFM that is U-BNIC, 1-SCP and {M-TD}-proof remains open.

4.7.2 Almost Miner Individual Rationality and Stability of Miner Revenue

In previous sections, we have discussed the *expected* miner revenue, which is in general a constant-fraction approximation of optimum, which naturally implies interim Miner Individual Rationality (MIR). In this section, we will be concerned about the guarantee on the *worst-case* miner revenue. We define the (ex-post) MIR as a worst-case specification for the miner, which requires that the miner revenue is always non-negative no matter how the users bid. Formally,

Definition 16 ((Ex-Post) Miner Individual Rationality)**.** *Suppose there are n users and the block size is k. A TFM $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ satisfies the Miner Individual Rationality (MIR) property if and only if $\tilde{r}(\mathbf{b}) \geq 0$ holds for all $\mathbf{b} \in [0,1]^n$.*

The MIR condition requires that the miner revenue is non-negative for *every* realization of the bidding vector $\mathbf{b}$. Unfortunately, our Mechanism 3 do not satisfy such a strong condition. Recall that, in Mechanism 3, we have $\tilde{r}(\mathbf{b}) = \frac{1}{2} h \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho (n-1)} \right)$ where $h > 0$. For the particular bidding vector $\mathbf{b} = (1, 1, \ldots, 1) \in [0,1]^n$, we have that $\tilde{r}(\mathbf{b}) = \frac{h}{2} \cdot \left(n - \frac{n(n-1)/2}{c_\rho (n-1)} \right) = \frac{hn}{2} \cdot \left(1 - \frac{1}{2c_\rho} \right)$.

We now observe that, for the user valuation distribution V_0 with $c_\rho \in (0, 1/2)$ (for example, the uniform distribution over $[0,1]$ leads to $c_\rho = 1/3$), our Mechanism 3 is not MIR. In fact, we have the following sufficient and necessary condition of when our mechanism is MIR, the proof of which is deferred to Appendix B.4.7.

Theorem 4.7. *Our Mechanism 3 is MIR if and only if $c_\rho \geq \frac{1}{2}$.*

Nevertheless, even when $c_\rho < 1/2$, so long as it is not too small, we are able to show that the miner gets a non-negative revenue with overwhelming probability, and the probability that the miner is "not lucky" to receive a negative revenue diminishes exponentially as the growth of n. Formally, we have the following theorem.

Theorem 4.8 (Concentration of Miner Revenue)**.** *Fix n to be the number of users. For any block size k, let the TFM $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ be defined according to Mechanism 3. Assume that each user's bid follows the* i.i.d. *distribution V_0 and let c_ρ be correspondingly defined according to Eq.* (4.18)*. For any $\lambda > 0$, we have that* $\Pr\left[\frac{\tilde{r}(\mathbf{b})}{\mathbb{E}[\tilde{r}(\mathbf{b})]} \leq 1 - \frac{\lambda}{c_\rho^2 n}\right] \leq 2\exp(-\lambda)$.

The proof of Theorem 4.8 is deferred to Appendix B.4.8. Let $\lambda = c_\rho^2 n$, we immediate get the *almost-MIR* property of our mechanism:

Corollary 4.1 (Almost Ex-Post Miner Individual Rationality)**.** *Following the setup in Theorem 4.8, we have that* $\Pr[\tilde{r}(\mathbf{b}) < 0] \leq 2\exp(-c_\rho^2 n)$.

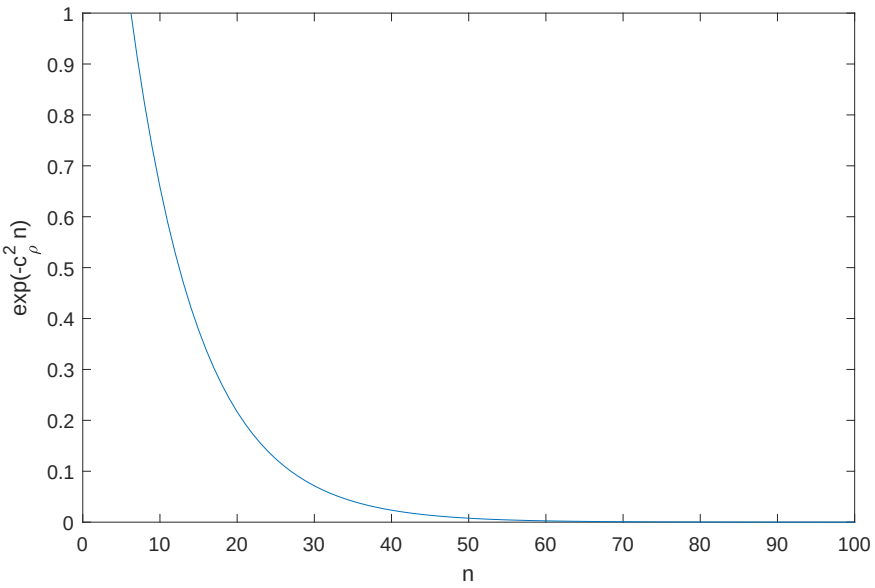

Figure 4.4: The diminishing probability of MIR being violated for the uniform distribution $b_i \sim \text{Unif}[0, 1]$.

From Corollary 4.1, we can see that the probability of $\tilde{r}(\mathbf{b}) < 0$ diminishes exponentially as n increases. As a demonstration, we show the plot of $\exp(-c_\rho^2 n)$ for the uniform distribution

$b_i \sim \text{Unif}[0,1]$ in which $c_\rho = \frac{1}{3}$. Considering the practical scenario that the Bitcoin and Ethereum blockchains typically have thousands of transactions in each block, in practice $\tilde{r}(\mathbf{b}) \geq 0$ would indeed hold with overwhelming probability.

Consideration of Mining Costs. In Section 4.3.3 we assume that the miner's utility is equal to the miner revenue and ignore the mining costs and block rewards. In actual cases, we denote r_0 as the block reward minus mining cost, and as long as $r_0 \geq 0$, the Corollary 4.1 is still valid. In case that $r_0 < 0$, from Theorem 4.8 we see that as long as $\mathbb{E}[\tilde{r}(\mathbf{b})] \geq (1+\Theta(1))|r_0|$, the probability that the miner gets a negative utility still diminishes exponentially with n.

4.8 Discussion

In this chapter, we model each transaction as constant-sized. However, in modern blockchain systems, transactions – especially with smart contracts in modern applications, have variant sizes. The knapsack auction problem [76] for the setting of blockchains is still open to future study.

Our mechanism mainly considers the valuation distributions of bounded supports. While our methodology may extend to distributions of unbounded supports as long as $c_\rho = \mathbb{E}[b_i^2]$ is finite, the bounded support assumption is necessary for our estimation of h and the revenue approximation guarantees. Whether it is possible to extend our approach to more general valuation distributions is also open to future study.

While we assume symmetry of the joint distribution of users' valuations, the actual distribution can still be correlated, which is not discussed in our work. Furthermore, in the real-world scenario, as blockchain users may value the blockchain space depending on the expectation of the market, they may even have interdependent valuations, as modeled by the work by Eden et al. [77]. Consideration of correlated or interdependent valuations in the design of blockchain TFMs can be a challenging but interesting future direction.

CHAPTER 5

POUW PROTOCOL DESIGN WITH AI MODEL TRAINING

5.1 Introduction

Blockchain, with prevailing examples as Bitcoin [78] and Ethereum [79], is an emerging technology that maintains decentralized consensus via a distributed ledger that utilizes cryptographic techniques to achieve trust and security. To prevent sybil attacks in the consensus mechanism, the earliest and most conventional way is Proof-of-Work (PoW) [80, 81, 82, 83] as Bitcoin uses: all "miners" attempt to solve a hash puzzle and the first miner getting a valid solution wins the access to the block.

However, the huge and inefficient use of energy and severe carbon footprint in the traditional PoW mechanism draws wide concern and is recognized as heavily controversial for the environmental impact of the blockchain system [84, 85]. Since May 2021, cryptocurrency mining and even cryptocurrency trading have been banned in China due to the ecological concern of energy inefficiency [86]. To address the energy issue, researchers propose alternative consensus mechanisms, e.g. Proof-of-Stake (PoS) [87, 88, 89] in order to substitute PoW, but they tend to have inherent drawbacks in security and centralization issues [90]. In the high-level view of economics, Piketty [91] argued that the phenomenon of $r>g$, i.e. the return rate on capital ("stake") being greater than the rate of economic growth ("work"), results in wealth concentration and social instability. Indeed, the heavy computation cost arguably binds the voting power with real-world productivity rather than intangible tokens. Were the computation made useful, the Proof-of-Useful-Work (PoUW) mechanism would indeed resolve the energy issue while preserving the decentralization and security of PoW [92, 93]. On the other hand, there are also positive views on the energy consumption of PoW mechanisms, e.g. the expansion of energy demand also motivates the development of

new energy solutions [94]. Since our PoUW mechanism essentially improves the efficiency of energy consumption instead of eliminating it, in contrast to PoS, our mechanism preserves this social benefit of PoW in the meantime of improving its sustainability.

In the age in which artificial intelligence (AI) has been becoming one of the most attractive topics in modern technology, researchers are actively attempting to incorporate machine learning tasks as PoUW challenges, i.e. Proof-of-Learning (PoL). As a consensus mechanism for the blockchain system, an ideal design of PoUW should satisfy the following properties:

1. **Security**: For the security and credibility of the blockchain system, an ideal PoUW mechanism should have theoretically provable security guarantees against dishonest behavior.

2. **Efficiency**: An ideal PoUW mechanism should have a low computational overhead (redundancy) for energy efficiency, as a main motivation of PoUW.

3. **Controllable Difficulty**[1]: As a stable block production time (BPT) is essential for the blockchain system's stability [95], an ideal PoUW mechanism should use challenges with predictable and controllable difficulty.

However, although there have been a series of PoL proposals in the literature (e.g., [96, 97, 98, 99]), as far as we are concerned, none of them could simultaneously satisfy the three properties above. Particularly, the methodologies of existing PoL mechanisms can be organized into two classes:

1. **Proof-of-Computation**: Proving that the training task is honestly done, e.g. [96];

2. **Proof-of-Performance**: Proving that the output model satisfies required accuracy on a test dataset, e.g. [97, 98, 99].

The difficulty of designing a desirable PoL mechanism is observed as follows. For Proof-of-Computation mechanisms, a recent work [100] shows the hardness of efficiently verifying the correctness of a Proof-of-Computation with provable security guarantees without a further

[1]The "difficulty" of a PoW challenge can be defined as the (expected) amount of computation needed to solve it.

theoretical understanding of deep learning — particularly, the work of Jia et al. [96] is subject to adversarial attacks [100, 101]. For Proof-of-Performance mechanisms, Hoffmann [92] argues that it is hard to evaluate the actual difficulty (even possibility) to achieve given accuracy, leading to a barrier to controllable difficulty. In summary of the existing PoL mechanisms, we observe a Trilemma of Proof-of-Learning as below:

> ***Trilemma of Proof-of-Learning***
>
> It is difficult to design a Proof-of-Learning mechanism that simultaneously satisfies **perfect security**, **efficiency** and **controllable difficulty**.

In this research, we are motivated to resolve the sustainability issue of blockchain systems via a Proof-of-Computation mechanism to machine learning model training, and tackle the trilemma via a delicate relaxation of the security notion. Instead of preventing all attacks from being conducted without getting detected (byzantine security), we aim to prevent the attacks from "being useful" with the *incentive-security* notion, i.e. an attacker cannot increase their utility via saving computational cost by cheating. Particularly, our mechanism in which the prover trains with designated random seeds and the verifier verifies random subsets of stages (as shown in Section 5.4.1-5.4.2) can prevent the attacks of [101] and [100] in the way as follows. From the stochastic nature of SGD, the verification protocol of [96] introduces a "tolerance" that allows small discrepancies in verification, which is exploited by these attacks. As our mechanism replaces the tolerance with designated random seeds[2], our mechanism is enabled to catch their exploits as "dishonest stages" successfully. Furthermore, our verification mechanism only has an $O(\frac{\log E}{E})$ relative computational overhead[3] for a total of E epochs with no staking requirement, or $O(\frac{1}{E})$ with a staking requirement comparable to the block reward, compared to $\Theta(1)$ in the work of Jia et al. [96]. For a model of size $|\mathscr{W}|$, we also improves the communication complexity from $\Theta(E|\mathscr{W}|)$ to $O(E + |\mathscr{W}| \log E)$

[2]Different types of machines or softwares may have different rounding behavior, but we can enforce high precision and set a tolerance low enough to prevent any "meaningful" attack.

[3]The ratio of computational power consumption in verification to computational power consumption in model training.

or $O(E + |\mathscr{W}|)$, respectively.

From another perspective, the recently rapid development of AI technologies also draws safety concerns on the trustworthiness of AI models [102, 103, 104, 105]. While studies on AI *alignment* (e.g., [106, 107, 108]) address the internal risks of *unrobust* AI models, attacks by malicious trainers via corrupting the training process may bypass the alignment measures. As a recent example, the adversarial attack on ByteDance LLM training by an intern, which leads to $1.1M loss [109], draws attention to the systematic security of AI model training. Compared to the Proof-of-Performance paradigm, our Proof-of-Computation mechanism offers additional practical value as a decentralized surveillance measure of AI model training. While the Proof-of-Performance mechanism is primarily motivated by the goal of improving the sustainability of blockchain PoW mining, thus ***improving blockchain with AI***, the Proof-of-Computation mechanism can also serve as a blockchain-based trustworthy AI platform, enhancing the security and credibility of machine learning, i.e. simultaneously ***securing AI with blockchain***.

Furthermore, while most recent research papers on PoUW explicitly or implicitly assume that problem providers are trusted — so that their proposed system is not completely decentralized, we are also motivated to consider *frontend incentive-security* against known-model and model-stealing attacks even when problem providers and provers are both untrusted, thus enabling full decentralization and more robustness of the system. (See discussion in Section 5.2.2)

Since the computational overhead of verification is low, our PoL protocol can be used for general applications in which the task provider would like to delegate the training/fine-tuning tasks for remote computation, as a Machine-Learning-as-a-Service (MLaaS) platform. Nevertheless, the functionality of verification makes the protocol particularly suitable for applications in which credibility of the model and/or training process is critical. Examples include AI grading [110, 111], where the transparency and accuracy of the grading model are essential for educational and hiring processes, and credit evaluation [112, 113], where the fairness and reliability of the model impact financial decisions. These applications benefit from PoL's verification mechanism, ensuring that the models are trained correctly and securely, thereby enhancing trust in their outputs.

In light of the security desiderata discussed above, in our paper, we propose an incentive-secure Proof-of-Learning mechanism with the following contributions consisting of:

1. With trusted verifiers (that are widely assumed in previous works), we propose our *interactive-proof*-based basic design satisfying computational efficiency, controllable difficulty, and incentive-security against dishonest provers for any stochastic optimization tasks, e.g. stochastic gradient descent (SGD), and also substantially improves the relative computational overhead of the previous work [96]. (Sections 5.3-5.4)

2. With untrusted verifiers, we propose a *capture-the-flag* protocol that preserves all desired properties in our basic design and additionally achieves incentive-security against dishonest verifiers. (Section 5.5)

3. We prove the theoretical incentive-security properties of our mechanisms. (Section 5.6)

Then, in Section 5.7, we perform experimental evaluations to show the performance of our mechanism on real-world ML tasks.

In Appendix C.4, we further discuss on potential augmentations of our mechanism to ensure model correctness against malicious attacks even from irrational attackers.

Rounds of interaction. Our basic mechanism needs one round of interaction between the prover and the verifier, and the full mechanism needs two rounds of interaction.

Limitation of our incentive model. While our novel modeling of *incentive-security* is a suitable relaxation both due to the Trilemma of Proof-of-Learning and the nature of blockchain systems whose security depends on economic incentives, our study focuses on the model of *individually* rational parties and does not consider collusions between the prover and the verifier. Nevertheless, the anonymity of blockchain reduces the risk of collusion due to the difficulty for the prover to predict or identify the identity of the verifier, and we would leave the expansion of more general incentive models with collusion-proofness for future study.

5.2 Background and Related Work

5.2.1 Proof-of-Useful-Work in Literature

The biggest concern of the traditional PoW mechanism is the computation, and essentially, energy consumption. As discussed by Chen et al. [114], the current energy consumption of the Bitcoin network is around 120TWh per year, comparable to a medium-sized country, but the consumption serves no social welfare apart from maintaining the security scheme, leading to severe social inefficiency. In recent years, the wasteful energy consumption of blockchains, particularly Bitcoin, has been widely criticized around the world. Particularly, Vranken [84] empirically discovered that the energy consumption of Bitcoin market is higher than its long-term benefit; Stoll et al. [85] also noticed the severe carbon footprint of Bitcoin for sustainability issues.

Aware of the energy and sustainability issues, previous research studied a wide variety of real-world problems that may serve as Proof-of-Useful-Work (PoUW) challenges. Hoffmann [92] surveyed the existing projects that incorporate number-theoretical, biological and machine learning problems into the PoUW mechanism. The survey shows a "more usefulness, more challenge" phenomenon in the existing works: while the Primecoin [115] has been the most developed and already deployed on chain, the number-theoretical problem may be of limited interest for the general public except mathematicians; the Coinami [116] proposes a solution to solve DNA sequencing problems for PoUW, but it needs a centralized authority and is not genuinely decentralized; the CoinAI [97] propose to develop a Proof-of-Learning system which uses the final performance as the certificate, but setting a reasonable "performance bar" to desired difficulty is a hard (if even possible) task.

In the specific area of Proof-of-Learning (PoL), Jia et al. [96] considered a setting of a specific *threat model*, and proposed a PoL mechanism to show that the verification of SGD training requires two types of parties as *provers* and *verifiers*. They aim to design a mechanism in which an honest certificate generated by the prover can be verified by the verifier at a low computational cost, while a dishonest certificate (spoof) *within the threat model* will be detected by the verifier at a low cost too. In their protocol, the provers report

the state every k epochs and the verifier checks the *largest updates*, arguing that within their threat model, the largest updates tend to be the most suspicious when the dishonest prover attempts to forge a fake certificate. However, when going beyond that specific threat model, Zhang et al. [101] showed that attackers can maliciously design spoofs that bypass the largest-update verification and exploit the tolerance. Furthermore, Fang et al. [100] claimed that the Proof-of-Learning "is more broken than you think" by demonstrating that realizing the desired security requirements reduces to solving hard open problems in learning theory, so that a provably Byzantine-secure PoL is not possible to design until significant progress in further understanding in deep learning.

In an economic view, the difficulty in designing a cheap but secure verification protocol of PoL is conceptually related to *Goodhart's Law*: "When a measure becomes a target, it ceases to be a good measure" [117]. Until further understanding of deep learning, no more efficient method has been found to verify the integrity of training than training it again. The work of Jia et al. [96], to reduce the computational overhead of the verification, chose to identify "most suspicious" parts to verify, but when the criteria for suspicion are deterministically designed, there would constantly be risks that cheaters adversarially design attacks to bypass the criteria. Therefore, designing an efficient method to deterministically (or with high probability) catch all cheats in PoL is indeed faced with major difficulties.

In contrast, our research relaxes the security requirement to "incentive-security" in a game-theoretical setting: we do not need to prevent all attacks, but only need to prevent attacks from being "worthy". Intuitively, while all attacks are considered equal in Byzantine security, they may have different degrees of effects in the economic view. In our design, our mechanism detects attacks in a stochastic way and "more severe" attacks that potentially benefit the attackers more, would be caught with higher chances and lead to heavier expected penalties. In this way, our incentive-secure PoL design can manage to disincentivize rational agents from cheating.

Another difference between the settings of Jia et al. [96] and our work is that: while the work of Jia et al. [96] mainly aims to prevent the spoof of a *specific* PoL to protect the copyright of the model, we aim to prevent all spoofs that try to cheat the verifier and claim that the training is correctly done, getting the training reward. Hence, while our work

adopts a relaxed notion of incentive security, it generally applies to a wider range of attacks (details discussed in Section 5.3.4).

5.2.2 Settings of Trusted or Untrusted Problem Providers in PoUW Protocols

	Cryptographic	Game-theoretic (existing)	Ours
Approach	Zero-knowledge Proofs	Verification Games	Verification Games
Example	zkML	opML, PoSP	Incentive-Secure PoL
Security	Cryptographic	Mixed-Strategy Nash Eq. (with few cheaters)	Pure-Strategy Nash Eq. (with no cheater)
Overhead	High ($\geq$ 1000x)	Moderate ($\geq$ 1x)	Low ($\lesssim$ 0.1x)
Challenges	High overhead, low scalability	Verifier's Dilemma	Communication cost (for extremely large models)

Table 5.1: Comparison of Trustworthy AI Protocols on Blockchain

In the traditional PoW mechanism, e.g. in Bitcoin, the hash puzzle is automatically generated from the previous block and is unpredictable before the previous block is confirmed. However, in the paradigm of PoUW, the problem should come from real-world providers, so can be indeed predictable or even controllable. In particular, malicious parties can conduct the following attacks:

- Known-model attack: submit a problem to which they already have a solution, and then submit the solution to claim the block.
- Model-stealing attack: submit a model trained by others (or based on it) and claim that they trained it on their own.

As far as we are concerned, most research in the literature of PoUW has not considered the credibility of the problems, i.e. implicitly assumed that the problems are *credible* and focus on the prevention of spurious certificates. Besides, Coinami [116] extensively discussed their system structure that depends on authority nodes and stated that their system is "not completely decentralized" and argued that it is necessary for usefulness; while the work

of [96] did not consider known-model attack, their solution to model-stealing attack is a chain-of-trust protocol that also relies on a sort of authorization.

Nevertheless, to build a robust blockchain system, we are motivated to design a mechanism in which both problem providers and provers can be *untrusted* but are incentivized to behave honestly, which we call *frontend-secure*. In consideration of frontend-security, Ball et al. [118] proposed a PoUW mechanism based on Orthogonal Vectors that adds an extra randomization layer to the PoUW challenge: instead of only requiring the prover to solve the problem, it requires the prover to solve the problem "in the way the system (randomly) specifies", so that even if the prover has a solution beforehand, the transcript may not meet the requirement of the challenge and the prover still has to compute the challenge again to pass the verification. The protocol works as follows:

- The system receives the problem A from an untrusted problem provider.
- The system generates a random seed ϕ and transform A to a PoUW challenge $C = \mathscr{C}(A, \phi)$.
- The prover solves the challenge and gets a certificate $c = S(C)$.
- The verifier verifies the certificate, expecting to get $V(C, c) = true$.
- The system recovers the solution $w = \mathscr{W}(C, \phi)$ and sends it to the problem provider.

On a high level, the frontend-security of the proposal is based on the one-way reduction from C to A: it is easy to generate a solution to A from a solution to C, but not in the inverse direction. While our design is generally different from this work, we indeed adopt the thought to introduce randomization in the design of PoUW challenges, which is naturally implementable due to the stochastic nature of the training of deep learning models.

5.2.3 Trustworthy AI and MLaaS on the Blockchain Platform

While the artificial intelligence (AI) has been becoming one of the most attractive topic in research and industry, the expansion of model sizes and computing source consumption

in machine learning tasks has raised significant concerns about security [119, 120] and sustainability [121]. The advent of Machine Learning as a Service (MLaaS) [122] has democratized access to powerful AI tools, enabling companies and individuals to integrate advanced machine learning models into their operations without extensive infrastructure.

However, this convenience comes with challenges in ensuring the transparency [123, 124] and security [125] of these services. Trustworthy AI principles are crucial in this context, as they advocate for the development and deployment of AI systems that are secure and accountable [126].

The blockchain, as a decentralized and transparent infrastructure, has an inherent affinity for applications in trustworthy AI [127]. Furthermore, the innate element of cryptocurrency tokens can also serve as economic incentives for participation [128].

Three recent methodologies that implement trustworthy AI in the blockchain platform are zero-knowledge machine learning (zkML) [129], optimistic machine learning (opML) [130] and Proof-of-Sampling (PoSP) [131]. The method of zkML utilizes the tool of zero-knowledge proof to secure the integrity of inference, but the nature of zero-knowledge proof makes the protocol extremely inefficient. The methods of opML and PoSP adopt economic incentives in the protocol and reduce the computational overheads to one or a few additional passes of computation, but opML effectively addresses the Verifier's Dilemma to prevent verifiers from being lazy when the fraction of dishonest provers is *arbitrarily* low[4], and the small challenging probability of PoSP leads to high staking requirements of verifiers and low detection probabilities of cheats, which undermine the user-friendliness and robustness of the protocol. In comparison, our mechanism has a computational overhead as low as a small fraction of one training pass, and it utilizes the capture-the-flag protocol to bypass the Verifier's Dilemma (See Section 5.5.1 and Theorem 5.1) and prevent lazy verifiers robustly when there are arbitrarily few or no cheating provers. We show the comparison of the related protocols in Table 5.1.

Hence, the family of Proof-of-Learning mechanisms, especially in the paradigm of Proof-of-Computation, not only serves as a fundamental mechanism to maintain the reliability of

[4]It utilizes constant penalty that works when the fraction ϵ of dishonest provers is at least a small constant, but does not work uniformly when $\epsilon \to 0$.

blockchain systems but also has the potential for the development of low-overhead decentralized computing power markets.

5.3 Preliminaries

In the Proof-of-Learning mechanism, we consider a situation where a prover tries to convince all parties via a "certificate" that she has honestly completed the training task and is thus eligible to claim the block reward; the verifier, in turn, is expected to verify the validity of the certificate to ensure the security of the system. In general, our protocol works as follows:

1. A PoL problem A is assigned.
2. One or more provers work on the problem A, either honestly or dishonestly, until one prover claims to have solved the problem and posts the PoL certificate c, winning the competition; other provers lose the competition and have their computing efforts lost as a sunk cost.
3. The verifier verifies the certificate c, possibly via interactions with the prover, and reports the verification result.
4. The system processes rewards and penalties accordingly.

In the rest of this section, we briefly discuss the basic components of the protocol.

5.3.1 Modeling of ML Training Tasks

Suppose there is a data distribution $\mathscr{D}$ in the form of $\mathscr{X} \times \mathscr{Y}$, in which $\mathscr{X}$ is the input space and $\mathscr{Y}$ is the output space. A machine learning model (abbreviated as "model") is a function $f : \mathscr{W} \times \mathscr{X} \to \mathscr{Y}$ in which $\mathscr{W}$ is the parameter space. In the ML practice, the parameters are commonly called *weights*.

The ML training task can be modeled as *empirical risk minimization*, in which a training dataset is sampled from the distribution as $D_{tr} \sim \mathscr{D}^n$, and we denote $D_{tr} = (d_1, \cdots, d_n)$ in which $d_i = (x_i, y_i)$. For any data point (x, y) and weight $w \in \mathscr{W}$, the model prediction is

$f(w, x)$, and the loss is defined as a *loss function* $\mathscr{L}(f(w, x), y)$. Then, the empirical risk to minimize is defined as:

$$\hat{L}(w) = \sum_{i \in [n]} \mathscr{L}(f(w, x_i), y_i). \tag{5.1}$$

The stochastic gradient descent (SGD) training process consists of a number E of *epochs*, and every epoch corresponds to one full pass of the training set. In each epoch $e \in [E]$, the training set is randomly divided into l batches of size m, with $n = l \cdot m$. In every step $s = (e-1)m + j$, the corresponding batch, denoted as a subset $b_e(j)$ of $[n]$, is processed, and the weight is updated as:

$$w_s = T_{\eta, b_e(j)}(w_{s-1}) = w_{s-1} - \eta \cdot \nabla \hat{L}_{b_e(j)}(w_{s-1}). \tag{5.2}$$

Here, η is a hyper-parameter of learning rate and $\hat{L}_{b_e(j)}$ is the empirical risk on the batch $b_e(j)$, defined as:

$$\hat{L}_{b_e(j)}(w) = \sum_{i \in b_e(j)} \mathscr{L}(f(w, x_i), y_i). \tag{5.3}$$

Therefore, given the batch division as $b_e \in \mathscr{B}$, the training process of epoch e can be formulated as a mapping $\mathscr{T}_\eta : \mathscr{B} \times \mathscr{W} \to \mathscr{W}$, with

$$\mathscr{T}_\eta(b_e, w) = T_{\eta, b_e(m)}(T_{\eta, b_e(m-1)}(\ \cdots\ T_{\eta, b_e(1)}(w)\ \cdots\ )). \tag{5.4}$$

In the rest of this paper, we regard η as a fixed hyper-parameter and denote $\mathscr{T}_\eta$ as $\mathscr{T}$ for simplicity.

5.3.2 Credible (Pseudo-)Randomness Generator

As described above, due to the random choice of batches $\{b_e(j)\}$, the training process $\mathscr{T}$ of ***stochastic*** gradient descent, is innately a ***stochastic process***. To verify the correctness of the training process, the paper of Jia et al. [96] leverages the concentration properties of the process and introduces *tolerance* for slight discrepancies in verification. However, the tolerance can, in turn, be exploited for adversarial attacks (See in [101]).

In Bitcoin, the randomness in the hash puzzle is essentially based on a pseudo-randomness generator (cryptographic hash) seeded with the last block, so that every party can have a consensus on the same pseudo-random PoW challenge.

A typical pseudo-randomness generator (PRG) works as follows. Given a random seed ϕ, the PRG generates a sequence of $r_\phi(1), r_\phi(2), \cdots$, and without loss of generality we assume they are uniformly distributed in $[0, 1)$. Since the PRG is typically based on a finite state machine, the sequence will eventually repeat after a period. Nevertheless, a "good" PRG would have a period long enough and pass certain randomness tests, and a PRG that meets the cryptographical criteria is called "cryptographically secure" [132].

In this paper, we would perform the SGD training with $\{b_e\}$ generated from a cryptographically secure PRG with seeds generated from the previous block, so that the prover and verifier would run with the same pseudo-random sequences and get exactly the same result for the same epoch. On the other hand, as the sequence is not predictable until the seed ϕ is generated, even if a strategic party submits a task with a known model and training process, as the protocol requires the prover to train with the given random seed, the prepared model or training process would not pass the verification and she still has to train it again to claim the reward.

5.3.3 Modeling of Prover's Incentive

For a fixed prover and a fixed task, we can assume the computational cost to honestly train an epoch is a deterministic constant m, and thus honestly training the task has a cost (aka. "difficulty") of $M = m \cdot E > 0$, which can be dynamically adjusted by adjustment of E. For each epoch, the prover may train it honestly or dishonestly (detailed discussion in Section 5.6). When dishonestly training an epoch, the prover may pay a significantly lower computational cost, and we assume it to be 0. We assume that dishonest training of one epoch does not affect the computational cost of further epochs. Therefore, if we honestly train a ρ portion of all epochs, the computational cost is (lower bounded by) ρM.

There can be competition among provers (or not, due to the allocation rule of the tasks) and only the first prover who submits a certificate wins, so if a prover does more honest

computation and consumes more time before submission, her probability of winning the competition does not increase. We define $P : [0, 1] \to (0, 1]$ as a non-increasing function that characterizes the competition: if the prover computes ρ portion of the task (i.e. ρE epochs) honestly, then she has a $P(\rho)$ probability of winning, in which $P(0) = 1$. If there is no competition, we just let $P(x) \equiv 1$.

When the prover wins the competition and submits her certificate, if $\rho < 1$, i.e., the prover does not act honestly, then there is a chance that she is caught. For any fixed ρ, as the prover may have multiple strategies to choose the $(1 - \rho)$ portion for cheating, we denote $Q(\rho)$ as the maximal probability among all such cheating strategies of passing the verification, in which we assume $Q(\cdot)$ is monotonic non-decreasing and $Q(1) = 1$. If passing the verification, the prover gets a reward of R at a computational cost of ρM, and the net utility is $R - \rho M$; if getting caught cheating, she will be penalized for γR, and the net utility is $-(\gamma R + \rho M)$. For a good PoL mechanism, we expect a low γ, ideally zero, to lower the staking requirement[5] and improve the convenience of participation.

If the prover loses the competition, the sunk cost in training the model is still paid, but she may find out that the task has been completed by another prover before she completes the computation, so the cost can be less than ρM. Hence, we denote her expected utility conditioned on losing as $-\mu(\rho) \in [-\rho M, 0]$. Assuming $P(\cdot)$ is a differentiable function, we can compute that (details in Appendix C.1):

$$\mu(\rho) = \frac{\int_0^\rho P(x)dx - \rho P(\rho)}{1 - P(\rho)} M. \tag{5.5}$$

In summary, the expected utility for the prover to honestly train a ρ portion of the task is

$$\begin{aligned} u(\rho) &= P(\rho)(Q(\rho) \cdot (R - \rho M) - (1 - Q(\rho)) \cdot (\gamma R + \rho M)) - (1 - P(\rho))\mu(\rho) \\ &= P(\rho)(Q(\rho) - \gamma(1 - Q(\rho)))R - \int_0^\rho P(x)dx \cdot M. \end{aligned} \tag{5.6}$$

[5]To ensure that the prover has enough tokens to pay the penalty, we have to require the prover to stake γR before participation. We can see that setting $\gamma \to +\infty$ makes the problem trivial as the prover gets an infinite penalty whenever she cheats; however, it needs the prover to stake an infinite amount of tokens, which is not possible.

To make the mechanism desirable for the prover and incentivize the prover to honestly train all the E epochs, we expect to satisfy the following (strict) interim individual-rationality (strictly interim IR) and basic incentive-security (BIS) properties:

Definition 17 (Strict interim individual-rationality)**.** *We call a PoL mechanism strictly interim individually-rational (strictly interim IR) if and only if honestly training the task earns a positive expected utility, i.e.,*

$$u(1) > 0, \tag{5.7}$$

assuming the verifier is honest.

Definition 18 (Strict interim basic incentive-security)**.** *We call a PoL mechanism strictly interim basic incentive-secure (strictly interim BIS) if and only if honestly training the task earns strictly more expected utility than dishonest training, i.e.,*

$$\forall \rho \in [0, 1), u(\rho) < u(1), \tag{5.8}$$

assuming the verifier is honest.

In the rest of this paper, without confusion, we omit the words "strict" and "interim", and call a mechanism γ-IR-BIS if it satisfies both of the properties above for parameter γ.

5.3.4 Threat Model

Jia et al. [96] introduce a threat model that consists of 4 types of attacks, as follows:

1. Retraining-based spoofing: the attacker aims to get the same PoL of the same model.
2. Stochastic spoofing: the attacker aims to get a different PoL of the same model.
3. Structurally correct spoofing: the attacker aims to get an invalid PoL of the same model that passes verification.
4. Distillation-based spoofing: the attacker aims to get a PoL of a (slightly) different model.

While our mechanism has some structural similarity to [96], our work has a different motivation. The work of Jia et al. [96] mainly aims to protect the copyright of an already trained *model*, but in our work the PoL serves as a Proof-of-Useful-Work, and our mechanism mainly aims to verify that the prover (as a miner) honestly did the computation, in which the attacker may have the interest to steal the copyright or not (if yes, we can just add the benefit of the copyright into the reward R in our analysis, so we essentially consider a wider attack space.) Nevertheless, as PoW miners typically compete for the blocks to earn *block rewards*, so we are motivated to mainly consider *rational* miners who would cheat to gain more economic utility.

In the paper of Jia et al. [96], the authors assume the attacker has the full information of the desired model, the full dataset, but does not have information of the random source of the model. In our paper, as the random seed is specified by the protocol, we consider an *even stronger* adversary that also has the random source. Formally, we assume that:

- The attacker has full information of the desired model $f(W, \cdot)$ trained with seed ϕ, but does not know the training process (for model-stealing attacks); she has also pre-trained a valid model $f(W', \cdot)$ with a different seed ϕ' (for known-model attacks).
- The attacker has full information on the dataset.
- The attacker also has the random source of the desired model, i.e. the random seed ϕ and the randomization guideline $\mathscr{G}$.

With our rational attacker assumption, the attack space contains a slightly modified version of 4 types of attacks. Actually, it is stronger because the structurally correct spoofing no longer requires to get the same model.

1. Retraining-based spoofing: the attacker aims to get the same PoL of the desired model $f(W, \cdot)$.
2. Stochastic spoofing: the attacker aims to get a different but valid PoL of the desired model $f(W, \cdot)$.

3. Structurally correct spoofing: the attacker aims to get an invalid PoL of any (correct or incorrect) model $f(W^{\#}, \cdot)$ that passes verification.

4. Distillation-based spoofing: the attacker aims to get a valid PoL of a (slightly) different model $f(W'', \cdot)$.

In Section 5.6 we will show the incentive-security property of our basic and full mechanisms against such attacks.

5.4 Basic Mechanism for Trusted Verifiers

In this section, we provide a general overview of our basic protocol for provers and verifiers, under the assumption of trusted verifiers which is widely adopted in previous literature.

5.4.1 Generation of PoL Certificate

The protocol is shown in Algorithm 2. For each block, we assume that there is an assigned problem $A = (D_{tr}, \mathscr{E}, \phi)$, in which D_{tr} is the training dataset, $\mathscr{E}$ is the environmental variables which include learning rate η, loss function $\mathscr{L}$, batch size m, number of epochs E, randomization guideline $\mathscr{G}$ that dictates how the randomness is generated from the seed, and other required specifications if needed (e.g. the initialization), and ϕ is the random seed generated from past blocks.

The prover is expected to solve the problem A by training E epochs following the given rule directed by $\mathscr{E}$, with the random seed ϕ. The initialization w_0 is specified by $\mathscr{E}$, and the prover is required to record the status after every k epochs, in which k is an integer parameter (either specified in the blockchain rule or specified in $\mathscr{G}$): smaller k leads to larger certificate size and prover storage consumption but lower computational overhead (see Section 5.6).

We assume that E is divisible by k, then the training process can consist of $T = \frac{E}{k}$ *stages*, in which each stage consists of $\tau = k \cdot l$ steps. For each stage $t \in [T]$, the prover is required to save the current weight $W_t = w_{t \cdot \tau}$. To save on-chain space, we only need the prover to a hash value of each W_t, and the required certificate is structured as $c = (c_1, \cdots, c_T)$ in

which $c_t = hash(W_t)$; In the verification stage, she also needs to post a subset of $\{W_t\}$ when queried by the verifier (see section 5.4.2).

Denote $|\mathscr{W}|$ as the model size, then the communication complexity is $O(\frac{E}{k})$ and the storage requirement for the prover is $O(\frac{E|\mathscr{W}|}{k})$ on this part.

Algorithm 2 Prover's certificate generation protocol in the basic mechanism

Input $A = (D_{tr}, \mathscr{E}, \phi), k, \alpha, \mathscr{L}, f$.
Initialize $w = w_0$ according to $\mathscr{E}$.
$T := \frac{E}{k}$.
$e := 0$
for $t := 1 \cdots T$ **do**
    for $x := 1 \cdots k$ **do**
        $e := e + 1$
        Draw b_e according to $(\mathscr{G}, \phi)$
        $w_{(e)} := \mathscr{T}_\eta(b_e, w_{(e-1)})$
    $W_t := w_{(e)}$
    $c_t := hash(W_t)$
Post $c := (c_1, \cdots, c_T)$.

5.4.2 Verification

The verification protocol is shown in Algorithm 3. The verifier is expected to *randomly*[6] verify α stages $\mathrm{t}_{ve} = \{t_1, \cdots, t_\alpha\}$ among T, in which α is a security parameter. For unpredictability to the prover, these stages should be drawn via uniform random sampling without replacement from her own secret (independent from ϕ). Then the verifier posts t_{ve}, requiring the prover to show corresponding weights.

Then, for each t_i, the prover is expected to post the weights before and after the stage, i.e. W_{t_i-1} and W_{t_i}. The verifier then checks whether the previously posted hashes are correct, and re-train the stage from W_{t_i-1} to see if the result is W_{t_i}. If and only if all tests are passed, then the basic verification is successful; otherwise, the verifier reports the detected cheating stages and indicates that the verification has failed.

[6]In this chapter, whenever we use the term "randomly", we refer to "randomly with a uniform distribution".

In this part, the communication complexity is $O(\alpha|\mathscr{W}|)$ and the relative computational overhead is $O(\frac{\alpha k}{E})$. In total, the communication complexity is $O(E + \alpha|\mathscr{W}|)$.

Algorithm 3 Verifier's verification protocol in the basic mechanism

Input $A = (D_{tr}, \mathscr{E}, \phi), k, \mathscr{L}, f, c = (c_1, \cdots, c_T)$.
Draw $\mathrm{t}_{ve} = \{t_1, \cdots, t_\alpha\}$ from $\{1, \cdots, T\}$ via her own secret.
Post t_{ve} to the prover, expecting to get $\{(W_{t_i-1}, W_{t_i})\}$ for each $t_i \in \mathrm{t}_{ve}$.
for $i \in 1 \cdots \alpha$ **do**
    if $c_{t_i-1} \neq hash(W_{t_i-1}) \vee c_{t_i} \neq hash(W_{t_i})$ **then**
        Return ("Fail", InvalidWeights(t_i))
    $w = W_{t_i-1}$
    for $e := k \cdot (t_i - 1) + 1, \ \cdots, \ k \cdot t_i$ **do**
        Draw b_e according to $(\mathscr{G}, \phi)$
        $w := \mathscr{T}_\eta(b_e, w)$
    if $w \neq W_{t_i}$ **then**
        Return ("Fail", ErrorInStage(t_i))
Return "Success"

5.5 Full Mechanism for Untrusted Verifiers

In this section, we discuss the verifier's incentive and augment our design to incentivize the verifier to verify honestly. On a high level, we introduce *safe deviations* as "flags" that do not affect the validity of the PoL but gain the verifier additional rewards that compensate for the verification cost, and design economic incentives to incentivize the verifier to find as many flags as possible within the α stages they inquire for their optimal utility, so that they would indeed verify α stages as supposed to.

5.5.1 Verifier's Strategy Space

In the previous works on Proof-of-Learning, it is typical that the systems only prevent the provers from cheating while assuming that verifiers are honest. However, in a fully decentralized and permissionless blockchain system, this is not necessarily true. While one may straightforwardly consider game-theoretic ways to incentivize verifiers to verify honestly, the Verifier's Dilemma [133, 134] would occur:

Verifier's Dilemma

- If a PoUW mechanism is (incentive-)secure against strategic provers, then no (rational) prover would cheat.
- If no prover would cheat and the verification has a non-zero computational cost, then the verifier's optimal strategy is to report "Success" without verification.
- If all verifiers are rational and would not actually verify, then the security properties no longer hold.

The Verifier's Dilemma indicates the difficulty in the design of a truthful mechanism with a *Nash equilibrium*[7] that both the prover and verifier act honestly.

Formally, we can model the verification game as follows:

Definition 19 (Verification Game)**.** *In a verification game, there is one prover P and $n_v \geq 1$ verifier(s) $V_1, \cdots, V_{n_v}$. The prover has an action space A_p, and a subset $A_p^H \subseteq A_p$ is denoted as* honest*. We denote $A_p^D = A_p \backslash A_p^H$ as the set of the prover's dishonest actions. For each action $a_p \in A_p$, the prover is incurred an initial cost $c_p(a_p)$.*

We assume n_v verifiers are independent and homogeneous. Any verifier also has an action space A_v with subsets A_v^H and A_v^D defined similarly. For any action $a_v \in A_v$, the verifier pays a cost of $c_v(a_v, a_p)$ and observes a result "Success" or "Fail", possibly attached with additional information in $\mathscr{I}$. Here, we denote $P_v(a_v, a_p)$ as the probability that the result is "Success".

In this work, we assume that the honest verification process may fail to detect cheats, but always passes honest proofs, i.e.,

$$a_p \in A_p^H \wedge a_v \in A_v^H \Rightarrow P_v(a_v, a_p) = 1.$$

[7] A Nash equilibrium refers to a situation in multi-party games in which no single party can benefit from individual deviation.

Finally, the prover and verifiers are rewarded or punished based on the verifiers' reports and the prover's action, given that the prover may dispute and future users may check the verification result and do slashing *for dishonest verification. Hence, the payment rule can be denoted as:*

$$\pi : (\{\text{“Success”, “Fail”}\} \times \mathscr{I})^{n_v} \times A_p \to \mathbb{R}^{n_v+1}.$$

For the slashing rule, since the honest verification always passes honest proofs, we assume that reporting "Fail" when $a_p \in A_p^H$ can be regarded as ***deliberately malicious*** *and will incur heavy penalties ($\to \infty$) for the verifier.*

From the modeling, we can show a formal negative result as:

Theorem 5.1 (Verifier's Dilemma)**.** *In a verification game in which the only information the verifier(s) report is "Success" or "Fail", i.e. $|\mathscr{I}| = 1$, and honest verification has a strictly positive cost, i.e.*

$$a_v \in A_v^H \Rightarrow c_v(a_v, a_p) > 0,$$

it is impossible to design a verification mechanism with a pure-strategy Nash equilibrium that the prover and verifier(s) simultaneously act honestly.

The proof is deferred to Appendix C.5.1.

To analyze the concern in the scope of our work, in the context of this paper, we classify the verifier's strategies into 3 types:

- Honest: Run the verification protocol honestly.
- Lazy: Verify a different (possibly stochastic) $\alpha' \leq \alpha$ of stages from designated, with $\Pr[\alpha' < \alpha] > 0$.
- Non-trivially Dishonest: Run any algorithm non-equivalent to Honest or Lazy.

We notice that any Honest or Lazy verification strategy essentially verifies a subset of the stages so that no honest proof would fail the verification. On the other hand, from our protocol in Section 5.4.2, when a verifier reports "Fail" she must indicate the stage that fails the verification; hence, if the prover is actually honest, she can clarify its honesty and

thus the verifier can be easily caught and heavily penalized by a "slashing" mechanism like in Ethereum. Therefore, we mainly consider the "benign" verification strategies, formally defined as follows:

Definition 20 (Benign verification strategy)**.** *A verification strategy is* ***benign*** *if and only if honest proofs pass the verification with probability 1.*

In the rest of this section, we only consider benign verification strategies for the verifier.

5.5.2 The Symmetric-Cheating Model and Failure of Basic Mechanism

While a dishonest prover may prefer certain stages over others for cheating in the real world, since every stage has the same computational cost and our verifier's protocol in Section 5.4.2 guarantees that the probability of getting caught only depends on the *number* of cheating stages, we can argue that a dishonest prover would be indifferent on the stages to cheat. Therefore, we consider a symmetric-cheating model in which a dishonest prover acts in the following way:

Definition 21 (Symmetric-cheating prover)**.** *A symmetric-cheating prover has a type* $\mathrm{p} = (p_0, p_1, \cdots, p_T)$ *in which* p_i *is the probability that she cheats in* i *stages, and* $\sum_i p_i = 1$*. When she is generating a PoL, she performs as follows:*

1. *Nature chooses* $m \sim \mathrm{p}$ *as the number of stages she would cheat.*
2. *She uniformly randomly draws* m *stages among the total* T *stages to cheat and compute the PoL in this way.*
3. *She submits the PoL.*

Now we assume that the prover is symmetric-cheating. Since we have shown the basic (prover-side) incentive-security of our mechanism, among the population of parties that may serve as provers, we assume that an overwhelming majority are honest, and only a small fraction ϵ may cheat. Define $\bar{\mathrm{p}} = (\bar{p}_0, \cdots, \bar{p}_T)$ as the mean of p in the population of provers, then we have

$$\bar{p}_0 \in (1 - \epsilon, 1). \tag{5.9}$$

Failure of the basic mechanism. While we may straightforwardly want to reward the verifier for catching cheats, unfortunately from the Verifier's Dilemma, as long as the reward for the verifier is bounded, we can see that our basic mechanism in Section 5.4.1-5.4.2 would not work. Formally, we have

Theorem 5.2. *In our basic mechanism in Section 5.4.1-5.4.2, if we assume that the verifier's maximum reward for finding a cheat is v_+ and the verifier's expected reward when the PoL passes the verification is v_0, then if $v_+ \leq v_0$ or $\epsilon \in (0, \frac{M}{T(v_+ - v_0)})$, the verifier's strictly optimal strategy is to report "Success" without verification.*

The proof of Theorem 5.2 is deferred to Appendix C.5.2. Therefore, for any fixed v_+, v_0, we always have $\epsilon > 0$ which makes the mechanism not incentive-secure for the verifier, because for ϵ small enough, the expected "additional reward" for catching a cheat would not cover the cost of verification. Therefore, we desire to modify the basic mechanism in a way that the verifier would maximize her expected utility by verifying and reporting honestly, uniformly for any ϵ small enough.

In this setting, we define verifier incentive-security (VIS) as follows:

Definition 22 (Verifier incentive-security)**.** *We call a PoL mechanism verifier incentive-secure if and only if, for some fixed $\epsilon > 0$, as long as the prover is honest with a probability greater than $1 - \epsilon$, the verifier gets the most expected utility via honestly performing the verification protocol among all benign verification strategies.*

Particularly, the mechanism discussed in this section is VIS if and only if the verifier is incentivized to honestly verify all α stages in t_{ve} honestly.

5.5.3 The Capture-The-Flag Protocol

As discussed in the parts above, we are aware that the Verifier's Dilemma only occurs in the scenario of $\epsilon \to 0$. Hence, a natural idea is to increase ϵ, i.e. insert deliberate invalid objects, or so-called "flags" to incentivize verifiers to find, as in the works of [135, 136, 137]. On the other hand, our Theorem 5.1 also shows the necessity for a desirable verification mechanism to let the verifier incorporate additional information into her report. Hence, the

most straightforward idea is to deliberately generate invalid PoL's into the pool that serve as flags. However, this approach also faces the following challenges:

- The cheaters in the pool can have complicated behavior, e.g., having different ρ's in their cheating patterns. It is difficult to set proper ρ's or analyze verifiers' behavior in the presence of both cheats and deliberately inserted flags.
- Particularly, if ρ is not close to 0, then the generation of invalid PoL's needs to contain a large portion of honest computation which has immense computational overhead, which not only undermines the efficiency but also complicates the protocol, e.g., in the allocation and compensation of such "chores".
- If ρ is close to 0, then the verifier would have a high probability of identifying the flags even if they only verify 1 stage (rather than α), which could incentivize a different dishonest strategy rather than the honest one.

In consideration of the issues above, we propose a variant to (let provers) insert the flags into each PoL certificate, i.e. designate a random subset of the stages as flags, and provers should make commitments about the flags inserted when submitting the PoL. However, due to the sequential nature of the SGD algorithm, inserting an invalid stage may affect the validity of the following stages and ultimately the resulting model; therefore, we insert *safe deviations* that serve as flags, which is implemented by computing honestly with a differently designated seed. In particular, given the (root) random seed ϕ, a stage t can have 4 possible types:

1. Normal: it is trained with random seed $r_\phi(3t)$, as defined in Section 5.3.2.
2. Flag F_1: it is trained with random seed $r_\phi(3t+1)$.
3. Flag F_2: it is trained with random seed $r_\phi(3t+2)$.
4. Dishonest: otherwise.

Notice that we do need two types of flags so that the verifier would be willing to check the type of the flag, instead of reporting "Flag" when the verification of "Normal" fails without

any attempt to differentiate it from a dishonest stage. In this setting, we assume that less than half of the stages are flagged, so that the verifier would first verify with seed $r_\phi(3t)$ for stage t. If the verification of seed $r_\phi(3t)$ fails, the verifier, who believes that the probability of cheating is sufficiently small, would believe that it is a flag and randomly choose one of the following actions:

- Verify with seed $r_\phi(3t+1)$. If successful report F_1, otherwise report F_2.[8]
- Verify with seed $r_\phi(3t+2)$. If successful report F_2, otherwise report F_1.

The verifier can alternatively randomly guess F_1 or F_2 without verification of either, but this would lead to a $\frac{1}{2}$ probability of reporting the wrong flag and getting penalized (for a higher amount than the flag reward). Hence, the verifier is incentivized to perform the verification as described above.

Therefore, if a cheater wants to disguise a dishonest stage as a flag, she must claim that it is F_1 or F_2 in the commitment, with a $\kappa = 1/2$ probability of being caught if the stage is verified.

The protocol of certificate generation and verification are shown in Algorithm 4 and Algorithm 5, respectively.

Intuitively, to incentivize the verifier to verify α stages among the total T, assume that we would like the prover to insert ηT (committed) flags in which $\eta \in [\frac{2\alpha}{T}, \frac{1}{2})$, then when the verifier verifies honestly, the expected number of flags she finds would be $\alpha\eta$. Since the verifier only has access to the α stages in t_{ve}, we would like to incentivize the verifier to find as many flags as possible so that the verifier would honestly verify all the α stages. Therefore, we award the verifier for each flag she detected. Particularly, recalling that the training cost of a stage is $\frac{M}{T}$ and noting that the discovery of a flag would take an additional $\frac{M}{T}$ cost of computation, we set positive parameters $R_0 \gg R_1 > \frac{M}{T}(\frac{2}{\eta} + 1)$. When the verifier finds u flags and D dishonest stages, the system gives the verifier a reward of $W_v(u)$:

$$W_v(u) = R_0[D > 0] + R_1 u. \tag{5.10}$$

[8]Since the stage is neither normal or F_1, it is either F_2 or dishonest. As the probability of cheating is sufficiently small, she would prefer to believe it is F_2 rather than take additional computational cost to distinguish them via verifying with seed $r_\phi(3t+2)$. Similar for the other case.

Algorithm 4 Prover's certificate generation protocol in the full mechanism

Input $A = (D_{tr}, \mathscr{E}, \phi), k, \alpha, \mathscr{L}, f, \eta$.
Initialize $w = w_0$ according to $\mathscr{E}$.
$T := \frac{E}{k}$
$e := 0$
Generate $\sigma = (\sigma_1, \cdots, \sigma_T)$ as a random permutation of $[T]$ from her own secret.
$\mathscr{H} := hash(\sigma)$
for $t := 1 \cdots T$ **do**
    if $\sigma_t \leq \eta T$ **then**
        if σ_t is odd **then** $s_t := r_\phi(3t+1)$ **else** $s_t := r_\phi(3t+2)$
    else
        $s_t := r_\phi(3t)$
    for $x := 1 \cdots k$ **do**
        $e := e + 1$
        Draw b_e according to $(\mathscr{G}, s_t)$, denoted as $b_e := B_e(s_t)$.
        $w_{(e)} := \mathscr{T}_\eta(b_e, w_{(e-1)})$
    $W_t := w_{(e)}$
    $c_t := hash(W_t)$
$c := (c_1, \cdots, c_T)$
Post $(c, \mathscr{H})$.

in which the notation $[statement]$ stands for

$$[statement] = \begin{cases} 1, & \text{if } statement \text{ is true;} \\ 0, & \text{otherwise.} \end{cases}$$

In Section 5.6, we prove that for values of α, β, T that satisfy certain conditions, there is a Nash equilibrium that the prover trains honestly, and the verifier verifies exactly α stages.

5.6 Theoretical Incentive-Security Analysis

In this section, we show the incentive-security properties of our mechanisms.

In Section 5.3.4, we model 4 types of attacks to the PoL mechanism. In the protocol defined in Section 5.4.1, the training task is divided into T stages. Even though it is a *stochastic* gradient descent task, since the random seeds are given by the protocol, the training process of each stage is deterministic.

In the prover's training process, the prover is expected to save the model weights W_t at each stage t, and post $c_t = hash(W_t)$. An honest prover should compute each W_t from the result W_{t-1} of the previous stage following the expected procedure.

For a possibly dishonest prover, in each stage t, she may compute W_t from W_{t-1} either honestly or dishonestly, or even does not compute a W_t at all while forging a fake c_t. In our definition, even if W_{t-1} may be dishonestly computed, as long as she follows the procedure and computes W_t from W_{t-1}, we say that she trains stage t "honestly"; otherwise, if either W_{t-1} or W_t is nonexistent or invalid, or the prover does not follow the procedure when computing W_t from W_{t-1}, we say that she trains the stage t "dishonestly". Hence, we can naturally define the ρ (as discussed in Section 5.3.3) as the fraction of stages trained honestly and say that the prover is honest if and only if $\rho = 1$, i.e., she trains all stages honestly.

As introduced in Section 5.4.2, the verifier randomly chooses α stages among the T stages to verify. For each chosen stage t, the verifier queries the prover for (W_{t-1}, W_t) and verifies if W_{t-1}, W_t match the hashes and W_t is the result of honest computation from W_{t-1}. Since the prover needs to post hashes of weights before the verification, all the weights have to be finalized before the verification. Hence, the prover would pass the verification with a probability of 1 if and only if all verified stages are trained honestly. In the full mechanism, if ξ verified stages are not trained honestly, the prover passes the verification with a probability of $2^{-\xi}$.

In Section 5.3.4 we discussed about 4 types of attacks. In retraining-based spoofing, the attacker aims to get the same PoL, while in the other 3 types of attacks, the attacker aims to get a different PoL. Due to the deterministic nature of our protocol, if the attacker aims to get a different PoL, she must train a subset of stages dishonestly, which is indeed classified as "dishonest" in our analysis[9]; for the retraining-based spoofing, since the attacker aims to get the same PoL, it can neither save any computational cost nor corrupt the model, so it only has interest in copyright protection and does not need to be considered for the motivation of out setting that aims to adopt PoL as a PoUW.

For prevention of the 3 types of attacks, under mild assumptions, we show that our

[9]In the augmentation of Section 5.5 there may exist different valid *safe deviations* but they could not save any computational cost.

mechanism is incentive-secure for small α compared to the number T of stages and a moderately large R_1, as characterized as below:

- Even with no penalty ($\gamma = 0$), an $\alpha = O(\log T)$ is sufficient as long as the reward R guarantees "just slightly more than" individual-rationality.
- With moderate penalty $\gamma = \Theta(1)$, an $\alpha = O(1/\gamma) = O(1)$ and a reward R guaranteeing IR are sufficient to guarantee γ-IR-BIS.
- With $\eta \in \left[\frac{2\alpha}{T}, \frac{1}{2}\right)$ and $R_1 \geq \frac{M}{T}\left(\frac{2}{\eta} + 1\right)$, our full mechanism is guaranteed to be VIS.

Formally, we have our main theorem on the prover side:

Theorem 5.3 (Main Theorem)**.** *Assume $T \geq 2$, and denote $\beta = \frac{M}{R}$. If the winning probability function $P(\cdot)$ is differentiable and its hazard rate is upper bounded by λ, i.e.,*

$$\frac{P'(\rho)}{P(\rho)} \in [-\lambda, 0], \forall \rho \in [0, 1], \tag{5.11}$$

in which $P'(\cdot)$ is denoted as the derivative of $P(\cdot)$; and in the verification protocol defined, a cheating stage has at least a $\kappa = \Theta(1)$ probability to be caught when verified[10]*, then the mechanisms defined as Algorithms 2-3 and Algorithms 4-5 are 0-IR-BIS if*

$$R > \frac{\int_0^1 P(\rho)d\rho \cdot M}{P(1) - (1-\kappa)^\alpha}, \tag{5.12}$$

$$\alpha \geq \max\left\{\frac{2(\lambda+\beta)}{\beta\kappa}, \frac{2\ln\frac{T}{\beta}}{\kappa}\right\}, \tag{5.13}$$

in which Eq. (5.12) *exponentially converges to $R > \frac{\int_0^1 P(\rho)d\rho \cdot M}{P(1)}$, the sufficient and necessary condition for IR, when α is moderately large.*

The proof is deferred to Appendix C.5.3. From the main theorem, we see that for a fixed P, the number of required stages for verification is $O(\log T)$ for bounded λ and $\kappa = \Theta(1)$, making the relative computational overhead as low as $O(\frac{\log T}{T}) = O(\frac{k \log E}{E})$.

[10] In the basic mechanism we have $\kappa = 1$, while in the full mechanism $\kappa \geq \frac{1}{2}$.

Furthermore, by inducing penalty $\gamma = \Theta(1)$, i.e. getting caught cheating leads to a penalty comparable to the block reward, we can lower the number of required stages to $O(1)$ and the relative computational overhead to $O(\frac{k}{E})$. Formally, we have:

Theorem 5.4. *For $\gamma > 0$, with the same definition of β, λ, κ as in Theorem 5.3, the mechanisms defined as Algorithms 2-3 and Algorithms 4-5 are γ-IR-BIS if*

$$R > \frac{\int_0^1 P(\rho)d\rho \cdot M}{P(1)}, \tag{5.14}$$

$$\alpha > \max\left\{\frac{\beta}{\gamma\kappa}, \frac{\lambda}{\kappa}\right\}. \tag{5.15}$$

The proof of Theorem 5.4 is deferred to Appendix C.5.4.

On the other hand, we show the verifier incentive-security property of our mechanism, which, combined with the basic incentive-security properties of our mechanism, guarantees a Nash equilibrium that both parties behave honestly:

Theorem 5.5. *Our full mechanism defined as Algorithms 4-5 is VIS if*

$$\eta \in \left[\frac{2\alpha}{T}, \frac{1}{2}\right), \tag{5.16}$$

$$R_1 \geq \frac{M}{T}\left(\frac{2}{\eta} + 1\right). \tag{5.17}$$

The proof of Theorem 5.5 is deferred to Appendix C.5.5.

5.7 Experimental Demonstration

In this section, we perform experiments to evaluate the practical performance on our proposed PoL mechanism. In our experiments, we test our mechanism with the CIFAR (CNN) and MNIST (MLP) training tasks on a computer with NVIDIA GeForce RTX 4090 and 24GB memory. Each task contains $T = E = 1000$ stages with each stage containing one epoch, and for robustness of the system, each task is independently verified by $n = 5$ verifiers. We set parameters $\eta = 0.2$ (20% stages flagged), $\gamma = 0$ (no penalty), $\beta = \frac{1}{2}$ (the reward is

2 times the computational cost), and $\alpha \in \{1, 2, 5, 8, 10, 50\}$ as the number of stages each verifier checks. In case of disagreement among verifiers, we employ the following reward rules:

Provers' rewards. To decide on the acceptance or rejection of a proof, we follow the vote of majority verifiers, i.e., the proof is accepted if and only if $v > \frac{n}{2}$ verifiers vote "Success". Then the prover's reward can be proceeded with one of the following rules:

1. **Proportional Rule**: The prover gets a $\frac{v}{n}$ fraction of the training reward, i.e., $\frac{v}{n}R$, regardless of the decision.

2. **Strict-Proportional Rule**: The prover gets $\frac{v}{n}R$ when accepted, and 0 when rejected.

Even if the proof is accepted, we do not pay full rewards when $v < n$ to ensure that even "slight" cheats are not (marginally) profitable. We can see that the Proportional Rule has the same prover incentive properties as the setting of one single verifier, and we defer detailed discussions to Appendix C.2.

Verifiers' rewards. While the design of more theoretically guaranteed reward rules to incentivize honest reports without reference to ground-truth information generally lies in the scope of *peer prediction* (e.g., [5, 138, 139, 140]), in this study we mainly focus on the design of PoL protocols and leave it to future work. Here, we reward the verifiers based on majority voting, and only verifiers whose reports agree with the majority get rewards as follows.

- If the proof is accepted, verifiers reporting "Success" are rewarded according to detected flags according to Section 5.5.3.

- If the proof is rejected, verifiers reporting "Fail" are given a constant reward as the expected verification reward if the proof were honest and accepted, i.e., $\alpha\eta R_1$.

5.7.1 Experimental Results

In the experiments, we perform the following groups of tests with different types of attacks as shown in Table 5.2. Among these attacks, only the partial spoofing attack shows non-zero success rates, as other attacks invalidate the output of every stage and will be detected even

if only one stage is checked. We notice that the attacks of [100, 101] essentially modify the training process to exploit the error tolerance in the work of Jia et al. [96] and lie in the scope of *distillation-based spoofing attack*, and hence are effectively prevented by our mechanism.

#	Attack Type	Success Rate
0	Honest: No cheating or attack.	1
1	Known-model Attack: The attacker submits a pretrained model obtained from external sources.	0
2	Model-stealing Attack: The attacker submits a model trained by others who received the same training task.	0
3	Stochastic Spoofing Attack: The attacker randomly generates formatmatched results as the certificate.	0
4	Structurally Correct Spoofing Attack: The attacker mimicks the format of a PoL, randomly updating the model's weight without doing the actual training.	0
5	Distillation-based Spoofing Attack: The attacker modifies some parameters or the training process. **Attacks of [100, 101] lie in this scope.**	0
6	Partial Spoofing Attack: The attacker trains partial of the stages honestly and partial dishonestly.	Depending on parameters.

Table 5.2: Types of attacks in the experiments.

Provers' rewards. In Figures 5.1, we show the experimental results for training CIFAR and MNIST datasets with different α, in which the Proportional Rule is used for prover's rewards and the reward ratio refers to the expected reward from the system compared to honest training. From the plots we show that the system can detect almost all partial spoofs with $\alpha = 50$, i.e. each verifier verifies 5% of all stages. For smaller α, the expected reward of a spoof increases with higher honesty ratios and decreases with larger α's.

Furthermore, in Figure 5.2 we show the incentive properties of our mechanism for the tasks. The "Utility Ratio" refers to the net utility (reward minus computational cost) compared honest training. From the results, we see that when there is no mining competition, training the model honestly yields the maximum utility for the prover even for $\alpha = 1$, i.e., the mechanism is incentive-secure. Furthermore, we see that for $\alpha \geq 10$, the prover gets negative utility unless at least 90% of the stages are honestly trained, showing the sharpness of our incentive guarantee even for small α's.

Since the experimental evaluation in the scenario with mining competition is complicated

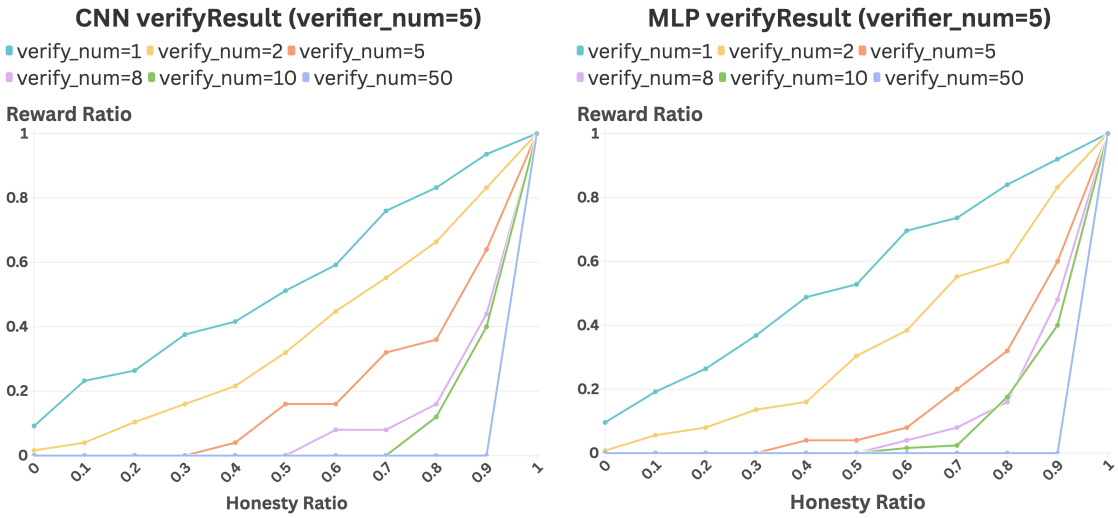

Figure 5.1: Experimental Results.

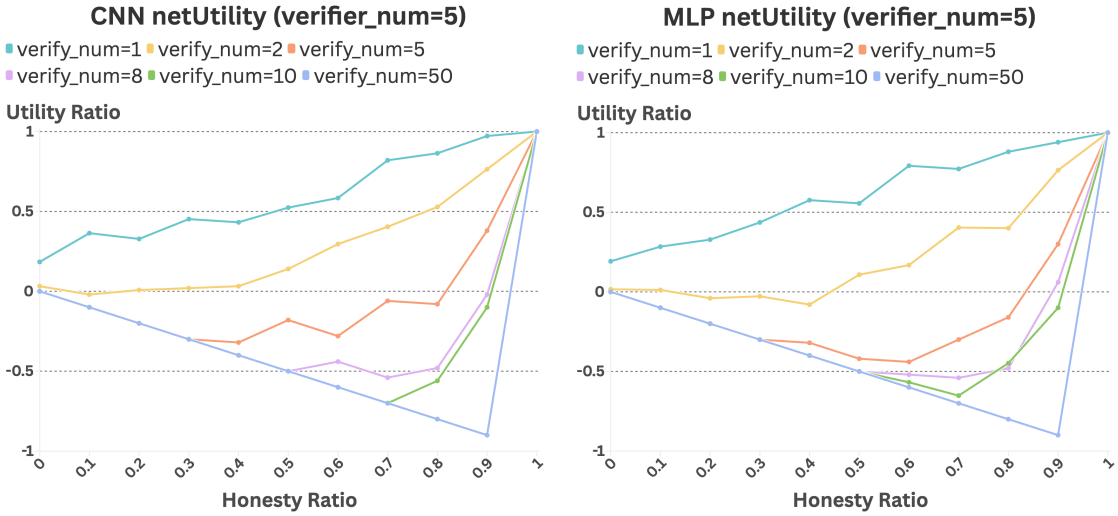

Figure 5.2: Prover Net Utilities.

with real ML training tasks, particularly for the estimation of sunk costs when losing the competition, we refer to Theorems 5.3-5.5 for theoretical guarantees and leave real-world experiment for future empirical study.

Verifiers' rewards. In the notion of Nash equilibria, we assume the honesty of the prover and all other verifiers. When we consider the case of $\alpha = 50$ in which there is an overwhelming probability that all other verifiers report the ground truth (as shown in previous experiments), the proof is accepted and the verifier's reward and utility are simply proportional to the honestly verified stages. Hence, the verifier is indeed incentivized to honestly verify all α stages. In Appendix C.3 we show the detailed experimental results and also demonstrate the necessity of the CTF protocol for the assurance of verifiers' incentive guarantees empirically.

Computational overheads. In Table 5.3, we show the average running time for training and verification, in which $\alpha = 50$ epochs are verified among a total of $E = T = 1000$. We can see that for each verifier, verifying an honest proof takes 6.1% of the training time, slightly higher than $\frac{\alpha}{T} = 5.0\%$ as the flag test in Algorithm 5 takes additional computation. Since we expect that most of the proofs are honest, our mechanism indeed achieves low computational overheads.

Communication overheads. The communication overheads are shown in Table 5.4. We can see that the communication overheads are worse than computational overheads because full parameters need to be transmitted for verification, but still within a reasonably small fraction ($< 20\%$) of all data generated during the training process. To further optimize the communication overheads, low-rank training techniques (e.g., GaLore [141, 142]) can be adopted to optimize the overall I/O overheads for the training tasks.

5.8 Discussion

In this chapter, we develop an incentive-secure PoL mechanism with provable incentive-security, efficiency and controllable difficulty that successfully bypasses the existing hardness results, and also tackles the Verifier's Dilemma via a capture-the-flag protocol that encourages honest verification, while improving the relative computational overhead from $\Theta(1)$ in [96]

Honest Ratio	Training (s)	Verification (s)	Overhead/Verifier (%)
0.0	169.5	282.3	166.5
0.1	500.5	274.5	54.8
0.2	775.5	264.4	34.1
0.3	984.0	253.1	25.7
0.4	1235.5	243.5	19.7
0.5	1521.0	228.6	15.0
0.6	1717.5	222.3	12.9
0.7	2027.5	213.0	10.5
0.8	2356.5	199.5	8.5
0.9	2642.5	185.8	7.0
1.0	2782.0	171.0	6.1

Table 5.3: Computational Overhead Analysis. ($\alpha = 50$)

	MNIST	CIFAR
Model Size (MB)	52.41	162.60
Data Generated in Training (MB)	2369	3595
Transmission/Verifier (MB), $\alpha = 10$	147	382
Overhead/Verifier (%), $\alpha = 10$	6.2	10.6
Transmission/Verifier (MB), $\alpha = 50$	333	658
Overhead/Verifier (%), $\alpha = 50$	14.0	18.3

Table 5.4: Communication Overhead Analysis.

to $O(\frac{\log E}{E})$ or $O(\frac{1}{E})$, and improving the communication complexity from $\Theta(E|\mathscr{W}|)$ in [96] to $O(E+|\mathscr{W}|\log E)$ or $O(E+|\mathscr{W}|)$, depending on different settings. On a high level, this paper not only provides an approach toward a secure and sustainable PoUW puzzle, but also has the potential to be a novel design for decentralized AI platforms.

While our mechanism can significantly improve the communication complexity compared to privious work, if the communication is implemented on-chain, it is only applicable for relatively small models. To enable models with larger sizes compared to block spaces, IPFS [143] or layer-2 techniques [144] can be used for cheaper storage.

In real-world applications where the trained model may have exogenous interests, the prover may gain additional utility from training an incorrect model. In this scenario, our mechanism can be augmented with a family of anomaly detection techniques for deep learning [145] and ensure that corrupting a small number of epochs would not significantly corrupt the output model. We defer high-level discussions to Appendix C.4 and leave the detailed

study for future research.

Algorithm 5 Verifier's verification protocol in the full mechanism

Input $A = (D_{tr}, \mathscr{E}, \phi), k, \mathscr{L}, f, c = (c_1, \cdots, c_T), \mathscr{H}$.
Draw $\mathrm{t}_{ve} = \{t_1, \cdots, t_\alpha\}$ from $\{1, \cdots, T\}$ via her own secret.
Post t_{ve} to the prover, expecting to get $\{(W_{t_i-1}, W_{t_i})\}$ for each $t_i \in \mathrm{t}_{ve}$.
for $i \in 1 \cdots \alpha$ **do**
    if $c_{t_i-1} \neq hash(W_{t_i-1}) \vee c_{t_i} \neq hash(W_{t_i})$ **then**
        Return ("Fail", InvalidWeights(t_i))
    $w = W_{t_i-1}$
    $w_1 = w$
    for $e := k \cdot (t_i - 1) + 1, \ \cdots, \ k \cdot t_i$ **do**
        $b_e^{(0)} = B_e(r_\phi(3t))$
        $b_e^{(1)} = B_e(r_\phi(3t+1))$
        $b_e^{(2)} = B_e(r_\phi(3t+2))$
        $w_1 := \mathscr{T}_\eta(b_e^{(0)}, w_1)$
    if $w_1 = W_{t_i}$ **then**
        $V_i := 0$
    else
        Draw $\xi \sim Uniform\{0, 1\}$
        if $\xi = 1$ **then**
            for $e := k \cdot (t_i - 1) + 1, \ \cdots, \ k \cdot t_i$ **do**
                $w := \mathscr{T}_\eta(b_e^{(1)}, w)$
            if $w = W_{t_i}$ **then** $V_i := 1$ **else** $V_i := 2$
        else
            for $e := k \cdot (t_i - 1) + 1, \ \cdots, \ k \cdot t_i$ **do**
                $w := \mathscr{T}_\eta(b_e^{(2)}, w)$
            if $w = W_{t_i}$ **then** $V_i := 2$ **else** $V_i := 1$
Post $V = \{V_i\}_{i \in [\alpha]}$, requesting the prover to post σ.
if $hash(\sigma) \neq \mathscr{H}$ **then**
    Return ("Fail", InvalidFlagCommitment)
for $i \in 1 \cdots \alpha$ **do**
    if $\sigma_{t_i} \leq \eta T$ **then**
        if σ_{t_i} is odd **then** $s_i := 1$ **else** $s_i := 2$
    else
        $s_i := 0$
    if $V_i \neq s_i$ **then**
        Return("Fail", ErrorInStage(t_i))
Return ("Success", t_{ve}, $\{s_i\}$).

CHAPTER 6

INCENTIVES FOR DECENTRALIZED VERIFICATION GAMES

Peindre l'amour, peindre la vie,

Pleurer en couleur.

— Clair Obscur: Expedition 33

6.1 Introduction

Blockchain, with prevailing examples as Bitcoin [78] and Ethereum [79], is an emerging technology that maintains decentralized consensus via a distributed ledger that utilizes cryptographic techniques to achieve trust and security. In recent years, the blockchain technology is drawing wide interest in the operations research community (see, Chen et al. [3], Davydiuk et al. [30], Iyengar et al. [31], Manzoor et al. [32], Whitaker and Kräussl [33], Gong et al. [146] and their references); on the other hand, it also has applications that empower traditional operations research studies, e.g., supply chain management (Keskin et al. [34], Cole et al. [147], Cui et al. [148]). Furthermore, following the current trend of artificial intelligence (AI), a frontier topic of *decentralized AI* [149] occurs with the motivation to leverage blockchain technologies for securing training and inference procedures of machine learning (ML) computation to ensure credibility and accountability of AI models, and has been drawing interest in both blockchain and AI communities (see, e.g., Zhao et al. [4], Conway et al. [130], Chen et al. [150]).

In the meantime, from a game-theoretic perspective, a well-designed incentive mechanism is crucial to motivate self-interested players to behave honestly in a decentralized ecosystem, and a line of recent studies (see, Chen et al. [3], Roughgarden [37], Hansjoerg and Pierre-Olivier [151], Chen and Golab [152], etc.) has formed an emerging research field of *blockchain*

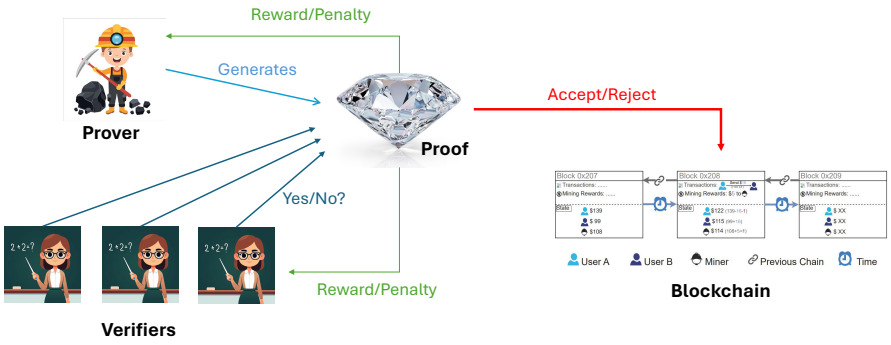

Figure 6.1: Illustration of decentralized verification games.

mechanism design that investigates the design and analysis of on-chain reward/penalty[1] mechanisms to incentivize honest behavior on blockchain platforms, including decentralized AI applications.

For concrete understanding of blockchain incentive mechanisms, let us look into how decentralized consensus is maintained. In a blockchain system, the "chain" is essentially a linked list of "blocks" growing with time, where each block stores a piece of data (aka. *transactions*). Every player stores a copy of the chain, and players who propose new blocks are supposed to simultaneously *validate* previous blocks. To achieve consensus in such decentralized systems, Bitcoin adopts the Proof-of-Work (PoW) that requires "miners" to expend significant computational effort (and energy) to earn access to blocks. Alternatively, Ethereum uses Proof-of-Stake (PoS) that requires validators to stake tokens for participation, and selects validators with probabilities in proportion to their staked tokens [153]. Furthermore, recent studies also propose Proof-of-Learning (PoL) to replace the PoW task with AI model training [4, 96]. On the one hand, the PoL protocol serves as a Proof-of-Useful-Work (PoUW) that addresses the energy and sustainability issues of traditional PoW protocols [84, 85] and the security and centralization issues of PoS [90]; on the other hand, the verification of PoL in turn serves as the certificate that the model is trained honestly, realizing the motivation of decentralized AI to prevent adversarial attacks on model training [109] and address *AI safety* concerns [102] in the current times.

While such mechanisms are motivated to incentivize the prover to behave honestly, they

[1]In the rest of this paper, we use terms "reward" and "penalty" interchangeably: a penalty can be regarded as a negative reward and vice versa.

can inadvertently introduce strategic concerns for validators, e.g., rational validators may act *lazily* or *maliciously* in the validation process. A notable phenomenon is the Verifier's Dilemma [133, 137, 154], showing that rational verifiers do not have the incentive to act honestly when a proposed block (or "proof") is honest with overwhelming probability and verification incurs nontrivial computational costs. While the Verifier's Dilemma does not appear to seriously undermine the security of Bitcoin or Ethereum in practice, it would become a prominent challenge in decentralized AI applications due to the heavy computational costs of ML verification [4, 130]. The Verifier's Dilemma is described as follows: If a mechanism could incentivize provers to behave honestly, then no (rational) provers would cheat; if no prover would cheat and the verification has a non-zero computational cost, then the verifier's optimal strategy is to lazily accept the proof without actual verification; when verifiers become lazy, the incentive guarantees for provers no longer hold. Formally, the Verifier's Dilemma can be formulated as the following theorem. The proof is deferred to Appendix D.6.1.

Theorem 6.1 (Verifier's Dilemma)**.** *In a verification game in which*

- *A verifier's report is binary, e.g. "Success" or "Fail";*
- *The verification result of an honest proof is always "Success";*
- *Honest verification has a strictly positive cost,*

It is impossible to design an incentive mechanism realizing a pure-strategy Nash equilibrium such that the prover and verifier(s) simultaneously act honestly.[2]

The Verifier's Dilemma arises in settings where the fraction of cheating provers tends to zero, making it optimal for verifiers to adopt lazy strategies. In such scenarios, any bounded reward for catching a cheat cannot, in expectation, offset the cost of verification. Hence, a traditional reward-based approach can only lower, but not eliminate, the rate of cheating in the pool of proofs. This issue becomes particularly prominent in applications involving costly

[2]Although the Verifier's Dilemma is sometimes understood as the tendency of verifiers to act *lazily*, our study is aimed for a general purpose to design robust mechanisms incentivizing *honest* verification—i.e., verifiers are incentivized not only to avoid laziness but also to refrain from acting maliciously.

verification—particularly for decentralized AI applications. For example, Conway et al. [130] propose the opML (Optimistic Machine Learning), a more straightforward protocol than PoL for decentralized trustworthy ML computation performed by at least two parties, in which one *prover* essentially runs the computation and one or more *verifiers* simply re-run the same procedure for verification. Conway et al. [130] show that their mechanism achieves a mixed-strategy Nash equilibrium in which the prover has a $\frac{C}{R+L}$ probability to cheat, where C is the verification cost and $R+L$ is essentially the penalty imposed to the verifier for failing to report a cheat plus the reward for successfully reporting one. In real-world blockchain systems, penalties are typically upper bounded by the required stake and excessive rewards may lead to inflation. As a result, higher verification costs exacerbate the dilemma: the system must either demand larger stakes, suffer severe inflation, or tolerate a higher fraction of dishonest proofs.

Recent studies are actively working on addressing the challenge of Verifier's Dilemma, but they mostly depend on certain extents of trusted authorities. A line of recent studies attempts to bypass the binary-report assumption of Theorem 6.1 via introducing "attention challenges" that contain extra information or deliberate errors, e.g., inserting deliberate objects (which can be valid or invalid) as so-called "flags" to incentivize verifiers to find and report, as in [4, 135, 136, 137, 154, 155], essentially bypassing the binary-report assumption in Theorem 6.1. For example, Zhao et al. [4] propose a incentive-secure PoL protocol in which the verifiers are supposed to verify a random subset of the training process, in which the provers may use different designated random seeds as "flags" for the verifiers to report, Sheng et al. [155] design the flags as traces of transaction computation, and Teutsch and Reitwies,s'ner [136] design the flags as deliberately invalid proofs. However, in the scenario where the verifiers may also be strategic or even *malicious*, the validity of verification results, particularly for ML computation, can also be costly to verify. Hence, in traditional decentralized ML verification protocols [4, 96], we generally need some credibility assumptions on verifiers. Alternatively, other proposals invoke additional phases of "committee voting" when disagreement occurs [130, 131], use heuristic reputation-based designs [156], or resort to *oracle*-like entities [157, 158]. In these proposals, the committees, high-reputation parties, or oracles are regarded as trusted authorities and act as *proxies of ground truths*,

	$Z_2 = 0$	$Z_2 = F_1$	$Z_2 = F_2$	$Z_2 = 1$
$Z_1 = 0$	$(1, 1)$	$(-1, -1)$	$(-1, -1)$	$(-1, -1)$
$Z_1 = F_1$	$(-1, -1)$	$(1, 1)$	$(-1, -1)$	$(-1, -1)$
$Z_1 = F_2$	$(-1, -1)$	$(-1, -1)$	$(1, 1)$	$(-1, -1)$
$Z_1 = 1$	$(-1, -1)$	$(-1, -1)$	$(-1, -1)$	$(1, 1)$

Table 6.1: The Simple-Agreement Scoring Rule

and these kinds of trusted authorities both lack theoretically guaranteed credibility (beyond heuristics) and undermine decentralization. As long as we want to design a fully decentralized system with no trusted authorities, the ground truth of proofs' validity may be inaccessible and payments can only be decided by consensus among the voting parties. For reference, we defer detailed discussion on existing designs of decentralized AI protocols to Appendix D.1.

In the operations literature, the technique of *peer prediction* [138] refers to a wide scope of incentive mechanisms to elicit honest information without access to ground truth, which is widely adopted in the applications of dataset acquisition [139], peer grading [159], and also recent blockchain applications [140, 160]. A general paradigm of peer prediction is to ask multiple players the same question (or overlapping question sets) and reward each player based on the comparison between her report Z_i and other players' reports $\mathbf{Z}_{-i}$ according to a subtly designed *scoring rule*. As a toy example, in a 2-player simple-agreement scoring rule, the two players receive $+1$ when their reports agree, and receive -1 otherwise. In this case, the scoring rule $R_i(Z_1, Z_2)$ is shown as Table 6.1.

Whereas the simple-agreement scoring rule may not theoretically incentivize truthful reporting in all scenarios, the general purpose of peer prediction studies (including our research) is to design refined scoring rules that secure such incentive guarantees (for example, Table D.4 in Appendix D.4.) Nevertheless, whereas existing peer prediction mechanisms are designed to elicit truthful reports in the absence of ground truth, the following challenges occur in our setting for blockchain and particularly decentralized AI applications:

- Costly observation: Most traditional peer prediction mechanisms are designed to elicit truthful *reporting* without considering observation costs, but in our setting we need to incentivize the verifiers to make costly computational efforts to verify on-chain contents, particularly for decentralized AI applications such as opML and PoL in which the verification processes are computationally intensive (as discussed in Appendix D.1).

- Robustness: Most traditional peer prediction mechanisms have strong assumptions that may not apply in the decentralized setting of blockchain ecosystems, particularly when players are anonymous and may be adversarial. Particularly, we need ***permutation-proofness*** to disincentivize malicious reporting, ***Byzantine robustness*** against adversarial peer verifiers, and ***distributional robustness*** against dishonest proofs.

A widely adopted paradigm in literature is using mutual-information-based scoring rules [8, 161]. While these types of scoring rules provide *permutation-proofness* and are convenient for usage in scenarios with known prior, they do not explicitly consider costly observation, and their Bayesian Nash equilibria do not ensure the *Byzantine robustness* if a small fraction of peers are malicious. Furthermore, they also have a gap from resolving the Verifier's Dilemma due to the lack of *distributional robustness*, as the logarithm-based scoring rules are sensitive to low probabilities. In the practice of blockchain systems, as the Verifier's Dilemma leads to an arbitrarily low cheating probability ϵ, its empirical value becomes difficult to estimate and that sensitivity will severely undermine the robustness of the reward mechanism.

In contrast to most peer prediction mechanisms that guarantee Bayesian Nash equilibria requiring a known prior, a recent work [10] designs a determinant-based mutual information (DMI) mechanism without the need of prior information. However, it does not satisfy *permutation-proofness*: a verifier who systematically flips all her reports can still obtain optimal rewards, rendering it inapplicable for blockchain verification. Conceptually, the DMI mechanism is motivated to elicit *informative* (non-lazy) feedback rather than *trustworthy* (non-lazy and also non-adversarial) ones. As the study of Kong and Schoenebeck [8] shows that no peer prediction mechanism can satisfy prior-free and permutation-proof properties simultaneously, the requirement of approximate prior knowledge remains necessary in the application of decentralized verification games. Hence, assuming approximate prior knowledge is theoretically justified in addressing the Verifier's Dilemma in blockchain applications.

6.1.1 Our Contribution

In this research, we develop a theoretical framework with modeling of *decentralized verification game (DVG)*, and initiate the study that combines the ideas of *flags* and peer prediction into our proposed mechanism, named capture-the-flag peer prediction (CTF-PP), which only needs one phase in its procedure, and incentivizes honest verifying and reporting via simultaneously satisfying the following properties that explicitly consider *observation costs*:

- Interim *unique* incentive compatibility (interim UniIC): A verifier, after performing the verification, maximizes her expected utility when she reports honestly. Furthermore, if she reports a different type from her observation, her expected utility is non-positive.
- Interim individual rationality (interim IR): A verifier, after performing the verification, gets a non-negative expected net utility, when she acts honestly.
- Interim no-free-lunch (interim NFL): A verifier cannot get a positive expected utility via any *uninformed strategy* [162], i.e., without doing the verification.

Combining all the desired properties, we characterize the notion of *incentive alignment* (δ-IA) as a general guideline for peer prediction mechanisms in decentralized verification games, which depicts the property that any pure strategy gains a positive interim utility if *and only if* it is honest, with a margin of δ (details in Section 6.3.3) aimed for Byzantine robustness and distributional robustness. With this stronger incentive guarantee, we can ensure that the peer prediction mechanism works as desired for the tricky setting of decentralized environments in blockchains, with an additional guarantee to *disincentivize free-riding behavior* in blockchain systems, particularly reinforcing economic foundations of decentralized AI ecosystems in which verification can be costly.

We show the comparison of our design to existing peer prediction mechanisms in Table 6.2. Beside theoretical derivations, we also perform extensive numerical experiments to show the effectiveness of our design in comparison with existing peer prediction mechanisms in different scenarios. (Section 6.7 and Appendix D.4-D.5)

	Prior	Observ. Cost	Perm. Proof	Byzan. Robust[3]	Distr. Robust[4]
Log-Based[5]	Needed	×	✓	×	×
DMI[6]	Free	×	×	$n-1$	1
Ours	Only Approximate	✓	✓	$\Theta(n)$	$\Theta(1)$

Table 6.2: Comparison of Peer Prediction Mechanisms for DVG.

Our technical contributions can be summarized as follows:

1. We formulated the desired incentive guarantees (δ-IA) of 2-verifier decentralized verification games (DVG) with a linear program (LP), and then illustrated the general feasibility of the linear program via a generalization of the Cremer-McLean mechanism [163]. Furthermore, we extend our methodology for general n-verifier DVG, showing a basic solution for DVG that incentivizes honest verification and reporting, considering *observation costs* and satisfying *permutation proofness* (Section 6.4).

2. More crucially, we discuss the *Byzantine robustness* properties and develop a general guideline of the *compactness* criteria that *wide incentive margins and relatively low rewards/penalties* realize good Byzantine robustness against malicious verifiers, and design an extended LP that additionally optimizes the compactness and achieves Byzantine robustness against an $\epsilon = O(1)$ fraction of malicious verifiers. We also show that the compactness criteria simultaneously realizes budget efficiency as such robustness guarantees only require an additional budget of $O(\epsilon)$. (See in Section 6.5)

3. Then, we observe the *Byzantine reduction* principle showing that inaccurate beliefs and priors can be statistically reduced to existence of malicious players via the *coupling argument*, and leverage this principle to connect the *distributional robustness* against inaccurate priors/beliefs with the desiderata of the Byzantine robustness, showing the generality of our compactness criteria for robust peer prediction mechanisms. (See in Section 6.6)

[3]The maximum number of malicious verifiers that can be tolerated, n denoting the total number of verifiers.

[4]The maximum noise in prior distributions that can be tolerated, in the sense of total variation (TV) distance.[Move to caption]

[5]Including family of Shannon entropy-based peer prediction mechanisms that use logarithm-based scoring rules, e.g., Kong and Schoenebeck [8], Zheng et al. [161].

[6]The mechanism proposed by Kong [10].

6.2 Background and Related Work

Since the emergence of Bitcoin [78], the concept of blockchain is inherently designed as an unalterable distributed ledger that maintains trustworthiness via decentralized consensus. The blockchain can be modeled as a growing linked list stored by decentralized nodes, in which each *block* contains its contents that consists of *transactions*, a hash reference of its previous block, and a certificate (e.g., PoW, PoS and etc.) that controls the access to the block. Conceptually, when a block producer, also called a *miner*, would like to pack and propose new contents on the blockchain, she needs to attach the block to a previous block, and pay certain efforts to gain access to produce the block. When a miner attaches a new block to an existing block, she is supposed to have also *verified* the validity of the previous block. This process also makes the previous block unalterable, because the new block would be stored and witnessed by all the nodes of the network.

Nevertheless, in real-world blockchain ecosystems, the verifiers may be economically rational or selfish. In this context, the Verifier's Dilemma occurs. For example, Cao et al. [164] propose an attack that leverages the Verifier's Dilemma to double spend in Bitcoin. Besides, the studies of Smuseva et al. [133], Alharby et al. [165] make extensive analyses on Ethereum and the results show that Ethereum verifiers are frequently incentivized not to verify the contents while they are supposed to, rendering the Ethereum protocol vulnerable.

That said, one may argue that in the original design of Bitcoin or Ethereum, the verification of a block has negligible costs as it only needs the miner to check if all the transactions and the Proof-of-Work (PoW) or Proof-of-Stake (PoS) is valid. Since invalid blocks can be detected easily, miners might practically decide to behave honestly even if it is (slightly) irrational. Nevertheless, the development of the blockchain technology generalized the usage of blockchain system from an unalterable ledger of monetary transactions to a general decentralized platform that guarantees the integrity of diverse contents, e.g., smart contracts [166, 167, 168], and furthermore, with the recent rapid development of AI technologies and the demands of trustworthy AI models, researchers are actively exploring to establish blockchain-based platforms that verify the computation of machine learning [4, 96, 130, 150, 169], which brings new motivations for blockchain studies as a novel paradigm of

decentralized trustworthy AI.

Unlike hash puzzles in the Bitcoin PoW, the verification of such complicated contents can be potentially costly. Particularly, in the context of ML verification, Fang et al. [100] show that efficient byzantine-secure verification of stochastic gradient descent (SGD) computation reduces to fundamentally hard open problems in deep learning theories. Even though the study of Zhao et al. [4] achieves substantially lower verification overheads via the relaxation to incentive-security, the verification protocol still needs to reproduce the training process of at least $\Theta(1)$ epochs which has non-negligible computational costs. Consequently, recent studies typically resort to weaker incentive properties for verification games. For example, the recent proposal of opML [130], a protocol that designs for trustworthy ML inference on blockchain, can only reach a mixed-strategy Nash equilibrium that a (small) constant fraction of provers and verifiers behave dishonestly, and this fraction scales up with the verification cost (as discussed in Introduction).

Because the Verifier's Dilemma, unless suitably addressed, appears as a fundamental vulnerability in the incentive structure of blockchains that may severely undermine the reliability of blockchain systems, the studies of Zhang et al. [131], Teutsch and Reitwies,s'ner [136] work on this issue via introducing deliberate invalid objects as *attention challenges* that incentivize verification. Nevertheless, their protocols are multi-phased as they need additional dispute processes and are potentially restricted to particular applications. In our work, we are motivated to design a one-phase general-purpose and oracle-free solution to the Verifier's Dilemma with theoretical incentive guarantees, expecting to resolve the critical incentive issue in decentralized verification games in a reliable and efficient paradigm.

6.3 Basic Modeling of Decentralized Verification Games

To initiate the study, we first formulate the modeling of decentralized verification games (DVG). In a n-verifier DVG, there are n homogeneous players (verifiers) $i = 1, \cdots, n$ independently verifying an on-chain *proof*, and we use the terms "player" and "verifier" interchangeably.

The proof has an underlying ground-truth type $\theta \in S$, which can be either "Honest"

($\theta = 0$), "Flag j" ($\theta = F_j$, for $j = 1, 2, \cdots, m$), or "Dishonest" ($\theta = 1$). We define $S_* = \{0, F_1, \cdots, F_m\}$ as the set of all non-dishonest types. The actual type θ is unknown to both the verifiers and the system. While the observations $\{X_i\}$ can potentially be noisy, every verifier's observation, when they actively verify the proof, is *i.i.d.* conditioned on θ with known distributions $\{P(X_i|\theta)\}$.[7] Since the system can insert flags to maintain a pre-set flag rate (as in Zhao et al. [4], Teutsch and Reitwies,s'ner [136], Smuseva et al. [154], etc.) that robustly incentivizes verification when no cheater occurs, we have *principal* prior probabilities $P(\theta = F_i) = p_{F_i}$ and $P(\theta = 0) = p_0$ as publicly known information, with $\sum_{\theta \in S^*} P(\theta) = 1$ and the principal probability of $\theta = 1$ is zero. Throughout the paper, the term "principal" refers exclusively to this cheater-free scenario.

However, in reality, the prior distribution slightly deviates from the principal scenario as there is a small but unknown probability $\epsilon \in [0, \epsilon_0]$ that the proof is dishonest, i.e., $P(\theta = 1) = \epsilon$, with a known upper bound ϵ_0. We assume that the appearance of dishonest proofs may take up the probabilities of types in S_* in an arbitrary way. Hence, for any $s \in S_*$ we have $P(\theta = s) = p_s - \epsilon_s$, in which $\sum_{s \in S_*} \epsilon_s = \epsilon$ but the exact values of $\{\epsilon_s\}$ remain unknown.

Similar to [4], we begin our discussion with the verification process in a *lossy-channel* model as follows, and then study the general case (as shown in Theorem 6.2). When verifier i verifies the proof, the distribution of the observation X_i is dependent on θ in this way:

- ***Completeness***: A honest proof is always observed as honest, i.e. $P(X_i = 0|\theta = 0) = 1$.
- ***Probabilistic soundness***: A dishonest proof can be observed as any type, but the probabilities are known to the public, and the probability of correct detection at least $\kappa > 0$, i.e. $P(X_i = 1|\theta = 1) \geq \kappa$.
- ***Benign flags***: A flag F_j can be detected with known probability $\mu_j > 0$ or missed and observed as honest, but will never be observed as dishonest or other flags, i.e. $P(X_i = F_j|\theta = F_j) = \mu_j$ and $P(X_i = 0|\theta = F_j) = 1 - \mu_j$.

[7] The symmetry/homogeneity among verifiers can be assumed both according to the fixed verification protocols and the anonymity of decentralized systems.

Verification protocol. For each verifier i, she first needs to *stake* a pre-specified amount L of tokens to the system, and is informed that she will be rewarded based on a public *scoring rule* denoted as $R_i(Z_i, \mathbf{Z}_{-i})$, in which Z_i is her report, $\mathbf{Z}_{-i}$ is the collection of other verifiers' reports, and the maximum possible penalty cannot exceed the staked amount, i.e., $R_i(Z_i, \mathbf{Z}_{-i}) \geq -L$. Then, the verifier makes a decision to follow one of the following strategies, or any mixture between them:

1. Informed (active) strategy: Actively verifies the proof and gets the observation, which gains her access to X_i but incurs a publicly-known cost $c(X_i) \geq 0$ which can depend on X_i.

2. Uninformed (lazy) strategy: Does not verify and has no access to X_i. For convenience of expression, we can denote $X_i = \perp$ in this case, and $c(\perp) = 0$.

Hence, verifier i's Bayesian belief $\mathscr{B}$ on $\mathbf{Z}_{-i}$ is the conditional distribution of $P(\mathbf{X}_{-i}|X_i)$ for the informed strategy, or the marginal distribution $P(\mathbf{X}_{-i}|\perp) = P(\mathbf{X}_{-i})$ for the uninformed strategy. Here, i's belief of the cheating probability can be an arbitrary $\epsilon^{(i)} \in [0, \epsilon_0]$ that can be different from the actual ϵ, with arbitrary $\{\epsilon_s^{(i)}\}$ such that $\sum_{s \in S_*} \epsilon_s^{(i)} = \epsilon^{(i)}$, and we desire to design a mechanism that uniformly satisfies the incentive guarantees for arbitrary $\{\epsilon_s^{(i)}\}$. In this context, we define the *principal belief* as the Bayesian belief given $\forall \epsilon_s^{(i)} = 0$, i.e., $\epsilon_0 = 0$.

Then, verifier i reports a Z_i that maximizes $\mathbb{E}_{\tilde{\mathbf{Z}}_{-i} \sim \mathscr{B}}[R_i(Z_i, \tilde{\mathbf{Z}}_{-i})]$ in which $\mathscr{B}$ is her belief of $\mathbf{Z}_{-i}$, and claims that Z_i is her observation. After each verifier i independently reports Z_i without seeing $\mathbf{Z}_{-i}$[8], the system has the information of $Z_1, \cdots, Z_n$, but not θ, and rewards each prover i according to the scoring rule $R_i(Z_i, \mathbf{Z}_{-i})$. If $R_i(Z_i, \mathbf{Z}_{-i}) < 0$, the penalty will be deducted from her staked tokens. Then, verifier i's net utility is $R_i(Z_i, \mathbf{Z}_{-i}) - c(X_i)$.

Formal characterization of strategies and utilities. For the action of any verifier i, she may first decide to be active (informed) or lazy (uninformed). If she chooses to be lazy, then she can choose a distribution $D \in \Delta(S)$, in which $\Delta(S)$ is the set of all convex combinations of elements in S, and report $Z_i \sim D$. If she chooses to be active, she may

[8]This can be implemented via a cryptographic commitment scheme.

observe X_i and report according to a respective distribution corresponding to each $X_i \in S$, so any informed strategy can be characterized by a mapping $D(\cdot) : S \to \Delta(S)$. Furthermore, the verifier can also randomly decide to be active or lazy. Hence, we formally characterize the verifiers' strategies as:

Definition 23. *The strategy space of any verifier can be characterized as*

$$\Omega = \Omega(S) = \Delta(\{\Delta(S), \Delta(S)^S\}),$$

and we denote the strategy of verifier i as s_i.

For example, $s_i \in \Delta(S)$ if i chooses an uninformed strategy, and $s_i \in \Delta(S)^S$ if i chooses an informed strategy. Otherwise, if s_i is a random choice between informed and uninformed strategies, we can represent s_i with the 3-tuple (λ, μ, α), denoted as

$$s_i \triangleq (\lambda, \mu, \alpha),$$

in which $\lambda \in \Delta(S), \mu \in \Delta(S)^S, \alpha \in (0, 1)$ and $s_i = \alpha \cdot \lambda + (1 - \alpha) \cdot \mu$.

For any verifier, she would maximize her expected utility based on her *belief* on the reports of other verifiers. The belief profile of verifier i can be characterized as $\mathbb{B}_i : (S \cup \{\perp\}) \to \Delta(S^{n-1})$ which maps her observation to a joint distribution of other verifiers' reports. For *Bayesian* verifiers, they always set their belief as $\mathbb{B}_i(X_i) = P(\mathbf{X}_{-i}|X_i)$. Then, with regard to belief $\mathbb{B}_i$, we can define the interim utility $u_i(s_i; \mathbb{B}_i)$ as:

$$u_i(s_i; \mathbb{B}_i) = \begin{cases} \mathbb{E}_{Z_i \sim s_i, \mathbf{Z}_{-i} \sim \mathbb{B}_i(\perp)}[R_i(Z_i, \mathbf{Z}_{-i})], & s_i \in \Delta(S); \\ \mathbb{E}_{X_i \sim P(X_i)}[\mathbb{E}_{Z_i \sim s_i(X_i), Z_{-i} \sim \mathbb{B}_i(X_i)}[R_i(Z_i, \mathbf{Z}_{-i}) - c(X_i)]], & s_i \in \Delta(S)^S; \\ \alpha \cdot u_i(\lambda; \mathbb{B}_i) + (1 - \alpha) \cdot u_i(\mu; \mathbb{B}_i), & s_i \triangleq (\lambda, \mu, \alpha). \end{cases} \tag{6.1}$$

In most parts of this paper (except for the discussion of Byzantine robustness and Appendix D.2), we always assume that all verifiers have principal Bayesian beliefs $\mathbb{B}_i(X_i) = P(\mathbf{X}_{-i}|X_i)$ assuming $\epsilon = 0$ (we justify using $\epsilon = 0$ in place of small unknown $\epsilon > 0$ in the robustness analysis of Section 6.6). Hence, we simplify the notation as $u_i(s_i)$. We call a strategy *honest*

or *truthful* if and only if $s_i = I$ is (induced by) the identity map on S, i.e. $s \in \Delta(S)^S$ and $s(X_i) \equiv X_i$.

Pure & mixed strategies. In the strategy space Ω, we define the subset $\Omega_d = S \cup S^S$ as the space of *pure* strategies, in which the verifier deterministically decides to be informed or uninformed, and reacts deterministically to her observation. From the linearity of the utility function, we can see that the optimal utility is always realized by a pure strategy for any fixed belief, and hence we mainly consider pure strategies throughout this paper.

6.3.1 IR and UniIC Constraints for Informed Verifiers

The IR constraint requires that given the verifier i observes X_i, truthfully reporting it gains her an expected reward no less than $c(X_i)$. Since X_i and X_{-i} are independent conditioned on θ, define $r_{X_i}(Z_i)$ as verifier i's expected reward of reporting Z_i conditioned on observing X_i, then $r_{X_i}(Z_i)$ can be computed as

$$r_{X_i}(Z_i) = \sum_{\mathbf{X}_{-i} \in S} R_i(Z_i, \mathbf{X}_{-i}) P(\mathbf{X}_{-i} | X_i) \tag{6.2}$$

$$= \sum_{\mathbf{X}_{-i} \in S} R_i(Z_i, \mathbf{X}_{-i}) \frac{P(X_i, \mathbf{X}_{-i})}{P(X_i)} \tag{6.3}$$

$$= \sum_{\mathbf{X}_{-i} \in S} R_i(Z_i, \mathbf{X}_{-i}) \frac{\sum_{\theta \in S} P(\theta) P(X_i|\theta) P(\mathbf{X}_{-i}|\theta)}{\sum_{\theta \in S} P(\theta) P(X_i|\theta)}. \tag{6.4}$$

While the probabilities are dependent on ϵ, as long as ϵ is small enough, for $X_i \in S_*$, the r_{X_i} is a continuous function w.r.t. ϵ, so the IR constraints on $X_i \in S_*$ can be implied by

$$\sum_{\mathbf{X}_{-i} \in S} R_i(X_i, \mathbf{X}_{-i}) \frac{\sum_{\theta \in S} \tilde{P}(\theta) \tilde{P}(X_i|\theta) \tilde{P}(\mathbf{X}_{-i}|\theta)}{\sum_{\theta \in S} \tilde{P}(\theta) \tilde{P}(X_i|\theta)} \geq c(X_i) + \delta, \qquad \forall X_i \in S_*, \tag{6.5}$$

in which $\tilde{P}$ denotes the *principal* probabilities assuming $\epsilon = 0$, and the margin δ is introduced to ensure the incentive guarantees even if the actual priors slightly deviate from the principal priors, so that for any $\delta > 0$, the constraints hold robustly for some $\epsilon_0 > 0$. The condition is also necessary when $\delta = 0$. In the rest of this paper, we say a condition is "sufficient and almost necessary" when it is sufficient with a $\epsilon_0 > 0$ depending on δ, and when we set $\delta = 0$,

it becomes a necessary condition. We defer the rigorous justification of the "sufficiency" and quantitative analysis on the relations between δ and ϵ_0 to Section 6.6.

For the case of $X_i = 1$, from the lossy-channel model, we know that $\theta = 1$. Therefore, we have

$$r_1(1) = \sum_{X_{-i} \in S} R_i(1, \mathbf{X}_{-i}) P(\mathbf{X}_{-i} | \theta = 1) \geq c(1) + \delta. \tag{6.6}$$

So Eqs. (6.5-6.6) are sufficient and almost necessary conditions that a CTF-PP mechanism is IR.

For the IC constraint, we need and only need $r_{X_i}(X_i) = \max_{Z_i \in S}\{r_{X_i}(Z_i)\}$. With similar arguments, we can also develop sufficient and almost necessary conditions that a CTF-PP mechanism is IC. Actually, given that the IR is satisfied we can define a stronger notion of *Uniquely-IC* as follows:

- Uniquely IC (UniIC): In addition to the IC requirement, given all other verifiers act honestly and a verifier actively performed the verification, then she gets a negative expected utility when she reports any type different from her observation.

Besides conventional IC notions, the UniIC requirement additionally rules out the possibility that a dishonest verifier cheats the system without losing money. Assuming that the IR constraints are already satisfied, the UniIC constraints can be formulated as:

$$r_{X_i}(Z_i) \leq c(X_i) - \delta, \qquad \forall X_i \in S,\ Z_i \neq X_i. \tag{6.7}$$

6.3.2 NFL Constraints for Uninformed Verifiers

We assume verifiers other than i are honest, i.e. they all decide on the informed strategy and $\mathbf{Z}_{-i} = \mathbf{X}_{-i}$. If verifier i performs the uninformed strategy, she has no information on θ and her strategy can only be reporting any type in $S = \{0, F_1, \cdots, F_m, 1\}$, or any convex combination of them. Hence, i's utility when she lazily reports Z_i is denoted as:

$$r_\perp(Z_i) = \sum_{\mathbf{X}_{-i} \in S} R_i(Z_i, \mathbf{X}_{-i}) P(\mathbf{X}_{-i}). \tag{6.8}$$

From the NFL requirement and assuming small $\epsilon \leq \epsilon_0$, a sufficient and almost necessary condition is the following linear constraints

$$\sum_{\mathbf{X}_{-i} \in S} R_i(Z_i, \mathbf{X}_{-i}) P(\mathbf{X}_{-i}) \leq -\delta, \qquad \forall Z_i \in S. \tag{6.9}$$

6.3.3 Incentive Alignment for Decentralized Verification Games

From the discussion in Section 6.3.1-6.3.2, we would like to design a mechanism for the decentralized verification game that simultaneously satisfies IR, UniIC, and NFL constraints. Combining the derivations above, we can summarize the sufficient and almost necessary conditions that satisfy all constraints above. Hence, we define the notion of *incentive alignment* (δ-IA) as follows:

Definition 24 (Incentive Alignment)**.** *A CTF-PP mechanism is δ-incentive-aligned (δ-IA) if and only if for any verifier i and pure strategy $s_i \in \Omega_d$,*

$$u_i(s_i) \begin{cases} \geq \delta, & s_i = I; \\ \leq -\delta, & s_i \neq I. \end{cases}$$

Equivalently,

$$r_{X_i}(Z_i) - c(X_i) \begin{cases} \geq \delta, & Z_i = X_i; \\ \leq -\delta, & Z_i \neq X_i. \end{cases}$$

Here, the δ-IA is a sufficient and almost necessary condition that IR, UniIC and NFL are simultaneously satisfied.

6.4 Theoretical Guarantee for DVG: LP Modeling and Feasibility

In this section, we show a basic result on the existence of incentive aligned CTF-PP mechanisms for any 2-verifier DVG that satisfies mild conditions, and then generalize our design to a general setting of n verifiers.

6.4.1 The 2-verifier Case

Assume $\epsilon = 0$, and define the *principal* belief matrix $B : S^2 \to \mathbb{R}$ as $B_{xy} = P(X_{-i} = y | X_i = x)$.[9] Besides, we define $B_\perp$ as the blind-belief (row) vector as $B_{\perp y} = P(X_{-i} = y)$ that describes the belief of verifier i when she does not verify the proof. Then, we can formulate the design of a δ-IA CTF-PP mechanism as a linear programming (LP) problem.

We define decision variable as the scoring matrix $T : S^2 \to \mathbb{R}$ with $T_{xy} = R_i(x, y)$, and denote

$$W = BT'. \tag{6.10}$$

Then $W_{xy} = r_x(y)$, which is the expected reward verifier i gets from the mechanism when she observes x and reports y. The IR and UniIC conditions are equivalent to the following:

$$W_{xx} \geq c(x) + \delta, \qquad \forall x \in S; \tag{6.11}$$

$$W_{xy} \leq c(x) - \delta, \qquad \forall x \in S, \quad y \in S - \{x\}. \tag{6.12}$$

Similarly, we denote

$$W_\perp = B_\perp T', \tag{6.13}$$

then $W_{\perp y} = r_\perp(y)$ is the expected reward verifier i gets when she does not verify and lazily reports y. Then the NFL conditions are equivalent to the following:

$$W_\perp \leq -\delta. \tag{6.14}$$

Hence, we only need to find a feasible solution of the linear system (6.10-6.14), i.e., solve the following linear program:

$$\begin{aligned} LP_0: \quad & minimize \quad 0 \\ & s.t. \quad \text{(6.10-6.14)}. \end{aligned}$$

Here, inspired by the Cremer-McLean mechanism [163], we propose our basic theorem that shows the feasibility of LP_0, with the proof deferred to Appendix D.6.2:

[9] $B_1 y$ can still be defined even if $P(\theta = 1) = \epsilon = 0$, e.g. $\theta = 1$ when a zero-measure set is drawn.

Theorem 6.2 (Basic Theorem)**.** *If B is invertible, and $P(X_i = y|\theta = 1) = 0$ for any $y \neq 1$ (i.e., a non-cheating proof is never observed as a cheat), then for any $\delta \geq 0$, we can find a δ-IA mechanism for the 2-verifier DVG as a feasible solution of LP_0.*

For some $\epsilon_0 > 0$, the mechanism is IR, NFL and UniIC for any $\epsilon \in [0, \epsilon_0]$.

Particularly, we show that our method always works for the DVG with the lossy-channel model as defined in Section 6.3 with the following proposition. The proof is deferred to Appendix D.6.3.

Proposition 6.3. *In the lossy-channel model defined in Section 6.3, the principal belief matrix B is invertible.*

6.4.2 General n-Verifier Case

We have just shown that under mild assumptions there always exists an incentive-aligned mechanism for any 2-verifier DVG. In this part we invoke the 2-verifier mechanism as a building block and construct our mechanism for the general setting of n verifiers.

Vectorized notation. In the 2-verifier game, the (Z_i, Z_j)-th entry of the matrix T, denoted as $T_{Z_iZ_j}$, depicts the reward of verifier i when she reports Z_i while the other verifier j reports Z_j. With a slight abuse of notation, if we regard each type in S as a unit one-hot column vector in the corresponding dimension, we can get $T_{Z_iZ_j} = Z_i'TZ_j$. In the general case of n verifiers, we use a pairwise-scoring mechanism that compares every verifier's report with the *average* of other verifiers', which can be formulated as:

$$R_i(Z_i, \mathbf{Z}_{-i}) = Z_i'T\overline{\mathbf{Z}_{-i}}. \tag{6.15}$$

Here, we denote

$$\overline{\mathbf{Z}_{-i}} = \frac{1}{n-1}\sum_{j\neq i} Z_j. \tag{6.16}$$

Then, we can show that the pairwise-scoring mechanism as described as Eq. (6.15) has equivalent incentive structures as the 2-verifier mechanism characterized as T. Formally, we have:

Theorem 6.4. *If the scoring matrix T satisfies the δ-IA property for the 2-verifier DVG, then the scoring rule as Eq.* (6.15) *also satisfies δ-IA for the general n-verifier DVG.*

Furthermore, if the 2-verifier mechanism characterized as T is IR, NFL and UniIC for any $\epsilon \in [0, \epsilon_0]$, then the n-verifier mechanism in Eq. (6.15) *also satisfies the same properties.*

The proof of Theorem 6.4 is deferred to Appendix D.6.4.

6.5 Byzantine Robustness via Margin Optimization

In the context of (Bayesian) Nash equilibria, we aim to design mechanisms in which no agent may benefit from *individual* deviations. In other words, we guarantee that each verifier is incentivized to be honest given that *all* others are honest. However, in decentralized ecosystems like blockchains, this assumption may be too strong as there may exist *malicious* players who would deliberately attack the system, i.e., trying to undermine the robustness of the system at the risk of losing their own utilities. Furthermore, just like the widely studied topic of blockchain *transaction fee mechanisms* (See, e.g., Chen et al. [3], Roughgarden [37], Chung and Shi [41], Roughgarden [42], Wu et al. [45]), blockchain players may also potentially collude with each other or create fake identities to increase their utilities. Because the blockchain consensus protocols, e.g., PoW or PoS, can inherently address the Sybil attack issue, in this study we mainly consider non-Sybil dishonest players who do not conduct Sybil attacks but may act adversarially otherwise.

Whereas it may be too strong to assume that all other players are individually rational, in the field of decentralized systems, the notion of *Byzantine robustness* (See, e.g., Yin et al. [170], Wu et al. [171], Chen et al. [172]), also called *Byzantine fault tolerance* or *Byzantine resilience*, is widely studied as a desired property that the system works robustly as expected even if a (small) portion of the system does not work correctly. In the works of Wang et al. [140], Schoenebeck et al. [173], the existence of colluding players is also considered for peer prediction mechanisms. Particularly, Schoenebeck et al. [173] consider the multi-task setting and tackle with it as a *robust learning* problem, and Wang et al. [140] focus on the specific *leader election* problem [174] for blockchain consensus. In another perspective, Frongillo

and Witkowski [175] look into the scenario of peer prediction with inaccurate distributional knowledge, and develop a margin optimization methodology to maximize the tolerance to inaccurate posterior beliefs.

Inspired by these studies, we are motivated to design a general-purpose solution for decentralized consensus with stronger *incentive alignment* guarantees, with an optimization framework of Byzantine robustness via the the *compactness* criteria (See in Section 6.5.2.) Furthermore, we show the budget efficiency of our design in Section 6.5.5, and will show in Section 6.6 the generality of our Byzantine robustness notion as it can also imply *distributional robustness*. Following the framework in the study of Schoenebeck et al. [173], the types of players can be generally classified into the following categories:

1. Altruistic ($\mathscr{A}$): Acting honestly without consideration of utilities;
2. Selfish ($\mathscr{S}$): Acting in the way that maximizes their own utilities;
3. Colluding ($\mathscr{C}$): Conducting collusions with other players (within $\mathscr{C}$) to maximize their joint utility;
4. Malicious ($\mathscr{M}$): Acting in arbitrarily manners that may not optimize their utilities, without access of non-malicious players' information.

While traditional game theory primarily focuses on the behavior of $\mathscr{A}$ and $\mathscr{S}$ players, $\mathscr{C}$ and $\mathscr{M}$ players typically fall outside the scope of its standard models. Therefore, we call $\mathscr{A}, \mathscr{S}$ players as **benign** and $\mathscr{C}, \mathscr{M}$ players as **rogue**.

Intuitively, we would like to design the mechanism in the following way: as long as rogue verifiers only constitute a small portion, all four types of verifiers are incentivized to act honestly, even though malicious players may actually act in different manners at the cost of their own utilities. However, we still assume that malicious players cannot access non-malicious players' information (e.g. observations and reports) as the unauthorized access of non-malicious players' information should be prevented by the system design, and also breaks the basic model of Bayesian games.

6.5.1 Characterization of Robust Incentive Properties

To ensure that arbitrary actions of malicious players do not affect the incentive guarantees of other players even in the worst case, we introduce the notion of *robust incentive properties* describing the scenario in which the incentive properties holding uniformly for any possible realization of malicious players' actions.

In this case, we can define ϕ as the *environmental variable* that depicts the prior probabilities, the number of $\mathscr{A}, \mathscr{S}, \mathscr{C}, \mathscr{M}$ players, and the strategies of $\mathscr{M}$ players. While players might not have the exact information of ϕ, they do have the knowledge that ϕ lies in a set Φ of *environmental assumptions*, e.g., the total number of $\mathscr{C}, \mathscr{M}$ players does not exceed a particular threshold.

The motivation of *robust incentive properties* is to guarantee that the players will not regret their honest actions even if they learn the existence and actions of the malicious players *ex-post*, so that these malicious behavior would not affect the incentive guarantees for the majority of non-malicious players. In other words, in a mechanism with robust incentive guarantees, the desired properties hold uniformly for any $\phi \in \Phi$, similar to the framework of *distributionally robust optimization* [176].

In this context, given different environment variables, the player i would have different beliefs of the other players' reports, and we can characterize the belief profile of player i as $\mathbb{B}_i : (S \cup \{\perp\}) \times \Phi \to \Delta(S^{n-1})$, which maps the tuple of her observation and the environment variable to a joint distribution of other players' reports. Different from the model in Section 6.3, the players do not have a prior distribution of $\phi \in \Phi$. Hence, similar to the characterization of partial distributional knowledge in [177], for any strategy s_i, player i would actually have a *belief interval* $[u_i^{\Phi-}(s_i; \mathbb{B}_i), u_i^{\Phi+}(s_i; \mathbb{B}_i)]$ of its utility among all possible ϕ's, formulated as

$$u_i^{\Phi-}(s_i; \mathbb{B}_i) = \inf_{\phi \in \Phi} u^{\phi}(s_i; \mathbb{B}_i),$$

$$u_i^{\Phi+}(s_i; \mathbb{B}_i) = \sup_{\phi \in \Phi} u^{\phi}(s_i; \mathbb{B}_i),$$

in which

$$u_i^{\phi}(s_i;\mathbb{B}_i)=\begin{cases}\mathbb{E}_{Z_i\sim s_i,\mathbf{Z}_{-i}\sim\mathbb{B}_i(\perp,\phi)}[R_i(Z_i,\mathbf{Z}_{-i})], & s_i\in\Delta(S);\\ \mathbb{E}_{X_i\sim P(X_i)}[\mathbb{E}_{Z_i\sim s_i(X_i),Z_{-i}\sim\mathbb{B}_i(X_i,\phi)}[R_i(Z_i,\mathbf{Z}_{-i})-c(X_i)]], & s_i\in\Delta(S)^S;\\ \alpha\cdot u_i(\lambda;\mathbb{B}_i)+(1-\alpha)\cdot u_i(\mu;\mathbb{B}_i), & s_i\triangleq(\lambda,\mu,\alpha).\end{cases}\tag{6.17}$$

In this section, we also mainly consider *Bayesian* players with $\mathbb{B}_i(X_i,\phi)=P(\mathbf{X}_{-i}|X_i,\phi)$, and denote $u_i^{\phi}(s_i)$,$u_i^{\Phi-}(s_i)$, $u_i^{\Phi+}(s_i)$ in this case for simplicity. Then with the general guideline that the incentive properties should uniformly hold for any $\phi\in\Phi$, we define the notion of robust utility maximization and robust IA as follows:

Definition 25 (Robust Utility Maximization)**.** *In a fixed mechanism, a strategy s_i robustly maximizes player i's utility w.r.t. environmental assumption Φ, if and only if for any strategy $s_i'\in\Omega$,*

$$u^{\phi}(s_i)\geq u^{\phi}(s_i'),\qquad\qquad\forall\phi\in\Phi.$$

Definition 26 (Robust Incentive Alignment)**.** *A mechanism satisfies* robust *δ-IA w.r.t. environmental assumption Φ, if and only if for any player i and pure strategy $s_i\in\Omega_d$,*

$$\begin{aligned}u_i^{\Phi-}(s_i)&\geq\delta, & s_i&=I;\\ u_i^{\Phi+}(s_i)&\leq-\delta, & s_i&\neq I.\end{aligned}$$

Here, we denote I as the honest informed strategy, i.e., $s_i\in\Delta(S)^s$ and $s_i(X_i)\equiv X_i$.

From the notions above, we formally define the notion of $f(n)$-Byzantine-robustness ($f(n)$-BR) as follows:

Definition 27 (Byzantine Robustness)**.** *For a DVG with n verifiers, we call a mechanism $f(n)$-BR if and only if: as long as Φ constrains that the total number of rogue ($\mathscr{C}$ and $\mathscr{M}$) verifiers does not exceed $f(n)$,*

- *Each $\mathscr{A},\mathscr{S},\mathscr{C}$ verifier robustly maximizes her interim utility via acting honestly with robust 0-IA guarantees, assuming that other $\mathscr{A},\mathscr{S},\mathscr{C}$ verifiers act honestly.*

- *Each colluding party in $\mathscr{C}$ robustly maximizes their total interim utility via acting honestly, assuming that all $\mathscr{A}, \mathscr{S}$ verifiers and other colluding parties in $\mathscr{C}$ act honestly.*
- *Each $\mathscr{M}$ verifier would robustly maximize their interim utilities with robust 0-IA guarantees if she acted honestly, even though she may actually act otherwise, assuming that all $\mathscr{A}, \mathscr{S}, \mathscr{C}$ verifiers act honestly.*

In the rest of this section, we show that under mild assumptions, the design in Section 6.4.2, as long as the scoring matrix T comes from a "good" solution of the linear system Eqs. (6.11-6.14), is $\Theta(n)$-BR, i.e., resilient against a constant portion of rogue verifiers.

6.5.2 Bang for the Buck: Compactness Criteria for Byzantine Robustness

When we look at the pairwise-scoring mechanism Eq. (6.15), the reward of each verifier i is essentially based on the comparison of her report Z_i and the *average* of other verifiers' reports $\overline{\mathbf{Z}_{-i}}$. Intuitively, if only a small portion of other verifiers may act dishonestly, since their contribution to $\overline{\mathbf{Z}_{-i}}$ is not significant, the actual expectation of $\overline{\mathbf{Z}_{-i}}$ conditioned on X_i would not deviate significantly from $\mathbb{E}\left[\overline{\mathbf{X}_{-i}} \middle| X_i\right]$, and the δ margin in our design would make the reward matrix of i still satisfy *incentive alignment* properties even with a slightly perturbed posterior distribution of $\overline{\mathbf{Z}_{-i}}$.

For simplicity of discussion, in the family of rogue verifiers, we first only consider *simple malicious* ones who could not create fake identities but may act unpredictably and report in any strategy, as in the *canonical Byzantine setting* defined in Definition 28 below. In later sections, we will show that (certain types of) collusions can also be reduced to the canonical Byzantine setting (details in Section 6.5.4). For Sybil attacks, while no voting-based protocols can effectively prevent them if the attacker has unlimited resources (e.g., 51% attack [178]), each Sybil identity can also be regarded as a (new) malicious agent and, as long as they only make up a small portion of all verifiers, our framework of Byzantine robustness can prevent them from harming the incentive structure of other verifiers. Furthermore, since gaining additional voting power in PoW or PoS protocols has additional costs, we can show that as long as the total resources (e.g., computing power for PoW or stakes for PoS) of

the verifier only makes up a small portion of the network, she could not gain a significant advantage via Sybil attacks compared to honest behavior.

Definition 28 (Canonical Byzantine Setting)**.** *In a canonical Byzantine setting, each verifier acts in one of the following strategies:*

- *No-Sybil selfish ($\mathscr{S}_*$): Acting in a way that maximizes their own utilities, but unable to create fake identities.*
- *No-Sybil malicious ($\mathscr{M}_*$): Reporting arbitrarily, but unable to create fake identities.*

From the pairwise-scoring mechanism, we can see that conditioned on the verifier i observing $X_i \in S \cup \{\perp\}$, the expected utility of reporting Z_i is

$$r_{X_i}(Z_i) - c(X_i) = \frac{1}{n-1} \sum_{j \neq i} \left(\sum_{Z_j \in S} P(Z_j|X_i) T_{Z_i Z_j} - c(X_i) \right). \tag{6.18}$$

When verifier j is honest, we see that $Z_j = X_j$ and $\sum_{X_j \in S} P(Z_j|X_i) T_{Z_i Z_j} - c(X_i) = (BT')_{X_i Z_i} - c(X_i) = W_{X_i Z_i}$, and the δ-IA condition ensures that W's diagonal entries are at least $+\delta$ and other entries are at most $-\delta$. If j is dishonest, then her report Z_j may deviate from X_j, resulting in a different $\sum_{X_j \in S} P(Z_j|X_i) T_{Z_i Z_j}$ and leading to a perturbation to $r_{X_i}(Z_i)$.

Intuitively, if the summation of all these perturbations is bounded below δ, then $\{r_{X_i}(Z_i) - c(X_i)\}$ still has positive diagonal entries and negative non-diagonal entries, satisfying the robust incentive alignment property (with a smaller margin). On the other hand, we notice that $\sum_{X_j \in S} P(Z_j|X_i) T_{Z_i Z_j}$ is a convex combination of $\{T_{Z_i Z_j} : Z_j \in S\}$. If we upper bound the magnitude of the scoring rule, i.e.

$$|T_{Z_i Z_j}| \leq K, \quad \forall Z_i, Z_j \in S, \tag{6.19}$$

then we can deduce that

$$\sum_{Z_j \in S} P(Z_j|X_i) T_{Z_i Z_j} \in [-K, K]. \tag{6.20}$$

Hence, each dishonest verifier j can perturb the value of $r_{X_i}(Z_i)$ by at most $\frac{2K}{n-1}$, so a large incentive margin δ with a relatively small K would achieve a good *"bang for the buck"* for desired Byzantine-robust guarantees. In this sense, we define (δ, K)-compactness as:

Definition 29 ((δ, K)-compactness)**.** *For fixed observation costs $c(\cdot)$ and principal belief matrix B (denoted as the (c, B)-environment), a pairwise scoring matrix T is called (δ, K)-compact if and only if its entries are bounded within $[-K, K]$ and the corresponding mechanism is δ-IA.*

For convenience, we also call a mechanism (or a pairwise scoring matrix) $\frac{\delta}{K}$-compact if it is (δ, K)-compact for some (δ, K).

Then, we immediately derive the following lemma, showing that (δ, K)-compactness implies the Byzantine robustness against a $\Theta(\frac{\delta}{K})$ fraction of malicious players:

Lemma 6.1. *If a CTF-PP mechanism has a (δ, K)-compact pairwise scoring matrix, then it is $\frac{\delta}{2K}(n-1)$-BR in the canonical Byzantine setting, as it is 0-IA even in the presence of up to $\frac{\delta}{2K}(n-1)$ malicious players.*

In the following parts we focus on the construction of (δ, K)-compact pairwise scoring matrices with optimized $\frac{\delta}{K}$.

6.5.3 LP Modeling for Byzantine Robustness

In Section 6.4, we formulated the δ-IA condition for the 2-verifier DVG as the linear system of (6.11-6.14), and showed that the linear system is generally feasible, so that a desirable mechanism can be found via linear programming, and further showed that the LP solution generalizes to the n-verifier setting, so that our proposal is a general-purposed solution for the design of DVG mechanisms.

Whereas the general paradigm of incentive requirements in peer prediction can be depicted with the linear system, the LP problem actually allows us to optimize an *objective function*, which is not specified in previous parts. Considering the motivation of Byzantine robustness, from Lemma 6.1 we would like to construct a (δ, K)-compact scoring matrix with a large $\frac{\delta}{K}$. Hence, an intuitive idea is to fix δ and minimize K. For fixed principal belief matrix B,

prior distribution vector $B_\perp$ and observation costs $c(\cdot)$, we can formulate the LP problem $LP_1(B, B_\perp, c, \delta)$ with decision variable $\left(K \in \mathbb{R}, T \in \mathbb{R}^{S^2}\right)$ as:

$$
\begin{aligned}
&LP_1(B, B_\perp, c, \delta): \\
&\textit{minimize} \qquad K && (6.21)\\
&\textit{s.t.} \qquad |T| \le K, && (6.22)\\
&\qquad (BT')_{xx} \ge c(x) + \delta,\ \forall x \in S && (6.23)\\
&\qquad (BT')_{xy} \le c(x) - \delta,\ \forall x \in S,\ y \in S - \{x\} && (6.24)\\
&\qquad B_\perp T' \le -\delta. && (6.25)
\end{aligned}
$$

In fact, denoting $K_* = (B, B_\perp, c, \delta)$ as the optimal objective value of $LP_1(B, B_\perp, c, \delta)$, we can show an upper bound on $K_*(B, B_\perp, c, \delta)$ as:

Theorem 6.5. *Denote $c_1 = \max_{x\in S}\{c(x)\}$ as the maximum observation cost, $p_1 = \max\{B_\perp\} = \max_{x\in S}\{P(X_i = x)\}$ as the maximum prior probability of any observation, and $k = |S| = m + 2$ as the number of types, then we have*

$$
K_*(B, B_\perp, c, \delta) \le \|B^{-1}\|_2 \cdot (c_1 \cdot g_1(k, p_1) + \delta \cdot g_2(k, p_1)), \tag{6.26}
$$

in which

$$
g_1(k, p_1) = \sqrt{\left(1 + (k-1)\frac{p_1}{1-p_1}\right)\left(1 + \frac{1}{1-p_1}\right)} = O\left(\frac{\sqrt{kp_1}}{1-p_1}\right), \tag{6.27}
$$

$$
g_2(k, p_1) = \sqrt{\left(k + (2k-2)\frac{p_1}{1-p_1}\right)\left(k + \frac{2}{1-p_1}\right)} = O\left(\max\left\{k\sqrt{\frac{p_1}{1-p_1}}, \frac{\sqrt{kp_1}}{1-p_1}\right\}\right). \tag{6.28}
$$

are only dependent on k, p_1 but independent to n.

Additionally, there exists a feasible solution satisfying (6.26) *that makes the equality hold in* (6.23).

The proof of Theorem 6.5 is deferred to Appendix D.6.5.

6.5.4 Reduction of Colluding Players

For the characterization of Byzantine players, it is intuitive that colluding behavior is within the scope of malicious behavior, and *the resilience against malicious players should infer the resilience against colluding players.* While this proposition is true, the reduction is actually non-trivial.

From the classification of players, we only consider the *external* effects and *individual* incentives of malicious players, i.e., their existence does not disrupt the incentive alignment guarantees of *other* players or benefit individual utilities. However, in the consideration of colluding players, we still need to prevent them from gaining *total* utility via collusion, i.e., we also need to consider *internal* effects of collusion which is not covered in previous discussion. Similar to the Side-Contract-Proofness (SCP) notion proposed by Chung and Shi [41], we define *weak-SCP* in the scope of DVGs as follows:

Definition 30 (weak-Side-Contract-Proofness (weak-SCP))**.** *We further define $\mathscr{C}_*$ players as:*

- *No-Sybil colluding ($\mathscr{C}_*$): Conducting collusions with other players (within $\mathscr{C}_*$) to maximize their joint utility, but unable to create fake identities.*

*We call a mechanism weak-Side-Contract-Proof (*weak-SCP*) under some conditions, if and only if as long as these conditions hold, in any collusion party, the players robustly maximize their total interim utility w.r.t. their individual (non-shared) Bayesian beliefs ($P(X_{-i}|X_i)$) via acting honestly.*

We call this notion *weak*-SCP because although players collude with each other, they still update their posterior beliefs only based on their own observations, not considering other colluders' observations as they might not "fully trust each other". In turn, we call the collusion-proofness against colluders who share their beliefs based on their aggregated observations *strong-SCP*. Nevertheless, there are additional challenges in the design of strong-SCP peer prediction mechanisms and we leave it to future work. The intuition behind such challenges is that *a belief-sharing colluding party would be incentivized to report the same type even though they may have different observations.* On the bright side, from

Proposition 6.6, although belief-sharing colluders may benefit from collusions, they do not disturb the incentive guarantees of other players as long as they only constitute a small portion of all players. We defer detailed discussions on strong-SCP to Appendix D.2.

In actual cases, there may exist multiple colluding parties, but from the argument in the player classification, for any selfish player or colluding party that intend to maximize their total utility, other colluding players outside the party do not have access to their actions or observations, and only need to be considered w.r.t. their external effects. Hence, we can deduce that

Proposition 6.6. *Colluding players can be regarded as malicious players from the perspective of players outside their colluding parties.*

With Proposition 6.6, we only need to consider the existence of one colluding party of collusion players, beside a (small) number of malicious players. Formally, we have the following theorem:

Theorem 6.7. *Assume that there are n players, among which are $|\mathscr{M}_*|$ no-sybil malicious players and a no-sybil colluding party of $|\mathscr{C}_*|$ players. If the CTF-PP mechanism has a (δ, K)-compact scoring matrix, then the mechanism is 0-IA and weak-SCP as long as*

$$|\mathscr{M}_*| + |\mathscr{C}_*| \leq \frac{\delta}{2K}(n-1).$$

The proof of Theorem 6.7 is deferred to Appendix D.6.6. From Theorem 6.7, we show that in the weak-SCP notion, colluding players can also be reduced to malicious players w.r.t. the weak-SCP notion for the Byzantine-robustness results of Theorem 6.5.

6.5.5 Budget and Cost of Robustness

In real-world information elicitation and crowdsourcing applications, the principal may also aim to minimize the total budget while maintaining the desired level of information quality. Even within blockchain ecosystems, where tokens can be minted at will, excessive token issuance should be avoided to prevent inflation and the consequent devaluation of the cryptocurrency.

In the specific context of decentralized verification games, our goal is ideally to compensate verifiers exactly according to their (expected) verification costs—this serves as a natural lower bound on the total budget, provided that individual rationality (IR) constraints are satisfied. Moreover, to ensure robustness, we may impose an additional δ-margin as specified by the (δ, K)-compactness criterion. Under this requirement, the lower bound on the budget becomes the expected verification cost plus δ, where the δ term can be interpreted as the cost of robustness.

For this analysis, we assume $\epsilon = 0$ and that all verifiers are honest. Let $\overline{c}$ denote the expected verification cost, and $\overline{r}$ the expected payment to a verifier, both taken over the distribution of ground-truth states θ and corresponding observations. We then define the *cost of robustness* as:

$$\mu = \overline{r} - \overline{c}.$$

It follows directly that $\mu \geq \delta$. Furthermore, from Theorem 6.5, we can infer the existence of a feasible solution to LP_1 such that equality holds in Eq. (6.23) (implying that $\mu = \delta$), and that $K \leq \|B^{-1}\|_2 \cdot (c_1 + O(\delta))$. Therefore, when $\delta \to 0$, the mechanism achieves $\Theta\left(\frac{\delta}{c_1 \cdot \|B^{-1}\|_2}\right)$-compactness and is robust against a $\Theta\left(\frac{\delta}{c_1 \cdot \|B^{-1}\|_2}\right)$ fraction of adversarial verifiers. Formally, we have:

Theorem 6.8. *For $\eta < \frac{1}{g_2(k,p_1)\|B^{-1}\|_2}$, in order to ensure η compactness, we only need a margin and cost of robustness*

$$\mu = \delta \leq \frac{\eta c_1 g_1(k, p_1)\|B^{-1}\|_2}{1 - \eta g_2(k, p_1)\|B^{-1}\|_2}. \tag{6.29}$$

The proof is deferred to Appendix D.6.7.

6.6 Byzantine Reduction for Inaccurate Beliefs and Priors

In Section 6.5, we discussed the construction of Byzantine-robust mechanisms for decentralized verification games in the presence of a small fraction of malicious players, assuming the prior

knowledge of the distribution of θ is accurate. However, due to the possible existence of cheating provers, as we discussed in Section 6.3, there is actually a "small but unknown" ϵ probability that a proof is invalid, which is regarded as 0 in the derivation of previous sections. While we may argue that the δ margin can indeed ensure the IA properties in the presence of "sufficiently small" perturbations of prior distributions, we still need explicit and quantitative results of *distributional robustness* on the relations between δ and ϵ_0 to show the practical robustness of our mechanism against dishonest provers.

In this section, we show that the (δ, K)-compactness criterion is not only effective for robustness against malicious players, but also for robustness against inaccurate knowledge of prior distributions. Formally, a (δ, K)-compact CTF-PP mechanism maintains its IA guarantees even if the actual distribution of θ has an $O(\delta/K)$ *total variation* (TV) distance from the $P(\theta)$ we use in the construction of scoring rules, with a *Byzantine reduction* argument that reduces inaccurate beliefs to the existence of malicious players. With this technique, we not only show the general robustness of our design against malicious verifiers and inaccurate priors ($\epsilon > 0$, which represents the fraction of malicious provers), but also show the generality of our (δ, K)-compactness criteria and Byzantine-robustness as general conceptual guidelines of robust peer prediction mechanisms.

In the rest of this section, we refer to the TV distance when we mention the distance between distributions.

6.6.1 Byzantine Reduction: Inaccurate Beliefs $\leq$ Malicious Players

Before deriving the robustness results against inaccurate priors, we first discuss the robustness properties against inaccurate beliefs.

In the notation of Section 6.5.1, we suppose that the system has a perception of the environmental variable as $\hat{\phi}$, while the actual environmental variable is ϕ. Assuming that the player i observes $X_i \in S \cup \{\perp\}$, her posterior belief on the distribution of $\mathbf{X}_{-i}$ is $\mathbb{B}_i(X_i, \hat{\phi}) = P(\mathbf{X}_{-i}|X_i, \hat{\phi})$. However, as the actual environmental variable ϕ is (slightly) different from $\hat{\phi}$, the actual posterior distribution of $\mathbf{X}_{-i}$ is $\mathbb{B}_i(X_i, \phi) = P(\mathbf{X}_{-i}|X_i, \phi)$.

In the pairwise-scoring mechanism defined in Section 6.4.2, for the environmental variable

ϕ, the expected utility of reporting Z_i when observing X_i is

$$r_{X_i}(Z_i) - c(X_i) = \frac{1}{n-1} \sum_{j \neq i} \left(\sum_{Z_j \in S} P(Z_j | X_i, \phi) T_{Z_i Z_j} - c(X_i) \right). \tag{6.30}$$

Assuming that all other players are honest, i.e., $Z_j = X_j$, from symmetry we have

$$r_{X_i}(Z_i) - c(X_i) = \frac{1}{n-1} \sum_{j \neq i} \left(\sum_{X_j \in S} P(X_j | X_i, \phi) T_{Z_i X_j} - c(X_i) \right) \tag{6.31}$$

$$= P(X_j | X_i, \phi) T_{Z_i X_j} - c(X_i), \tag{6.32}$$

in which the j in (6.32) can be an arbitrary player different from i. Intuitively, if $P(X_j | X_i, \hat{\phi})$ is close to $P(X_j | X_i, \phi)$, then even if the scoring matrix $\{T_{Z_i Z_j}\}$ is designed for the δ-IA property according to $\hat{\phi}$, i.e.,

$$\sum_{X_j \in S} P(X_j | X_i, \hat{\phi}) T_{Z_i X_j} - c(X_i) \begin{cases} \geq \delta, & Z_i = X_i; \\ \leq -\delta, & Z_i \neq X_i. \end{cases} \tag{6.33}$$

the margin of δ can still make the mechanism 0-IA for the actual environment of ϕ, i.e.,

$$\sum_{X_j \in S} P(X_j | X_i, \phi) T_{Z_i X_j} - c(X_i) \begin{cases} \geq 0, & Z_i = X_i; \\ \leq 0, & Z_i \neq X_i. \end{cases} \tag{6.34}$$

Actually, if the scoring matrix T is (δ, K)-compact, we only need that the total variation (TV) distance between $P(X_j | X_i, \hat{\phi})$ and $P(X_j | X_i, \phi)$ is bounded below $\Theta(\delta / K)$. Formally, we have:

Lemma 6.2. *If Eq.* (6.33) *holds,* $\max\{|T_{Z_i Z_j}|\} \leq K$, *and* $TV_{X_j}(P(X_j | X_i, \hat{\phi}), P(X_j | X_i, \phi)) \leq \frac{\delta}{2K}$, *then Eq.* (6.34) *holds.*

The proof of Lemma 6.2 is deferred to Appendix D.6.8. From Lemma 6.2 we immediately deduce the following theorem:

Theorem 6.9. *Assume that there are no rogue players in environments ϕ and $\hat{\phi}$. If a pairwise-scoring CTF-PP mechanism is (δ, K)-compact for environment $\hat{\phi}$, and for a player*

$j \neq i$ it holds that the total variation distance $TV(P(X_j|X_i,\hat{\phi}), P(X_j|X_i,\phi)) \leq \frac{\delta}{2K}$ for every possible observation $X_i \in S \cup \{\perp\}$, then the mechanism is 0*-IA for environment ϕ.*

Actually, we can see that if $TV_{X_j}(P(X_j|X_i,\hat{\phi}), P(X_j|X_i,\phi)) = \alpha \cdot \frac{\delta}{2K}$ for $\alpha \in [0,1]$, then the mechanism is $(1-\alpha)\delta$-IA for environment ϕ, and hence according to Lemma 6.1, it is still Byzantine-robust to $(1-\alpha)\cdot\frac{\delta}{2K}(n-1)$ malicious players, showing that the compactness of δ/K serves as a "reservoir" of robustness against both inaccurate beliefs and malicious players. Intuitively, it indicates that the robustness against a Δ fraction of malicious players *implies* the robustness against a Δ level of noise in posterior beliefs. This intuition can be interpreted via *coupling argument*: if the posterior distribution is different from the belief, it can be equivalently regarded as the players "within the total variation" reporting dishonestly. The detailed discussion is deferred to Appendix D.3.

6.6.2 Robustness for Inaccurate Priors

In this part, we look into the effects of inaccurate prior distributions on the posterior beliefs, and derive the robustness of our mechanism in the presence of inaccurate priors. Under mild assumptions, we can show that an $O(\Delta)$ TV distance between two different priors is generally equivalent to an $O(\Delta)$ TV distance between corresponding posterior beliefs; hence, a (δ, K)-compact mechanism is also robust in the presence of a $\Theta(\frac{\delta}{K})$ noise in prior distributions. Formally,

Theorem 6.10. *Assume that environments ϕ and $\hat{\phi}$ have no rogue players and are identical except for different prior distributions $P(\theta|\phi) \neq P(\theta|\hat{\phi})$, and $P(X_i = 1|\theta \neq 1) = 0$. If a pairwise-scoring CTF-PP mechanism is (δ, K)-compact for environment $\hat{\phi}$ and*

$$TV_\theta(P(\theta|\phi), P(\theta|\hat{\phi})) \leq \frac{\delta}{4K} \cdot \min_{X_i \in S^*, \varphi \in \{\phi,\hat{\phi}\}} \{P(X_i|\varphi)\}. \tag{6.35}$$

then the mechanism is 0*-IA for environment ϕ.*

Here, the "min" represents the minimum probability that any non-dishonest observation (i.e., "Honest" or any flag) is observed, which is a positive constant dependent on the

verification protocol. The proof is deferred to Appendix D.6.9. From this result, we particularly show that the incentive guarantees robustly hold when at most an $\epsilon_0 = \Theta(\frac{\delta}{K})$ fraction of proofs are dishonest.

6.7 Experimental Evaluation

In this section, we perform numerical experiments and compare our mechanism with existing peer prediction mechanisms to show the effectiveness of our design.

6.7.1 Benchmarks

Elicitation environments. To evaluate the performance of our mechanism compared to existing mechanisms, we introduce two information elicitation environments.

The first environment is "Coin" in which an unfair coin may have a type $\theta = h$ with head probability $P(X_i = H|\theta = h) = 0.8$ or type $\theta = l$ with head probability $P(X_i = H|\theta = l) = 0.2$, the principal prior is $P(\theta = h) = 0.4, P(\theta = l) = 0.6$, and the observation cost is $c_H = c_T = 1$. This environment represents a standard scenario of information elicitation.

The second environment is Proof-of-Learning (PoL) as described in Appendix D.4 with principal prior $P(\theta = 0) = \frac{1}{2}, P(\theta = F_1) = P(\theta = F_2) = \frac{1}{4}, P(\theta = 1) = 0$, observation matrix as Table D.1 and observation costs $c(0) = \frac{1}{3}, c(F_1) = c(F_2) = c(1) = 2$. Here, "0" means "valid", "1" means "invalid" and F_1, F_2 stand for flags.

Player strategies. In our experiments, we consider three player strategies: "Honest", "Lazy", and "Adversarial".

In the Honest strategy, the player honestly observes and reports her observation. In the Lazy strategy, the player does not observe and reports a type that maximizes her expected reward among uninformed strategies. In the Adversarial (permutation) strategy, the player observes but reports a flipped type, that is: in Coin, reporting H when observing T and vice versa; in PoL, reporting 0 when observing 1, reporting F_1 when observing F_2, and vice versa.

Experiment schemes. We perform two experiments to evaluate the basic performance

and robustness to inaccurate priors. In the first experiment, we simulate a 2-verifier DVG in which the principal prior is accurate, and the peer is Honest. In the second experiment, we simulate a 2-verifier DVG in which the prior has an ϵ distance from the principal and the peer is Honest.

Evaluation rubrics. In each experiment, we evaluate three properties of our mechanism compared to baseline mechanisms: incentive guarantees, variance, and budget, described as follows:

- Incentive guarantees: We want to ensure that the Honest strategy yields non-negative utility, while the Lazy strategy (and Adversarial, if possible) yields non-positive utility.
- Variance: As real-world players are typically risk-aversive, we report the standard deviation of players' net utilities given they play Honestly. (This concept is also studied by Xu et al. [179].)
- Budget: We report the expected amount of money the system needs to pay players. To save the cost, it is preferably as little over expected verification cost as possible.

6.7.2 Baselines

To evaluate the performance of our design, we compare its incentive guarantees, variance, and budget to the baseline mechanisms as follows. In the third experiment, we compute the scores via pairwise average as described in Section 6.4.2.

- Simple Agreement (SA): The player is rewarded $+r$ if her report agrees with the peer, and $-r$ otherwise.
- Logarithmic Scoring Rule (Log): The player is rewarded $\log P(X_{-i} = Z_{-i} | X_i = Z_i)$ when she reports Z_i and her peer reports Z_{-i}.
- Pointwise Mutual Information Scoring Rule (PMI): The player is rewarded $\log \frac{P(X_{-i}=Z_{-i}, X_i=Z_i)}{P(X_{-i}=Z_{-i}) \cdot P(X_i=Z_i)}$ when she reports Z_i and her peer reports Z_{-i} [161].

- DMI Mechanism (DMI): The multi-task mechanism proposed by Kong [10]. As the DMI mechanism needs at least $2k$ tasks in which k is the number of different observations, we perform the experiments with $2k$ and $10k$ tasks to show its performance with different number of tasks.

To ensure a fair comparison, we apply an affine transformation $f(x) = ax + b$ to the scores of each baseline mechanism. We choose the smallest possible a (and corresponding b) such that the incentive-alignment guarantees hold (for DMI, we do not enforce UniIC and allow adversarial utilities to be positive), thereby giving each baseline the best opportunity to minimize variance and budget.[10] Furthermore, in the experiment for inaccurate priors, *we assume that the accurate prior is known by the system for all baselines (but not in our design)*, and is the same across all tasks for the DMI baseline. While these assumptions may not be realistic, we are allowing these baselines to operate under their most favorable conditions.

6.7.3 Experiment Results

In this section, we show the results of the first experiment (accurate prior, honest peer) in Tables 6.3-6.4, and defer the second to Appendix D.5.

In the Coin benchmark, we show that if we do not enforce a δ margin, our mechanism achieves optimal budget that equals the observation cost, and also achieves the smallest variance among all listed mechanisms, showing that our objective of magnitude minimization also *implicitly minimizes the variance* as it is upper bounded by the magnitude of scores. On the other hand, the PMI mechanism is the most competitive among all the baselines. Meanwhile, the DMI mechanism, though achieving desirable budget and prior-free incentive compatibility, has the worst variance and significantly worse compactness (robustness) than our mechanism under the same δ and budget. Furthermore, even though we enforce a δ margin for honest and lazy strategies, its inherent non-permutation-proof property renders

[10] In the multi-task DMI mechanism, the "budget" we report is the budget per task and the "variance" is divided by $\sqrt{T}$ in which T is the number of tasks. If we run a single-task mechanism T times, the standard deviation of total utility is $\sqrt{T}$ times the standard deviation for a single task. Hence, we divide the standard deviation by $\sqrt{T}$ for fair comparison.

	Budget	Variance	Compactness	Honest Utility	Lazy Utility	Adversarial Utility
Ours ($\delta = 0$)	**1.00**	**2.87**	0.000	0.00	0.00	-2.13
Ours ($\delta = 0.2$)	1.20	3.92	**0.038**	0.20	-0.20	-2.69
SA	1.57	6.11	0.000	0.57	0.00	-4.18
Log	1.38	4.26	0.000	0.38	0.00	-2.65
PMI	1.12	3.29	0.000	0.12	0.00	-2.37
DMI ($2k, \delta = 0$)	**1.00**	18.47	0.000	0.00	0.00	0.00
DMI ($2k, \delta = 0.2$)	1.20	25.86	0.001	0.20	-0.20	**0.20**
DMI ($10k, \delta = 0$)	**1.00**	6.78	0.000	0.00	0.00	0.00

Table 6.3: Experiment Results for Coin Benchmark

	Budget	Variance	Compactness	Honest Utility	Lazy Utility	Adversarial Utility
Ours ($\delta = 0$)	**0.75**	**2.56**	0.000	0.00	0.00	-2.85
Ours ($\delta = 0.2$)	0.95	3.65	**0.027**	0.20	-0.20	-3.30
SA	**(Infeasible)**					
Log	4.94	23.06	0.000	4.19	0.00	$-\infty$
PMI	1.25	3.84	0.000	0.50	0.00	$-\infty$
DMI	**(Infeasible)**					

Table 6.4: Experiment Results for PoL Benchmark

it subject to adversarial reports. Hence, our mechanism achieves better performance than the DMI mechanism in the standard setting.

The results for the PoL benchmarks are similar, in which the PMI mechanism is also the most competitive among all the baselines. Nevertheless, the DMI mechanism does not work in the case of $\epsilon = 0$ (no cheating provers) as the reward is always zero. Particularly, the DMI mechanism requires a full-rank "answer matrix" to distribute non-zero rewards, so the verifiers' rewards would be zero unless at least one of the tasks are done by a cheating prover, whether or not "flags" are adopted. Hence, it *only rewards the verifiers when cheats are detected, similar to opML [130]*, which is not robust for $\epsilon \to 0$ and fails to resolve the Verifier's Dilemma.

6.8 Discussion

In this paper, we develop a theoretical framework for the decentralized verification game on decentralized validation protocols and get theoretical results to robustly resolve the Verifier's Dilemma in a fully decentralized environment, potentially reinforcing the backbone

of decentralized AI incentive systems. On the other hand, we also explore the design of peer prediction mechanisms with broader agent strategy spaces and more general settings and dive into its robustness issue. In future work, we will improve and broaden the study in the following aspects:

1. Although the PoW/PoS protocols minimize the influence of Sybil attacks, they do not eliminate them completely. In our future work, we will look into more precise economic models w.r.t. PoW/PoS protocols and discuss the resilience against Sybil attacks of our mechanisms.

2. While this paper is mainly on the elicitation of truthful verification results, we will also develop back-end voting/aggregation mechanisms that (optimally) make decisions on whether to accept the proof.

3. Beside the applications of blockchain and decentralized verification games, we will explore broader scopes of potential applications of Byzantine-robust peer prediction for decentralized consensus (e.g. Ethereum slashing), and human feedback elicitation for RLHF and AI model training/inference.

CHAPTER 7

CONCLUSION: FOR THOSE WHO COME AFTER

In this thesis, I studied the problem of incentive design for a wide scope of digital economic platforms, from traditional ridesharing platforms to the futuristic vision of decentralized AI. In all of the parts, despite diverse technical contributions, they are all motivated by the same societal objective: to maintain reliability (running as expected even under complicated real-world environments) and sustainability (achieving long-term socially desirable outcomes when working as expected), while also keeping the revenue for the principals who deploy the mechanisms.

For future research directions, I am actively looking at the following scopes, to broaden the fields of my research's real-world applications:

- Mechanism design for *risk-averse players*: While most standard game-theoretical studies assume linear utility functions, i.e., risk-neutral players, in the real world most people are actually risk-averse, particularly in current years of declining economy. By considering such non-linear utility models, like Zhang et al. [180], we may fit theoretical studies better into practical use.

- *Bad-luck prevention* in collusion-proofness: Most studies on collusion-proof mechanism design depend on the assumption that colluders take up a small fraction of *current participants*, but in the real decentralized world, e.g., the PoS protocol in Ethereum, the participants are usually random drawn from a *latent pool*, and in tail cases the current "committee" may contain substantially more colluders than expectation, leading to occasional vulnerabilities in the system. To address the issue, we may add certain *disputing* mechanisms in which honest participants may pay some cost to *reroll* a new committee in such cases, and design proper incentives to make sure only honest

participants would be incentivized to dispute, like my working project of [181]. In this way, we can transform the assumption on *honest majority in committees* into *honest majority in pools*, reinforcing the security in real-world environments.

- *Scaffolding* decentralization: While decentralized AI has great potential to address reliability and safety issues in AI development, we need to admit the current monopolizing power in centralized ecosystems. To make my study *immediately* benefit the society before the maturity of decentralized AI ecosystems, I am actively working on AI systems that are still partially centralized, but capture some conceptual essence in decentralized AI, such as *incentive-compatible crowdsourcing*. The future is not built in a day, hence we need to build a smooth path towards it.

When conducting all these research topics, I have kept thinking over the question: There are so many applications for mechanism design, but what kind of topics are most valuable? What should be the ultimate objective, to do mechanism design?

- To benefit the entire society at large?
- Or to benefit the principal who ***deploys*** *the mechanism*?
- Or to benefit yourself, the person who ***designs*** *the mechanism*?

In an ideal world, I believe the first objective should take precedence: those who design the rules of society should ideally target social welfare. However, a critical issue remains: *the principal is also a strategic player*. If one manages to design a perfectly fair mechanism that achieves societal optimality, yet an alternative mechanism offers the principal a competitive advantage (albeit unfair), a *rational* principal is likely to choose the latter. In such a scenario, the altruistic mechanism designer would find their work unimplemented, unable to contribute to the society they aim to serve. Hence, to achieve tangible impact in the real world, *even the most altruistic and omniscient mechanism designer* must incorporate the principal's selfish incentives into consideration, ensuring that social welfare is optimized within the constraints of adoption.

At the time I am finishing this thesis, I am facing a critical stage of my life—PhD graduation. I have always been enjoying my research with the starry-eyed passion to change the world with my designs: to make even the most selfish people behave in a way that benefits the society, wishing to finally resolve the *Prisoner's Dilemma* that makes people fight against each other in disguise of "rationality". That would certainly be the most exciting and fulfilling achievement of my PhD.

Yet, I am frequently asked: "You have done brilliant work, but why choose these topics over trending ones that might secure a more prestigious position or higher compensation?" This question reveals why many researchers follow the hype: *the mechanism designer is also a strategic player*. If a mechanism cannot earn sufficient rewards for its designer, it is likely never to be designed at all.

Hence, an *altruistic and omniscient* mechanism designer may only consider the first two objectives; nevertheless, if you are *altruistic but* ***not omniscient***, then you need to consider the third, because working on a mechanism that does not benefit the designer will not attract others to join you, and it will be significantly harder to do these work all on your own, *unless you are omniscient*.

Hence, an ultimate objective of mechanism design, should be balancing the three. A PhD degree may already show that I am a *reliable* mechanism designer. After graduation, I should strive to remain *sustainable* as well.

The handwritten dedication at the beginning of this thesis translates to "keep on loving you".

One essential thing is being able to love; the other is, being able to keep on.

APPENDIX A

APPENDIX FOR CHAPTER 3

A.1 Omitted Proofs

A.1.1 Proof of Theorem 3.2

Proof. In this proof, we set all costs to be zero, so the reward is equivalent to the revenue.

We notice that even for states that are not adjacent, the reward function $r(n; e^{(\mathrm{w})}_{s,s'})$ is still well-defined if for all states t on a path from s to s', all drivers visiting t must visit s before and visit s' after. In this concept, we can regard it as a "virtual arc" as long as they do not intervene with each other. Then, we reduce Set Cover [182] to Maximum Revenue Car Dispatching.

Lemma A.1. *For $n, A \in \mathbb{N}, n \geq 1$, we can construct an arc (s, s') in polynomial time and size with $r(x; e^{(\mathrm{w})}_{s,s'}) = A \cdot 1_{x \geq n} + C_1$ for $x \in [n]$, in which C_1 is a constant only dependent on A and n.*

Proof of Lemma A.1. We create an arc (s, s') with n orders of valuation $\frac{n-1}{1}A, \frac{n-1}{2}A, \cdots, \frac{n-1}{n-1}A, A$. Then it satisfies the condition with $C_1 = A \cdot (n-1)$. **Q.E.D.**

Lemma A.2. *For $n, A \in \mathbb{N}, n \geq 1$, we can construct a virtual arc (s, s') in polynomial time and size with $r(x + (n-1); e^{(\mathrm{w})}_{s,s'}) = A \cdot \max\{0, x-1\} + C_2$ for $0 \leq x \leq n$, in which C_2 is a constant only dependent on A and n. We call it a (A, n)-virtual arc.*

Proof of Lemma A.2. For each $i \in [n-1]$, we construct an arc (s, s_i) with $r(x+(n-1); e^{(\mathrm{w})}_{s,s_i}) = A(i \cdot 1_{x_i \geq i+2} + C_1^{(i)})$ for $x_i > 0$ by Lemma A.1, and an arc (s_i, s') with no reward. Then, as it is straightforward to see $i \cdot 1_{x_i \geq i+1} \leq x_i - 1 - 1_{x_i \geq 2}$ for $x_i > 0$ and $C_1^{(i)} - 1 = i(i+1) - 1 > 0$ for $x_i = 0$, we always have

$$r\left(x_i; e^{(\mathrm{w})}_{s,s_i}\right) \leq A(x_i - 1 - 1_{x_i \geq 2} + C^{(i)}_1).$$

Therefore,

$$\begin{aligned} & r\left(\sum_{i=1}^{n-1} x_i; e^{(\mathrm{w})}_{s,s'}\right) = \sum_{i=1}^{n-1} r(x_i; e^{(\mathrm{w})}_{s,s_i}) \\ & \leq A \sum_{i=1}^{n-1} (x_i - 1 - 1_{x_i \geq 2} + C^{(i)}_1) \\ & = A\left(\sum_{i=1}^{n-1} x_i - (n-1) - \sum_{i=1}^{n-1} 1_{x_i \geq 2} + \sum_{i=1}^{n-1} C^{(i)}_1\right). \end{aligned}$$

We let $x = \sum_{i=1}^{n-1} x_i - (n-1)$, $C_2 = A \sum_{i=1}^{n-1} C^{(i)}_1$, and notice that $\sum_{i=1}^{n-1} 1_{x_i \geq 2} \geq 1_{x \geq 1}$. Then we get:

$$\begin{aligned} & r(x + (n-1); e^{(\mathrm{w})}_{s,s'}) \\ & \leq A(x - 1_{x \geq 1}) + C_2 \\ & = A \cdot \max\{0, x-1\} + C_2, 0 \leq x \leq n. \end{aligned}$$

When we let $x_i = 1 + x \cdot 1_{i=x-1}$, the equality holds. **Q.E.D.**

Now consider an instance of the set cover problem with the set $A = \{a_1, \cdots, a_n\}$, a family $\mathscr{K} = \{K_1, \cdots, K_m\}$ of subsets of A. Now we construct the Maximum Revenue Car Dispatching problem with $S = A \cup \mathscr{K} \cup \{O\}$. In the initiation, on each $a_i \in A$ we assign 1 driver, and for each $K_j \in \mathscr{K}$, we assign $|K_j| - 1$ drivers. Then, for each $a_i \in K_j$, we add an edge (a_i, K_j) with one order of valuation 1, and for each K_j, we add an $(1, |K_j|)$-virtual arc (K_j, O) as in Lemma A.2, with the respective C_2 denoted as $C_{(j)}$. Then, the drivers initially in K_j will go straight to O getting $C_{(j)}$ reward, for a total of $C_3 := \sum_{j=1}^{m} C_{(j)}$.

Now we consider the routes of drivers initially in A. Each order from A to $\mathscr{K}$ earns 1, and the reward from any K_j to O is non-decreasing, so in an optimal plan all drivers must reach O, and all nodes in $\mathscr{K}$ visited by some drivers from A form a set cover of A. For each fixed K_j, if $x \geq 1$ drivers from A visit K_j, the virtual arc (K_j, O) will earn an additional

reward of $x - 1$. Therefore, if totally k nodes in $\mathscr{K}$ are visited, the total revenue is:

$$2n + C_3 - k.$$

Therefore, if we can compute the optimal plan for this instance of Maximum Revenue Car Dispatching, we find the optimal solution to Set Cover, so there is no polynomial time algorithm for general Maximum Revenue Car Dispatching unless P = NP. **Q.E.D.**

A.1.2 Proof of Lemma 3.1

Proof. We firstly prove the "only if" direction, i.e. a reward re-allocation is fair only if there exists a potential function satisfying the condition (1-4):

We assume that the plan $\mathscr{P}$ is fair. Because $\mathscr{P}$ is envy-free, for every driver who visits the same state s, their utility obtained from s to the end must be the same, and we denote it as $P(s)$. We only consider states visited by at least one driver. For all s not visited by any driver, we assign $P(s) = 0$.

In condition 3, let d be a driver who drives through the arc (s, s'). By construction rule of P, the utility of d from s to the end is $P(s)$ and the utility of d from s' to the end is $P(s')$, so the net income of d driving from s to s' must be $P(s) - P(s')$. According to the non-negative producer surplus requirement in fair re-allocation, $P(s) - P(s') \geq 0$. So condition 3 holds.

If condition 2 is violated, then from condition 3 we know that $F(s, s') = 0$. Then, for any driver at s, when he/she deviates the route and drives to s' instead, he/she will benefit from the deviation. Contradiction. So condition 2 holds.

In condition 1, we consider a driver d leaving the platform at s, then d will not earn any net income further. By construction rule of P we know $P(s) = 0$.

In condition 4, the LHS is the summation of total net income of all drivers and the RHS is the summation of revenue of the platform, so it is equivalent to budget-balance condition. Therefore condition 4 holds.

In conclusion, we can construct a potential function P from any fair reward re-allocation.

Then we prove the "if" direction, i.e. a reward re-allocation is fair if there exists a potential

function satisfying the condition (1-4):

If a re-allocation scheme is not fair, then at least one of budget-balance, non-negative producer surplus, subgame-perfectness and envy-freeness is violated. We assume there still exists a potential function P satisfying all conditions.

If envy-freeness is violated, then there must exist a state s such that two drivers earn different net incomes from s to the end. By condition 3, all drivers at state s earn a net income of $P(s)$ from s until he/she leaves the platform if he/she follows the dispatching plan. Contradiction.

If subgame-perfectness is violated, then there must be a state s where a driver may deviate and improve his/her utility. Then condition 2 is violated. Contradiction.

If budget-balance is violated, then condition 4 is violated. Contradiction.

If non-negative producer surplus is violated, then the "≥ 0" constraint in condition 3 is violated. Contradiction.

In conclusion, if a re-allocation is not fair, such P satisfying all conditions does not exist. **Q.E.D.**

A.2 Approximate NLWC Algorithm for Non-regular Cases

To make the edge reward function concave, we essentially construct the *concave envelope* of $r(\cdot; e^{(w)}_{s,s'})$, as the least-valued concave function $\overline{r}(\cdot; e^{(w)}_{s,s'})$ not less than it. To compute $\overline{r}(\cdot; e^{(w)}_{s,s'})$, we only need to solve a linear program with decision variables $\{\overline{r}(i; e^{(w)}_{s,s'})\}_{i\in[o(s,s')]}$:

$$\text{Minimize} \quad \sum_{i=1}^{o(s,s')} \overline{r}(i; e^{(w)}_{s,s'})$$

$$\text{Subject to} \quad \overline{r}(i; e^{(w)}_{s,s'}) \geq r(i; e^{(w)}_{s,s'}), \quad \forall i \in [o(s,s')] \tag{A.1}$$

$$2\overline{r}(i; e^{(w)}_{s,s'}) \geq \overline{r}(i-1; e^{(w)}_{s,s'}) + \overline{r}(i+1; e^{(w)}_{s,s'}), \tag{A.2}$$

$$i = 2, \cdots, o(s,s') - 1.$$

Then, in Line 7 of Algorithm A1, we compute w with $\overline{r}$ instead, thus making w non-increasing and getting an approximated regular NLWC instance for computation of approximated

Maximum Revenue Car Dispatching.

A.3 An Example for Merit of Two-Phase Pricing

Through Wuhan City runs the Yangtze River, across which there had been only two bridges in 2000s. During rush hour, people traveling across the river crowded the bridges and made the traffic extremely heavy. As taxis would struggle in crossing the river, which would take a long time and increase both fuel and opportunity costs, taxi drivers frequently refused trips and made citizens complain.[1]

Consider three locations W_1, W_2, H, in which W_1, W_2 are far apart but both in Wuchang on the same side of the river, while H is just opposite to W_1 across the No.2 Yangtze Bridge in Hankou. By the taxi pricing system, the fare from W_1 to W_2 is $p(W_1, W_2) = 20$, while $p(W_1, H) = 10$. However, due to distinct traffic conditions, the costs are $c(W_1, W_2) = 10, c(W_1, H) = 8$. As $p(W_1, W_2) - c(W_1, W_2) > p(W_1, H) - c(W_1, H)$, taxi drivers would not be willing to cross the river.

Suppose there were two riders who would travel from W_1 to W_2 and H, respectively, with valuations the same as taxi fares. In the conventional mechanism, two drivers would take one order each, but one earns 10 while the other earns only 2, making the latter complain or even refuse the trip. If we enforce envy-freeness, as we cannot increase the $p(W_1, H)$ (otherwise the price exceeds the rider's valuation and the rider would not take the trip), we can only lower $p(W_1, W_2)$ to 12, which just lowers the revenue and makes drivers **equally unsatisfied**.

In our two-phase pricing mechanism, we can compute the potentials $P(W_1) = 6, P(W_2) = P(H) = 0$. Therefore, $r(W_1, W_2) = 16, r(W_1, H) = 14$, so no matter which trip they choose, they always earn 6. This kind of redistribution has not been possible until ridesharing platforms occur, but does make drivers envy-free and alleviate the problem that drivers are not willing to cross the river (and other situations of heavy traffic) without modifying rider-side pricing or total revenue, improving both parties' experience.

[1] See https://www.wsj.com/articles/SB10001424052702303330
20457924773153283669 4

A.4 Computing the Reward Functions for the Gaussian-Poisson Demand Distribution

A.4.1 The Gaussian-Poisson Demand Distribution.

In literature on pricing under stochastic demands, it is common to model the stochastic arrivals as Poisson processes [183, 184, 185, 186], and assume that the agents' undisclosed valuations of follow the normal distribution [183, 187]. In light of this, we define the parametric *Gaussian-Poisson distribution* for $\mathscr{D}(s, s')$, which is both practically useful and easy to learn. If $\mathscr{D}(s, s')$ is the Gaussian-Poisson distribution with parameters $(\mu_{s,s'}, \sigma_{s,s'}, \lambda_{s,s'})$, we have that $x_{s,s'} \sim \text{Pois}(\lambda_{s,s'})$ and each $v_t \sim \mathscr{N}(\mu_{s,s'}, \sigma^2_{s,s'})$, where $\text{Pois}(\lambda)$ and $\mathscr{N}(\mu, \sigma^2)$ respectively denote the Poisson and the Gaussian distribution.

For parametrization of the demand distribution, in our experiments in stochastic setting and online learning setting, we assume the latent orders obey Gaussian-Poisson distribution.

A.4.2 Computation of Reward Functions

Fix any arc (s, s') and the distribution parameters $(\mu_{s,s'}, \sigma_{s,s'}, \lambda_{s,s'})$. Denote $\Phi(\cdot)$ as the cumulative distribution function (cdf) of standard Gaussian distribution. If we offer a price p, since valuations of the latent orders on the arc obey $\mathscr{N}(\mu_{s,s'}, \sigma^2_{s,s'})$, each latent order has a valuation greater than p independently with probability $(1 - \Phi(\frac{p-\mu_{s,s'}}{\sigma_{s,s'}}))$. For convenience, we also refer to these orders as *qualified*.

The following lemma characterizes the number of the qualified orders on the given arc.

Lemma A.3. *Let $\tilde{x}(p)$ denote the the number of the qualified orders on the arc (s, s'). $\tilde{x}(p)$ follows* $\text{Pois}(\tilde{\lambda}(p; s, s'))$ *where*

$$\tilde{\lambda}(p; s, s') := \left(1 - \Phi(\frac{p - \mu_{s,s'}}{\sigma_{s,s'}})\right) \lambda_{s,s'}.$$

Proof. We equivalently prove the following statement: for $x \sim \text{Pois}(\lambda)$ we toss x coins each with head probability p, then the number of heads of all coins tossed obeys distribution $\text{Pois}(\lambda p)$.

The probability generating function of Pois(λ) is

$$\begin{aligned} G_1(t) &= \sum_{i=0}^{+\infty} e^{-\lambda} \frac{\lambda^i}{i!} \\ &= e^{\lambda(t-1)}. \end{aligned} \tag{A.3}$$

For every coin the probability generating function of the number of heads is

$$G_2(t) = (1-p) + pt. \tag{A.4}$$

Then the probability generating function of the total number of tails is

$$\begin{aligned} G_1(G_2(t)) &= e^{\lambda((1-p+pt)-1)} \\ &= e^{\lambda p(t-1)}, \end{aligned} \tag{A.5}$$

identical to the probability generating function of Pois(λp).

Therefore, the number of heads obeys the distribution Pois(λp). **Q.E.D.**

For $\tilde{\lambda} = \tilde{\lambda}(p; s, s')$, we define the function

$$\Theta(n, \tilde{\lambda}) := n - \sum_{i=0}^{n-1} (n-i) \frac{\tilde{\lambda}^i}{i!} e^{-\tilde{\lambda}}.$$

We have that $\Theta(n, \tilde{\lambda})$ is the expected number of the fulfilled orders on the arc if we dispatch n drivers. This is because the expected number of fulfilled orders is

$$\begin{aligned} &\sum_{i=0}^{\infty} \min\{n, i\} \cdot \Pr[\tilde{x}(p) = i] \\ &= \sum_{i=0}^{\infty} (n \cdot \Pr[\tilde{x}(p) = i]) - \sum_{i=0}^{n-1} ((n-i) \cdot \Pr[\tilde{x}(p) = i]) \\ &= n - \sum_{i=0}^{n-1} (n-i) \frac{\tilde{\lambda}^i}{i!} e^{-\tilde{\lambda}} = \Theta(n, \tilde{\lambda}). \end{aligned}$$

Finally, we use the functions defined above to compute $\mathscr{R}(n, p; s, s')$, and derive the

calculation methods for the edge reward function $r(\cdot; e^{(\mathrm{w})}_{s,s'})$ as follows.

Theorem A.1. *If $\mathscr{D}(s,s')$ follows the Gaussian-Poisson distribution with parameters $(\mu_{s,s'}, \sigma_{s,s'}, \lambda_{s,s'})$, then*

$$r(i; e^{(\mathrm{w})}_{s,s'}) = \max_{p \in \mathbb{R}^{\geq 0}} \left\{ \Theta\left(i, \tilde{\lambda}(p; s, s')\right) p \right\} - c(s,s')i. \tag{A.6}$$

A.5 Our Online Learning Algorithm

When the distributions $\{\mathscr{D}(s,s')\}$ of the latent orders are not known beforehand, our scheduling algorithm needs to actively collect data and learn these distributions with better accuracy while pursuing higher revenue. Note that a key component of $\mathscr{D}(s,s')$ is the riders' valuation distribution on each arc. The scheduling algorithm has to learn the distribution from the partial information that whether a rider has accepted the proposed price on the arc. On the other hand, the amount of partial information revealed about the valuation distribution critically depends on the scheduling algorithm's pricing strategy, as a too high or too low price would result in the riders always accepting or rejecting the offer, which is little useful information. Therefore, the learning-and-optimization algorithm has to carefully price the arcs to balance the two goals of obtaining enough information and securing high revenue. This is also known as the *exploration vs. exploitation* dilemma in online learning and decision-making.

The Thompson Sampling Framework. To address this challenge, we adopt Thompson sampling (TS), a general online learning and decision-making algorithmic design principle that dates back to [188] and proves to be useful in many practical tasks (e.g., [189, 190, 191]). Suppose that the scheduling algorithm will run for a time horizon of $\mathscr{T}$ days, and each day forms an independent scheduling task with the identical distributions $\{\mathscr{D}(s,s')\}$. The scheduling algorithm on day τ may use the information observed during the first $(\tau - 1)$ days to learn $\{\mathscr{D}(s,s')\}$, and has to make scheduling decisions for day τ, generating revenue as well as new data for future learning. The TS framework usually works with parametric distributions (where we assumed that $\{\mathscr{D}(s,s')\}$ are Gaussian-Poisson distributions with

parameters $\{(\mu_{s,s'}, \sigma_{s,s'}, \lambda_{s,s'})\}$), and maintain a prior distribution for the parameters. On each day τ, TS samples the parameters $\{(\hat{\mu}_{s,s'}^{(\tau)}, \hat{\sigma}_{s,s'}^{(\tau)})\}$ ($\hat{\lambda}_{s,s'}^{(\tau)}$ can be obtained from direct estimation as it is not involved in the exploration-exploitation dilemma) from the prior and correspondingly constructs the estimation $\{\hat{\mathscr{D}}^{(\tau)}(s, s')\}$. An optimal scheduling policy is computed based on $\{\hat{\mathscr{D}}^{(\tau)}(s, s')\}$ and the riders' responses (accept or reject) are observed. The TS algorithm then computes the posterior distribution for the parameters based on the new observation, which also serves as the prior on the next day.

Gaussian Priors and Laplace Approximation. A key choice we have to make in designing the TS algorithm is the specific form of the prior distributions that should simultaneously guarantee the learning performance and facilitate the posterior calculation. In our algorithm, we set the prior distributions for both $\mu_{s,s'}$ and $\sigma_{s,s'}$ to be independent Gaussian distributions:

$$\mu_{s,s'} \sim \mathscr{N}(\mu_{s,s'}^{\mu}, (\sigma_{s,s'}^{\mu})^2), \quad \sigma_{s,s'} \sim \mathscr{N}(\mu_{s,s'}^{\sigma}, (\sigma_{s,s'}^{\sigma})^2),$$

where $\mu_{s,s'}^{\mu}$, $\sigma_{s,s'}^{\mu}$, $\mu_{s,s'}^{\sigma}$, $\sigma_{s,s'}^{\sigma}$ can be estimated based on the intrinsic properties of the trip (s, s') (e.g., length, tolls, road quality, etc) without any interaction with the riders.

However, even with the above assumption, the posterior distributions of $\mu_{s,s'}$ and $\sigma_{s,s'}$ may become a complicated form other than Gaussian, which may lead to further description and computational complexity as the algorithm runs after multiple days. To address this challenge, we adopt the Laplace's method to approximate the potentially complicated posterior by another Gaussian distribution. Such a Laplace approximation method, first proposed by Chapelle and Li [189], is able to maintain the conjugacy properties for the priors and therefore greatly facilitates the computation. The detailed approximation procedure is derived in Appendix A.6.

Algorithm Description. In Algorithm 6, we describe the details of our TS algorithm. At Line 5, the $\lambda_{s,s'}$ parameter is not involved in the exploration-exploitation dilemma and therefore is learned directly via the maximum likelihood estimate. At Line 7, the approximate Gaussian posterior is done via the Laplace's method.

Algorithm 6 TS for Maximum Revenue Car Dispatching

1: **for** each $(s, s') \in Q$: initialize

$$(\mu_{s,s'}^{\mu,(1)}, \sigma_{s,s'}^{\mu,(1)}, \mu_{s,s'}^{\sigma,(1)}, \sigma_{s,s'}^{\sigma,(1)}) \leftarrow (\mu_{s,s'}^{\mu}, \sigma_{s,s'}^{\mu}, \mu_{s,s'}^{\sigma}, \sigma_{s,s'}^{\sigma}).$$

2: **for** $\tau \leftarrow 1, 2, \ldots, \mathscr{T}$ **do**

3: **for** $(s, s') \in Q$ **do**

4: Sample $\hat{\mu}_{s,s'}^{(\tau)} \sim \mathscr{N}(\mu_{s,s'}^{\mu,(\tau)}, (\sigma_{s,s'}^{\mu,(\tau)})^2)$, $\hat{\sigma}_{s,s'}^{(\tau)} \sim \mathscr{N}(\mu_{s,s'}^{\sigma,(\tau)}, (\sigma_{s,s'}^{\sigma,(\tau)})^2)$.

5: Estimate $\hat{\lambda}_{s,s'}^{(\tau)}$ as the daily average of the number of the latent orders on (s, s').

6: Compute the optimal Stochastic Maximum Revenue Car Dispatching (Definition 6) plan for the Gaussian-Poisson demand distribution with parameters $\{(\hat{\mu}_{s,s'}^{(\tau)}, \hat{\sigma}_{s,s'}^{(\tau)}, \hat{\lambda}_{s,s'}^{(\tau)})\}_{s,s'}$, and execute the plan on day τ.

7: Observe the riders' responses and compute the parameters for the approximate Gaussian posterior $\{(\mu_{s,s'}^{\mu,(\tau+1)}, \sigma_{s,s'}^{\mu,(\tau+1)}, \mu_{s,s'}^{\sigma,(\tau+1)}, \sigma_{s,s'}^{\sigma,(\tau+1)})\}_{s,s'}$.

A.6 Laplace Approximation for Posterior Computation in the Thompson Sampling Algorithm

At the end of day τ, for each arc (s, s'), suppose $\{p_i, y_i\}_{i \in n_{s,s'}^{(\tau)}}$ is the set of prices and rider responses on the arc in history. We compute the likelihood function

$$\mathscr{L}(\mu, \sigma) = \phi\left(\frac{\mu - \mu_{s,s'}^{\mu}}{\sigma_{s,s'}^{\mu}}\right) \phi\left(\frac{\sigma - \mu_{s,s'}^{\sigma}}{\sigma_{s,s'}^{\sigma}}\right) \prod_i L\left(\frac{p_i - \mu}{\sigma}, y_i\right),$$

where $\phi(\cdot)$ is the probability density function (pdf) of the standard Gaussian and $L(x, y) := \begin{cases} \Phi(x), & y = 0 \\ 1 - \Phi(x), & y = 1 \end{cases}$.

We adopt the Laplace approximation method for multi-variate likelihood [192] to approximate $\mathscr{L}$ by a product of Gaussian distributions of μ and σ. We firstly find the mode of $\log \mathscr{L}$:

$$(\tilde{\mu}^{\mu}, \tilde{\mu}^{\sigma}) = \arg\max \log \mathscr{L}(\mu, \sigma).$$

Then we use the symmetric difference quotient method [193] to compute the numerical Hessian of $\log \mathscr{L}(\mu, \sigma)$ at $(\tilde{\mu}^{\mu}, \tilde{\mu}^{\sigma})$ as

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \left.\begin{bmatrix} \frac{\partial^2 \log \mathscr{L}}{\partial \mu^2} & \frac{\partial^2 \log \mathscr{L}}{\partial \mu \partial \sigma} \\ \frac{\partial^2 \log \mathscr{L}}{\partial \mu \partial \sigma} & \frac{\partial^2 \log \mathscr{L}}{\partial \sigma^2} \end{bmatrix}\right|_{(\mu,\sigma)=(\tilde{\mu}^{\mu}, \tilde{\mu}^{\sigma})}.$$

Finally, we set $\mu_{s,s'}^{\mu,(\tau+1)} = \tilde{\mu}^{\mu}$, $\mu_{s,s'}^{\sigma,(\tau+1)} = \tilde{\mu}^{\sigma}$, and $\sigma_{s,s'}^{\mu,(\tau+1)} = \sqrt{-H_{11}^{-1}}$, $\sigma_{s,s'}^{\sigma,(\tau+1)} = \sqrt{-H_{22}^{-1}}$ as the parameters for the approximate Gaussian posterior on day τ, as well as the Gaussian prior on day $(\tau + 1)$. Note that H is a negative semi-definite matrix and therefore both $-H_{11}$ and $-H_{22}$ are non-negative.

A.7 Additional Experiments

A.7.1 Regularity of the Gaussian-Poisson Distribution

In this part, we perform numerical experiments to show that the edge reward function is concave for Gaussian-Poisson distributions. Without loss of generality we can set $\mu = 1$ (up to normalization). We then choose different (σ, λ) and verify the convexity of edge reward function. We have made a 330×330 grid for $\sigma \in [0, 1.5]$ and $\lambda \in [0, 33]$, and verified that at all grid points observe the regularity condition. The range of this grid covers the data appearing in the dataset of Section 3.6 and the resolution is fine, so it empirically verifies the regularity of the Gaussian-Poisson distribution.

In Figure A.1, we plot the marginal rewards v_k' with $\mu = 1$ and a few representative σ and λ values. It is easy to see that all curves are monotonically non-increasing with k.

A.7.2 Experiments for online learning

Online Setting. When the model parameters are not known before hand, we run our online learning algorithm (in Section A.5) for 50 days. We refer to the revenue of the algorithm as the *Thompson Sampling value* (TS). We also introduce the baseline *exploration-and-exploitation* (EE), another common strategy in online learning. In the first 19 days, EE performs exploration where the prices are chosen uniformly in a pre-defined interval, and on day 20 we learn the model parameters using the first 19-day data, then compute the optimal plan based on the learned parameters for the rest of days.

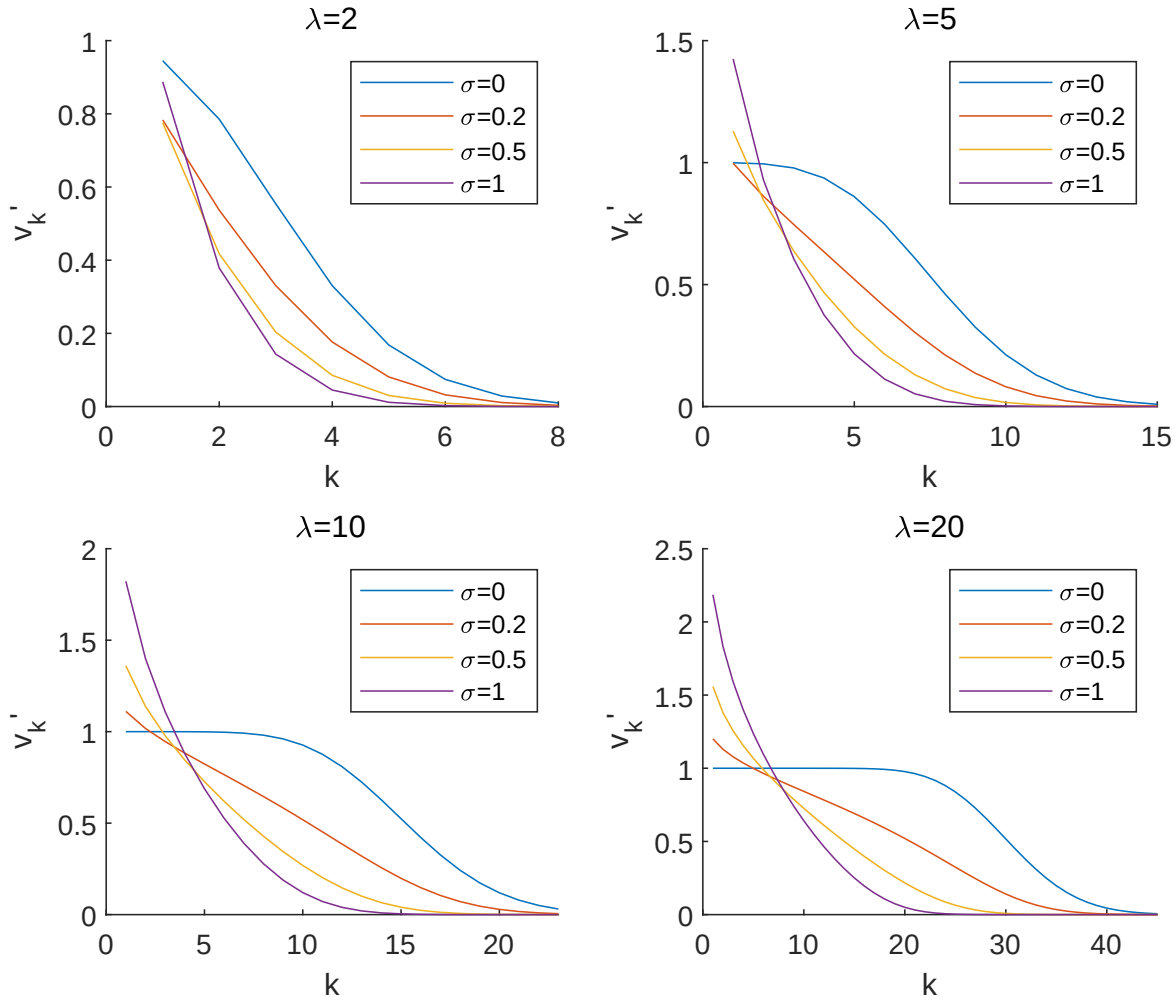

Figure A.1: Marginal rewards v'_k with different parameters.

Results (Online). We present the learning curves (the revenue collected on each day) of the online methods in Figure A.2. In the figure, we also plot OV for reference. We see that TS approaches the target OV much faster than EE.[2] For each online algorithm $\mathsf{A} \in \{\mathsf{TS}, \mathsf{EE}\}$, we define its *average regret* to be $\mathsf{Reg}(\mathsf{A}) := \frac{1}{50}\sum_{i=1}^{50}(\mathsf{OV} - \mathsf{A}(i))$, where $\mathsf{A}(i)$ denotes the revenue of A on day i. $\mathsf{Reg}(\mathsf{A})$ is a standard metric in online learning that measures the average price paid by A on each day to learn and approach the target OV. We report that $\mathsf{Reg}(\mathsf{TS}) = 1.29 \times 10^4$ and $\mathsf{Reg}(\mathsf{EE}) = 3.38 \times 10^4$. Our TS algorithm incurs a much smaller regret than the baseline EE.

Robustness. To evaluate the generalization ability of our algorithm, we modify the

[2]To reduce the computational burden, we only update the policy for TS in a subset of the 50 days, which results in the observable non-smoothness of the learning curve. If the policy were updated everyday, the performance would be slightly better.

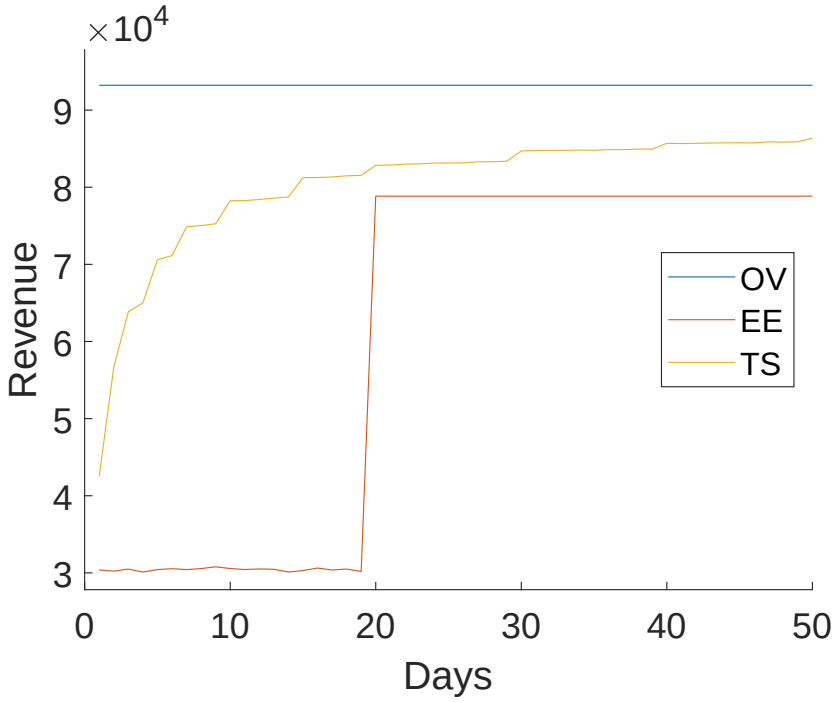

Figure A.2: Comparison of learning curves

following two key parameters in experiments: the number of drivers and the standard deviations of the riders' valuations. We report the experimental results showing that our algorithms still perform well under these different experimental environments.

In Table A.1, we modify the number of drivers. In the 50% drivers setting we remove each driver from the system with 50% independent probability and in the 200% drivers setting we duplicate every driver. In Table A.2, we modify the variations of the riders' valuations. Compared to the original dataset, we modify the standard deviations of valuations by 0.5 and 1.5 times respectively. We see that in all settings, our TS algorithm consistently performs better than other baselines.

For EE and TS, we present the revenue on the 50th day (Rev) and average regrets (Reg) during the period. Learning curves of experiments with modified parameters are shown in Figures A.3-A.6.

#drivers	6655		13411		26822	
	Rev	Reg	Rev	Reg	Rev	Reg
OV	6.82	–	9.32	–	11.17	–
FP	5.54	–	7.56	–	9.02	–
EE	5.88	2.47	7.88	3.38	9.27	4.14
TS	**6.40**	**0.87**	**8.64**	**1.29**	**10.25**	**1.65**

Table A.1: Rev/Reg with different numbers of drivers ($\times 10^4$).

stddev	0.5σ		1.0σ		1.5σ	
	Rev	Reg	Rev	Reg	Rev	Reg
OV	10.36	–	9.32	–	8.61	–
FP	7.90	–	7.56	–	7.25	–
EE	8.41	4.11	7.88	3.38	7.45	2.90
TS	**9.49**	**1.72**	**8.64**	**1.29**	**7.98**	**1.11**

Table A.2: Rev/Reg with modified standard deviations ($\times 10^4$).

A.7.3 An Illustrative Example for Fair Re-allocation

In this part, we show the properties of fair re-allocation for running the Phase 2 algorithm on DiDi dataset in the deterministic setting. Due to the large size of the dataset, we draw a representative subset of the whole dataset to show its behavior, and impose the budget-balance constraint on this subset instead of the whole dataset of rides.

In this example, we consider four positions in Chengdu city in China. Position A is the South Railway Station of Chengdu; position B is Tianfu Square, the leisure and business center located in the center of Chengdu; positions C and D are in two residential districts (Shuangqiaozi and Caojia Alley respectively). We then consider the traces of 10 drivers initiating from C, and three consecutive time stamps $1, 2, 3$ representing the time period of 8:00am to 8:45am. The trip from each position to another takes one time step, but as position A is relatively far from the cluster of $\{B, C, D\}$, trips to or from A typically earn more revenues. Figure A.7 shows the numbers of rides and net incomes of each arc from the Maximum Revenue Car Dispatching algorithm.

On the riders' side, riders traveling from or to A are not expected to complain about higher prices, because they do have longer trips. However when it comes to drivers, they may prefer to take longer rides from or to A than traveling among B, C, D. Particularly, one driver J_1 is assigned the trip $C \to A \to D$ and gains a net income of 6.76, and another

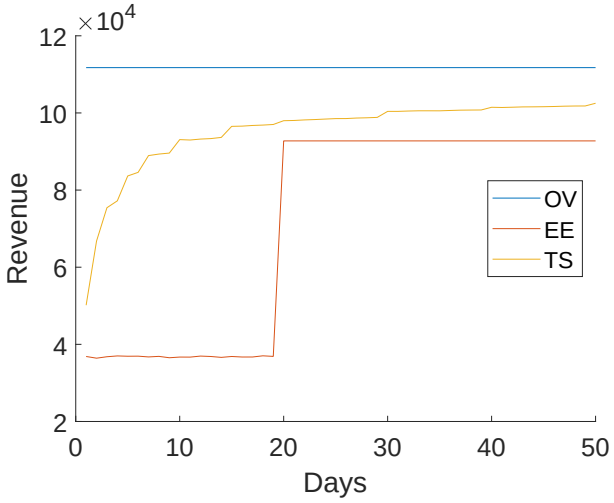

Figure A.3: Learning curves with 200% drivers.

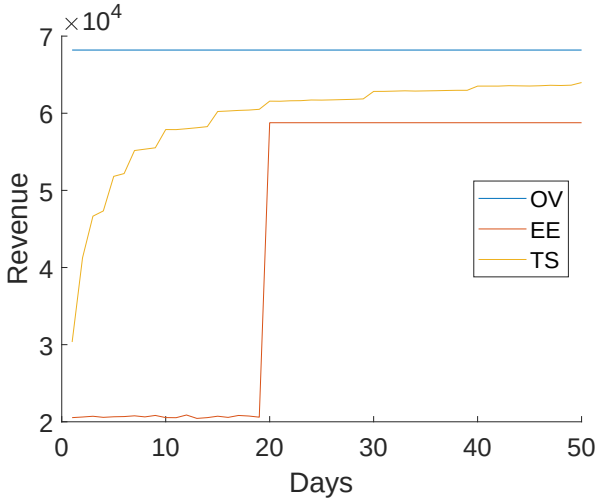

Figure A.4: Learning curves with 50% drivers.

driver J_2 is assigned the trip $C \to D \to D$ and gains a net income of 3.71. Then, J_2 may envy J_1 for earning more merely because assigned a "better" route.

In the same example, after we run the re-allocation algorithm, we re-allocate the money collected from riders to drivers, so that the net utilities for drivers of rides are shown in Figure A.8. In this way, no matter which route is assigned, a driver always gets a total net income of 4.81 (ignoring rounding errors) within the same total budget, and they cannot improve their net income by deviation, so fairness among drivers are guaranteed while the total revenue is still optimized.

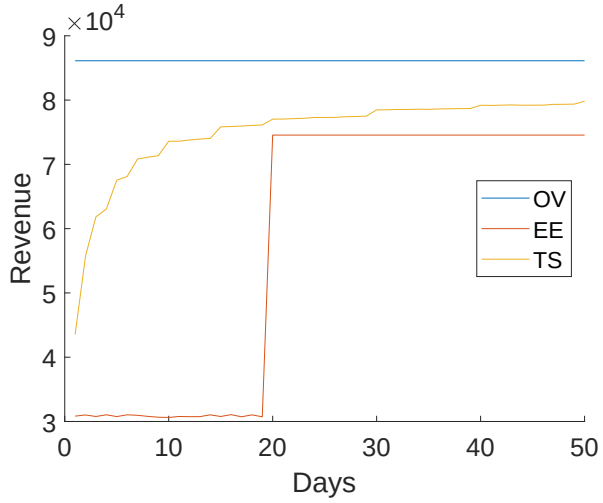

Figure A.5: Learning curves with 150% standard deviations of valuations.

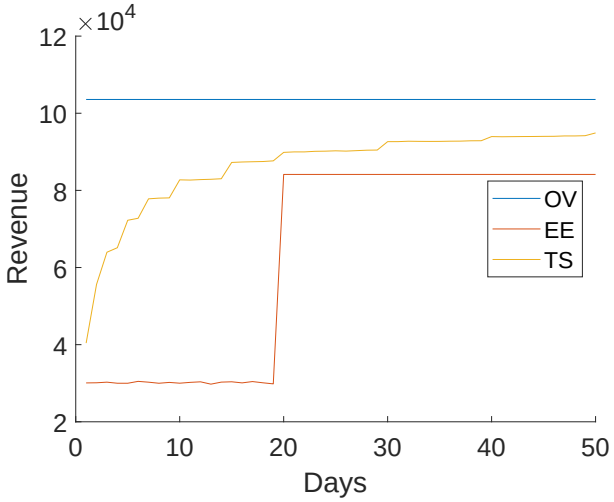

Figure A.6: Learning curves with 50% standard deviations of valuations.

A.7.4 A simple analysis for influence of number of drivers on unfairness without re-allocation

In fairness evaluation of our experiments, we notice that without the re-allocation phase, the relative unfairness increases with numbers of drivers. Intuitively, when there is only one driver, he/she would just pick the most profitable route; when more drivers join in, if they all choose to pick the most profitable routes for themselves, there may not be enough latent orders for all the routes, and some drivers would have to drive through sub-optimal routes for their income. This phenomenon increases with the number of drivers, which leads to the

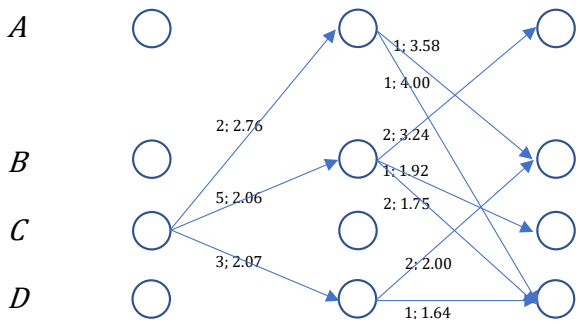

Figure A.7: Rider-side pricing for the example in Appendix A.7.3

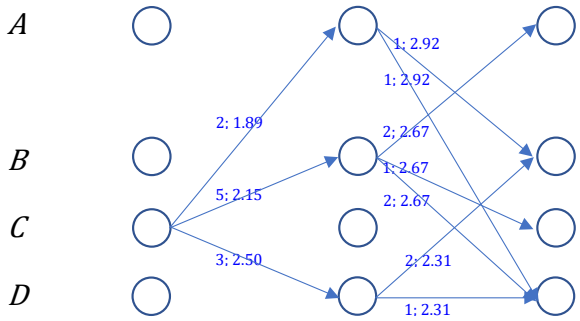

Figure A.8: Driver-side re-allocation for the example in Appendix A.7.3

increase of relative unfairness.

As an simple example, when there are only 5 latent orders from A to B, C, D, E, F, with profits $10, 9, 8, 7, 6$ respectively. When there are 2 drivers initially at A, they will be dispatched with $A \to B$ and $A \to C$ trips, and their profits are 10 and 9, so the relative unfairness is 0.053. If there are 5 drivers instead, then all trips will be taken and there is a wider spread in profits of individual drivers, and the relative unfairness is 0.177.

However, although it is the general tendency, the relative unfairness is **not** guaranteed to monotonically increase with the number of drivers. Consider an example in which there are also 45 latent orders from A to G with profit 2, then the relative unfairness for 25 drivers is 0.776 while the relative unfairness for 50 drivers is 0.713. This phenomenon occurs because in the case of 50 drivers, most drivers can only get the same low profit, so it becomes "less unfair" than the case of 25 drivers.

APPENDIX B

APPENDIX FOR CHAPTER 4

B.1 Cryptographic Protocols for On-Chain Implementation

Ideally, to design a credible blockchain TFM, we seek to discourage all kinds of dishonest behavior by either *systematically* preventing them from being conducted, or *economically* discouraging them by making them non-profitable.

Fortunately, the transparency property of a blockchain [194] and its implementation of many cryptographic protocols [195] have already helped prevent several types of dishonest behaviors. For example, since the blockchain is public, it is not possible for the miner to behave in a Byzantine manner via commuting different bidding vectors to different users (see the discussion in [39]), and the slashing rule in the Ethereum blockchain also discourage the miner from conducting certain classes of dishonest behavior via monetary penalties [196].

Also, [39] propose to adopt a secure commitment scheme, which uses cryptographic protocols to guarantee that a bid cannot be modified after proposal. This scheme has the following advantages:

1. It restricts the strategy space of the miner to merely adding fake transactions and concealing transactions, ruling out strategies for the miner to collude with users and change existing bids.

2. It implements a sealed-bidding auction format that not only makes the Bayesian game modeling valid but also guarantees fairness among users' information sets, restricting users' strategy space and preventing the MEV issue in which the miners strategically manipulate transaction orders to increase their utility.

Remark 1. *while we only need to prevent individual user deviations in the interim setting, for c-SCP and MIC properties we want a stronger ex-post version.*

Particularly, we can implement the *commitment scheme* in the way as follows:

1. Users submit the (salted) *hash values* of their transactions.
2. The miner packs and broadcasts all the hash values of the transactions that compete for the block, following by a hash value of the all packed hash values.
3. The users reveal their transactions and the miner uploads them. If the uploaded transactions deviates from the hash values too much ($\Delta \geq \epsilon_3 n$ for a pre-set $\epsilon_3 \in (0, 1)$, with Δ defined in Section 4.7.1), the miner is penalized.
4. The system processes the TFM.

For the miner-only deviation, the miner may behave dishonestly in Steps 1-3, and the number of deviations can be restricted in the way as follows:

1. The miner may submit fake transactions in Step 1, without seeing the honest transactions (interim `M-FT`). The system can restrict the number of transactions proposed by an identity in any block, and require any identity to have a deposit before proposing any transaction, so that the miner cannot create a large number of identities to submit too many fake transactions. We assume that the miner would not afford to inject more than $\epsilon_1 n$ transactions.
2. The miner may ignore some hashes in Step 2, without seeing their bids (interim `M-TD`). In this way, the system effectively runs with a smaller n. But if we set the parameter h in the way described in Section 4.7.1, reducing n cannot benefit the miner's revenue. Besides, the users who have their hashes ignored can also report this behavior and get the miner penalized. We assume that the miner will be caught if she ignores more than $\epsilon_2 n$ hashes.
3. The miner may insert or ignore transactions after she sees the bids in Step 3 (ex-post `M-FT` and `M-TD`), but this type of behavior will be detected. If the number of deviations

goes beyond an acceptable level, the miner will be penalized. On the other hand, an acceptable level $\epsilon_3 > 0$ is necessary because a missing transaction might also be simply due to the unstable connection from the user.

Hence, our protocol can restrict the miner individual deviation into a low level compared to n, and from the argument in Section 4.7.1, the relative advantage in miner revenue from {`M-FT`, `M-TD`} is bounded below $O\left(\left(\frac{\epsilon_1+\epsilon_3}{1-\epsilon_2}\right)^{4/3}\right)$. However, the miner-user collusion cannot be effectively prevented in this way, as they may conduct the collusion off-chain before Step 1.

Therefore, we can remark that:

Remark 2. *Existing cryptographic protocols can effectively prevent miner individual deviations, but can only prevent part of miner-user collusions.*

On the other hand, one may feel that the individual user's deviation is a "least destructive" honest behavior, because it happens in users' minds and does not seemingly disrupt the blockchain system. Hence, it also cannot be detected or prevented on the system level at all. However, we still argue that a desirable TFM should satisfy *truthfulness*, i.e., no individual user's deviation should be profitable. One key reason to design truthful mechanisms is the Revelation Principle [70, 71]: informally, for any non-truthful mechanism, we can construct an "equivalent" direct truthful mechanism that incorporates agents' optimal strategies into the mechanism itself, so that agents would maximize their utilities by reporting their true types (bidding their valuations). It renders untruthfulness unable to gain more advantage revenue.[1] Additionally, by the argument of the Revelation Principle, we also only need to consider single-round mechanisms. Hence, we remark that:

Remark 3. *The optimal revenue for any single-round truthful TFM is optimal even considering the class of non-truthful and multi-round mechanisms.*

Furthermore, due to the anonymity of the blockchains [197], it is difficult for users to collude with each other, as argued by [41]. Thus, user-user collusion is not a critical issue

[1] As long as there exists a mechanism whose outcome can achieve certain desired properties, we can indeed construct the equivalent truthful mechanism that both prevents agents from strategic behavior, and simplify the analysis as we can assume rational agents who seek to maximize their individual utilities will indeed follow the mechanism as we expect.

in the design of blockchain transaction fee mechanisms. Therefore, the remaining challenge to resolve is the prevention of user individual deviation and miner-user collusion, but as we have discussed, such dishonest behavior cannot be effectively prevented at the systematic level, so we have to discourage them in an economic way. In conclusion, we can remark that:

Remark 4. *To design a desirable blockchain transaction fee mechanism, the most critical challenge is to* ***discourage individual user's deviation and miner-user collusion via economic methods****.*

B.2 Impossibility Result on Deterministic TFM

In this section, we propose an impossibility result that under certain conditions, any deterministic TFM which is U-BNIC and 1-SCP cannot have positive miner revenue. Here we additionally introduce several notions. Although this impossibility does not fully rule out deterministic mechanisms, it does motivate us to introduce randomness into our main mechanism.

Deterministic. When bids are distinct, the outcome of the auction is deterministic, i.e., $a_i \in \{0, 1\}$.

Symmetric. When we swap the bids of two users, their allocations and payments are exactly swapped.

Continuous. $\mathbf{p}$ and r are continuous functions of $\mathbf{b}$, and V has bounded, strictly positive PDF on a simply connected support $\mathrm{dom}(V)$.

Strongly Monotone. If we raise the bid of bidder i while leave other bids unchanged, a_i, p_i do not decrease and $a_j (\forall j \neq i)$ does not increase.

Theorem B.1. *For all deterministic, symmetric, continuous, strongly monotone, user-individually-rational and budget-feasible TFMs, if* $\mathbf{0} \in V$*, then U-BNIC and 1-SCP implies non-positive miner revenue.*

B.2.1 Proof of Theorem B.1

Proof sketch. To prove the non-positive-miner-revenue property of all satisfying mechanisms, we first show that all satisfying mechanisms must obey certain restrictive conditions, as the

payment (Sec. B.2.1) and revenue (Sec. B.2.1) rules both must follow corresponding closed-form formulas; then we show that this type of mechanisms have non-positive miner revenue.

In this section, we introduce the δ-function with

$$\int_{-\epsilon}^{\epsilon} \delta(t)dt = 1, \qquad \forall \epsilon > 0. \tag{B.1}$$

We assume there exists a transaction fee mechanism $M_0(\mathbf{a}, \mathbf{p}, r)$ that satisfies all conditions.

Pinning down the payment rule

From definition we know that if M_0 is BNIC, then

$$\left.\frac{\partial E_{\mathbf{v}_{-i}\sim V_{-i}}\left[u_i(b_i, v_i, \mathbf{v}_{-i})\right]}{\partial b_i}\right|_{b_i=v_i} = 0, \qquad \forall v_i \tag{B.2}$$

i.e.,

$$\begin{aligned}\int_{\mathbf{v}_{-i}} \Bigg(&(v_i - p_i(v_i, \mathbf{v}_{-i}))\, \frac{\partial a_i(v_i, \mathbf{v}_{-i})}{\partial v_i} \\ &- a_i(v_i, \mathbf{v}_{-i}) \frac{\partial p_i(v_i, \mathbf{v}_{-i})}{\partial v_i} \Bigg) \rho_{-i}(\mathbf{v}_{-i}) d\mathbf{v}_{-i} = 0,\end{aligned} \tag{B.3}$$

in which $\rho_{-i}(\cdot)$ is the pdf of V_{-i}.

For fixed $\mathbf{v}_{-i}$, since the mechanism is deterministic, we have that $a_i(\cdot, \mathbf{v}_{-i}) \in \{0, 1\}$ almost everywhere. Additionally because $a_i(\cdot, \mathbf{v}_{-i})$ is monotonic increasing, we have

$$a_i(v_i, \mathbf{v}_{-i}) = \begin{cases} 0, \ v_i < \theta(\mathbf{v}_{-i}) \\ 1, \ v_i > \theta(\mathbf{v}_{-i}), \end{cases} \tag{B.4}$$

in which $\theta(\mathbf{v}_{-i})$ is a constant for fixed $\mathbf{v}_{-i}$. Therefore,

$$\frac{\partial a_i(v_i, \mathbf{v}_{-i})}{\partial v_i} = \delta(v_i - \theta(\mathbf{v}_{-i})). \tag{B.5}$$

Now we have a lemma:

Lemma B.1. *For* $\forall \mathbf{v}_{-i}$,

$$p_i(\theta(\mathbf{v}_{-i}), \mathbf{v}_{-i}) = \theta(\mathbf{v}_{-i}). \tag{B.6}$$

Proof. Proof. If $p_i(\theta(\mathbf{v}_{-i}), \mathbf{v}_{-i}) > \theta(\mathbf{v}_{-i})$, let $t = p_i(\theta(\mathbf{v}_{-i}), \mathbf{v}_{-i}) - \theta(\mathbf{v}_{-i})$. Then by continuity, there exists a small $\epsilon > 0$ s.t. $p_i(\theta(\mathbf{v}_{-i}) + \epsilon, \mathbf{v}_{-i}) > \theta(\mathbf{v}_{-i}) + \frac{t}{2}$ and $a_i(\theta(\mathbf{v}_{-i}) + \epsilon, \mathbf{v}_{-i}) = 1$, and the user i would have negative utility. In this scenario, the miner would want to collude with user i and ask him to change his bid to $\theta(\mathbf{v}_{-i}) - \epsilon$, so that user i would now have 0 utility.

But by continuity, the change of the miner's revenue is arbitrarily small, increasing their total utility. So the 1-SCP property is violated.

If $p_i(\theta(\mathbf{v}_{-i}), \mathbf{v}_{-i}) < \theta(\mathbf{v}_{-i})$, similarly there exists a scenario where user i has valuation $\theta(\mathbf{v}_{-i}) - \epsilon$ but the miner would want to let her bid $\theta(\mathbf{v}_{-i}) + \epsilon$ instead, also violating 1-SCP.

Therefore, it must hold that $p_i(\theta(\mathbf{v}_{-i}), \mathbf{v}_{-i}) = \theta(\mathbf{v}_{-i})$.

□ **Q.E.D.**

From Lemma B.1 we have

$$\int_{\mathbf{v}_{-i}} \left((v_i - p_i(v_i, \mathbf{v}_{-i})) \frac{\partial a_i(v_i, \mathbf{v}_{-i})}{\partial v_i} \right) \rho_{-i}(\mathbf{v}_{-i}) d\mathbf{v}_{-i} = 0, \tag{B.7}$$

so

$$\int_{\mathbf{v}_{-i}} \left(a_i(v_i, \mathbf{v}_{-i}) \frac{\partial p_i(v_i, \mathbf{v}_{-i})}{\partial v_i} \right) \rho_{-i}(\mathbf{v}_{-i}) d\mathbf{v}_{-i} = 0. \tag{B.8}$$

Since monotonicity implies $\frac{\partial p_i(v_i, \mathbf{v}_{-i})}{\partial v_i} \geq 0$, we know that $\forall v_i > \theta(\mathbf{v}_{-i})$, $\frac{\partial p_i(v_i, \mathbf{v}_{-i})}{\partial v_i} = 0$. Therefore,

$$\forall b_i > \theta(\mathbf{v}_{-i}), \epsilon > 0, \quad p_i(b_i, \mathbf{v}_{-i}) = p_i\left(\theta(\mathbf{v}_{-i}) + \epsilon, \mathbf{v}_{-i}\right). \tag{B.9}$$

Combined with Lemma B.1, from continuity we get

$$\forall b_i \geq \theta(\mathbf{v}_{-i}), \quad p_i(b_i, \mathbf{v}_{-i}) = \theta(\mathbf{v}_{-i}). \tag{B.10}$$

Pinning down the miner revenue rule

In this part, we mainly use the 1-SCP property to prove that the miner revenue is a constant with regard to any user. To show this, we prove a lemma:

Lemma B.2. *If $v_i \neq \theta(\mathbf{v}_{-i})$, then $\frac{\partial r(v_i, \mathbf{v}_{-i})}{\partial v_i} = 0$.*

Proof. Proof. We recall that the total utility of the miner and user i is

$$C_i(b_i, v_i, \mathbf{v}_{-i}) = a_i(b_i, \mathbf{v}_{-i})(v_i - p_i(b_i, \mathbf{v}_{-i})) + r(b_i, \mathbf{v}_{-i}). \tag{B.11}$$

From 1-SCP we know that

$$\begin{aligned} 0 = & \left.\frac{\partial C_i(b_i, v_i, \mathbf{v}_{-i})}{\partial b_i}\right|_{b_i = v_i} && \text{(B.12)} \\ = & \left((v_i - p_i(v_i, \mathbf{v}_{-i})) \frac{\partial a_i(v_i, \mathbf{v}_{-i})}{\partial v_i} \right. \\ & \left. - a_i(v_i, \mathbf{v}_{-i}) \frac{\partial p(v_i, \mathbf{v}_{-i})}{\partial v_i} \right) + \frac{\partial r(v_i, \mathbf{v}_{-i})}{\partial v_i}. && \text{(B.13)} \end{aligned}$$

From Eq. (B.10) we know $a_i(v_i, \mathbf{v}_{-i}) \frac{\partial p(v_i, \mathbf{v}_{-i})}{\partial v_i} \equiv 0$, and from Eq. (B.5) we know $v_i \neq \theta(\mathbf{v}_{-i}) \Rightarrow \frac{\partial a_i(v_i, \mathbf{v}_{-i})}{\partial v_i} = 0$. So we deduce

$$v_i \neq \theta(\mathbf{v}_{-i}) \Rightarrow \frac{\partial r(v_i, \mathbf{v}_{-i})}{\partial v_i} = 0. \tag{B.14}$$

□ **Q.E.D.**

Because the continuity condition guarantees $r(\mathbf{b})$ is a continuous function of $\mathbf{b}$, from Lemma B.2 we know that for fixed $\mathbf{v}_{-i}$, $r(\cdot, \mathbf{v}_{-i})$ is a constant, hence

$$r(v_i, \mathbf{v}_{-i}) = r(0, \mathbf{v}_{-i}). \tag{B.15}$$

By iteratively apply Eq. (B.15) to all components of $\mathbf{v}$, we get

$$r(\mathbf{v}) = r(\mathbf{0}). \tag{B.16}$$

We notice that from UIR,

$$r(\mathbf{0}) \leq \sum_{i=1}^{n} a_i(\mathbf{0}) p_i(\mathbf{0}) \tag{B.17}$$

$$\leq \sum_{i=1}^{n} a_i(\mathbf{0}) \cdot 0 \tag{B.18}$$

$$= 0. \tag{B.19}$$

Therefore, we have

$$r(\mathbf{v}) \leq 0, \qquad \forall \mathbf{v}. \tag{B.20}$$

Here we prove Theorem B.1.

B.3 Additional Perspectives of Auxiliary Mechanism Method

B.3.1 A Failed Example: the First-Price Auction

In this part, we use a simple example to help readers understand the constraints for an admissible variation term. In particular, we will demonstrate a $\boldsymbol{\theta}$ function that cannot be coupled with any $\tilde{r}$ to form an admissible variation term. The $\boldsymbol{\theta}$ function is constructed

based on the natural first-price auction. As an interesting by-product, this example also shows that, although the first-price auction mechanism can be adapted to satisfy U-BNIC, it cannot be combined with a miner payment rule $\tilde{r}$ to further enjoy the 1-SCP property.

We now define $\boldsymbol{\theta}$ based on the first-price auction. For simplicity, we consider only $n = 2$ users and the block size $k = 1$. The first-price auction for the single block entry defines the following allocation rule $\mathbf{a}$ (both first-price and second-price auctions confirm the highest-bid user, also note that b_{-i} is a scalar since there are only 2 users):

$$a_i(b_i, b_{-i}) = \begin{cases} 1, \ b_i > b_{-i} \\ \frac{1}{2}, \ b_i = b_{-i} \\ 0, \ b_i < b_{-i} \end{cases} . \tag{B.21}$$

We then consider the payment rules that will help us to finally define $\boldsymbol{\theta}$. The first payment rule $\mathbf{p}$ is the dominant association of $\mathbf{a}$. We calculate $\mathbf{p}$ via Eq. (4.7) as follows.

$$p_i(b_i, b_{-i}) = \begin{cases} b_{-i}, & b_i \geq b_{-i} \\ 0, & b_i < b_{-i} \end{cases} . \tag{B.22}$$

Indeed, $\mathbf{a}$ and $\mathbf{p}$ form the second-price auction which is DSIC.

We now turn to the second payment rule $\tilde{\mathbf{p}}$ which is adapted from the payment rule of the first-price auction. It is well-known that the first-price auction is not truthful (DSIC) [37]: users would prefer to bid lower than their valuations, which is necessary for them to get any surplus even if they get the item. Nevertheless, there exist Bayesian Nash equilibria for specific settings when distributions of valuations are known. For example, when there are n users with *i.i.d.* uniformly random valuations over $[0, 1]$, it is a Bayesian Nash equilibrium for each bidder to bid $\frac{n-1}{n} v_i$. By the Revelation Principle [70, 71], we can derive a payment

rule $\tilde{p}$ to make the confirmed user pay $\frac{n-1}{n}$ times her bid. For $n = 2$, we derive $\tilde{p}$ as follows.

$$\tilde{p}_i(b_i, b_{-i}) = \begin{cases} \frac{1}{2}b_i & b_i \geq b_{-i} \\ 0 & b_i < b_{-i} \end{cases}. \tag{B.23}$$

Finally, we define $\boldsymbol{\theta}$ according to Eq. (4.8) and get that

$$\theta_i(b_i, b_{-i}) = \begin{cases} \frac{1}{2}b_i - b_{-i} & b_i > b_{-i} \\ -\frac{1}{4}b_i & b_i = b_{-i} \\ 0 & b_i < b_{-i} \end{cases}. \tag{B.24}$$

When the user valuation is uniformly random over $[0, 1]$, we have that $\mathbb{E}_{b_{-i}\sim U[0,1]}[\theta_i(0, b_{-i})] = 0$ for $i \in \{1, 2\}$, indicating that $\boldsymbol{\theta}$ satisfies the second condition (Eq. (4.11)) of the admissibility property. Suppose that we could find a miner revenue function $\tilde{r}$ such that $T = (\boldsymbol{\theta}, \tilde{r})$ is admissible. Let $M = (\mathbf{a}, \mathbf{p}, 0)$. According to Theorem 4.2 and by the definition of $\boldsymbol{\theta}$, we have that the composed TFM

$$\tilde{M} = M + T = (\mathbf{a}, \mathbf{p}, 0) + (\boldsymbol{\theta}, \tilde{r}) = (\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$$

is U-BNIC and 1-SCP. Then we could get the TFM $\tilde{M}$ which is a natural adaptation of the first-price auction (since its payment rule $\tilde{\mathbf{p}}$ is adapted from the first-price payment rule).

On the other hand, however, we show that this is impossible – there exists no $\tilde{r}$ such that $(\boldsymbol{\theta}, \tilde{r})$ is admissible. We prove this by contradiction. Suppose there exists such an $\tilde{r}$, we compute $\tilde{r}(1, 1)$ in two different ways. By the first condition of admissibility (Eq. (4.10)), we have that

$$\begin{aligned} \tilde{r}(1,1) &= \tilde{r}(0,0) + (\tilde{r}(1,0) - \tilde{r}(0,0)) + (\tilde{r}(1,1) - \tilde{r}(1,0)) \\ &= \tilde{r}(0,0) + \theta_1(1,0) + \theta_2(1,1) \\ &= \tilde{r}(0,0) + 0.5 - 0.25 \\ &= \tilde{r}(0,0) + 0.25. \end{aligned}$$

We can also invoke Eq. (4.10) and compute $\tilde{r}(1,1)$ via a different path:

$$\begin{aligned}
&\tilde{r}(1,1)\\
&= \tilde{r}(0,0) + (\tilde{r}(0.5,0) - \tilde{r}(0,0)) + (\tilde{r}(0.5,1) - \tilde{r}(0.5,0))\\
&\qquad + (\tilde{r}(1,1) - \tilde{r}(0,1)) - (\tilde{r}(0.5,1) - \tilde{r}(0,1))\\
&= \tilde{r}(0,0) + \theta_1(0.5,0) + \theta_2(0.5,1) + \theta_1(1,1) - \theta_1(0.5,1)\\
&= \tilde{r}(0,0) + 0.25 + 0 - 0.25 - 0\\
&= \tilde{r}(0,0) + 0.
\end{aligned}$$

Now we reach the contradiction. This example shows that using our auxiliary mechanism method, we are not able to extend the natural first-price auction to a U-BNIC and 1-SCP TFM.[2] We will need to carefully design a different $\boldsymbol{\theta}$ to satisfy the admissibility conditions.

B.3.2 A Conservative-field Perspective of the Payment Difference Function $\{\theta_i\}$

In this part, we distill our experience in the trial in Appendix B.3.1 and provide an additional perspective for the design of $\boldsymbol{\theta}$. From the example, we see that if we sum up the differences of $\boldsymbol{\theta}$ along any path that consists of axis-aligned arcs, the summation should only depend on the two terminals of the path. This suggests the path-independence property of the $\boldsymbol{\theta}$ function. In particular, for any $\boldsymbol{\theta}$ in an admissible variation term $(\boldsymbol{\theta}, \tilde{r})$, if we define the vector field

$$\mathscr{D}_{\boldsymbol{\theta}}(\mathbf{b}) = \left(\frac{\partial}{\partial b_1}\theta_1(b_1, \mathbf{b}_{-1}), \cdots, \frac{\partial}{\partial b_n}\theta_n(b_n, \mathbf{b}_{-n}) \right), \tag{B.25}$$

[2]It is possible to prove a stronger statement: there exists no $\tilde{r}$ such that the TFM $(\mathbf{a}, \tilde{\mathbf{p}}, \tilde{r})$ is 1-SCP (where $\mathbf{a}$ and $\tilde{\mathbf{p}}$ are defined based on the first-price auction as in Section B.3.1). Therefore, the does not exist a U-BNIC and 1-SCP TFM extension based on the first-price auction. We omit the detailed proof of this statement since it is not directly related to the construction and the analysis of our TFM.

then $\mathscr{D}_{\boldsymbol{\theta}}$ should be a conservative field [198]. In other words, for any closed curve C (with parametrization $\mathbf{z}$), we have the following equality for the integration

$$\oint_C \mathscr{D}_{\boldsymbol{\theta}} \cdot d\mathbf{z} = 0. \tag{B.26}$$

According to Eq. (4.10), $\tilde{r}$ is actually the potential of $\mathscr{D}_{\boldsymbol{\theta}}$. From this conservative-field perspective, we see that in order to successfully construct an admissible variation term, we may consider first constructing a $\tilde{r}$ (as the *potential* that determines the field), while guaranteeing the $\boldsymbol{\theta}$ functions satisfies Eq. (4.11). This intuition helps our design of the admissible variation term. Nevertheless, it is still quite challenging to construct a good variation term. Thanks to the almost-modular property of the auxiliary mechanism and the variation term, we can re-use an admissible variation term in different settings, as we do in Sections 4.5-4.6.

B.3.3 Intuition of Variation Term Construction in Section 4.5.2

From the admissibility condition Eq. (4.10), i.e., $\theta_i(b_i, \mathbf{b}_{-i}) = \tilde{r}(b_i, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i})$, we get

$$\tilde{r}(b_i, \mathbf{b}_{-i}) = \tilde{r}(b_i, \mathbf{0}) + \theta(b_i, \mathbf{b}_{-i}) \tag{B.27}$$

From another admissibility condition of Eq. (4.11), i.e., $\mathbb{E}_{\mathbf{b}_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0$, for convenience we decouple b_i and $\mathbf{b}_{-i}$ and construct θ_i in the following form

$$\theta_i(b_i, \mathbf{b}_{-i}) = h \cdot \alpha(b_i) \cdot \beta(\mathbf{b}_{-i}), \tag{B.28}$$

in which $\beta(\mathbf{b}_{-i})$ is a symmetric expression on $\mathbf{b}_{-i}$ and

$$\mathbb{E}_{\mathbf{b}_{-i}}[\beta(\mathbf{b}_{-i})] = 0. \tag{B.29}$$

Now we consider the case of $\mathbf{b}_{-i} = \mathbf{0}$ and m is large, i.e., the situation is close to a second-price auction in which all other users bid zero, and the user i's payment in the auxiliary

mechanism is close to zero.

However, as long as $b_i > 0$, by intuition user i is capable of paying more. From the allocation rule, for any fixed m we can actually find a $K > 0$ in which $a_i(b_i, \mathbf{0}) \geq Kb_i$, and hence user i is able to pay at least $a_i(b_i, \mathbf{0}) \cdot b_i \geq Kb_i^2$. On the other hand, from Myerson's Lemma (Lemma 4.1), in the auxiliary mechanism we also have $a_i(b_i, \mathbf{0})p_i(b_i, \mathbf{0}) = \Theta(b_i^2)$ when $b_i \to 0$, but quickly "saturating" when $b_i > \Theta(\frac{1}{m})$ and $a_i(b_i, \mathbf{0})$ become close to 1. Hence, to uniformly exploit payment from user i for different values of b_i, we would like to construct[3]

$$\alpha(b_i) = \frac{1}{2}b_i^2. \tag{B.30}$$

On the other hand, since the expression of $\theta_i(\cdot)$ will appear in the expression of $\tilde{r}(\cdot)$, and $\tilde{r}(\cdot)$ is a symmetric expression. In order to ensure symmetry, we construct

$$\beta(\mathbf{b}_{-i}) = 1 - \mu \sum_{j:j\neq i} b_j^2. \tag{B.31}$$

Even if it indicated less payment when $\mathbf{b}_{-i}$ are large on the users' side, the negative fourth order terms in the expression of $\tilde{r}(\mathbf{b})$ are "halved" compared to the sum of $\{\theta_i(b_i, \mathbf{b}_{-i})\}$, yielding a positive expected miner revenue.

From Eq.(B.29), we have

$$\mu = \frac{1}{\mathbb{E}_{b_{-i}}[\sum_{j:j\neq i} b_j^2]} \tag{B.32}$$

$$= \frac{1}{c_\rho(n-1)}. \tag{B.33}$$

From Eqs. (B.28,B.30,B.31,B.33) we get the construction of the variation term as Eqs. (4.19,4.20).

[3]We introduced a coefficient $\frac{1}{2}$ because we initially constructed the variation term via partial derivatives.

B.4 Omitted Proofs

B.4.1 Proof of Theorem 4.2

First, we observe a sufficient condition for a TFM to be U-BNIC.

Observation 1. *$\tilde{M}$ is U-BNIC if*

$$\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0. \tag{B.34}$$

Proof. Proof. Because user i's expected utility $\tilde{u}(b_i, \mathbf{b}_{-i}; v_i) = u(b_i, \mathbf{b}_{-i}; v_i) - \theta(b_i, \mathbf{b}_{-i})$, if $\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0$, then for any bidding vector $\mathbf{b}$ and i's valuation v_i, mechanisms M and $\tilde{M}$ have the same expected utility

$$\mathbb{E}_{b_{-i}\sim V_{-i}}[u(b_i, \mathbf{b}_{-i}; v_i)] = \mathbb{E}_{b_{-i}\sim V_{-i}}[\tilde{u}(b_i, \mathbf{b}_{-i}; v_i)]. \tag{B.35}$$

As mechanism M is U-BNIC, it holds that $\tilde{M}$ is also U-BNIC.

Q.E.D.

As we have characterized a sufficient condition for U-BNIC, now we consider the condition for 1-SCP. We first introduce a lemma as a sufficient and necessary condition for a TFM to be 1-SCP:

Lemma B.3. *The mechanism $M = (\mathbf{a}, \mathbf{p}, r)$ is 1-SCP if and only if the following conditions are satisfied:*

- *Monotone allocation: $a_i(\cdot, \mathbf{b}_{-i})$ is monotonic non-decreasing,*
- *Constrained payment function:*

$$\begin{aligned} &a_i(b_i, \mathbf{b}_{-i})p_i(b_i, \mathbf{b}_{-i}) - r(b_i, \mathbf{b}_{-i}) \\ &= \int_0^{b_i} t\frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t}dt + a_i(0, \mathbf{b}_{-i})p_i(0, \mathbf{b}_{-i}) - r(0, \mathbf{b}_{-i}). \end{aligned} \tag{B.36}$$

Proof. Proof. Consider another mechanism $M' = (\mathbf{a}, \mathbf{p} - \frac{r}{\mathbf{a}}, 0)$. Since M' has zero miner revenue, it is 1-SCP if and only if it is U-DSIC.

From Lemma 4.1, M' is U-DSIC if and only if the given conditions hold. So M' is 1-SCP if and only if the conditions hold.

Notice that for the same bidding vector $\mathbf{b}$, the miner and user i have the same total utilities in mechanisms M and M'. So M is 1-SCP if and only if the conditions hold.

Q.E.D.

From Lemma B.3 we know that for an 1-SCP mechanism $(\mathbf{a}, \tilde{\mathbf{p}}, \tilde{\mathbf{r}})$, if we fix $\mathbf{b}_{-i}$, the difference of $a_i(\cdot, \mathbf{b}_{-i})\tilde{p}_i(\cdot, \mathbf{b}_{-i})$ and $\tilde{r}_i(\cdot, \mathbf{b}_{-i})$ is a constant. Furthermore, since $a(\cdot, \mathbf{b}_{-i})$ is monotonic increasing, if we want $\tilde{M}$ to be 1-SCP, from Lemma B.3 we need and only need:

$$\begin{aligned} & a_i(b_i, \mathbf{b}_{-i})\tilde{p}_i(b_i, \mathbf{b}_{-i}) - \tilde{r}(b_i, \mathbf{b}_{-i}) \\ & = \int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt + a_i(0, \mathbf{b}_{-i})\tilde{p}_i(0, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i}). \end{aligned} \tag{B.37}$$

From the construction of $\mathbf{p}$ we have

$$a_i(b_i, \mathbf{b}_{-i})p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt. \tag{B.38}$$

Since we set the boundary condition $\tilde{p}_i(0, \mathbf{b}_{-i}) = 0$, and the definition of $\{\theta_i\}$ as $\theta_i(b_i, \mathbf{b}_{-i}) = a_i(b_i, \mathbf{b}_{-i})(\tilde{p}_i(b_i, \mathbf{b}_{-i}) - p_i(b_i, \mathbf{b}_{-i}))$, we get a sufficient condition of 1-SCP as:

$$\theta_i(b_i, \mathbf{b}_{-i}) = \tilde{r}(b_i, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i}), \quad \forall i. \tag{B.39}$$

So $\tilde{M}$ is indeed U-BNIC and 1-SCP if M is U-DSIC and 1-SCP and T is admissible.

B.4.2 Proof of Lemma 4.2

We have

$$\tilde{r}(b_i, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i}) = \frac{1}{2}h\left(b_i^2 - \frac{\sum_{j\neq i} b_i^2 b_j^2}{c_\rho(n-1)}\right) \tag{B.40}$$

$$= \frac{1}{2}hb_i^2\left(1 - \frac{\sum_{j\neq i} b_j^2}{c_\rho(n-1)}\right) \tag{B.41}$$

$$= \theta_i(b_i, \mathbf{b}_{-i}) \tag{B.42}$$

and

$$\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}\theta_i(b_i, \mathbf{b}_{-i})$$

$$= \mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}\left[-\frac{1}{2}hb_i^2\left(\frac{\sum_{j\neq i} b_j^2}{c_\rho(n-1)} - 1\right)\right] \tag{B.43}$$

$$= -\frac{1}{2}hb_i^2\left(\frac{\sum_{j\neq i} \mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}[b_j^2]}{c_\rho(n-1)} - 1\right) \tag{B.44}$$

$$= -\frac{1}{2}hb_i^2\left(\frac{\sum_{j\neq i} c_\rho}{c_\rho(n-1)} - 1\right) \tag{B.45}$$

$$= 0. \tag{B.46}$$

Therefore, the variation term T is admissible.

B.4.3 Proof of Theorem 4.3

From the auxiliary mechanism method, the mechanism $\tilde{M} = (\mathbf{a}, \tilde{\mathbf{p}}, r)$ is U-BNIC and 1-SCP from Theorem 4.2. Now we prove the UIR, BF and U-SP properties.

Proof of UIR and BF

From Eq. (4.16) we know $p_i(0, \mathbf{b}_{-i}) = 0$. Then for $n \to \infty$, from Lemma 4.1 and $b_i \in [0, 1]$ we get:

$$a_i(b_i, \mathbf{b}_{-i}) p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt \tag{B.47}$$

$$= \int_0^{b_i} t \cdot \frac{e^t \sum_{j \neq i} e^{b_j}}{\left(e^t + \sum_{j \neq i} e^{b_j}\right)^2} dt. \tag{B.48}$$

Since $t \in [0, 1]$, it holds that

$$\frac{e^t \sum_{j \neq i} e^{b_j}}{e^t + \sum_{j \neq i} e^{b_j}} = \left(\frac{1}{e^t} + \frac{1}{\sum_{j \neq i} e^{b_j}}\right)^{-1} \tag{B.49}$$

$$\geq \left(\frac{1}{1} + \frac{1}{n-1}\right)^{-1} \tag{B.50}$$

$$= \frac{n-1}{n}. \tag{B.51}$$

Combined with $e^t + \sum_{j \neq i} e^{b_j} \leq en$, we have

$$\frac{e^t \sum_{j \neq i} e^{b_j}}{\left(e^t + \sum_{j \neq i} e^{b_j}\right)^2} \geq \frac{n-1}{en^2}. \tag{B.52}$$

Hence,

$$a_i(b_i, \mathbf{b}_{-i}) p_i(b_i, \mathbf{b}_{-i}) \geq \int_0^{b_i} t \frac{n-1}{en^2} dt \tag{B.53}$$

$$= \frac{n-1}{2en^2} \cdot b_i^2. \tag{B.54}$$

Therefore, the difference of the total collected fee and miner revenue in $\tilde{M}$ is

$$
\begin{aligned}
&\sum_{i=1}^{n} a_i(b_i, \mathbf{b}_{-i})\tilde{p}_i(b_i, \mathbf{b}_{-i}) - \tilde{r}(\mathbf{b}) \\
=& \sum_{i=1}^{n} a_i(b_i, \mathbf{b}_{-i}) p_i(b_i, \mathbf{b}_{-i}) + \sum_{i=1}^{n} \theta_i(b_i, \mathbf{b}_{-i}) - \tilde{r}(\mathbf{b}) && \text{(B.55)} \\
\geq& \frac{n-1}{2en^2} \cdot \sum_{i=1}^{n} b_i^2 - \left(h \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho(n-1)} - \frac{h}{2} \sum_{i=1}^{n} b_i^2 \right) \\
&- \frac{1}{2} h \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho(n-1)} \right) && \text{(B.56)} \\
=& \frac{n-1}{2en^2} \cdot \sum_{i=1}^{n} b_i^2 - \frac{h}{2c_\rho(n-1)} \sum_{1 \leq i < j \leq n} b_i^2 b_j^2 && \text{(B.57)} \\
\geq& \frac{n-1}{2en^2} \cdot \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n} b_i^2 \left(\frac{h}{4c_\rho(n-1)} \sum_{i=1}^{n} b_i^2 \right) && \text{(B.58)} \\
\geq& \frac{n-1}{2en^2} \cdot \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n} b_i^2 \left(\frac{h}{4c_\rho(n-1)} \cdot n \right) && \text{(B.59)} \\
=& \sum_{i=1}^{n} b_i^2 \cdot \left(\frac{n-1}{2en^2} - \frac{hn}{4c_\rho(n-1)} \right). && \text{(B.60)}
\end{aligned}
$$

So $\tilde{M}$ is budget feasible as long as $h \leq \frac{2c_\rho(n-1)^2}{en^3} = \Theta(c_\rho/n)$.

For user individual rationality,

$$
\begin{aligned}
&b_i - \tilde{p}_i(b_i, \mathbf{b}_{-i}) \\
=& b_i - p_i(b_i, \mathbf{b}_{-i}) - \frac{\theta_i(b_i, \mathbf{b}_{-i})}{a_i(b_i, \mathbf{b}_{-i})} && \text{(B.61)} \\
=& \frac{1}{a_i(b_i, \mathbf{b}_{-i})} \Big[b_i \left(a_i(0, \mathbf{b}_{-i}) + \int_0^{b_i} \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt \right) \\
&\qquad - \int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} \Big] && \text{(B.62)} \\
=& \frac{1}{a_i(b_i, \mathbf{b}_{-i})} \Big[b_i a_i(0, \mathbf{b}_{-i}) + \int_0^{b_i} (b_i - t) \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt \\
&\qquad + \frac{1}{2} h b_i^2 \left(\frac{\sum_{j \neq i} b_j^2}{c_\rho(n-1)} - 1 \right) \Big]. && \text{(B.63)}
\end{aligned}
$$

From Eq.(B.52), we also have

$$\begin{aligned}
&\int_0^{b_i}(b_i-t)\frac{\partial a_i(t,\mathbf{b}_{-i})}{\partial t}dt \\
&=\int_0^{b_i}(b_i-t)\cdot\frac{e^t\sum_{j\neq i}e^{b_j}}{\left(e^t+\sum_{j\neq i}e^{b_j}\right)^2}dt && \text{(B.64)}\\
&\geq\int_0^{b_i}(b_i-t)\cdot\frac{n-1}{en^2}dt && \text{(B.65)}\\
&=\frac{n-1}{2en^2}\cdot b_i^2. && \text{(B.66)}
\end{aligned}$$

Therefore, when $h=\frac{2c_\rho(n-1)^2}{en^3}$, since $c_\rho\leq 1$, we have

$$\begin{aligned}
&\tilde{u}_i(b_i,\mathbf{b}_{-i};b_i)\\
&=b_ia_i(0,\mathbf{b}_{-i})+\int_0^{b_i}(b_i-t)\frac{\partial a_i(t,\mathbf{b}_{-i})}{\partial t}dt\\
&\quad+\frac{1}{2}hb_i^2\left(\frac{\sum_{j\neq i}b_j^2}{c_\rho(n-1)}-1\right) && \text{(B.67)}\\
&\geq\frac{b_i}{en}+\int_0^{b_i}(b_i-t)\frac{\partial a_i(t,\mathbf{b}_{-i})}{\partial t}dt\\
&\quad+\frac{1}{2}hb_i^2\left(\frac{\sum_{j\neq i}b_j^2}{c_\rho(n-1)}-1\right) && \text{(B.68)}\\
&\geq b_i^2\left(\frac{1}{en}+\frac{n-1}{2en^2}-\frac{h}{2}\right) && \text{(B.69)}\\
&=b_i^2\left(\frac{1}{en}+\frac{n-1}{2en^2}-\frac{c_\rho(n-1)^2}{en^3}\right) && \text{(B.70)}\\
&\geq b_i^2\left(\frac{1}{en}+\frac{n-1}{2en^2}-\frac{1}{en}\right) && \text{(B.71)}\\
&\geq 0. && \text{(B.72)}
\end{aligned}$$

So the UIR also holds for $h=\frac{2c_\rho(n-1)^2}{en^3}$.

Therefore, we have shown $h_*(n,c_\rho)\geq\frac{2c_\rho(n-1)^2}{en^3}=\Omega(c_\rho/n)$.

Proof of U-SP

We first consider the auxiliary mechanism $(\mathbf{a}, \mathbf{p}, r)$. Denote $w_{-i} = \sum_{j \neq i} e^{mb_j}$, then we have

$$a_i(b_i, \mathbf{b}_{-i}) = \frac{e^{mb_i}}{e^{mb_i} + w_{-i}} \tag{B.73}$$

$$p_i(b_i, \mathbf{b}_{-i}) = b_i - \frac{e^{mb_i} + w_{-i}}{me^{mb_i}} \ln \frac{e^{mb_i} + w_{-i}}{1 + w_{-i}}. \tag{B.74}$$

The utility of identity i is $u_i(b_i, \mathbf{b}_{-i}; v_i) = a_i(b_i, \mathbf{b}_{-i})(v_i - p_i(b_i, \mathbf{b}_{-i}))$. We can also regard as it as a function of (b_i, w_{-i}, v_i), then we have

$$\left.\frac{\partial u_i}{\partial w_{-i}}\right|_{b_i = v_i} = \frac{-\frac{1}{1+w_{-i}} + \frac{1}{e^m + w_{-i}}}{m} \leq 0. \tag{B.75}$$

As injecting fake bids is equivalent to increasing w_{-i} for identity i in the auxiliary mechanism, it cannot increase identity i's utility in the auxiliary mechanism.

However, the injected fake bids can influence user i's utility in two more aspects, as:

- The variation term.
- The utilities of fake identities.

We denote h as the scaling parameter for total user number $n + l$, hence, we have

$$h \leq \frac{2c_\rho(n + l - 1)^2}{e(n + l)^3}. \tag{B.76}$$

Without fake identities, the expectation of $\theta_i(b_i, \mathbf{b}_{-i})$ is zero. Therefore, denote $\Omega = \{i\} \cup \{n + 1, \cdots, n + l\}$, then Ω is the set of all identities that the user has access to, and we only need to show that

$$\mathbb{E}_{\mathbf{b}_{-i} \sim V_{-i}} \Bigg[\sum_{j=n+1}^{n+l} a_j(b_j, \mathbf{b}_{-j}^+) p_j(b_j, \mathbf{b}_{-j}^+) + \sum_{j \in \Omega} \theta(b_j, \mathbf{b}_{-j}^+) \Bigg] \geq 0. \tag{B.77}$$

For a refined analysis of constants, we denote the Sybil attacker has real identity i, and submits fake bids with identities $n+1, \cdots, n+l$. We denote that:

$$\sigma = \sum_{j \le n, j \neq i} b_j^2,$$
$$\sigma_{\#} = \sum_{j=n+1}^{n+l} b_j^2,$$

Then σ is a random variable independent to any b_j for $j \in \Omega$, and it holds that

$$\mathbb{E}_{\mathbf{b}_{-i} \sim V_{-i}}[\sigma] = c_\rho(n-1).$$

From Eq. (B.54) , we have

$$\sum_{j=n+1}^{n+l} a_j(b_j, \mathbf{b}_{-j}^+) p_j(b_j, \mathbf{b}_{-j}^+) \ge \frac{n+l-1}{2e(n+l)^2} \sum_{j=n+1}^{n+l} b_j^2 \tag{B.78}$$
$$\ge \frac{1}{2e(n+l+2)} \cdot \sigma_{\#}. \tag{B.79}$$

For $j \in \Omega$, we have

$$\theta(b_j, \mathbf{b}_{-j}^+) = -\frac{1}{2} h b_j^2 \left(\frac{\sum_{t \le n+l, t \neq j} b_t^2}{c_\rho(n+l-1)} - 1 \right) \tag{B.80}$$
$$= -\frac{1}{2} h b_j^2 \left(\frac{\sigma + \sigma_{\#} + b_i^2 - b_j^2}{c_\rho(n+l-1)} - 1 \right). \tag{B.81}$$

Here, σ is the only random variable in the expression, and

$$\begin{aligned}
&\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}\left[\sum_{j\in\Omega}\theta(b_j,\mathbf{b}^+_{-j})\right]\\
&=\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}\left[-\sum_{j\in\Omega}\frac{1}{2}hb_j^2\left(\frac{\sigma+\sigma_\#+b_i^2-b_j^2}{c_\rho(n+l-1)}-1\right)\right] &\text{(B.82)}\\
&=-\frac{h}{2}\sum_{j\in\Omega}b_j^2\cdot\left(\frac{\mathbb{E}_{\mathbf{b}_{-i}\sim V_{-i}}[\sigma]+\sigma_\#+b_i^2}{c_\rho(n+l-1)}-1\right)\\
&\quad+\frac{h}{2}\sum_{j\in\Omega}\frac{b_j^4}{c_\rho(n+l-1)} &\text{(B.83)}\\
&=-\frac{h}{2}(\sigma_\#+b_i^2)\cdot\frac{\sigma_\#+b_i^2-c_\rho l}{c_\rho(n+l-1)}+\frac{h}{2}\cdot\frac{\sum_{j\in\Omega}b_j^4}{c_\rho(n+l-1)} &\text{(B.84)}\\
&=\frac{h}{2c_\rho(n+l-1)}\Bigg(-\sigma_\#^2-2b_i^2\sigma_\#-b_i^4\\
&\qquad\qquad+c_\rho l(\sigma_\#+b_i^2)+b_i^4+\sum_{j=n+1}^{n+l}b_j^4\Bigg) &\text{(B.85)}\\
&\geq\frac{h}{2c_\rho(n+l-1)}\left(-\sigma_\#^2-2b_i^2\sigma_\#\right) &\text{(B.86)}\\
&\geq\frac{n+l-1}{e(n+l)^3}\left(-\sigma_\#^2-2\sigma_\#\right). &\text{(B.87)}
\end{aligned}$$

From Eqs. (B.79,B.87), Eq. (B.77) is implied by

$$\frac{1}{2e(n+l+2)}\sigma_\#\geq\frac{n+l-1}{e(n+l)^3}(\sigma_\#^2+2\sigma_\#). \tag{B.88}$$

Noticing that $\forall b_i\leq 1$, so $\sigma_\#\leq l$. We only need

$$(n+l)^3\geq 2(n+l+2)(n+l-1)(l+2). \tag{B.89}$$

Now for any $C\in[0,1)$, we assume $n\geq\frac{6C+5}{1-C^2}$, then denote $\varphi=\frac{l}{n}\leq C$, and we have

$$(1+\varphi)^3 n^3 - 2((1+\varphi)n+2)((1+\varphi)n-1)(\varphi n+2) \tag{B.90}$$

$$= (1+\varphi)((1-\varphi^2)n - (6\varphi+4))n - 4)n + 8. \tag{B.91}$$

Since $n \geq \frac{6C+5}{1-C^2}$, we see that $n \geq 5$, and $(1-\varphi^2)n - (6\varphi+4) \geq (1-C^2)n - (6C+4) \geq 1$. Hence,

$$(1+\varphi)((1-\varphi^2)n - (6\varphi+4))n - 4)n + 8 \tag{B.92}$$

$$\geq 1 \cdot (1 \cdot n - 4)n + 8 \tag{B.93}$$

$$\geq 13 \tag{B.94}$$

$$> 0. \tag{B.95}$$

Now we prove that the mechanism is $(C, \frac{6C+5}{1-C^2})$-U-SP for any $C \in [0, 1)$.

B.4.4 Proof of Theorem 4.4

From the auxiliary mechanism method, we have the U-BNIC and 1-SCP properties as long as the allocation rule is monotone. Hence, our proof for Theorem consists of 3 parts:

- Proof of monotonicity of allocation rule.
- Proof of UIR and BF.
- Proof of U-SP.

Proof of Monotonicity of Allocation Rule.

For monontonicity, we just need to show that for any $\mathbf{b}_{-i}$, $a_i(b_i, \mathbf{b}_{-i}) \geq a_i(b_i', \mathbf{b}_{-i})$ if $b_i \geq b_i'$.

If $1 \leq n \leq k$, we have $a_i(b_i, \mathbf{b}_{-i}) = a_i(b_i', \mathbf{b}_{-i}) = 1$, so the monotonicity holds. Now we consider $n > k$.

For convenience denote $w_i = e^{mb_i}$ and without loss of generality we assume $i = n$. Now For any map $X : \mathbb{N}_+ \to [0, 1)$, vector $\mathbf{t}$ s.t. $0 = t_0 < t_1 < t_2 < \cdots < t_{n-1} < t_n = 1$, $B_0 \subseteq [0, 1)$ and $k \leq n-1$, define an algorithm as Algorithm 7:

Algorithm 7 $Draw(X, \mathbf{t}, B_0, k)$

Input $X, \mathbf{t}, B_0, k$;
$B \leftarrow B_0$; $S \leftarrow \emptyset$;
$u \leftarrow 1$; $v \leftarrow 1$;
while $v \leq k$ **do**
    $x \leftarrow X(u)$;
    if $x \notin B$ **then**
        Find i s.t. $x \in [t_{i-1}, t_i)$;
        $S \leftarrow S \cup i$;
        $v \leftarrow v + 1$;
    $B \leftarrow B \cup [t_{x-1}, t_x)$;
    $u \leftarrow u + 1$;
Output S;

Now we denote $W_i = \frac{w_i}{\sum_{i=1}^n w_i}$ for $1 \leq i \leq n$, and

$$W_i' = \begin{cases} W_i, & i \leq n-1 \\ \frac{e^{mb_n'}}{\sum_{i=1}^n w_i}, & i = n. \end{cases}$$

Then, we define $\mathbf{t}, \mathbf{t}'$ as

$$t_i = \sum_{j=1}^{i} W_j, \quad 0 \leq i \leq n$$

$$t_i' = \begin{cases} \sum_{j=1}^{i} W_j', & 0 \leq i \leq n \\ 1, & i = n+1. \end{cases}$$

Then when X is a *i.i.d.* uniform random sequence in $[0, 1)$, we can see that

- $Draw(X, \mathbf{t}, \emptyset, k)$ randomly samples k items among $\{1, \cdots, n\}$ with weights $\{W_i\}$ without replacement.
- $Draw(X, \mathbf{t}', [t_n', 1), k)$ randomly samples k items among $\{1, \cdots, n\}$ with (relative) weights $\{W_i'\}_{i \in [n]}$ without replacement.

In fact, Algorithm 7 performs random drawing without replacement in the following way. Every round an item in $\{1, \cdots, n\}$ is drawn, and in the second scenario the total weights is less than 1 so that a "placeholder" item $n+1$ with weight $1 - t_n'$ is added. If the item is

already drawn or is the "placeholder", we draw again; other wise, we finalize it and add it to S.

In the rest of the proof, we prove that

$$\begin{aligned}&\Pr[n \in Draw(X, \mathbf{t}, \emptyset, k)]\\ &\quad \geq \Pr[n \in Draw(X, \mathbf{t}', [t'_n, 1), k)]\end{aligned}$$

by actually showing

$$n \in Draw(X, \mathbf{t}', [t'_n, 1), k) \Rightarrow n \in Draw(X, \mathbf{t}, \emptyset, k).$$

In fact, assume $n \in Draw(X, \mathbf{t}', [t'_n, 1), k)$. By the time the drawing process $Draw(X, \mathbf{t}', [t'_n, 1), k)$ stops, if no value $X(u) \in [t'_n, 1)$ is obtained, then $Draw(X, \mathbf{t}, \emptyset, k)$ has exactly the same outcome, so it also contains n.

If in some round $X(u) \in [t'_n, 1)$ is obtained in $Draw(X, \mathbf{t}', [t'_n, 1), k$, we consider the first round that happens.

Before that round, $Draw(X, \mathbf{t}, \emptyset, k)$ have the same outcome, so it is not stopped either. In that round, $Draw(X, \mathbf{t}, \emptyset, k)$ adds n to S, so $n \in Draw(X, \mathbf{t}, \emptyset, k)$.

So we have shown that $n \in Draw(X, \mathbf{t}', [t'_n, 1), k) \Rightarrow n \in Draw(X, \mathbf{t}, \emptyset, k)$, implying the monotonicity of the allocation rule.

Proof of UIR and BF.

From Lemma 4.1, similar to the case of block size 1, we essentially need to derive a lower bound on $\frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t}$, in order to lower bound the total payment. Therefore, we only need to analyze the partial derivative of $\delta_t(i; j)$ on w_i.

When we fix j and $\mathbf{b}_{-i}$ (i.e., $\mathbf{w}_{-i}$), we can regard $\delta_t(i; j)$ as a function of w_i. Here we make a notation of X_s for $0 \leq s \leq k-1$ as

$$X_s = W - w_i - \sum_{z=1}^{s} w_{j_z}, \tag{B.96}$$

then X_s is a constant.

From Eq. (4.24) we get (note that $W - \sum_{z=1}^{s} w_{j_s} = X_s + w_i$)

$$\frac{\partial \delta_t(i;j)}{\partial w_i} = \left(\prod_{s=1}^{t-1} w_{j_s}\right) \cdot \frac{\partial}{\partial w_i} \frac{w_i}{\prod_{s=0}^{t-1}(X_s + w_i)}, \tag{B.97}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial w_i} \frac{w_i}{\prod_{s=0}^{t-1}(X_s + w_i)} \\
&= \frac{\partial}{\partial w_i} \left(w_i \cdot \prod_{s=0}^{t-1} \frac{1}{X_s + w_i} \right) && \text{(B.98)} \\
&= \prod_{s=0}^{t-1} \frac{1}{X_s + w_i} + w_i \cdot \frac{\partial}{\partial w_i} \prod_{s=0}^{t-1} \frac{1}{X_s + w_i} && \text{(B.99)} \\
&= \prod_{s=0}^{t-1} \frac{1}{X_s + w_i} - w_i \cdot \left(\sum_{s=0}^{t-1} \frac{1}{X_s + w_i} \right) \cdot \left(\prod_{s=0}^{t-1} \frac{1}{X_s + w_i} \right) && \text{(B.100)} \\
&= \left(\prod_{s=0}^{t-1} \frac{1}{X_s + w_i} \right) \cdot \left(1 - w_i \sum_{s=0}^{t-1} \frac{1}{X_s + w_i} \right) && \text{(B.101)}
\end{aligned}$$

Notice that $X_s + w_i$ is a sum of $(n-s)$ weights, each one no less than 1, so $\frac{1}{X_s + w_i} \leq \frac{1}{n-s} \leq \ln \frac{n-s}{n-s-1}$, and $w_i = e^{mb_i} \leq e^m$. Therefore,

$$\begin{aligned}
1 - w_i \sum_{s=0}^{t-1} \frac{1}{X_s + w_i} &\geq 1 - e^m \sum_{s=0}^{t-1} \ln \frac{n-s}{n-s-1} && \text{(B.102)} \\
&= 1 - e^m \ln \frac{n}{n-t}. && \text{(B.103)}
\end{aligned}$$

Denote

$$D(m, \lambda) = 1 - e^m \ln \frac{\lambda}{\lambda - 1}, \tag{B.104}$$

then $\forall \frac{n}{k} < \frac{e}{e-1}$, $\exists m > 0$ s.t. $D\left(m, \frac{n}{k}\right) > 0$.

Therefore, from Eq. (B.97) we have

$$\frac{\partial \delta_t(i;j)}{\partial w_i} \geq D\left(m, \frac{n}{k}\right)\left(\prod_{s=1}^{t-1} w_{j_s}\right)\left(\prod_{s=0}^{t-1} \frac{1}{X_s + w_i}\right) \tag{B.105}$$

$$= D\left(m, \frac{n}{k}\right) \cdot \frac{w_{j_1}}{X_0 + w_i} \cdot \frac{w_{j_2}}{X_1 + w_i} \cdot \cdots \cdot \frac{1}{X_{t-1} + w_i}. \tag{B.106}$$

We notice that $\frac{w_{j_1}}{X_0+w_i} \cdot \frac{w_{j_2}}{X_1+w_i} \cdots \cdots \frac{w_{j_{t-1}}}{X_{t-2}+w_i}$ is just the probability that the sampling outcome of the first $t-1$ rounds are $(j_1, j_2, \cdots, j_{t-1})$, denoted as $P(j_{[t-1]})$. Furthermore, from $X_{t-1}+w_i \leq e^m \cdot n$, we have

$$\frac{\partial \delta_t(i;j)}{\partial w_i} \geq \frac{D\left(m, \frac{n}{k}\right)}{e^m n} P(j_{[t-1]}). \tag{B.107}$$

Therefore from Eq. (4.23):

$$\frac{\partial \delta_t(i)}{\partial w_i} = \sum_{j \in J_t(i)} \frac{\partial \delta_t(i;j)}{\partial w_i} \tag{B.108}$$

$$\geq \frac{D\left(m, \frac{n}{k}\right)}{e^m n} \sum_{j \in J_t(i)} P(j_{[t-1]}). \tag{B.109}$$

For $j \in J_t(i)$, we observe that $j_{[t-1]}$ iterates through all $(t-1)$-permutations of $[n]$ that does not contain element i. Therefore, $\sum_{j \in J_t(i)} P(j_{[t-1]})$ is the probability that i is not chosen in the first $(t-1)$ rounds.

To compute the probability that i is not chosen in the first $(t-1)$ rounds, we consider each round. In each round, there are at least $(n-k)$ users each with weight at least 1, and user i has weight at most e^m, so i is chosen with probability at most $\frac{e^m}{n-k}$. Therefore for t rounds, the probability that i is not ever chosen is at most $\left(1 - \frac{e^m}{n-k}\right)^t \geq \left(1 - \frac{e^m}{n-k}\right)^k = (1 - o(1))e^{-\frac{e^m k}{n-k}}$. That implies:

$$\frac{\partial \delta_t(i)}{\partial w_i} \geq (1-o(1))\frac{D\left(m,\frac{n}{k}\right)}{e^m n}e^{-\frac{e^m k}{n-k}}, \tag{B.110}$$

so

$$\frac{\partial a_i(b_i,\mathbf{b}_{-i})}{\partial b_i} = \frac{\partial w_i}{\partial b_i}\cdot\frac{\partial a_i(b_i,\mathbf{b}_{-i})}{\partial w_i} \tag{B.111}$$

$$= me^{mb_i}\cdot\sum_{t=1}^{k}\frac{\partial \delta_t(i)}{\partial w_i} \tag{B.112}$$

$$\geq me^{mb_i}\cdot\sum_{t=1}^{k}\left((1-o(1))\frac{D\left(m,\frac{n}{k}\right)}{e^m n}e^{-\frac{e^m k}{n-k}}\right) \tag{B.113}$$

$$= \frac{k}{n}\left((1-o(1))me^{mb_i}\frac{D\left(m,\frac{n}{k}\right)}{e^m}e^{-\frac{e^m k}{n-k}}\right) \tag{B.114}$$

For any fixed $\lambda_0 > \frac{e}{e-1}$, let $\lambda = \frac{n}{k}$. If $\lambda \geq \lambda_0$, let

$$m = m_{\#}(\lambda_0) = \min\left\{\frac{1}{2}\ln\frac{1}{\ln\frac{\lambda_0}{\lambda_0-1}}, 1\right\} \tag{B.115}$$

be a constant. Then we have:

$$D(m,\lambda) = \max\left\{1-\sqrt{\ln\frac{\lambda}{\lambda-1}}, 1-e\ln\frac{\lambda}{\lambda-1}\right\}. \tag{B.116}$$

Because $m_{\#}(\cdot)$ and $D(m,\cdot)$ are non-decreasing, we have

$$\frac{\partial a_i(b_i,\mathbf{b}_{-i})}{\partial t} \geq \frac{k}{n}\left((1-o(1))me^{mb_i}\frac{D\left(m,\frac{n}{k}\right)}{e^m}e^{-\frac{e^m k}{n-k}}\right) \tag{B.117}$$

$$\geq \frac{k}{n}\left((1-o(1))m\frac{D\left(m,\lambda_0\right)}{e}e^{-\frac{e^m}{\lambda_0-1}}\right). \tag{B.118}$$

Because $m, \lambda_0, D(m,\lambda_0)$ are all positive constants, we get

$$\frac{\partial a_i(b_i,\mathbf{b}_{-i})}{\partial t} \geq \frac{k}{n}f(\lambda_0)(1-o(1)). \tag{B.119}$$

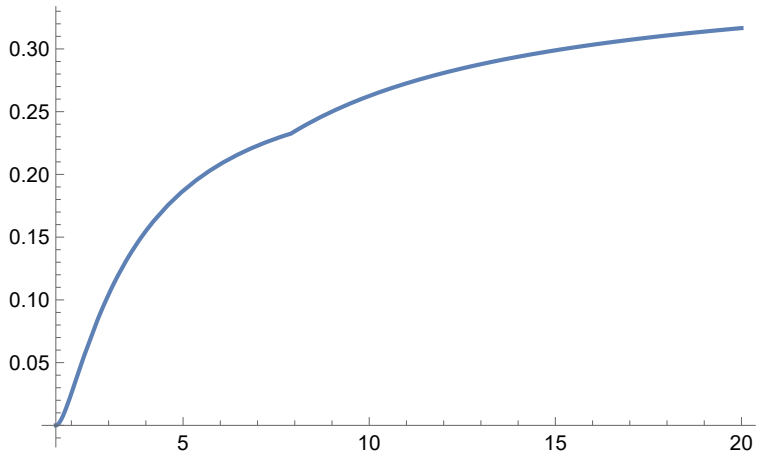

Figure B.1: The plot of $f(\cdot)$.

Therefore, from Lemma 4.1 and $p_i(0, \mathbf{b}_{-i}) = 0$, we get

$$a_i(b_i, \mathbf{b}_{-i}) p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} t \frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t} dt \tag{B.120}$$

$$\geq \int_0^{b_i} t \frac{k}{n} f(\lambda_0)(1 - o(1)) dt \tag{B.121}$$

$$= f(\lambda_0) \Theta \left(\frac{k}{n} b_i^2 \right). \tag{B.122}$$

Here, the expression of $f(\cdot)$ is given by

$$f(\lambda) = \frac{m_{\#}(\lambda) D(m_{\#}(\lambda), \lambda)}{e^{m_{\#}(\lambda)}} \tag{B.123}$$

and can be plotted as in Figure B.1. It can be noticed that $f(\cdot)$ is monotonic increasing and

$$\lim_{\lambda \to +\infty} f(\lambda) = \frac{1}{e}. \tag{B.124}$$

Then, when we let $\tilde{p}_i(b_i, \mathbf{b}_{-i}) = p_i(b_i, \mathbf{b}_{-i}) + \frac{\theta_i(b_i, \mathbf{b}_{-i})}{a_i(b_i, \mathbf{b}_{-i})}$ while using the variation term of Eqs. (4.19-4.20), similar to the argument of Eqs. (B.55-B.63), we can get the UIR and BF properties.

Detailed constant analysis.

From the assumption that $n \geq 30$ and $n > \frac{e}{e-1}k$, we have $n - k \geq 3 > e \geq e^m$. Since

$$(1-\alpha)^k = (1 + \frac{\alpha}{1-\alpha})^{-k} \geq e^{-\frac{k\alpha}{1-\alpha}}, \quad \alpha \in [0, 1)$$

we have

$$\left(1 - \frac{e^m}{n-k}\right)^k \geq e^{-\frac{e^m k}{n-k-e^m}}. \tag{B.125}$$

Then we get that

$$\frac{\partial \delta_t(i)}{\partial w_i} \geq \frac{D\left(m, \frac{n}{k}\right)}{e^m n} e^{-\frac{e^m k}{n-k-e^m}}, \tag{B.126}$$

Since $n \geq 30$, we have

$$\begin{aligned}
\frac{\partial a_i(b_i, \mathbf{b}_{-i})}{\partial t} &\geq \frac{k}{n}\left(m e^{m b_i} \frac{D\left(m, \frac{n}{k}\right)}{e^m} e^{-\frac{e^m k}{n-k-e^m}}\right) && \text{(B.127)} \\
&> \frac{k}{n}\left(m e^{m b_i} \frac{D\left(m, \frac{n}{k}\right)}{e^m} e^{-\frac{e^m k}{(n-3)-k}}\right) && \text{(B.128)} \\
&\geq \frac{k}{n}\left(m e^{m b_i} \frac{D\left(m, \frac{n}{k}\right)}{e^m} e^{-\frac{e^m k}{0.9n-k}}\right) && \text{(B.129)} \\
&\geq \frac{k}{n}\left(m \frac{D\left(m, \lambda_0\right)}{e^m} e^{-\frac{e}{0.9\lambda_0 - 1}}\right) && \text{(B.130)} \\
&= \frac{k}{n}\left(f(\lambda_0) e^{-\frac{e}{0.9\lambda_0 - 1}}\right). && \text{(B.131)}
\end{aligned}$$

Here, we can let

$$g(\lambda) = e f(\lambda) e^{-\frac{e}{0.9\lambda - 1}}, \tag{B.132}$$

then g is increasing and

$$\lim_{\lambda \to \infty} g(\lambda) = 1. \tag{B.133}$$

It holds that

$$a_i(b_i, \mathbf{b}_{-i})p_i(b_i, \mathbf{b}_{-i}) = \int_0^{b_i} t\frac{\partial a_i(t, \mathbf{b}_{-i})}{\partial t}dt \tag{B.134}$$

$$\geq \frac{g(\lambda)}{2e} \cdot \frac{k}{n} b_i^2. \tag{B.135}$$

Similar to the argument of Eqs. (B.55-B.63), the UIR and BF hold when

$$h_* = g(\lambda_0) \cdot \frac{2kc_\rho(n-1)}{en^2}. \tag{B.136}$$

Proof of U-SP.

Since the variation term of the mechanism for block size k has the same form as block size 1, we can show that the effects of the variation term do not influence the U-SP property in the same way as Appendix B.4.3. Furthermore, because fake transactions have zero valuation and non-negative payment, we only need to prove the following proposition:

Proposition B.2. *For any user i, adding a fake bid will not benefit her utility in the Auxiliary Mechanism M for block size k.*

Actually when $p_i(0, \mathbf{b}_{-i}) = 0$, the payment function in Myerson's Lemma has an equivalent form [41]:

$$a_i(b_i, \mathbf{b}_{-i})p_i(b_i, \mathbf{b}_{-i}) = a_i(b_i, \mathbf{b}_{-i})b_i - \int_0^{b_i} a_i(t, \mathbf{b}_{-i})dt. \tag{B.137}$$

Therefore, the utility of user i when truthfully bidding in the auxiliary mechanism is:

$$u_i(b_i, \mathbf{b}_{-i}; b_i) = \int_0^{b_i} a_i(t, \mathbf{b}_{-i})dt. \tag{B.138}$$

Now we only need to show that when we inject a fake transaction, the probability that a user (bidding arbitrary t) is confirmed would not increase, as the following lemma:

Lemma B.4. *In a weighted random sampling without replacement, if we add a new item, the probability that any already existing item is chosen does not increase.*

Proof. Proof.

Consider the Algorithm 7. Now we assume there are n items $1, \cdots, n$ with weights $w_1, \cdots, w_n$ and without loss of generality we assume $\sum_{i=1}^{n} = 1$, and define $t_j = \sum_{i=1}^{j} w_i$, then when X is a *i.i.d.* uniform random sequence in $[0, 1)$, we can see that

- $Draw(X, \mathbf{t}, \emptyset, k)$ randomly samples k items among $\{1, \cdots, n\}$ without replacement.
- $Draw(X, \mathbf{t}, [t_{n-1}, 1), k)$ randomly samples k items among $\{1, \cdots, n-1\}$ without replacement.

We recall that Algorithm 7 performs random drawing without replacement in the following way. Every round an item in $\{1, \cdots, n\}$ is drawn. If the item is already drawn or does not exist, we draw again; otherwise, we finalize it and add it to S.

In the rest of the proof, we prove that $\forall i \in \{1, \cdots, n-1\}$,

$$\begin{aligned} &\Pr[i \in Draw(X, \mathbf{t}, [t_{n-1}, 1), k)] \\ &\geq \Pr[i \in Draw(X, \mathbf{t}, \emptyset, k)] \end{aligned}$$

by actually showing

$$i \in Draw(X, \mathbf{t}, \emptyset, k) \;\Rightarrow\; i \in Draw(X, \mathbf{t}, [t_{n-1}, 1), k).$$

In fact, for fixed X, because

$$P \subseteq Q \quad \Rightarrow \quad P \cup R \subseteq Q \cup R,$$

after each round of drawing, the B in $Draw(X, \mathbf{t}, \emptyset, k)$ is always a subset of the B in $Draw(X, \mathbf{t}, [t_{n-1}, 1), k)$. Therefore, $Draw(X, \mathbf{t}, [t_{n-1}, 1), k)$ would draw no less rounds than $Draw(X, \mathbf{t}, \emptyset, k)$.

Besides, we see that when $i \neq n$, i is drawn if and only if a $x \in [t_{i-1}, t_i)$ appears by the time the drawing completes, so if $i \in Draw(X, \mathbf{t}, \emptyset, k)$, we indeed have $i \in Draw(X, \mathbf{t}, [t_{n-1}, 1), k)$.

Hence we have shown that $\Pr[i \in Draw(X, \mathbf{t}, [t_{n-1}, 1), k)] \geq \Pr[i \in Draw(X, \mathbf{t}, \emptyset, k)]$.

□

Q.E.D.

From Lemma B.4 we prove that our TFM for block size k is U-SP.

For the corresponding constants, we note that in the mechanism of block size 1, the expected payment of a user bidding b_i is lower bounded by $\sim \frac{b_i^2}{2en}$ and $h \lesssim \frac{2c_\rho}{en}$, and it is $\big(C, O(\frac{1}{1-C})\big)$-U-SP for any $C < 1$. In the mechanism of block size k, the expected payment of user i is lower bounded by $g(\lambda_0)k \cdot \frac{b_i^2}{2en}$, and $h \lesssim g(\lambda_0)k \cdot \frac{2c_\rho}{en}$. Hence, it can be shown in a similar way that Mechanism 3 is also $\big(C, O(\frac{1}{1-C})\big)$-U-SP for any $C < 1$.

B.4.5 Proof of Theorem 4.5

Without loss of generality, we can assume the miner will conduct the deviation in this way: in Stage 1 the miner deletes transactions one by one, and then in Stage 2 inject fake transactions one by one. Then we introduce two lemmas before proving the theorem: firstly analyze the robustness of the miner revenue function $\tilde{r}$, then upper bound the advantage the miner may gain in each stage.

Robustness analysis of the miner revenue function.

Firstly, we assume that the mean of b_i^2 is close to $c_\rho = \Theta(1)$, which holds with high probability with large n and $\Delta = o(n)$. Here we define $H = Lc_\rho$, then we prove the following lemma, showing that as long as the average of $\{b_i^2\}$ is close to c_ρ, adding or deleting a transaction would not have a significant impact on the miner revenue:

Lemma B.5. *If* $\left|\frac{\sum_{i=1}^n b_i^2}{n} - c_\rho\right| < \delta$, *and recall that*

$$\tilde{r}(\mathbf{b}) = \frac{Hk}{2n}\left(\sum_{i=1}^n b_i^2 - \frac{\sum_{1\le i<j\le n} b_i^2 b_j^2}{c_\rho(n-1)}\right), \tag{B.139}$$

then for $n \ge 3$, *there exists a constant* $C_{LB.5}$ *s.t.* $\forall j \in [n]$,

$$|\tilde{r}(\mathbf{b}_{-j}) - \tilde{r}(\mathbf{b})| \le C_{LB.5}\delta \cdot \frac{Hk}{c_\rho n}. \tag{B.140}$$

Proof. Proof. Without loss of generality we assume $j = n$. Then, we compute that

$$\begin{aligned}
&\frac{\tilde{r}(\mathbf{b}_{-n}) - \tilde{r}(\mathbf{b})}{\frac{1}{2}Hk} \\
&= \frac{\sum_{i=1}^{n-1} b_i^2}{n-1} - \frac{\sum_{1\le i<j\le n-1} b_i^2 b_j^2}{c_\rho (n-2)(n-1)} \\
&\qquad - \frac{\sum_{i=1}^{n} b_i^2}{n} + \frac{\sum_{1\le i<j\le n} b_i^2 b_j^2}{c_\rho (n-1)n} \qquad \text{(B.141)} \\
&= \frac{1}{n(n-1)} \sum_{i=1}^{n-1} b_i^2 - \frac{b_n^2}{n} \\
&\qquad - \frac{2\sum_{1\le i<j\le n-1} b_i^2 b_j^2}{c_\rho n(n-1)(n-2)} + \frac{b_n^2 \sum_{i=1}^{n-1} b_i^2}{c_\rho n(n-1)} \qquad \text{(B.142)} \\
&= \frac{1}{n(n-1)} \left(\sum_{i=1}^{n-1} b_i^2 - \frac{2\sum_{1\le i<j\le n-1} b_i^2 b_j^2}{c_\rho (n-2)} \right) \\
&\qquad + \frac{b_n^2}{n} \left(\frac{\sum_{i=1}^{n-1} b_i^2}{c_\rho (n-1)} - 1 \right) \qquad \text{(B.143)}
\end{aligned}$$

From the assumption we see that $\left| \frac{\sum_{i=1}^{n-1} b_i^2}{c_\rho (n-1)} - 1 \right| = O(\delta / c_\rho)$ and $b_n^2 \le 1$, we have

$$\left| \frac{b_n^2}{n} \left(\frac{\sum_{i=1}^{n-1} b_i^2}{c_\rho (n-1)} - 1 \right) \right| = O(\delta / c_\rho n). \qquad \text{(B.144)}$$

Now we only need to prove that $\left| \sum_{i=1}^{n-1} b_i^2 - \frac{2\sum_{1\le i<j\le n-1} b_i^2 b_j^2}{c_\rho (n-2)} \right| = O(\delta n / c_\rho)$.

In fact, we notice that

$$2 \sum_{1\le i<j\le n-1} b_i^2 b_j^2 = \left(\sum_{i=1}^{n-1} b_i^2 \right)^2 - \sum_{i=1}^{n-1} b_i^4. \qquad \text{(B.145)}$$

Hence,

$$\begin{aligned}
&\left|\sum_{i=1}^{n-1} b_i^2 - \frac{2\sum_{1\le i<j\le n-1} b_i^2 b_j^2}{c_\rho(n-2)}\right| \\
&= \left|\sum_{i=1}^{n-1} b_i^2 - \frac{\left(\sum_{i=1}^{n-1} b_i^2\right)^2 - \sum_{i=1}^{n-1} b_i^4}{c_\rho(n-2)}\right| && \text{(B.146)} \\
&= \left|\sum_{i=1}^{n-1} b_i^2 \cdot \left(1 - \frac{\sum_{i=1}^{n-1} b_i^2}{c_\rho(n-2)}\right) - \frac{\sum_{i=1}^{n-1} b_i^4}{c_\rho(n-2)}\right| && \text{(B.147)} \\
&\le \sum_{i=1}^{n-1} b_i^2 \cdot \left|\left(1 - \frac{\sum_{i=1}^{n-1} b_i^2}{c_\rho(n-2)}\right)\right| + \left|\frac{\sum_{i=1}^{n-1} b_i^4}{c_\rho(n-2)}\right| && \text{(B.148)} \\
&= O(n)\cdot O(\delta/c_\rho) + O(1) && \text{(B.149)} \\
&= O(\delta n/c_\rho). && \text{(B.150)}
\end{aligned}$$

Q.E.D.

Advantage analysis of `M-TD`.

Now we analyze the advantage in revenue the miner can get after conducting all the transaction deletions. Intuitively, we first show that for large n and $\delta = \omega(\Delta/n)$, the condition $\left|\frac{\sum_{i=1}^{n} b_i^2}{n} - c_\rho\right| < \delta$ holds with high probability at each step in the $\Delta = o(n)$ deletions. Then we use Lemma B.5 to bound the advantage.

First, we deduce the following concentration lemma.

Lemma B.6. *For any* i.i.d. *random variable* $\{b_i\}$ *in* $[0,1]$ *satisfying* $\mathbb{E}[b_i^2] = c_\rho$ *and given* $\delta > 0$*, we have*

$$\Pr\left[\left|\frac{\sum_{i=1}^{n} b_i^2}{n} - c_\rho\right| \ge \frac{\delta}{2}\right] \le 2\exp\left(-\frac{\delta^2 n}{2}\right). \tag{B.151}$$

Proof. Proof. Hoeffding's inequality [199] states that when $\{x_i\}$ are independent random variables with $l_i \le x_i \le r_i$, and denoting $s_n = \sum_{i=1}^{n} x_i$, it holds that

$$\Pr[|s_n - \mathbb{E}[s_n]| \geq t] \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(r_i - l_i)^2}\right). \tag{B.152}$$

Let $x_i = b_i^2, l_i = 0, r_i = 1, t = \frac{\delta n}{2}$, then $\mathbb{E}[s_n] = c_\rho n$ and we get:

$$\Pr\left[\left|\sum_{i=1}^{n} b_i^2 - c_\rho n\right| \geq \frac{\delta n}{2}\right] \leq 2\exp\left(-\frac{\frac{1}{2}\delta^2 n^2}{n}\right), \tag{B.153}$$

i.e.,

$$\Pr\left[\left|\frac{\sum_{i=1}^{n} b_i^2}{n} - c_\rho\right| \geq \frac{\delta}{2}\right] \leq 2\exp\left(-\frac{\delta^2 n}{2}\right). \tag{B.154}$$

Q.E.D.

Then we upper bound the impact of transaction deletion on the average of $\{b_i^2\}$. Without loss of generality, we assume the miner deletes $b_n, b_{n-1}, \cdots, b_{n-t+1}$ sequentially[4] for $t \leq \Delta = o(n)$, and we want that $\left|\frac{\sum_{i=1}^{n} b_i^2}{n} - \frac{\sum_{i=1}^{n-t+1} b_i^2}{n-t+1}\right| \leq \frac{\delta}{2}$.

In fact, we have $\mathbf{b}_i \in [0, 1]$, so

$$\frac{\sum_{i=1}^{n} b_i^2}{n-t+1} \leq \frac{\sum_{i=1}^{n-t+1} b_i^2}{n-t+1} \leq \frac{(\sum_{i=1}^{n} b_i^2) - t}{n-t+1} \tag{B.155}$$

Therefore, for $t \leq \Delta$, There exists constant C_{MIC1} s.t. for $n \geq C_{MIC1}\frac{\Delta}{\delta}$ and $n-t+1 \geq 3$, we indeed have

$$\left|\frac{\sum_{i=1}^{n} b_i^2}{n} - \frac{\sum_{i=1}^{n-t+1} b_i^2}{n-t+1}\right| \leq \frac{\delta}{2}. \tag{B.156}$$

Combined with Lemma B.5, we deduce that when $n \geq C_{MIC1}\frac{\Delta}{\delta}$, with probability at least

[4]Notice that the argument holds for any subset and order of deletion, via re-permutations of $\{b_i\}$.

$1-2\exp\left(\delta^2 n/2\right)$, the advantage of M-TD with t deletions is at most $O(\delta)\cdot\frac{Hkt}{c_\rho n}$. We also see that when we require $\delta\in(0,1]$, then because $\Delta\geq 1$, $n-t+1\geq 3$ is guaranteed. Formally:

Theorem B.3 (Our mechanism is almost-{M-TD}-proof)**.** *Denote $B_\Delta^-(\mathbf{b})$ as the family of all bidding vectors generated via deleting at most Δ bids from $\mathbf{b}$. Then for universal constant $C_{MIC1}>0$ and $\delta\in(0,1], n\geq C_{MIC1}\frac{\Delta}{\delta}$, we have*

$$\begin{aligned}&\Pr_{\mathbf{b}}\left[\sup_{\mathbf{b}'\in B_\Delta^-(\mathbf{b})}\left(\tilde{r}(\mathbf{b}')-\tilde{r}(\mathbf{b})\right)>O(\delta)\frac{Hk\Delta}{c_\rho n}\right]\\&\leq 2\exp\left(-\frac{\delta^2 n}{2}\right).\end{aligned}\tag{B.157}$$

Advantage analysis of M-FT.

Finally we analyze the miner advantage of the miner's injection of fake transactions. The advantage a miner can get consists of two parts: increase of the miner revenue $\tilde{r}(\cdot)$, and the utility of fake identities. We notice that the robustness analysis of $\tilde{r}(\cdot)$ not only holds for transaction deletion, but also injection. So we can upper bound the miner advantage in the immediate revenue via very similar arguments. Formally, we have (proof omitted):

Corollary B.1. *Denote $B_\Delta(\mathbf{b})$ as the family of all bidding vectors generated via injecting and deleting a total of at most Δ bids to/from $\mathbf{b}$. Then for universal constants $C_{M0}, C_{M0'}, C_{MIC2}, C_{MIC3}>0$ and $\delta\in(0,1], n\geq C_{MIC2}\frac{\Delta}{\delta}$,*

$$\begin{aligned}&\Pr_{\mathbf{b}}\left[\sup_{\mathbf{b}'\in B_\Delta(\mathbf{b})}\left(\tilde{r}(\mathbf{b}')-\tilde{r}(\mathbf{b})\right)>C_{M0}\delta\frac{Hk\Delta}{c_\rho n}\right]\\&\leq C_{M0'}\exp\left(-C_{MIC3}\delta^2 n\right).\end{aligned}\tag{B.158}$$

Hence, we only need to further upper bound the advantage from the utility of fake identities. We notice that the fake transactions do not have intrinsic values, so the valuations of fake transactions are zero.

Now we consider the total utility of fake identities. Because the valuations are zero, their total utility are just the opposite of their payment. So for $b'_j\in\mathbf{b}'\backslash\mathbf{b}$, the utility of identity

j' is

$$\begin{aligned}
&\tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0) \\
&= -a_j(b'_j, \mathbf{b}'_{-j})\tilde{p}_j(b'_j, \mathbf{b}'_{-j}) && \text{(B.159)} \\
&= -a_j(b'_j, \mathbf{b}'_{-j})p_j(b'_j, \mathbf{b}'_{-j}) - \theta_j(b'_j, \mathbf{b}'_{-j}). && \text{(B.160)}
\end{aligned}$$

From Eq. (B.122)[5], and denote that the number of bids in $\mathbf{b}'$ is $n' \in [n - \Delta, n + \Delta]$, we get:

$$a_j(b'_j, \mathbf{b}'_{-j})p_j(b'_j, \mathbf{b}'_{-j}) = \Theta\left(\frac{k{b'_j}^2}{n}\right). \tag{B.161}$$

From $h = \frac{Hk}{n'}$ we get:

$$\theta_j(b'_j, \mathbf{b}'_{-j}) = -\frac{Hk}{2n}{b'_j}^2\left(\frac{\sum_{i\neq j} {b'_i}^2}{c_\rho(n'-1)} - 1\right) \tag{B.162}$$

Similar to the argument in Appendix B.4.5, as long as $c_\rho = \Theta(1)$ and $\left|\frac{\sum_{i=1}^n b_i^2}{n} - c_\rho\right| < O(\delta)$, we have $\left|\frac{\sum_{i\neq j} {b'_i}^2}{c_\rho(n'-1)} - 1\right| \leq O(\delta/c_\rho)$ for any $(\mathbf{b}' \in B_\Delta(b), j \in \mathbf{b}'\backslash\mathbf{b})$, which happens with probability at least $1 - \exp(-\Theta(\delta^2 n))$.

In this case, we have:

$$\left|\theta_j(b'_j, \mathbf{b}'_{-j})\right| \leq O(\delta) \cdot \frac{Hk}{c_\rho n}. \tag{B.163}$$

[5]let $\lambda_0 = 1.582$ and compute $f(\lambda_0), m$ accordingly.

Therefore, with probability at least $1-\exp(-\Theta(\delta^2 n))$, for any $\mathbf{b}' \in B_\Delta(b)$

$$\begin{aligned}
&\sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} \tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0) \\
&= \sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} -a_j(b'_j, \mathbf{b}'_{-j}) p_j(b'_j, \mathbf{b}'_{-j}) - \theta_j(b'_j, \mathbf{b}'_{-j}) && \text{(B.164)} \\
&= \sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} \left(-\Theta\left(\frac{k {b'_j}^2}{n} \right) + O(\delta) \cdot \frac{Hk}{c_\rho n} \right) && \text{(B.165)} \\
&\leq \sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} O(\delta) \cdot \frac{Hk}{c_\rho n} && \text{(B.166)} \\
&\leq O(\delta) \cdot \frac{Hk\Delta}{c_\rho n}. && \text{(B.167)}
\end{aligned}$$

Combined with Corollary B.1, we deduce that for universal constants $C_{M0}, C_{MIC2}, C_{MIC3} > 0$, $C_{M0'} > 1$ and $\delta \in (0, 1], n \geq C_{MIC2} \frac{\Delta}{\delta}$,

$$\begin{aligned}
\Pr_{\mathbf{b}} \Bigg[\sup_{\mathbf{b}' \in B_\Delta(\mathbf{b})} \left(\tilde{r}(\mathbf{b}') - \tilde{r}(\mathbf{b}) + \sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} \tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0) \right) \\
> C_{M0} \delta \cdot \frac{Hk\Delta}{c_\rho n} \Bigg] < C_{M0'} \exp(-C_{MIC3} \delta^2 n). && \text{(B.168)}
\end{aligned}$$

Particularly, we can let $\delta = (\Delta/n)^{1/3}$, then for $n \geq C_{MIC2}^{3/2} \Delta$,

$$\begin{aligned}
\Pr_{\mathbf{b}} \Bigg[\sup_{\mathbf{b}' \in B_\Delta(\mathbf{b})} \left(\tilde{r}(\mathbf{b}') - \tilde{r}(\mathbf{b}) + \sum_{b'_j \in \mathbf{b}' \setminus \mathbf{b}} \tilde{u}_j(b'_j, \mathbf{b}'_{-j}; 0) \right) \\
> C_{M0} \frac{Hk\Delta^{4/3}}{c_\rho n^{4/3}} \Bigg] < C_{M0'} \exp(-C_{MIC3} \Delta^{2/3} n^{1/3}). && \text{(B.169)}
\end{aligned}$$

Therefore, because $\Delta \geq 1$, for any $\epsilon > 0$ when $n \geq \max\{C_{MIC2}^{3/2} \Delta, C_{MIC3}^{-3} \log^3 \frac{C_{M0'}}{\epsilon}\}$, we

have

$$\Pr_{\mathbf{b}}\left[\sup_{\mathbf{b}'\in B_\Delta(\mathbf{b})}\left(\tilde{r}(\mathbf{b}')-\tilde{r}(\mathbf{b})+\sum_{b'_j\in\mathbf{b}'\setminus\mathbf{b}}\tilde{u}_j(b'_j,\mathbf{b}'_{-j};0)\right)\right.$$
$$\left.>C_{M0}\frac{Hk\Delta^{4/3}}{c_\rho n^{4/3}}\right]<\epsilon. \tag{B.170}$$

For $\epsilon\in(0,1/2)$, we have

$$\begin{aligned}\log\frac{C_{M0'}}{\epsilon}&=\log C_{M0'}+\log\frac{1}{\epsilon}\\&=\log\frac{1}{\epsilon}\left(1+\frac{\log C_{M0'}}{\log\frac{1}{\epsilon}}\right)\\&<\log\frac{1}{\epsilon}\left(1+\frac{\log C_{M0'}}{\log 2}\right).\end{aligned}$$

Just let $C_{M1}=C_{MIC2}^{3/2}$, $C_{M2}=C_{MIC3}^{-3}\left(1+\frac{\log C_{M0'}}{\log 2}\right)^3$, and from $H=Lc_\rho$, we have proven Theorem 4.5.

B.4.6 Proof of Theorem 4.6

For convenience we let $t=|\mathbf{b}|-1$. Denote $M(\mathbf{a},\mathbf{p},r)$ and $T(\theta,\tilde{r})$ is the auxiliary-variation decomposition of an 1-SCP mechanism $\tilde{M}$, then from Lemma 4.1 and Lemma B.3 we know that

$$\theta_i(b_i,\mathbf{b}_{-i})-\theta_i(0,\mathbf{b}_{-i})=\tilde{r}(b_i,\mathbf{b}_{-i})-\tilde{r}(0,\mathbf{b}_{-i}). \tag{B.171}$$

User i's utility in $\tilde{M}$ is

$$\begin{aligned}&\tilde{u}(b_i,\mathbf{b}_{-i};v_i)\\&=a_i(b_i,\mathbf{b}_{-i})(v_i-p_i(b_i,\mathbf{b}_{-i}))-\theta_i(b_i,\mathbf{b}_{-i}) &\text{(B.172)}\\&=u(b_i,\mathbf{b}_{-i};v_i)-\theta_i(b_i,\mathbf{b}_{-i}). &\text{(B.173)}\end{aligned}$$

From U-BNIC of $\tilde{M}$ we know that $\mathbb{E}_{\mathbf{b}_{-i}}[\tilde{u}(b_i+\delta, \mathbf{b}_{-i}; b_i)] \leq \mathbb{E}_{\mathbf{b}_{-i}}[\tilde{u}(b_i, \mathbf{b}_{-i}; b_i)]$ and $\mathbb{E}_{\mathbf{b}_{-i}}[\tilde{u}(b_i, \mathbf{b}_{-i}; b_i+\delta)] \leq \mathbb{E}_{\mathbf{b}_{-i}}[\tilde{u}(b_i+\delta, \mathbf{b}_{-i}; b_i+\delta)]$, i.e.,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{b}_{-i}}[u(b_i, \mathbf{b}_{-i}; b_i) - \theta_i(b_i, \mathbf{b}_{-i})] \\
&\geq \mathbb{E}_{\mathbf{b}_{-i}}[u(b_i+\delta, \mathbf{b}_{-i}; b_i) - \theta_i(b_i+\delta, \mathbf{b}_{-i})] && \text{(B.174)} \\
&\mathbb{E}_{\mathbf{b}_{-i}}[u(b_i, \mathbf{b}_{-i}; b_i+\delta) - \theta_i(b_i, \mathbf{b}_{-i})] \\
&\leq \mathbb{E}_{\mathbf{b}_{-i}}[u(b_i+\delta, \mathbf{b}_{-i}; b_i+\delta) - \theta_i(b_i+\delta, \mathbf{b}_{-i})]. && \text{(B.175)}
\end{aligned}$$

From U-BNIC (implied by U-DSIC) of M we get:

$$\begin{aligned}
\mathbb{E}_{\mathbf{b}_{-i}}[u(b_i, \mathbf{b}_{-i}; b_i)] &\geq \mathbb{E}_{\mathbf{b}_{-i}}[u(b_i+\delta, \mathbf{b}_{-i}; b_i)] && \text{(B.176)} \\
\mathbb{E}_{\mathbf{b}_{-i}}[u(b_i, \mathbf{b}_{-i}; b_i+\delta)] &\leq \mathbb{E}_{\mathbf{b}_{-i}}[u(b_i+\delta, \mathbf{b}_{-i}; b_i+\delta)]. && \text{(B.177)}
\end{aligned}$$

By integration on b_i for fixed $\mathbf{b}_{-i}$, we know that

$$\mathbb{E}_{\mathbf{b}_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] - \mathbb{E}_{\mathbf{b}_{-i}}[\theta_i(0, \mathbf{b}_{-i})] = 0. \quad \text{(B.178)}$$

From NFL we know that $\theta_i(0, \mathbf{b}_{-i}) = 0$, so

$$\mathbb{E}_{\mathbf{b}_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0. \quad \text{(B.179)}$$

Combined with Eq. (B.171), we know that

$$\begin{aligned}
&\mathbb{E}_{\mathbf{b}_{-i}}[\tilde{r}(b_i, \mathbf{b}_{-i}) - \tilde{r}(0, \mathbf{b}_{-i})] && \text{(B.180)} \\
&= \mathbb{E}_{\mathbf{b}_{-i}}[\theta_i(b_i, \mathbf{b}_{-i}) - \theta_i(0, \mathbf{b}_{-i})] && \text{(B.181)} \\
&= \mathbb{E}_{\mathbf{b}_{-i}}[\theta_i(b_i, \mathbf{b}_{-i})] = 0. && \text{(B.182)}
\end{aligned}$$

From assumption we know that $\mathbb{E}_{\mathbf{b}_{-i}}[r(0, \mathbf{b}_{-i})] \leq \mathbb{E}_{\mathbf{b}_{-i}}[r(\mathbf{b}_{-i})]$, so we have

$$\mathbb{E}_{\mathbf{b}}[\tilde{r}(b_i, \mathbf{b}_{-i})] = \mathbb{E}_{b_i}[\mathbb{E}_{\mathbf{b}_{-i}}[\tilde{r}(b_i, \mathbf{b}_{-i})]] \tag{B.183}$$

$$= \mathbb{E}_{b_i}[\mathbb{E}_{\mathbf{b}_{-i}}[\tilde{r}(0, \mathbf{b}_{-i})]] \leq \mathbb{E}_{b_i}[\mathbb{E}_{\mathbf{b}_{-i}}[\tilde{r}(\mathbf{b}_{-i})]] \tag{B.184}$$

$$= \mathbb{E}_{\mathbf{b}_{-i}}[\tilde{r}(\mathbf{b}_{-i})]. \tag{B.185}$$

Hence the expected revenue for $t+1$ users is at most the expected revenue for t users. We notice that when there is zero user the expected revenue is non-positive, so by induction, the expected revenue for arbitrary n users is non-positive.

B.4.7 Proof of Theorem 4.7

Necessity. Let $\forall b_i = 1$, then it has already been shown that $\tilde{r}(\mathbf{b}) = \Theta(k)\left(1 - \frac{1}{2c_\rho}\right)$. If $c_\rho < \frac{1}{2}$, then in this case $\tilde{r}(\mathbf{b}) < 0$, violating MIR.

Sufficiency. Because $\forall b_i \in [0, 1]$, we have $b_i^2 \geq b_i^4$. Therefore,

$$\tilde{r}(\mathbf{b}) = \Theta\left(\frac{k}{n}\right) \cdot \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho(n-1)}\right) \tag{B.186}$$

$$\geq \Theta\left(\frac{k}{n}\right) \cdot \left(\sum_{i=1}^{n} b_i^4 - \frac{\sum_{1 \leq i < j \leq n} b_i^2 b_j^2}{c_\rho(n-1)}\right) \tag{B.187}$$

$$= \Theta\left(\frac{k}{n}\right) \cdot \left(\left(1 - \frac{1}{2c_\rho}\right) \sum_{i=1}^{n} b_i^4 + \frac{1}{4c_\rho(n-1)} \sum_{i=1}^{n} \left(b_i^2 - b_j^2\right)^2\right). \tag{B.188}$$

If $c_\rho \geq \frac{1}{2}$, then $1 - \frac{1}{2c_\rho} \geq 0$, so $\tilde{r}(\mathbf{b})$ is lower bounded by a sum of squares. Therefore, $\tilde{r}(\mathbf{b})$ is non-negative for any $\mathbf{b} \in [0, 1]^n$, proving the MIR property of the mechanism.

B.4.8 Proof of Theorem 4.8

We have that

$$\tilde{r}(\mathbf{b}) = \frac{h}{2} \cdot \left(\sum_{i=1}^{n} b_i^2 - \frac{\sum_{1 \le i < j \le n} b_i^2 b_j^2}{c_\rho (n-1)} \right) \tag{B.189}$$

$$= \frac{h}{2} \cdot \left(\sum_{i=1}^{n} b_i^2 - \frac{1}{2c_\rho (n-1)} \left(\left(\sum_{i=1}^{n} b_i^2 \right)^2 - \sum_{i=1}^{n} b_i^4 \right) \right). \tag{B.190}$$

By the Cauchy–Schwarz inequality, we have $(\sum_{i=1}^{n} b_i^4) \cdot (\sum_{i=1}^{n} 1) \ge (\sum_{i=1}^{n} b_i^2)^2$, i.e.,

$$\sum_{i=1}^{n} b_i^4 \ge \frac{1}{n} \left(\sum_{i=1}^{n} b_i^2 \right)^2. \tag{B.191}$$

Therefore,

$$\tilde{r}(\mathbf{b}) \ge \frac{h}{2} \left(\sum_{i=1}^{n} b_i^2 - \frac{1}{2c_\rho n} \left(\sum_{i=1}^{n} b_i^2 \right)^2 \right) \tag{B.192}$$

$$= \frac{h}{2} \sum_{i=1}^{n} b_i^2 \left(1 - \frac{1}{2c_\rho n} \sum_{i=1}^{n} b_i^2 \right) \tag{B.193}$$

$$= \frac{hc_\rho n}{4} \left(1 - \frac{1}{c_\rho^2 n^2} \left(\sum_{i=1}^{n} b_i^2 - c_\rho n \right)^2 \right). \tag{B.194}$$

We know that $\mathbb{E}[\tilde{r}(\mathbf{b})] = \frac{hc_\rho n}{4}$, so

$$\frac{\tilde{r}(\mathbf{b})}{\mathbb{E}[\tilde{r}(\mathbf{b})]} \ge 1 - \frac{1}{c_\rho^2 n^2} \left(\sum_{i=1}^{n} b_i^2 - c_\rho n \right)^2. \tag{B.195}$$

Hoeffding's inequality [199] states that when $\{x_i\}$ are independent random variables with $l_i \le x_i \le r_i$, and denoting $s_n = \sum_{i=1}^{n} x_i$, it holds that

$$\Pr[|s_n - \mathbb{E}[s_n]| \ge t] \le 2 \exp\left(-\frac{2t^2}{\sum_{i=1}^{n} (r_i - l_i)^2} \right). \tag{B.196}$$

We let $x_i = b_i^2, l_i = 0, r_i = 1, t = \sqrt{\frac{\lambda n}{2}}$, and get:

$$\Pr\left[\left|\sum_{i=1}^{n} b_i^2 - c_\rho n\right| \geq \sqrt{\frac{\lambda n}{2}}\right] \leq 2\exp(-\lambda). \tag{B.197}$$

Combined with Eq. (B.195), we get:

$$\Pr\left[\frac{\tilde{r}(\mathbf{b})}{\mathbb{E}[\tilde{r}(\mathbf{b})]} \leq 1 - \frac{\lambda}{c_\rho^2 n}\right] \leq 2\exp(-\lambda). \tag{B.198}$$

APPENDIX C

APPENDIX FOR CHAPTER 5

C.1 Computation of Prover's Sunk Cost $\mu(\rho)$ on Losing Competition

Define $P_{-}(t)$ as the probability that another prover would have finished the computation by the time the fixed prover computes a t portion of the task. Then by definition, we have

$$P_{-}(t) = 1 - P(t). \tag{C.1}$$

Denote X as the random variable of the portion the fixed prover has done to the task when another prover would submit the work, then $P_{-}(\cdot)$ is essentially the CDF of X, and the PDF of X is $P'_{-}(\cdot)$.

Given that the fixed prover would stop computing when some other prover submits the task, we get that

$$\frac{\mu(\rho)}{M} = \mathbb{E}[X|X<\rho] \tag{C.2}$$
$$= \frac{\mathbb{E}[X\cdot\mathbf{1}_{[X<\rho]}]}{\Pr[X<\rho]} \tag{C.3}$$
$$= \frac{\int_0^\rho tP'(t)dt}{1-P(\rho)} \tag{C.4}$$
$$= \frac{\int_0^\rho \int_0^t P'_-(t)dxdt}{1-P(\rho)} \tag{C.5}$$
$$= \frac{\int_0^\rho \int_t^\rho P'_-(t)dtdx}{1-P(\rho)} \tag{C.6}$$
$$= \frac{\int_0^\rho (P_-(\rho)-P_-(x))dx}{1-P(\rho)} \tag{C.7}$$
$$= \frac{\int_0^\rho (P(x)-P(\rho))dx}{1-P(\rho)} \tag{C.8}$$
$$= \frac{\int_0^\rho P(x)dx-\rho P(\rho)}{1-P(\rho)}. \tag{C.9}$$

Therefore,

$$\mu(\rho) = \frac{\int_0^\rho P(x)dx-\rho P(\rho)}{1-P(\rho)}M. \tag{C.10}$$

C.2 Discussion on Reward Design for Multiple Verifiers

In the prover's reward design in Section 5.7, we decide on the acceptance or rejection of the proof based on the majority vote of verifiers, and only pay partial rewards $\frac{v}{n}R$ to the prover, instead of the full reward R, if $v \in (\frac{n}{2}, n)$ verifiers accept the proof. In this section, we discuss the rationale of this rule.

C.2.1 Majority Vote or One-Vote-Veto?

Assuming that the verifiers are honest, we can see that when any verifier rejects the proof, its certain that the proof is dishonest. Hence, in the case of honest verifiers, the one-vote-veto

rule can optimize the decision-making of the mechanism.

However, in the case where the verifiers may be dishonest, the one-vote-veto rule could render the mechanism vulnerable, as even one all-reject verifier can manipulate the system to reject all proofs. Hence, it is more robust to make the system reject the proof only when more than one verifier rejects it.

While other rules, e.g., two-vote-veto may also work or even work better in certain scenarios, we leave the detailed discussions in future work and use the simplest majority vote for the decision-making.

C.2.2 Why Partial Rewards?

If the prover gets the full rewards whenever the proof is rejected, then the prover may benefit from "slight" cheats as the probability to be caught by a majority of verifiers is sub-linearly low. For example, if there are $n = 1000$ stages in which $\alpha = 50$ stages are verified, and the prover cheats for the 1 stage (disguised as a random flag), saving $\frac{1}{1000}$ computational power, then each verifier has an independent $\frac{1}{40}$ probability to detect the cheat. If there is only one verifier, the probability that the proof is rejected is $\frac{1}{40}$.

Then we consider the majority vote of 5 verifiers. The probability that the proof is rejected is:

$$\sum_{i=3}^{5} \binom{5}{i} \left(\frac{1}{40}\right)^{i} \left(\frac{39}{40}\right)^{5-i} \approx 0.00015 < \frac{1}{1000}.$$

Hence, the mechanism is no longer BIS. The rationale is that if the prover cheats a $\delta \to 0$ fraction of the proof, then each verifier has a $\Theta(\alpha\delta)$ probability to detect the cheat. Hence in a $(2z-1)$-player majority vote, the probability of rejection is $\Theta\left(\binom{2z-1}{z}(\alpha\delta)\right) = o(\delta)$, rendering the mechanism not BIS for the case that δ is small enough.

On the other hand, in the Proportional Rule, it can be regarded that each verifier's report independently contributes to a $\frac{1}{2z-1}$ fraction of the prover's reward, so that the prover's reward is the same as the case of only 1 verifier, hence it is BIS as long as the basic 1-verifier mechanism is BIS.

In the Strict-Proportional Rule, the prover's reward is always no greater than in the Proportional Rule, with the equality holding at $\delta = 0$. Hence, cheating provers get less rewards while honest provers get the same, so it is also BIS as long as the basic 1-verifier mechanism is BIS.

C.3 Experiments on Verifiers' Incentives

We consider the case of $\alpha = 50$ that the mechanism almost always makes the correct decision, as shown in Section 5.7, and we set the expected verification reward to be 2 times the verification cost of honest proofs. We can expect that there is an overwhelming probabilities that other players are honest. Hence, we assume that other 4 of the 5 verifiers are honest, and the proof is honest with probability $p_{proof} \in [0, 1]$ in increments of 0.2; dishonest provers conduct partial spoof attacks with honest ratio $\rho = 0.9$ (which is relatively hard to detect). Then, we run numerical simulations and plot the verifier's expected utility when she honestly verifies $\alpha' \in [0, 50]$ stages in Figure C.1.

From Figure C.1 we see that for $p_{proof} \geq 0.4$, the CTF protocol incentivizes the verifier to honestly verify all $\alpha = 50$ stages via the flag rewards. For low p_{proof} (which is unlikely to occur due to prover-side incentive-security), the verifier is incentivized to verify fewer stages. The intuitive explanation is that verification rewards for rejected proofs are irrelevant to flags, and verifying 20 to 30 stages is already enough to detect the cheats with high probability.

Ablation analysis. To empirically show the necessity of our CTF protocol, we also plot the verifiers' utilities in Figure C.2 when we use the basic mechanism (Algorithms 2-3) with verifiers' rewards given by simple majority vote. In the figure, we see that particularly for $p_{proof} = 1$, the verifier would be incentivized to lazily accept the proof even if all other verifiers are honest, demonstrating the phenomenon of the Verifier's Dilemma. Hence, we show the practical effectiveness and necessity of our CTF protocol for the incentive guarantees on the verifier's side.

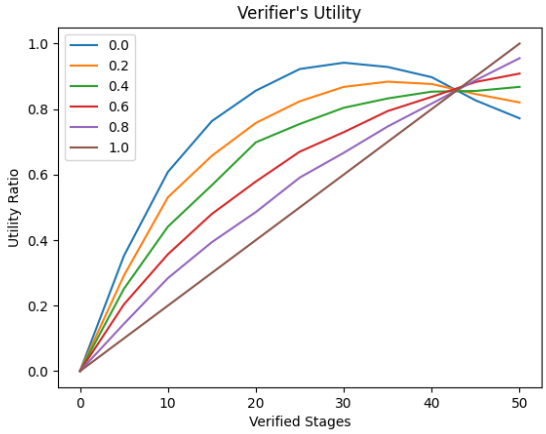

Figure C.1: Verifier's Utility

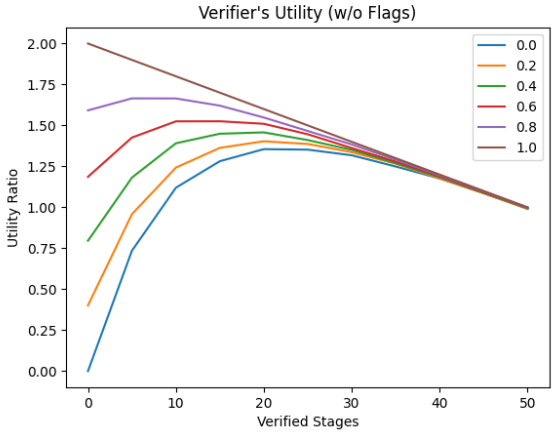

Figure C.2: Verifier's Utility without CTF Protocol

C.4 Discussions on Malicious Provers and Anomaly Detection

Throughout the paper, we mainly consider the scenario in which strategic provers are motivated solely by the block rewards for the training task, with their utility defined as the block reward minus computational costs. Nevertheless, in reality, there are indeed *malicious* trainers who may have incentives to adversarially sabotage the model for their own benefit [109]. While a detailed investigation of such cases is deferred to future work, we discuss here how our mechanism could be augmented for resilience against such *malicious* trainers.

C.4.1 Upper Bounds on Dishonest Stages

To circumvent the PoL Trilemma (as discussed in Section 5.1), our mechanism relaxes the requirement of Byzantine security to *incentive security*. In essence, we no longer demand that the mechanism be "absolutely secure" against *all* attacks. Instead, we only require it to be "secure enough" so that an attack is detected with sufficiently high probability to deter *rational* players from attacking. Consequently, for an attack that is "less severe" and yields small utility to the attacker, even a relatively small detection probability can suffice to ensure incentive security.

A potential concern with this model is the possible *underestimation* of the incentives to attack, as *malicious* players may have external motivations to benefit from training an incorrect model. In that case, an attacker might still find it worthwhile to mount an attack if the benefits from corrupting the model outweigh the lost block rewards, provided that a dishonest PoL can pass verification with non-negligible probability. Nonetheless, while our security notion is relaxed, it still essentially preserves Byzantine security in most practical settings: as long as the number of dishonest stages is not too small, our mechanism can detect the attack with overwhelming probability. Formally,

Proposition C.1. *In our full mechanism of Algorithms 4-5, if the prover cheats in more than* $\frac{2T}{\alpha}\ln\frac{1}{\epsilon}$ *stages, then the probability of passing verification (by one verifier) is at most* ϵ*.*

The proof of Proposition C.1 is deferred to Appendix C.5.6. From the proposition, we see that our mechanism effectively preserves Byzantine security against attacks involving more than $\Theta\left(\frac{T}{\alpha}\right)$ dishonest stages. Therefore, if compromising only a small number of stages cannot substantially degrade the trained model, then any model that passes verification in our PoL mechanism can be considered effectively correct.

In particular, if we set $\alpha = \Theta(T)$ (i.e., allowing a constant-ratio overhead in the mechanism), then an adversary can only corrupt a constant number of stages with a non-negligible probability of passing the verification.

C.4.2 Approaches for Anomaly Detection

From the above discussion, we demonstrate that our mechanism effectively limits the number of dishonest stages in a PoL that can pass verification. Consequently, if we can ensure that each dishonest stage is unable to significantly corrupt the output model, we can guarantee the correctness of the trained model even in the presence of (potentially irrational) malicious provers.

Most existing work addressing this issue falls in the scope of *anomaly detection*, whose primary aim is to detect significant errors at low cost [145]. In the context of PoL, we want to ensure that the weight updates from dishonest stages do not deviate excessively from the correct updates, so that the final model remains close to one trained honestly. Although more sophisticated approaches may exist, a simple strategy is to monitor the magnitudes of weight updates: under the smoothness conditions typical of many ML problems, gradients are not expected to grow arbitrarily large. Therefore, if verifiers observe unexpectedly large updates in certain stages, they would prioritize verifying those stages to detect potential attacks (similar to [96]).

Nevertheless, in our original PoL mechanism, the verifier does not receive model weights until they select which stages to verify and obtain the corresponding weights from the prover, thus saving communication costs. To address this limitation, the PoL certificate can be augmented with a *compressed* representation of the model weights that approximates the relevant distance information. According to the Johnson–Lindenstrauss lemma (Lemma C.1), this representation can be realized via a random low-dimensional projection. The projection direction is determined by the hash of the original PoL certificate, ensuring that it cannot be manipulated or known in advance before the training is completed.

Lemma C.1 (Johnson–Lindenstrauss)**.** *Let X be a set of n points in $\mathbb{R}^D$. Consider a random projection from $\mathbb{R}^D$ to $\mathbb{R}^d$ where $d = \Theta\big(\frac{\log n}{\epsilon^2}\big)$. With high probability, this projection preserves all pairwise Euclidean distances in X up to a multiplicative factor of $(1 \pm O(\epsilon))$.*

With this augmentation, we propose an approach to limit the effects of each dishonest stage to the output model, in order to ensure the model correctness in our PoL mechanism. We leave the detailed implementation and analysis for future work.

C.5 Omitted Proofs

C.5.1 Proof of Theorem 5.1

Assume we have such a mechanism. By the definition of Nash equilibrium, we consider a fixed verifier. Given that the prover and all other verifiers (if exist) act honestly, that verifier should be incentivized to do the honest verification.

Since the prover is honest, when that verifier performs honest verification, the result should always be "Success". However, if the verifier simply reports "Success" without verification, the outcome is the same but the verifier saves the verification cost, so that the verifier is incentivized to deviate from the honest strategy.

That leads to a contradiction. So no such mechanism exists.

C.5.2 Proof of Theorem 5.2

Notice that if the verifier verifies at least one stage, then she has a computational cost of $\frac{M}{T}$.

If $v_+ \leq v_0$, then the verifier does not have any incentive to find a cheat, so her strict optimal strategy is reporting "Success". Now we assume $v_+ > v_0$.

If the verifier verifies at least one stage, then as the probability that the proof is dishonest is at most ϵ, she catches a cheat with a probability upper bounded by ϵ. Therefore, her expected utility is at most $v_+\epsilon + v_0(1-\epsilon) - \frac{M}{T}$.

If the verifier just report "Success", her utility is v_0.

Since $\epsilon < \frac{M}{T(v_+ - v_0)}$, we have

$$v_0 > v_+\epsilon + v_0(1-\epsilon) - \frac{M}{T}. \tag{C.11}$$

Therefore the verifier's strict optimal strategy is to report "Success" without actual verification.

C.5.3 Proof of Theorem 5.3

We assume $\alpha \geq 2$. From Eq. 5.6 and $Q(1) = 1$ we see that

$$u(1) = P(1)R - \int_0^1 P(x)dx \cdot M. \tag{C.12}$$

So Eq. (5.12) implies that $u(1) > 0$, i.e. the mechanism is IR. From $P(x) \geq P(1)$ we also deduce that $M < R$, i.e. the reward must be greater than the honest computation cost.

Now we estimate $Q(\rho)$ for $\rho \in [0, 1)$.

Since each cheating stage has an independent κ probability to be caught when verified, we can equivalently model that the verification of each stage has an independent κ probability to be *effective*. In other words, a cheating stage is caught if and only if it is verified and the verification happens to be effective.

Then, we denote $\alpha^{\#}$ as a random variable of the total number of effectively verified stages. Hence we have:

$$\Pr[\alpha^{\#} = s] = \binom{\alpha}{s} \kappa^s (1-\kappa)^{\alpha - s}. \tag{C.13}$$

For the total of T stages, there are ρT stages trained honestly, and $Q(\rho)$ is the probability that all $\alpha^{\#}$ effectively verified stages are honest. Denote $Q_s(\rho)$ as the conditional probability that the proof passes the verification given $\alpha^{\#} = s$, then

$$\begin{aligned} Q_s(\rho) &= \frac{\binom{\rho T}{s}}{\binom{T}{s}} && \text{(C.14)} \\ &= \frac{\rho T(\rho T - 1) \cdots (\rho T - s + 1)}{T(T-1)\cdots(T-s+1)} && \text{(C.15)} \\ &\leq \frac{\rho T(\rho T - \rho) \cdots (\rho T - (s+1)\rho)}{T(T-1)\cdots(T-s+1)} && \text{(C.16)} \\ &= \rho^s. && \text{(C.17)} \end{aligned}$$

Therefore, we have

$$Q(\rho) = \sum_{s=0}^{\alpha} \Pr[\alpha^{\#} = s] Q_s(\rho) \tag{C.18}$$

$$\leq \sum_{s=0}^{\alpha} \binom{\alpha}{s} \kappa^s (1-\kappa)^{\alpha-s} \rho^s \tag{C.19}$$

$$= \sum_{s=0}^{\alpha} \binom{\alpha}{s} (\kappa\rho)^s (1-\kappa)^{\alpha-s} \tag{C.20}$$

$$= (1 - \kappa + \kappa\rho)^{\alpha}. \tag{C.21}$$

Let $\gamma = 0$, from Eq. (5.6) we see that

$$u(\rho) = P(\rho)(Q(\rho) - \gamma(1 - Q(\rho)))R - \int_0^{\rho} P(x)dx \cdot M \tag{C.22}$$

$$\leq (1 - \kappa + \kappa\rho)^{\alpha} P(\rho) R - \int_0^{\rho} P(x)dx \cdot M, \tag{C.23}$$

with equality holding at $\rho = 1$.

Now we define $\beta = \frac{M}{R} \in (0, 1)$and

$$\overline{u}(\rho) = (1 - \kappa + \kappa\rho)^{\alpha} P(\rho) - \beta \int_0^{\rho} P(x)dx. \tag{C.24}$$

Notice that $P(\cdot)$ is a non-increasing function, so for $x \in [0, \rho]$, $P(x) \geq P(\rho)$. Hence, we have

$$\overline{u}(\rho) = (1 - \kappa + \kappa\rho)^{\alpha} P(\rho) - \beta \int_0^{\rho} P(x)dx \tag{C.25}$$

$$\leq (1 - \kappa + \kappa\rho)^{\alpha} P(\rho) - \beta \int_0^{\rho} P(\rho)dx \tag{C.26}$$

$$= ((1 - \kappa + \kappa\rho)^{\alpha} - \beta\rho) P(\rho). \tag{C.27}$$

Since ρ is defined as the fraction of honest stages, which in practice must be multiples of

$\frac{1}{T}$, we only need to prove that if Eqs. (5.12)-(5.13) hold, then

$$\forall \rho \in \{0\} \cup [\frac{1}{T}, 1), \quad \overline{u}(\rho) < \overline{u}(1). \tag{C.28}$$

Now we prove (C.28) for $\rho = 0$, $\rho \in [\frac{1}{T}, \frac{1}{2}]$, and $\rho \in (\frac{1}{2}, 1)$, respectively.

(i) Case of $\rho = 0$.

Since $\rho = 0$, we have $\overline{u}(0) = (1-\kappa)^{\alpha} P(0) = (1-\kappa)^{\alpha}$. From (5.12) we see that $\overline{u}(0) < \overline{u}(1)$.

(ii) Case of $\rho \in [\frac{1}{T}, \frac{1}{2}]$.

From (C.27) we only need to prove $(1 - \kappa + \kappa\rho)^{\alpha} - \beta\rho \leq 0$ to deduce $\overline{u}(\rho) \leq 0 < \overline{u}(1)$.

Define

$$\psi(\rho) = (1 - \kappa + \kappa\rho)^{\alpha} - \beta\rho.$$

From $\alpha \geq 2$ we get $\psi''(\rho) = (1 - \kappa + \kappa\rho)^{\alpha-2} \geq 0$, so $\phi(\cdot)$ is concave and we only need to show $\psi(\frac{1}{T}) \leq 0$ and $\psi(\frac{1}{2}) \leq 0$.

Actually, for $\rho \in \left[\frac{1}{T}, \frac{1}{2}\right]$ we have

$$\begin{aligned}
\psi(\rho) &= (1 - \kappa + \kappa\rho)^{\alpha} - \beta\rho && \text{(C.29)} \\
&\leq \left(1 - \kappa + \frac{\kappa}{2}\right)^{\alpha} - \frac{\beta}{T} && \text{(C.30)} \\
&\leq e^{-\frac{\kappa}{2}\alpha} - \frac{\beta}{T} && \text{(C.31)} \\
&\leq e^{-\frac{\kappa}{2} \cdot \frac{2 \ln \frac{T}{\beta}}{\kappa}} - \frac{\beta}{T} && \text{(C.32)} \\
&\leq e^{-\ln \frac{T}{\beta}} - \frac{\beta}{T} && \text{(C.33)} \\
&= 0. && \text{(C.34)}
\end{aligned}$$

(iii) Case of $\rho \in (\frac{1}{2}, 1)$.

From Eq. (C.24) we get

$$\overline{u}'(\rho) = \alpha\kappa(1 - \kappa + \kappa\rho)^{\alpha-1} P(\rho) + (1 - \kappa + \kappa\rho)^{\alpha} P'(\rho) - \beta P(\rho). \tag{C.35}$$

From Eq. (5.11) we have $P'(\rho) \geq -\lambda P(\rho)$, hence

$$\overline{u}'(\rho) \geq \alpha\kappa(1-\kappa+\kappa\rho)^{\alpha-1}P(\rho) - \lambda(1-\kappa+\kappa\rho)^{\alpha}P(\rho) - \beta P(\rho) \tag{C.36}$$

$$= ((1-\kappa+\kappa\rho)^{\alpha-1}(\alpha\kappa - \lambda(1-\kappa+\kappa\rho)) - \beta)P(\rho). \tag{C.37}$$

Now we define $t = 1-\kappa+\kappa\rho$, then we have $\rho = \frac{t+(1-\kappa)}{\kappa}$ and

$$t \in (1-\frac{\kappa}{2}, 1). \tag{C.38}$$

We denote

$$\begin{aligned} V(t) &= (1-\kappa+\kappa\rho)^{\alpha-1}(\alpha\kappa - \lambda(1-\kappa+\kappa\rho)) - \beta \\ &= -\lambda t^{\alpha} + \alpha\kappa t^{\alpha-1} - \beta, \\ U(t) &= ((1-\kappa+\kappa\rho)^{\alpha} - \beta\rho) \\ &= t^{\alpha} - \frac{\beta}{\kappa}t + \frac{\beta(1-\kappa)}{\kappa}, \end{aligned}$$

then from (C.37) we see that

$$\overline{u}'(\rho) \geq V(t)P(\rho), \tag{C.39}$$

and from (C.27) we see that

$$\overline{u}(\rho) \leq U(t)P(\rho). \tag{C.40}$$

For $\rho \in [\frac{1}{2}, 1]$ we define

$$\overline{\overline{u}}(\rho) = \overline{u}(1) - \int_{\rho}^{1} V(1-\kappa+\kappa x)P(x)dx,$$

then

$$\overline{\overline{u}}'(\rho) = V(t)P(\rho) \leq \overline{u}'(\rho). \tag{C.41}$$

and from (C.39) we deduce

$$\begin{aligned}
\overline{\overline{u}}(\rho) &= \overline{u}(1) - \int_{\rho}^{1} \overline{\overline{u}}'(x)dx && \text{(C.42)}\\
&\geq \overline{u}(1) - \int_{\rho}^{1} \overline{u}'(x)dx && \text{(C.43)}\\
&= \overline{u}(\rho). && \text{(C.44)}
\end{aligned}$$

From (5.13) we have $\alpha \geq \frac{2(\lambda+\beta)}{\beta\kappa} \geq \frac{\lambda}{\kappa} + 1$, thus we get

$$\begin{aligned}
V'(t) &= \alpha t^{\alpha-2}((\alpha-1)\kappa - \lambda t) && \text{(C.45)}\\
&\geq \alpha t^{\alpha-2}(\lambda - \lambda t) && \text{(C.46)}\\
&\geq 0. && \text{(C.47)}
\end{aligned}$$

Hence $V(t)$ has at most one zero point on $(\frac{1}{2}, 1)$, and from (C.41), $\overline{\overline{u}}(\rho)$ has at most one stationary point on $(\frac{1}{2}, 1)$. Because $V(1) = -\lambda + \alpha\kappa - \beta \geq 0$, we deduce that $\overline{\overline{u}}(\rho)$ must satisfy one of the following:

- Monotonic increasing on $(\frac{1}{2}, 1)$, or
- Monotonic decreasing on $(\frac{1}{2}, \xi)$ and increasing on $(\xi, 1)$, in which $\xi \in (\frac{1}{2}, 1)$.

In the first case, it holds that $\overline{u}(\rho) \leq \overline{\overline{u}}(\rho) < \overline{\overline{u}}(1) = \overline{u}(1)$ for $\rho \in (\frac{1}{2}, 1)$ and we prove (C.28). Now we consider the second case.

Since $\overline{\overline{u}}(\rho)$ is increasing on $(\xi, 1)$, we see that $\forall \rho \in [\xi, 1)$, $\overline{u}(\rho) \leq \overline{\overline{u}}(\rho) < \overline{\overline{u}}(1) = \overline{u}(1)$. On the other hand, when $\rho \in (\frac{1}{2}, \xi)$, we prove that $\overline{u}(\rho) \leq 0$.

Actually, because $\overline{\overline{u}}(\cdot)$ is decreasing at $\rho \in (\frac{1}{2}, \xi)$, we deduce that $\overline{\overline{u}}'(\rho) \leq 0$, thus from (C.41) we have $V(t) \leq 0$.

Additionally, we have

$$tV(t) - \alpha\kappa U(t) \tag{C.48}$$
$$= (-\lambda t^{\alpha+1} + \alpha\kappa t^{\alpha} - \beta t) - (\alpha\kappa t^{\alpha} - \alpha\beta t + \alpha\beta(1-\kappa)) \tag{C.49}$$
$$= -\lambda t^{\alpha+1} - \beta t + \alpha\beta t - \alpha\beta(1-\kappa) \tag{C.50}$$
$$= \alpha\beta(t+\kappa-1) - (\lambda t^{\alpha+1} + \beta t). \tag{C.51}$$

From (5.13) we have $\alpha \geq \frac{2(\lambda+\beta)}{\beta\kappa}$, and from (C.38) we have $1 - \frac{\kappa}{2} < t < 1$. Therefore,

$$tV(t) - \alpha\kappa U(t) > \frac{2(\lambda+\beta)}{\beta\kappa}\beta\left(1 - \frac{\kappa}{2} + \kappa - 1\right) - (\lambda+\beta) \tag{C.52}$$
$$= \frac{2(\lambda+\beta)}{\kappa} \cdot \frac{\kappa}{2} - (\lambda+\beta) \tag{C.53}$$
$$= 0. \tag{C.54}$$

Combined with $V(t) \leq 0$, we deduce that $U(t) \leq 0$, and from (C.40) we get $\overline{u}(\rho) \leq 0 < \overline{u}(1)$.

Here we finish the proof for all three cases of (C.28). Now we have proven Theorem 5.3.

C.5.4 Proof of Theorem 5.4

It is straightforward to see that Eq. (5.14) holds if and only iff the mechanism is IR. Similar to the proof in Appendix C.5.3, we have

$$u(\rho) = P(\rho)(Q(\rho) - \gamma(1 - Q(\rho)))R - \int_0^{\rho} P(x)dx \cdot M \tag{C.55}$$
$$\leq ((1+\gamma)(1-\kappa+\kappa\rho)^{\alpha} - \gamma)P(\rho)R - \int_0^{\rho} P(x)dx \cdot M. \tag{C.56}$$

Hence, we can similarly define

$$\overline{u}(\rho) = ((1+\gamma)(1-\kappa+\kappa\rho)^{\alpha} - \gamma)P(\rho) - \beta\int_0^{\rho} P(x)dx, \tag{C.57}$$

and only need to prove that

$$\overline{u}(\rho) < \overline{u}(1), \quad \rho \in [0, 1).$$

For Eq. (C.57) we see that

$$(1 - \kappa + \kappa\rho)^\alpha \leq \frac{\gamma}{1+\gamma} \Rightarrow \overline{u}(\rho) \leq 0. \tag{C.58}$$

Now we consider two cases of $(1-\kappa)^\alpha < \frac{\gamma}{1+\gamma}$ and $(1-\kappa)^\alpha \geq \frac{\gamma}{1+\gamma}$ separately.

(i) Case of $(1-\kappa)^\alpha < \frac{\gamma}{1+\gamma}$.

In this case, we define $\rho_{th} = \frac{(\frac{\gamma}{1+\gamma})^{\frac{1}{\alpha}} + \kappa - 1}{\kappa}$, then for $\rho \in [0, 1]$, we have

$$\rho \leq \rho_{th} \iff (1 - \kappa + \kappa\rho)^\alpha \leq \frac{\gamma}{1+\gamma}. \tag{C.59}$$

From Eq.(C.58) and IR guarantee we have that $\overline{u}(\rho) \leq 0 < \overline{u}(1)$ when $\rho \in [0, \rho_{th}]$. Now we consider $\rho \in (\rho_{th}, 1)$.

From Eq.(C.57) we have

$$\begin{aligned}
\overline{u}'(\rho) &= \alpha\kappa(1+\gamma)(1-\kappa+\kappa\rho)^{\alpha-1}P(\rho) \\
&\quad + ((1+\gamma)(1-\kappa+\kappa\rho)^\alpha - \gamma)P'(\rho) - \beta P(\rho) && \text{(C.60)} \\
&\geq ((1+\gamma)(\alpha\kappa(1-\kappa+\kappa\rho)^{\alpha-1} \\
&\quad - \lambda(1-\kappa+\kappa\rho)^\alpha) + \lambda\gamma - \beta)P(\rho) && \text{(C.61)} \\
&= ((1+\gamma)(1-\kappa+\kappa\rho)^{\alpha-1}(\alpha\kappa - \lambda(1-\kappa+\kappa\rho)) + \lambda\gamma - \beta)P(\rho). && \text{(C.62)}
\end{aligned}$$

From (5.15) we have $\alpha\kappa \geq \lambda$, hence

$$\alpha\kappa - \lambda(1 - \kappa + \kappa\rho) \geq 0. \tag{C.63}$$

From Eq. (C.59) and $1 - \kappa + \kappa\rho \in [0, 1]$, we have

$$\rho > \rho_{th} \Rightarrow (1-\kappa+\kappa\rho)^{\alpha} \geq \frac{\gamma}{1+\gamma}$$
$$\Rightarrow (1-\kappa+\kappa\rho)^{\alpha-1} \geq \frac{\gamma}{1+\gamma}.$$

Therefore, for $\rho \in (\rho_{th}, 1)$, we have

$$\overline{u}'(\rho) \geq ((1+\gamma) \cdot \frac{\gamma}{1+\gamma} \cdot (\alpha\kappa - \lambda(1-\kappa+\kappa\rho)) + \lambda\gamma - \beta)P(\rho) \tag{C.64}$$
$$= (\gamma(\alpha\kappa - \lambda(1-\kappa+\kappa\rho)) + \lambda\gamma - \beta)P(\rho) \tag{C.65}$$
$$\geq (\gamma(\alpha\kappa - \lambda) + \lambda\gamma - \beta)P(\rho) \tag{C.66}$$
$$= (\alpha\gamma\kappa - \beta)P(\rho). \tag{C.67}$$

From 5.15 we have $\alpha > \frac{\beta}{\gamma\kappa}$, and as $\gamma, \kappa > 0$, we have $\alpha\gamma\kappa - \beta > 0$, hence $\overline{u}'(\rho) > 0$.

Therefore, $\overline{u}(\cdot)$ is monotonic increasing on $(\rho_{th}, 1)$, deducing that $\overline{u}(\rho) < \overline{u}(1)$ for $\rho \in (\rho_{th}, 1)$.

(ii) Case of $(1-\kappa)^{\alpha} \geq \frac{\gamma}{1+\gamma}$.

In this case, we have $(1-\kappa+\kappa\rho)^{\alpha} \geq \frac{\gamma}{1+\gamma}$ for $\rho \in [0, 1)$, so it holds that $\overline{u}(\cdot)$ is monotonic increasing on $[0, 1)$. Hence, we prove that $\overline{u}(\rho) < \overline{u}(1)$ for $\rho \in [0, 1)$.

C.5.5 Proof of Theorem 5.5

We first assume $\epsilon = 0$. Then, we only need to prove a fact: assuming the prover is honest, then as long as the verifier has verified less than α stages, she would increase her expected utility if she verifies one more stage.

Denote $\alpha' \leq \alpha - 1$ as the number of stages the verifier has verified, and she has found m flags, then $m \leq \alpha'$.

Then, among the $T - \alpha'$ remaining stages not verified yet, there are $\eta T - m \geq \eta T - \alpha'$ flags. Therefore, the probability that the verifier finds a flag in an additional stage is

$$p = \frac{\eta T - m}{T - \alpha'} > \frac{\eta T - \alpha}{T}. \tag{C.68}$$

Since $\eta \geq \frac{2\alpha}{T}$, we have $\alpha \leq \frac{\eta T}{2}$, so it holds that

$$p > \frac{\eta T/2}{T} = \frac{\eta}{2}. \tag{C.69}$$

If the verifier finds a flag, according to the CTF protocol, she re-trains the stage with two different seeds, taking a computational cost of $\frac{2M}{T}$ and gaining a reward of R_1. If she does not find a flag, she re-trains the stage with one seed, taking a computational cost of $\frac{M}{T}$ and getting no reward. Hence, the expected gain of the utility in verifying an additional stage is

$$\Delta u = p\left(R_1 - \frac{2M}{T}\right) - (1-p)\frac{M}{T} \tag{C.70}$$

$$= p\left(R_1 - \frac{M}{T}\right) - \frac{M}{T}. \tag{C.71}$$

From Eq. (5.17), we have

$$\Delta u \geq p\left(\frac{M}{T}\left(\frac{2}{\eta} + 1\right) - \frac{M}{T}\right) - \frac{M}{T} \tag{C.72}$$

$$= \frac{M}{T} \cdot \left(\frac{2}{\eta}p - 1\right) \tag{C.73}$$

$$> \frac{M}{T} \cdot \left(\frac{2}{\eta} \cdot \frac{\eta}{2} - 1\right) \tag{C.74}$$

$$= 0. \tag{C.75}$$

Hence, the verifier would always gain additional expected utility via verifying an additional stage as long as $\alpha' < \alpha$. On the other hand, the verifier only has access to α stages in $\mathbf{t}_{ve}$. Hence, given that the prover is honest, the verifier would maximize her expected utility when she honestly verifies all stages she requests.

Since the inequalities are strict, and the utilities are continuous functions of ϵ, it also holds

for any ϵ small enough. Therefore, the mechanism is VIS.

C.5.6 Proof of Proposition C.1

From Eq. (C.21) in Appendix C.5.3, denoting ρ as the fraction of honestly trained stages, the probability of passing the verification is

$$Q(\rho) \leq (1 - \kappa + \kappa\rho)^{\alpha}. \tag{C.76}$$

In our full mechanism we have $\kappa = \frac{1}{2}$, and denote Δ as the number of dishonest stages, then we have $\rho = 1 - \frac{\Delta}{T}$. Hence, we deduce that

$$Q(\rho) \leq \left(1 - \frac{\Delta}{2T}\right)^{\alpha} \tag{C.77}$$

$$\leq e^{-\frac{\alpha}{2T}\cdot\Delta}. \tag{C.78}$$

Since $\Delta \geq \frac{2T}{\alpha} \ln \frac{1}{\epsilon}$, we have

$$Q(\rho) \leq e^{-\frac{\alpha}{2T}\cdot\frac{2T}{\alpha}\ln\frac{1}{\epsilon}} \tag{C.79}$$

$$= e^{-\ln\frac{1}{\epsilon}} \tag{C.80}$$

$$= \epsilon. \tag{C.81}$$

APPENDIX D

APPENDIX FOR SECTION 6

D.1 Introduction of Decentralized AI Verification Protocols

Amid the rapid development of AI technologies in the LLM era, *decentralized AI (DeAI)* has emerged as a promising paradigm that aims to deploy AI infrastructure on decentralized platforms such as blockchains [149]. A key motivation behind DeAI is to ensure the *trustworthiness* of AI systems—specifically, to verify that training and inference processes are faithfully executed and free from adversarial tampering.

In addition to mitigating the risk of malicious attacks on AI models [109], DeAI also addresses growing concerns about *AI safety* [102]. While much of the existing AI safety literature focuses on *internal* risks—particularly issues of *alignment* [200, 201]—these approaches typically assume that the models are correctly trained and executed. However, due to the black-box nature of AI models, *external* risks arise—namely, that model developers may have incentives to manipulate the system for their own benefit, especially when model outputs influence high-stakes decisions. Thus, verifying and certifying the *integrity* of AI models —that they are properly trained and function as intended—is essential.

Centralized AI corporations may be incentivized to manipulate AI systems in the absence of transparency. In contrast, decentralized verification offers a trustless approach to ensure model integrity. Therefore, DeAI plays a crucial role in mitigating external risks by ensuring model integrity through decentralized technologies [202].

From a methodological perspective, existing approaches to decentralized verification of AI models can be broadly categorized into two types: *cryptographic* and *game-theoretic* methods. Cryptographic methods aim to provide strong, provable guarantees of training and inference integrity, typically through mechanisms such as zero-knowledge proofs (e.g., zkML, Chen

et al. [150]) to ensure verifiability without revealing sensitive information. However, these methods often incur substantial computational overhead (typically exceeding 1000x) which severely undermines system efficiency and poses a significant barrier to practical deployment, particularly for large-scale models.

Alternatively, game-theoretic approaches aim to leverage economic incentives to ensure that all participants (e.g., trainers and verifiers) act honestly as a strategic equilibrium behavior—namely, that truthful actions constitute a Nash equilibrium.

D.1.1 opML: Optimistic Machine Learning for Model Inference

A representative example is the mechanism of *opML* (Optimistic Machine Learning, Conway et al. [130]), which secures the correctness of AI model *inference* via the *Optimistic Rollup* framework [203]. In this context, the term "optimistic" refers to the principle that *all submitted computations are presumed valid unless proven otherwise.* Verifiers are thus incentivized to verify the outputs and are rewarded for successfully identifying incorrect computations.

Specifically, when a verifier verifies a submitted ML task, they re-execute the computation and compare the results:

- If the results match, the task is accepted as valid.
- If the results do not match, a *committee voting* procedure is invoked, wherein a designated committee determines the validity of the task through majority vote.

Economic issues. Due to the optimistic assumption of correctness, it is essential that verifiers in opML are sufficiently incentivized to invest the necessary computational resources for verification. Hence, the mechanism should reward the efforts verifiers make to offset the computational cost.

Assuming that the opML mechanism works as expected, we can expect that most provers act honestly and an overwhelming majority of submitted computations are valid. Then, we may expect a *lazy* verifier to accept every proof without actual verification, unless the reward for detecting an invalid proof makes a difference.

To simplify the discussion, we assume that the committee voting can always correctly determine if the proof is valid. From the perspective of the verifier,

- If she acts honestly, she accepts a valid proof with $(1-\epsilon)$ probability and detects an invalid proof with ϵ probability.
- If she acts lazily, she accepts a valid proof with $(1-\epsilon)$ probability and accepts an invalid proof with ϵ probability.

We see that the outcomes only differs in the scenario that the prover cheats, which comes with a small probability of ϵ. Hence, if the verification cost is C, the reward R of detecting and penalty L for failing to detect must sum up to $R + L \geq \frac{C}{\epsilon}$ to incentivize honest verification. Actually, Conway et al. [130] show that for given $\{R, L, C\}$, the protocol would suffer an $\epsilon = \frac{C}{R+L}$ rate of invalid computation at Nash equilibrium, which resembles the Verifier's Dilemma and undermines the trustworthiness of the ecosystem particularly when the verification cost C is substantial.

Attention challenges. To address the Verifier's Dilemma, Conway et al. [130] propose a mechanism known as *attention challenges*, which operates as follows. Suppose the prover has address A_s and the output is $f(x)$:

- The prover first reveals the hash value $H(f(x), A_s)$ and issues a challenge to all verifiers v such that $H(f(x), A_v) < T$, where A_v is the address of verifier v and T is a predefined threshold.
- After a fixed time window, the prover reveals the full output $f(x)$ and computes $H(f(x), A_v)$ for each verifier. Any verifier for whom $H(f(x), A_v) < T$ but who did not respond is marked as non-participating.
- If the submitted output $f(x)$ is ultimately deemed valid, all such non-participating verifiers are penalized, and a portion of their penalties is awarded to the prover.

Nevertheless, this design is based on the assumption that computing (and verifying) $H(f(x), A_v)$ is computationally negligible. While this assumption holds for model *inference*

tasks—where $f(x)$ is a simple prediction vector or classification result, it fails for training tasks, where $f(x)$ represents an entire trained model and can be prohibitively large. In this case, the Verifier's Dilemma occurs to the prover in turn.

D.1.2 Proof-of-Learning (PoL): Lightweight Verification For Model Training

Whereas the designs of zkML and opML mainly apply to ML model inference, additional challenges occur in the development of verification protocols for ML training. Besides the fact that the training process is substantially more computationally intensive than inference, the output of the training task—the trained models—also have large sizes and even simple operations on them (e.g., hashing or comparison) take non-negligible computational costs, so that it would be harder to obtain "cheaply-shared ground truths" (like the $H(f(x), A_v)$ discussed above) to bypass the Verifier's Dilemma via cheap verification. Furthermore, the re-execution method in opML would incur an at least 1x computational overhead. For ML training tasks with heavy computational costs, we are still motivated to lower this computational overhead.

In light of this, Jia et al. [96] propose a "vanilla" Proof-of-Learning (PoL) mechanism in which the prover is supposed to train the model while leaving checkpoints during the training process, and the verifier chooses the "most suspicious" parts of the training process to verify via re-execution. Nevertheless, the vanilla PoL leaves a substantial gap to decentralized AI verification as it assumes the credibility of verifiers (which is unrealistic especially in the presence of the Verifier's Dilemma), and its criteria of "most suspicious" parts is also subject to adversarial attacks [100, 101], resembling the Goodhart's Law (*"When a measure becomes a target, it ceases to be a good measure"*, Goodhart [117]).

To adapt PoL for decentralized AI applications, Zhao et al. [4] introduce a refined mechanism called *incentive-secure PoL*, which replaces selective re-execution with *random sampling*. They demonstrate that, under mild assumptions, this protocol satisfies *incentive-security* for rational provers: dishonest behavior is detected with high probability unless the prover deviates during only a negligible fraction of training steps—insufficient to meaningfully affect

performance or yield economic gain. This approach retains the lightweight nature of PoL while aligning with the incentive constraints of decentralized verification.

In addition, Zhao et al. [4] propose a *capture-the-flag* mechanism to further strengthen verifier engagement. Here, flags—introduced as randomness by the prover—serve as verifiable tokens that honest verifiers are incentivized to detect and report, even when all proofs are valid. This incentivizes verifier efforts regardless of adversarial behavior.

The incentive-secure PoL protocol operates as follows:

- The prover runs a multi-stage stochastic training process (e.g., stochastic gradient descent), recording model weights and commit hashes after each stage. At a subset of stages, cryptographic *flags* are randomly inserted and committed using distinct random seeds.
- The verifiers randomly choose a small fraction of stages and request the prover to reveal the model weights before and after each selected stage.
- The prover responds by revealing the requested model weights.
- Verifiers check the correctness of these stages and privately commit to two reports: (1) whether the proof is accepted and (2) which flags, if any, they detect.
- The prover reveals the list of inserted flags.
- Verifiers reveal their reports and detected flags.
- Provers and verifiers receive rewards or penalties based on the consistency of reports and flag detections.

While Zhao et al. [4] do not conduct a formal game-theoretic analysis of the verifiers' scoring rules, our study fills this theoretical gap—especially in the context where invalid proofs can only be detected probabilistically. We provide a rigorous incentive analysis that characterizes equilibrium behavior under this capture-the-flag framework, offering a foundation for incentive-aligned training verification in decentralized AI.

D.2 Discussion on Strong-SCP Peer Prediction Mechanisms

In the scope of peer prediction mechanisms, it is assumed that players report in a way that maximizes their expected utilities w.r.t. their *beliefs* on other players' reports. Hence, different beliefs may lead to different estimations of utilities and different strategies. In the canonical setting of individual non-colluding players, their beliefs are the Bayesian posteriors conditioned on their observations, i.e. $P(\mathbf{X}_{-i}|X_i)$.

When the collusions occur, nevertheless, the beliefs may differ in different settings as the level of information and utility sharing may vary. In the notion of side-contract-proofness (SCP) in the transaction fee mechanism design [41], it is assumed that the utilities are transferable via side payments, so that rather than a local Pareto improvement that weakly benefits everyone, a successful collusion only needs to improve the total utility of all players in the colluding party.

While we inherit the SCP notion that allows side payments, in the context of information elicitation, there can also be different settings on whether the players' beliefs are shared or not. In the weak-SCP notion, we consider the non-sharing-belief setting that the players estimate their utilities based on their individual observations $P(X_i|\mathbf{X}_{-i})$, and we show in Theorem 6.7 that weak-SCP can be achieved via our design that optimizes (δ, K)-compactness. In contrast, a "strong-SCP" notion considers the sharing-belief setting with players estimating their utilities based on the collective observations of the colluding party. Namely, their beliefs on the reports inside the party are just their true reports, and their beliefs on the reports outside the party are computed by $P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})$.

Nevertheless, a series of difficulties occur in the design for strong-SCP mechanisms.

D.2.1 Challenges in the Design of Strong-SCP Mechanisms

Free-riding at low noises. Consider the scenario when the observation noise is low, i.e., the players observe the true type θ with high probability. Then in a colluding party, one observation is sufficient to secure a high confidence that other members would also observe that result, which is likely to be the ground-truth. Hence, it is rational for other players

to lazily report that result too, saving the observation cost (which is typically high in ML verification contexts).

Preference towards agreeing reports. Even if the observation cost is low enough to keep the players willing to do active observation, as a wide scope of practically used peer prediction mechanisms, e.g., the correlated agreement (CA) mechanism [162] and mutual-information-based mechanisms [139], typically rewards agreeing reports, there can be a tendency that all colluding players report the same even if they observe differently, violating strong-SCP properties. This challenge generalizes to the family of *pairwise-scoring* mechanisms, in which the colluding party's total utility is approximately linear to their average report when n is large.[1] Hence, it is impossible to design a pairwise-scoring peer prediction mechanism that satisfies a "strict" strong-SCP property with sensitivity guarantees (truthful reporting yields at least h more utility than reporting the same when disagreement occurs). Formally, we have:

Theorem D.1. *In an n-player decentralized verification game (DVG), for any pairwise-scoring mechanism with a pairwise scoring matrix T such that the reward of player i is given by*

$$R_i(Z_i, \mathbf{Z}_{-i}) = Z_i' T \overline{\mathbf{Z}_{-i}}, \tag{D.1}$$

and $-K \leq T \leq K$, it holds that:

***(i)** For a collusion party $\mathscr{C} = \{i_1, \cdots, i_c\}$ with observations $\mathbf{X}_{\mathscr{C}} = \{X_{i_1}, \cdots, X_{i_c}\}$, there exists an all-same report $\mathbf{Z}^*_{\mathscr{C}} = \{X_{i^*}, \cdots, X_{i^*}\}$ such that $i^* \in \mathscr{C}$ and*

$$\begin{aligned} &\mathbb{E}_{\mathbf{Z}_{\mathscr{C}} = \mathbf{Z}^*_{\mathscr{C}},\ \mathbf{Z}_{-\mathscr{C}} \sim P(\mathbf{X}_{-\mathscr{C}} | \mathbf{X}_{\mathscr{C}})} \left[\sum_{i \in \mathscr{C}} R_i(X_i^*, \mathbf{Z}_{-i}) \right] \\ &\geq \mathbb{E}_{\mathbf{Z}_{\mathscr{C}} = \mathbf{X}_{\mathscr{C}},\ \mathbf{Z}_{-\mathscr{C}} \sim P(\mathbf{X}_{-\mathscr{C}} | \mathbf{X}_{\mathscr{C}})} \left[\sum_{i \in \mathscr{C}} R_i(X_i, \mathbf{Z}_{-i}) \right] - h, \end{aligned} \tag{D.2}$$

in which

$$h = \frac{2c(c-1)}{n-1} K. \tag{D.3}$$

[1] The approximate linearity property is actually not restricted to pairwise-scoring mechanisms, so the results can potentially be further generalized in practice.

(ii) *If for any types $s_1, s_2 \in S$, $s_1 \neq s_2$, it holds that*

$$T_{s_1 s_1} \geq T_{s_1 s_2},$$

i.e., the scoring rule favors agreement, then Eq. (D.2) *holds for $h \leq 0$.*

Furthermore, if it is possible for the colluding party $\mathscr{C}$ to observe $\mathbf{X}_{\mathscr{C}}$ with two different observations X_i, X_j s.t. $T_{X_i X_i} > T_{X_i X_j}$, then Eq. (D.2) *holds for $h < 0$, indicating that the mechanism is not strong-SCP.*

The proof is deferred to Appendix D.6.10. We can interpret part (i) as that: when $c \ll n$, the approximate linearity implies the impossibility to disincentivize reporting the same type with significant incentive margins, as assuming the expected reward of an honest player is $\Theta(K)$, the expected reward of the party is $\Theta(cN) \gg h$. Furthermore, part (ii) shows that as long as the mechanism favors agreement (as most existing peer prediction mechanisms typically do), this type of collusion will be strictly profitable, rendering them non-strong-SCP. While we conjecture that it may be generally impossible to design a strong-SCP single-task peer prediction mechanism under mild assumptions, Theorem D.1 shows that if it is actually possible, we need to bypass the approximate linearity and may need extremely tricky designs. On the other hand, other possible approaches to address the collusion issue in the shared-belief setting may include:

- Multi-task settings: assigning different task sets to different players and make partial-copying strategies unprofitable.
- Aggregation design: while failing to prevent collusion in the front-end elicitation phase, it may still be possible to design aggregation mechanisms to minimize its impact to the back-end decision-making (e.g., whether to accept the proof/block/etc.)

Nevertheless, we can notice that for the phenomenon that colluding players tend to copy the same report, its practical effect on the functionality of back-end decision-making could be rather "benign" as at least one of them did the honest observation, which still bypasses the Verifier's Dilemma and can be informationally sufficient particularly in the low-noise scenario: in the back-end perspective, we can also regard such colluders as one player

with more voting power in an almost unanimous voting. We leave detailed analyses and discussions on these aspects for future work.

D.3 A Coupling Interpretation of Byzantine Reduction

D.3.1 Coupling Argument and Total Variation Distance

In probability theory and statistics, *coupling* is a technique to compare characteristics of two distributions, especially when they are close to each other. In general, for two given distributions D_1, D_2, we can construct two dependent random variables V_1, V_2 a with a joint distribution $P(V_1, V_2)$ such that the marginal distributions satisfy

$$P(V_1) = D_1, P(V_2) = D_2.$$

While the construction of the joint distribution $P(V_1, V_2)$ is not unique, in the particular case that D_1 and D_2 are close to each other, we would like to make $V_1 = V_2$ with a high probability.

As an intuitive interpretation, we can regard that in the "main world" Ω_1, a random event V_1 happens according to D_1; in a parallel "alternative world" Ω_2, the event is tampered to V_2 which has a different distribution D_2. We can imagine that such magical manipulation is costly, so that the manipulator would like to tamper as little as possible, i.e., minimize $P(V_1 \neq V_2)$ as long as $V_2 \sim D_2$. Actually, this model falls into the scope of *optimal transport*. From optimal transport theories, it holds that

$$\min_{V_1 \sim D_1, V_2 \sim D_2} P(V_1 \neq V_2) = TV(D_1, D_2). \tag{D.4}$$

Example. An academic institute has recently recruited 50 new tenure-track assistant professors, among which 5 are expected to get tenure, yielding a tenure rate of 10%. However, due to a sudden cut in funding, the tenure rate has to be lowered to 6%, hence some of the would-be decisions have to be changed. From Eq. (D.4), the minimum number of changed

decisions is

$$50 \cdot TV(Bern(0.10), Bern(0.06)) = 50 \cdot 0.04 = 2. \tag{D.5}$$

In fact, to change the fewest decisions, the tenure decisions for 2 unfortunate candidates will be revoked.

D.3.2 Interpretation for Robust Peer Prediction

In the context of (Byzantine)-robust peer prediction for n players, we consider a mechanism that is (δ, K)-compact for the environment $\hat{\phi}$ that has no rogue players. From Theorem 6.7 we know that this mechanism keeps the 0-IA incentive guarantee even if an arbitrary subset of at most $\frac{\delta}{2K}(n-1)$ players, i.e. a $\frac{\delta}{2K}$ fraction of other players, become malicious. For simplicity of discussion, we assume $n \to \infty$.

We assume that in the main world, the environment is $\hat{\phi}$. Conditioned on player i observing X_i, the (expected) distribution[2] of other players' observations is $P(X_j|X_i, \hat{\phi})$. From the Byzantine robustness results, the 0-IA guarantee holds as long as at least an $1 - \frac{\delta}{2K}$ fraction of them report honestly.

Then, we consider the alternative world in which the environment is ϕ. Assuming that the player i still observes X_i, the distribution of other players' observations is $P(X_j|X_i, \phi)$. From the coupling argument discussed above, a minimum of $TV_{X_j}(P(X_j|X_i, \hat{\phi}), P(X_j|X_i, \phi))$ fraction of other players have different observations between two worlds.

Now we assume that all other players report honestly in the alternative world, while in the main world, they also report their observations in the alternative world. Then we see that in the main world, an exact $TV_{X_j}(P(X_j|X_i, \hat{\phi}), P(X_j|X_i, \phi))$ fraction of other players are reporting dishonestly. Hence in the main world, the 0-IA guarantee holds as long as $TV_{X_j}(P(X_j|X_i, \hat{\phi}), P(X_j|X_i, \phi)) \leq \frac{\delta}{2K}$.

On the other hand, in the alternative world all players are reporting honestly, but the actual environment is ϕ instead of $\hat{\phi}$. Since all other players report identically in two worlds, in the perspective of player i, the 0-IA guarantee still holds in the alternative world as long as

[2]We consider the distribution ensemble-wise, i.e., among all possible ground-truth θ's. The term "(conditional) distribution" in this part is always interpreted in this way.

$TV_{X_j}(P(X_j|X_i,\hat{\phi}), P(X_j|X_i,\phi)) \leq \frac{\delta}{2K}$. Hence we see that the Byzantine-robustness against a $\frac{\delta}{2K}$ fraction of malicious players implies the tolerance of a $\frac{\delta}{2K}$ error of posterior beliefs.

An intuitive interpretation is that, from the perspective of a (self-centric) player i, even if the actual posterior distribution of others' observations differs from her belief, she can interpret this difference as arising from some players reporting dishonestly. In this interpretation, the player regards her belief (the ideal posterior distribution) as correct, while the fraction of players causing the discrepancy corresponds to the error measured by the total variation distance. This ensures that as long as the error remains below the tolerance threshold $\frac{\delta}{2K}$, the robustness guarantee holds.

D.4 Demonstration of th PoL Benchmark

In this section, we empirically demonstrate the process of designing a CTF-PP mechanism, for one set of parameters that is useful for practical interest.

D.4.1 Construction of the Scoring Rule

We consider the 2-verifier DVG which captures the case of one stage in [4]. Here, we set the distribution θ as $P(\theta = F_1) = P(\theta = F_2) = \frac{1}{4}$, which means that half of all stages are flagged. Then we consider the lossy-channel model in which $\mu_1 = \mu_2 = \frac{1}{2}$, as each verifier independently chooses half of all stages [3] and each flag is detected with probability 1 when verified. According to the CTF protocol, when a cheating stage is chosen by a verifier, it has an $\frac{1}{2}$ chance to be correctly detected, and a $\frac{1}{4}$ chance to be observed as F_1 or F_2 respectively. Hence, $P(X_i|\theta)$ is shown as in Table D.1, and assuming $\epsilon = 0$, we compute the marginal distribution of X_i as $B_\perp = [\frac{3}{4}, \frac{1}{8}, \frac{1}{8}, 0]$.

Then, assuming $\epsilon = 0$, from $P(X_i, X_{-i}) = \sum_\theta P(\theta)P(X_i|\theta)P(X_{-i}|\theta)$ we can compute the joint probabilities $P(X_1, X_2)$, as shown in Table D.2. We accordingly compute the post-observation belief $P(X_2|X_1) = \frac{P(X_1,X_2)}{P(X_1)}$ for $X_1 \neq 1$, and $P(X_2|X_1 = 1) = P(X_2|\theta = 1)$, getting the principal belief matrix B as Table D.3.

[3] It is significantly more than needed, but does work.

	$X_i = 0$	$X_i = F_1$	$X_i = F_2$	$X_i = 1$
$\theta = 0$	1	0	0	0
$\theta = F_1$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
$\theta = F_2$	$\frac{1}{2}$	0	$\frac{1}{2}$	0
$\theta = 1$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

Table D.1: $P(X_i|\theta)$, the observation distribution conditioned on θ.

	$X_2 = 0$	$X_2 = F_1$	$X_2 = F_2$	$X_2 = 1$
$X_1 = 0$	$\frac{5}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	0
$X_1 = F_1$	$\frac{1}{16}$	$\frac{1}{16}$	0	0
$X_1 = F_2$	$\frac{1}{16}$	0	$\frac{1}{16}$	0
$X_1 = 1$	0	0	0	0

Table D.2: Joint probabilities $P(X_1, X_2)$ ($\epsilon = 0$).

From the nature of the CTF mechanism, in which the observation of F_1, F_2 of 1 takes twice the computational cost of a stage, we set $c(F_1) = c(F_2) = c(1) = 2c$. On the other hand, the "observation" of a 0 has two cases: one is that the verifier has verified the stage that is not cheated or flagged, which has a $\mu(1-\eta) = \frac{1}{4}$ probability, and one is that the verifier does not verify the stage from the random verification protocol, which has a $1 - \mu = \frac{1}{2}$ probability. Hence, we have an "amortized" $c(0) = \frac{1}{3}c$. Without loss of generality, we set $c = 1$.

With the knowledge of $B, B_\perp$ and c, Eqs. (6.11)-(6.14) are the constraints that a desirable scoring rule for a CTF-PP mechanism should satisfy. For the robustness of our mechanism, we do not want the payments to have extremely large absolute values. Hence, we construct the linear program as:

$$\begin{aligned} \textit{minimize} \quad & K \\ s.t. \quad & \text{(6.11)-(6.14)}, \quad -K \leq T \leq K. \end{aligned}$$

	$X_2 = 0$	$X_2 = F_1$	$X_2 = F_2$	$X_2 = 1$
$X_1 = 0$	$\frac{5}{6}$	$\frac{1}{12}$	$\frac{1}{12}$	0
$X_1 = F_1$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
$X_1 = F_2$	$\frac{1}{2}$	0	$\frac{1}{2}$	0
$X_1 = 1$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

Table D.3: The principal belief matrix B.

We set a margin of $\delta = 0.2$, and compute a numerical solution to this LP, getting a scoring rule as shown in Table D.4.

	$Z_{-i} = 0$	$Z_{-i} = F_1$	$Z_{-i} = F_2$	$Z_{-i} = 1$
$Z_i = 0$	$+2.0690$	-7.1451	-7.1451	-2.2507
$Z_i = F_1$	-2.0446	$+6.4446$	-4.7421	-2.0022
$Z_i = F_2$	-2.0446	-4.7421	$+6.4446$	-2.0022
$Z_i = 1$	-2.2000	$+5.8000$	$+5.8000$	$+7.4000$

Table D.4: $T_{Z_i Z_{-i}} = R_i(Z_i, Z_{-i})$ as a numerical solution.

D.4.2 Evaluation

With this scoring rule, given verifier $-i$ acts honestly, we report the expected utility of verifier i is in Table D.5, assuming $\epsilon = 0$, showing that the verifier gets a positive expected utility if and only if she verifies and reports honestly.

Furthermore, we consider the case $\epsilon > 0$. We plot the maximum expected utility of dishonest actions and the minimum expected utility of honest actions in Figure D.1. From the plot, we show that the introduction of the margin keeps the IR, UniIC and NFL properties of our mechanism as long as $\epsilon < 0.045$. Hence, we demonstrate that our design of the CTF-PP mechanism for the 2-verifier DVG can incentivize honest verification even if there is no dishonest prover, thus bypassing the Verifier's Dilemma and achieving a pure-strategy Nash equilibrium that the prover and verifiers simultaneously act honestly.

	Reporting 0	Reporting F_1	Reporting F_2	Reporting 1
Observing 0	$+\mathbf{0.2000}$	-1.8953	-1.8953	-1.2000
Observing F_1	-4.5381	$+\mathbf{0.2000}$	-5.3933	-0.2000
Observing F_2	-4.5381	-5.3933	$+\mathbf{0.2000}$	-0.2000
Observing 1	-3.3144	-3.3100	-3.3100	$+\mathbf{0.2000}$
Uninformed	-0.2345	-1.3206	-1.3206	-0.2000

Table D.5: Verifier's expected utility, $\epsilon = 0$.

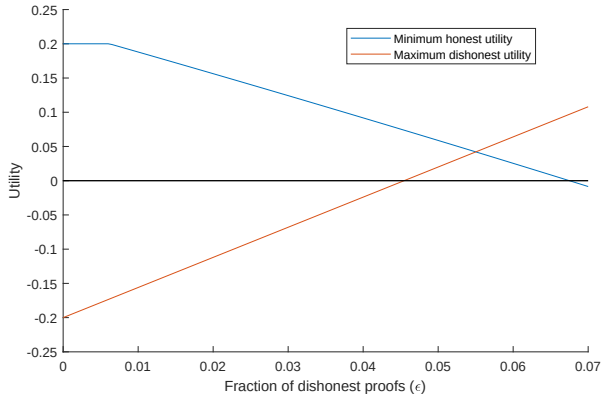

Figure D.1: Verifier's expected utility, $\epsilon > 0$.

D.5 Additional Experiments

In this part, we report the results of the second experiment as described in Section 6.7.1.

In this experiment, we consider the case of a 2-verifier DVG, in which the peer is honest but the actual prior has an ϵ TV distance from the principal distribution, simulating the case in which a small fraction of proofs is dishonest.

As the PMI baseline has been shown as the most competitive among the baselines in Section 6.7, and SA and DMI are infeasible even for the noise-free case in the PoL benchmark, in the PoL benchmark we mainly compare our design with PMI. To show the potential of re-scaling PMI scoring rules for an incentive margin (and robustness), we introduce a variation of the PMI mechanism:

- PMI-Oracle (PMI-O): The scoring rule is computed with the actual prior (which should not have been accessible in practice) and scaled accordingly.

For convenience in the computation of utilities, we truncate the infinite entries to ± 20 in the Log and PMI scoring rules. Since we have already shown in Section 6.7 that the DMI mechanism always has the same adversarial utility as honest utility (which is not desired in our setting), we omit the experiments for the DMI mechanism.

Here, we let $\epsilon = 0.01$ and $\epsilon = 0.03$, and show the results for Coin and PoL benchmarks in Tables D.6-D.9.

From the experiment results, we can see that the introduction of δ margin can ensure positive honest utility and negative dishonest utility even in the presence of inaccurate prior

	Budget	Honest Utility	Lazy Utility	Adversarial Utility
Ours ($\delta = 0$)	1.00	-0.004	0.04	-2.13
Ours ($\delta = 0.2$)	1.20	0.20	-0.15	-2.67
SA	1.57	0.57	0.08	-4.18
Log	1.39	0.39	0.07	-2.64
PMI	1.11	0.11	0.04	-2.37
PMI-O	1.12	0.12	0.00	-2.37

Table D.6: Experiment 2, Coin Benchmark, $\epsilon = 0.01$

	Budget	Honest Utility	Lazy Utility	Adversarial Utility
Ours ($\delta = 0$)	0.99	-0.01	0.12	-2.12
Ours ($\delta = 0.2$)	1.20	0.20	-0.04	-2.67
SA	1.57	0.57	0.24	-4.17
Log	1.41	0.41	0.20	-2.62
PMI	1.09	0.09	0.12	-2.37
PMI-O	1.16	0.16	0.00	-2.51

Table D.7: Experiment 2, Coin Benchmark, $\epsilon = 0.03$

of an $\epsilon = 0.03$ TV distance. In the Coin baseline, our design with $\delta = 0.2$ pays lower budget than SA and Log mechanisms, while the PMI-O baseline can achieve slightly lower budget than our design (with carefully tuned affine transformations assuming that the ϵ is known in advance). However, in the trickier PoL benchmark in which the Verifier's Dilemma actually occurs, our design yields better robustness and lower budgets than the PMI mechanism, even if we allow the PMI mechanism to optimally adjust the scaling factors with the accurate ϵ. Hence, we have shown that our design achieves a more robust and cost-efficient solution for the Verifier's Dilemma than existing peer prediction mechanisms listed above.

	Budget	Honest Utility	Lazy Utility	Adversarial Utility
Ours ($\delta = 0$)	0.75	-0.004	0.00	-2.85
Ours ($\delta = 0.2$)	0.95	0.20	-0.20	-3.30
PMI	1.24	0.49	0.01	-6.99
PMI-O	1.24	0.49	0.00	-7.98

Table D.8: Experiment 2, PoL Benchmark, $\epsilon = 0.01$

	Budget	Honest Utility	Lazy Utility	Adversarial Utility
Ours ($\delta = 0$)	0.74	-0.01	0.00	-2.83
Ours ($\delta = 0.2$)	0.94	0.19	-0.18	-3.28
PMI	1.22	0.47	0.02	-6.87
PMI-O	1.21	0.46	0.00	-7.64

Table D.9: Experiment 2, PoL Benchmark, $\epsilon = 0.03$

D.6 Deferred Proofs

D.6.1 Proof of Theorem 5.1

Assume we have such a mechanism. By the definition of Nash equilibrium, we consider a fixed verifier. Given that the prover and all other verifiers act honestly, that verifier should be incentivized to do the honest verification.

Since the prover is honest, when that verifier performs honest verification, the result should always be "Success". However, suppose the verifier simply reports "Success" without verification. In that case, the outcome is the same but the verifier saves the verification cost, so the verifier is incentivized to deviate from the honest strategy.

That leads to a contradiction, so no such mechanism exists.

D.6.2 Proof of Theorem 6.2

Notice that if B is invertible, for any given $W : S^2 \to \mathbb{R}$, we can compute a $T = (B^{-1}W)'$ that satisfies $W = BT'$. Since we have

$$B_{\perp y} = P(X_{-i} = y) \tag{D.6}$$

$$= \sum_{x \in S} P(X_i = x) \cdot P(X_{-i} = y | X_i = x) \tag{D.7}$$

$$= \sum_{x \in S} B_{\perp x} \cdot B_{xy}, \tag{D.8}$$

It holds that $B_\perp = B_\perp B$. Hence, $W_\perp = B_\perp T' = B_\perp B T' = B_\perp W$. Since $B_\perp \geq 0$, $W_\perp$ is a convex combination of rows in W. Therefore, we only need to construct a W that satisfies

Eq. (6.11) with non-diagonal entries small enough.

Here, for a constant $M > 0$ large enough, we construct W as:

$$W_{xx} = c(x) + \delta, \qquad \forall x \in S; \tag{D.9}$$

$$W_{xy} = -M, \qquad \forall x \in S, \quad y \in S - \{x\}. \tag{D.10}$$

Then,

$$W_{\perp y} = \sum_{x \in S} B_{\perp x} W_{xy} \tag{D.11}$$

$$= B_{\perp y} W_{yy} + \sum_{x \in S - \{y\}} B_{\perp x} W_{xy} \tag{D.12}$$

$$= B_{\perp y}(c(y) + \delta) - (1 - B_{\perp y})M. \tag{D.13}$$

Since the existence of flags introduces randomness in the observation, we have $\max\{B_\perp\} < 1$. Hence, denote $B^*_\perp = \max\{B_{\perp y}\}$, we only need to let

$$M \geq \frac{B^*_\perp \cdot (c(y) + \delta) + \delta}{1 - B^*_\perp}, \tag{D.14}$$

Then the required constraints of Eqs. (6.11)-(6.14) are satisfied with a margin of δ.

Then we consider the scenario that $\epsilon > 0$ but is small enough. In this case, define $B(\epsilon)$ and $B_\perp(\epsilon)$ as the belief matrix and blind-belief matrix considering the influence of ϵ. We can see that for any $x \neq 1$, since $\tilde{P}(X_i = x) > 0$, the influences of ϵ on $P(X_{-i}|X_i = x)$ and $P(X_{-i})$ are upper bounded by $O(\epsilon)$, and because $\theta \neq 1 \Rightarrow X_i \neq 1$, Eq. (6.6) always holds for any ϵ. Therefore, the margin of δ ensures that the constraints are not violated as long as ϵ_0 is small enough.

Finally, let $T = (B^{-1}W)'$, then T is a scoring rule that satisfies the requirements.

D.6.3 Proof of Proposition 6.3

In the 2-verifier DVG, By definition $B_{xy} = P(X_{-i} = y | X_i = x)$. We sort the elements of S in the order $(0, F_1, \cdots, F_m, 1)$. For convenience, we define the observation matrix (aka. confusion matrix) O and inference matrix E as:

$$O_{xy} = P(X_i = y | \theta = x), \tag{D.15}$$

$$E_{xy} = P(\theta = y | X_i = x). \tag{D.16}$$

From the lossy-channel model, we immediately see that O is an upper triangular matrix with non-zero diagonals, so O is invertible.

Furthermore, given $\epsilon = 0$, we can see that:

- If $X_i = 0$, then the ground truth θ may be 0 or any flag F_j, and $P(\theta = 0 | X_i = 0) > 0$.
- If $X_i = F_j$, then $\theta = F_j$.
- If $X_i = 1$, then $\theta = 1$.

Hence, E is a lower triangular matrix with non-zero diagonals, so E is invertible. Then, because X_i, X_{-i} are independent conditioned on θ, we have

$$B_{X_i X_{-i}} = P(X_{-i} | X_i) \tag{D.17}$$

$$= \sum_{\theta \in S} P(\theta | X_i) P(X_{-i} | \theta) \tag{D.18}$$

$$= \sum_{\theta \in S} E_{X_i \theta} \cdot O_{\theta X_{-i}}. \tag{D.19}$$

Therefore, $B = EO$.

Since E, O are invertible, we deduce that B is invertible.

D.6.4 Proof of Theorem 6.4

In the context of Bayesian Nash equilibrium, we can assume each verifier $j \neq i$ is honest, i.e., $\mathbf{Z}_{-i} = \mathbf{X}_{-i}$. Hence, given that verifier i observes $X_i \in S \cup \{\perp\}$ and reports $Z_i \in S$, the interim expected reward is:

$$r_{X_i}(Z_i) = \mathbb{E}\Big[Z_i' T \overline{\mathbf{X}_{-i}} \Big| X_i\Big] \tag{D.20}$$

$$= Z_i' T \cdot \mathbb{E}\Big[\overline{\mathbf{X}_{-i}} \Big| X_i\Big] \tag{D.21}$$

$$= Z_i' T \cdot \mathbb{E}\Big[\frac{1}{n-1}\sum_{j\neq i} X_j \Big| X_i\Big] \tag{D.22}$$

$$= \frac{1}{n-1}\sum_{j\neq i} Z_i' T \mathbb{E}\Big[X_j \Big| X_i\Big] \tag{D.23}$$

$$= \frac{1}{n-1}\sum_{j\neq i} Z_i' T \sum_{X_j \in S} P(X_j|X_i) X_j \tag{D.24}$$

$$= \frac{1}{n-1}\sum_{j\neq i}\sum_{X_j \in S} P(X_j|X_i) T_{Z_i X_j}. \tag{D.25}$$

With similar arguments as Section 6.4, we assume $\epsilon = 0$, and $P(X_j|X_i)$ is the (i,j)-th entry of the principal belief matrix B for any $j \neq i$. Hence, we have

$$r_{X_i}(Z_i) = \frac{1}{n-1}\sum_{j\neq i}\sum_{X_j \in S} P(X_j|X_i) T_{Z_i X_j} \tag{D.26}$$

$$= \sum_{X_j \in S} B_{X_i X_j} T_{Z_i X_j} \tag{D.27}$$

$$= (BT')_{X_i Z_i}. \tag{D.28}$$

Hence, the linear program of Eqs. (6.11-6.14) works equivalently for the n-verifier DVG when we use the linear average mechanism as Eq. (6.15) with exactly the same incentive structure. So any incentive property satisfied in the 2-verifier mechanism T is also satisfied in the linear average mechanism in Eq. (6.15).

D.6.5 Proof of Theorem 6.5

We first observe that any feasible solution of $LP_1(B, B_\perp, c, \delta)$ can be constructed with feasible solutions of $LP_1(B, B_\perp, c, 0)$ and $LP_1(B, B_\perp, 0, 1)$, i.e.,

Observation 2. *If (K_c, T_c) is a feasible solution of $LP_1(B, B_\perp, c, 0)$ and (K_δ, T_δ) is a feasible solution of $LP_1(B, B_\perp, 0, 1)$, then $(K_c+\delta K_\delta, T_c+\delta T_\delta)$ is a feasible solution of $LP_1(B, B_\perp, c, \delta)$.*

Hence, we can estimate upper bounds of optimal K_c and K_δ separately. Here we denote $\|\cdot\|_2$ as the matrix ℓ_2-norm, and denote $W = BT'$. From the assumption in Theorem 6.2 that B is invertible, T can be constructed as $(B^{-1}W)'$ and it holds that $\forall x, y \in S$, $|T_{xy}| \leq \|T\|_2 = \|(B^{-1}W)'\|_2 \leq \|B^{-1}\|_2 \cdot \|W\|_2$. Hence, we can estimate the upper bounds on entrywise maximums of T via ℓ_2 norms of W, respectively.

For $LP_1(B, B_\perp, c, 0)$, if we construct

$$W_{xy} = \begin{cases} c, & y = x; \\ -\frac{B_{\perp y}}{1-B_{\perp y}} \cdot c, & y \neq x. \end{cases} \tag{D.29}$$

Then the corresponding $T_c = (B^{-1}W)'$ obviously satisfies conditions (6.23-6.24). For condition (6.25), we have

$$\begin{aligned} (B_\perp W)_y &= \sum_{x \in S} B_{\perp x} W_{xy} && \text{(D.30)} \\ &= B_{\perp y} W_{yy} + \sum_{x \in S - \{y\}} B_{\perp x} W_{xy} && \text{(D.31)} \\ &= B_{\perp y} \cdot 1 + (1 - B_{\perp y}) \cdot \left(- \frac{B_{\perp y}}{1 - B_{\perp y}} \right) && \text{(D.32)} \\ &= 0. && \text{(D.33)} \end{aligned}$$

So T_c is feasible for $LP_1(B, B_\perp, c, 0)$, and we analyze K_c later.

Before analysis for $LP_1(B, B_\perp, 0, 1)$, we prove a lemma:

Lemma D.1. *$B'_\perp$ is an eigenvector of B' with eigenvalue 1, i.e., $B_\perp B = B_\perp$.*

Proof. Proof

Let j be an arbitrary verifier other than j. From the discussion of the uninformed strategy (in Section 6.3), we have

$$B_{\perp y} = P(X_j = y | X_i = \perp) \tag{D.34}$$

$$= P(X_j = y). \tag{D.35}$$

On the other hand,

$$(B_\perp B)_y = \sum_{x \in S} B_{\perp x} B_{xy} \tag{D.36}$$

$$= \sum_{x \in S} P(X_i = x) \cdot P(X_j = y | X_i = x) \tag{D.37}$$

$$= \sum_{x \in S} P(X_i = x, X_j = y) \tag{D.38}$$

$$= P(X_j = y). \tag{D.39}$$

Hence we have $B_{\perp y} = (B_\perp B)_y$ for $\forall y \in S$, so $B_\perp B = B_\perp$.

□ **Q.E.D.**

From Lemma D.1, the $LP_1(B, B_\perp, 0, 1)$ can be reformulated as:

$LP_2(B, B_\perp, 0, 1)$:

$$\textit{minimize} \qquad K \tag{D.40}$$

$$\textit{s.t.} \qquad |B^{-1}W| \leq K, \tag{D.41}$$

$$W_{xx} \geq 1, \qquad \forall x \in S \tag{D.42}$$

$$W_{xy} \leq -1, \qquad \forall x \in S, \quad y \in S - \{x\} \tag{D.43}$$

$$B_\perp W \leq -1. \tag{D.44}$$

Here, we can construct

$$W_{xy} = \begin{cases} 1, & y = x; \\ -\frac{1+B_{\perp y}}{1-B_{\perp y}}, & y \neq x. \end{cases}$$

From the construction we immediately see that conditions (D.42-D.43) are satisfied. For condition (D.44), we have

$$\begin{aligned} (B_\perp W)_y &= \sum_{x \in S} B_{\perp x} W_{xy} && \text{(D.45)} \\ &= B_{\perp y} W_{yy} + \sum_{x \in S - \{y\}} B_{\perp x} W_{xy} && \text{(D.46)} \\ &= B_{\perp y} \cdot 1 + (1 - B_{\perp y}) \cdot \left(- \frac{1 + B_{\perp y}}{1 - B_{\perp y}} \right) && \text{(D.47)} \\ &= -1. && \text{(D.48)} \end{aligned}$$

Hence, the W is feasible for $LP_2(B, B_\perp, 0, 1)$. Now we estimate an upper bound on $\|W\|_2$. We first show a lemma:

Lemma D.2. *For any matrix A,*

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}. \tag{D.49}$$

Proof. Proof Denote A^* as the conjugate transpose of A, which is equal to A' when A is real, and denote $\lambda_{max}(\cdot)$ as the maximum eigenvalue. Then it holds that

$$\|A\|_2 = \sqrt{\lambda_{max}(A^* A)}. \tag{D.50}$$

Because the maximum eigenvalue is a lower bound on the ℓ_∞ norm, we have

$$\lambda_{max}(A^*A) \leq \|A^*A\|_\infty \tag{D.51}$$
$$\leq \|A^*\|_\infty \|A\|_\infty \tag{D.52}$$
$$= \|A\|_1 \|A\|_\infty. \tag{D.53}$$

Hence we prove $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$.

□ **Q.E.D.**

Now we sort $\{B_{\perp y} : y \in S\}$ as $p_1 \geq p_2 \geq \cdots \geq p_k$, in which $k = |S| = m + 2$. Then we have

$$\|W\|_1 = \max_{y \in S} \sum_{x \in S} |W_{xy}| \tag{D.54}$$
$$= \max_{1 \leq j \leq k} \left\{ 1 + (k-1) \frac{1 + p_j}{1 - p_j} \right\} \tag{D.55}$$
$$= 1 + (k-1) \frac{1 + p_1}{1 - p_1} \tag{D.56}$$
$$= k + (2k-2) \frac{p_1}{1 - p_1}, \tag{D.57}$$

and

$$\|W\|_\infty = \max_{x \in S} \sum_{y \in S} |W_{xy}| \tag{D.58}$$
$$= \max_{1 \leq i \leq k} \left\{ 1 + \sum_{j \neq i} \frac{1 + p_j}{1 - p_j} \right\} \tag{D.59}$$
$$\leq \sum_{i=1}^{k} \frac{1 + p_j}{1 - p_j} \tag{D.60}$$
$$= k + 2 \sum_{i=1}^{k} \frac{p_j}{1 - p_j} \tag{D.61}$$
$$\leq k + 2 \sum_{i=1}^{k} \frac{p_j}{1 - p_1} \tag{D.62}$$
$$= k + \frac{2}{1 - p_1}. \tag{D.63}$$

Therefore, we have

$$\|W\|_2 \leq \sqrt{\Big(k + (2k-2)\frac{p_1}{1-p_1}\Big)\Big(k + \frac{2}{1-p_1}\Big)} \tag{D.64}$$

and

$$K_\delta = \|B^{-1}\|_2 \cdot \sqrt{\Big(k + (2k-2)\frac{p_1}{1-p_1}\Big)\Big(k + \frac{2}{1-p_1}\Big)} \tag{D.65}$$

is feasible for $LP_2(B, B_\perp, 0, 1)$.

Similarly, denote $\tilde{W}$ as the matrix given by (D.29), then we have

$$\|\tilde{W}\|_1 = \max_{y \in S} \sum_{x \in S} |\tilde{W}_{xy}| \tag{D.66}$$

$$= \max_{1 \leq j \leq k} \Big\{1 + (k-1)\frac{p_j}{1-p_j}\Big\} \tag{D.67}$$

$$= 1 + (k-1)\frac{p_1}{1-p_1} \tag{D.68}$$

and

$$\|\tilde{W}\|_\infty = \max_{x \in S} \sum_{y \in S} |W_{xy}| \tag{D.69}$$

$$= \max_{1 \leq i \leq k} \Big\{1 + \sum_{j \neq i} \frac{p_j}{1-p_j}\Big\} \tag{D.70}$$

$$\leq \max_{1 \leq i \leq k} \Big\{1 + \sum_{j} \frac{p_j}{1-p_j}\Big\} \tag{D.71}$$

$$\leq 1 + \sum_{j} \frac{p_j}{1-p_1} \tag{D.72}$$

$$= 1 + \frac{\sum_j p_j}{1-p_1} \tag{D.73}$$

$$= 1 + \frac{1}{1-p_1} \tag{D.74}$$

Therefore, we have

$$\|\tilde{W}\|_2 \leq \sqrt{\Big(1 + (k-1)\frac{p_1}{1-p_1}\Big)\Big(1 + \frac{1}{1-p_1}\Big)} \tag{D.75}$$

and

$$K_c = \|B^{-1}\|_2 \cdot \sqrt{\left(1 + (k-1)\frac{p_1}{1-p_1}\right)\left(1 + \frac{1}{1-p_1}\right)}. \tag{D.76}$$

Hence, $K_c + \delta K_\delta$ is feasible for $LP_1(B, B_\perp, c, \delta)$. Because our constructions for both parts make the equality hold in (6.23), the final construction makes the equality hold naturally.

D.6.6 Proof of Theorem 6.7

For the 0-IA property, according to Proposition 6.6 we only need to equivalently consider the case of $|\mathscr{M}_*| + |\mathscr{C}_*|$ malicious players in the canonical Byzantine setting. From Lemma 6.1, the mechanism is 0-IA even if up to $\frac{\delta}{2N}(n-1)$ malicious players are considered. Since $|\mathscr{M}_*| + |\mathscr{C}_*| \leq \frac{\delta}{2N}(n-1)$, it is indeed 0-IA.

Then we consider the colluding party. If all players in $\mathscr{C}_*$ act honestly, since the mechanism is (δ, K)-compact, each of them would get an interim utility of at least δ if there were no malicious players. As there are $\mathscr{M}_*$ malicious players and each can perturb $r_{X_i}(Z_i)$ by at most $\frac{2N}{n-1}$, the actual interim utility of each player is at least $\delta - \frac{2N}{n-1}|\mathscr{M}_*|$, so the total interim utility of the colluding party is

$$u_{\mathscr{C}_*}^{honest} \geq |\mathscr{C}_*| \cdot \left(\delta - \frac{2N}{n-1}|\mathscr{M}_*|\right). \tag{D.77}$$

Assuming the mechanism is not weak-SCP, then there exists a case in which $1 \leq d \leq |\mathscr{C}_*|$ players in $\mathscr{C}_*$ act dishonestly and increase the total interim utility of the colluding party. Hence, compared to the case that all players in $\mathscr{C}_*$ act honestly, we can model this scenario as colluding players in $\mathscr{C}_*$ change their actions, and consider the increment of their utilities.

As we assumed, d players in $\mathscr{C}_*'$ change their actions from honest to dishonest. Since there are now at most $|\mathscr{M}_*| + d$ dishonest players, the interim utility of each player in $\mathscr{C}_*'$ is at most $-\delta + \frac{2N}{n-1}(|\mathscr{M}_*| + d)$; on the other hand, d players changing their actions may increase the interim utility of each player in $\mathscr{C}_* - \mathscr{C}_*'$ by at most $\frac{2N}{n-1}d$. Hence, the increment of the

total interim utility in $\mathscr{C}_*$ is

$$\Delta \leq d \cdot \left((-\delta + \frac{2N}{n-1}(|\mathscr{M}_*| + d)) - (\delta - \frac{2N}{n-1}|\mathscr{M}_*|) \right) + (|\mathscr{C}_*| - d) \cdot \frac{2N}{n-1} d \tag{D.78}$$

$$= \left(-2\delta + \frac{2N}{n-1}(2|\mathscr{M}_*| + |\mathscr{C}_*|) \right) d \tag{D.79}$$

$$\leq \left(-2\delta + \frac{4N}{n-1}(|\mathscr{M}_*| + |\mathscr{C}_*|) \right) d \tag{D.80}$$

$$= \left(-2\delta + \frac{4N}{n-1} \cdot \frac{\delta}{2N}(n-1) \right) d \tag{D.81}$$

$$= 0. \tag{D.82}$$

Therefore, we show that the deviation cannot increase the colluding party's total utility, i.e. the mechanism is SCP when $|\mathscr{M}_*| + |\mathscr{C}_*| \leq \frac{\delta}{2N}(n-1)$.

D.6.7 Proof of Theorem 6.8

From Theorem 6.5 we see that for any $\delta \geq 0$, LP_1 has a feasible solution with the equality in Eq. (6.23) holding and objective value

$$K \leq \|B^{-1}\|_2 (c_1 \cdot g_1(k, p_1) + \delta \cdot g_2(k, p_1)). \tag{D.83}$$

To ensure a compactness of at least η, we only need $\delta \geq \eta K$.

In fact, assuming $\eta < \frac{1}{g_2(k,p_1)\|B^{-1}\|_2}$, if we let $\delta = \frac{\eta c_1 g_1(k,p_1)\|B^{-1}\|_2}{1-\eta g_2(k,p_1)\|B^{-1}\|_2}$ as in Eq. (6.29), then

$$\delta - \eta K \geq \frac{\eta c_1 g_1(k,p_1)\|B^{-1}\|_2}{1-\eta g_2(k,p_1)\|B^{-1}\|_2} - \eta \cdot \|B^{-1}\|_2(c_1 g_1(k,p_1) + \delta \cdot g_2(k,p_1)) \tag{D.84}$$

$$= \frac{\eta c_1 g_1(k,p_1)\|B^{-1}\|_2 - \eta \cdot \|B^{-1}\|_2(c_1 g_1(k,p_1) + \delta g_2(k,p_1))(1-\eta g_2(k,p_1)\|B^{-1}\|_2)}{1-\eta g_2(k,p_1)\|B^{-1}\|_2} \tag{D.85}$$

$$= \frac{-\eta\|B^{-1}\|_2 \cdot (-c_1 g_1(k,p_1)\eta g_2(k,p_1)\|B^{-1}\|_2 + \delta g_2(k,p_1) - \eta\delta(g_2(k,p_1))^2\|B^{-1}\|_2)}{1-\eta g_2(k,p_1)\|B^{-1}\|_2} \tag{D.86}$$

$$= \frac{-\eta g_2(k,p_1)\|B^{-1}\|_2 \cdot (-\eta c_1 g_1(k,p_1)\|B^{-1}\|_2 + \delta \cdot (1-\eta g_2(k,p_1)\|B^{-1}\|_2))}{1-\eta g_2(k,p_1)\|B^{-1}\|_2} \tag{D.87}$$

$$= \frac{-\eta g_2(k,p_1)\|B^{-1}\|_2}{1-\eta g_2(k,p_1)\|B^{-1}\|_2} \cdot (-\eta c_1 g_1(k,p_1)\|B^{-1}\|_2 + \eta c_1 g_1(k,p_1)\|B^{-1}\|_2) \tag{D.88}$$

$$= 0. \tag{D.89}$$

Hence, the η compactness is satisfied.

Now we only need to show that $\mu = \delta$. Actually, whenever a verifier gets an observation x she pays the cost of $c(x)$, and because the equality holds in Eq. (6.23), she gets an expected reward of $c(x) + \delta$ over the (conditional) distribution of other verifiers' observations. Hence we see that the expected payment to any verifier is δ plus the expected verification cost, and that $\mu = \delta$ holds indeed.

D.6.8 Proof of Lemma 6.2

We only need to prove that

$$\left| \sum_{X_j \in S} P(X_j|X_i,\hat{\phi}) T_{Z_i X_j} - \sum_{X_j \in S} P(X_j|X_i,\phi) T_{Z_i X_j} \right| \leq \delta.$$

In fact,

$$\left|\sum_{X_j\in S} P(X_j|X_i,\hat{\phi})T_{Z_iX_j} - \sum_{X_j\in S} P(X_j|X_i,\phi)T_{Z_iX_j}\right|$$

$$= \left|\sum_{X_j\in S} T_{Z_iX_j}\left(P(X_j|X_i,\hat{\phi}) - P(X_j|X_i,\phi)\right)\right| \tag{D.90}$$

$$\leq \sum_{X_j\in S} |T_{Z_iX_j}| \cdot \left|P(X_j|X_i,\hat{\phi}) - P(X_j|X_i,\phi)\right| \tag{D.91}$$

$$\leq \sum_{X_j\in S} N \cdot \left|P(X_j|X_i,\hat{\phi}) - P(X_j|X_i,\phi)\right| \tag{D.92}$$

$$= N \cdot \sum_{X_j\in S} \left|P(X_j|X_i,\hat{\phi}) - P(X_j|X_i,\phi)\right|. \tag{D.93}$$

From the definition of TV distance, we have

$$TV_{X_j}(P(X_j|X_i,\hat{\phi}), P(X_j|X_i,\phi))$$
$$= \frac{1}{2}\sum_{X_j\in S} \left|P(X_j|X_i,\hat{\phi}) - P(X_j|X_i,\phi)\right|. \tag{D.94}$$

Hence,

$$N \cdot \sum_{X_j\in S} \left|P(X_j|X_i,\hat{\phi}) - P(X_j|X_i,\phi)\right|$$

$$= 2N \cdot TV_{X_j}(P(X_j|X_i,\hat{\phi}), P(X_j|X_i,\phi)) \tag{D.95}$$

$$\leq 2N \cdot \frac{\delta}{2N} \tag{D.96}$$

$$= \delta. \tag{D.97}$$

Here we prove Lemma 6.2.

D.6.9 Proof of Theorem 6.10

From Theorem 6.4, without loss of generality we consider the 2-verifier DVG. Because environments $\phi, \hat{\phi}$ are identical except for priors, we define the environmental constraint

Φ as all environments identical to ϕ except for different priors. Then for $\varphi \in \Phi$, $P(X_i|\theta, \varphi)$ is a constant irrelevant to φ and we regard $\Theta_s = P(\theta = s, \varphi)$ as the variable. We omit the φ for simplicity in the following parts of the proof. Then, we can see that

$$P(X_{-i}|X_i) = \frac{\sum_\theta P(\theta)P(X_i|\theta)P(X_{-i}|\theta)}{\sum_\theta P(\theta)P(X_i|\theta)} \tag{D.98}$$

is a function of $\Theta = \{\Theta_s\}$. We denote $\mathbf{Q} : S \to \mathbb{R}^S$ as:

$$\mathbf{Q}_{X_{-i}}(X_i, \Theta) = P(X_{-i}|X_i).$$

Then, we derive stability of $\mathbf{Q}_{X_{-i}}(X_i, \cdot)$ via ℓ_1-Lipschitz properties. While in Eq. (D.98) there is a natural constraint that $\sum_{s\in S} \Theta_s = 1$, here we relax this constraint and allow arbitrary $\Theta \in \mathbb{R}^S$ for convenience in analysis. We have

$$\frac{\partial \mathbf{Q}_{X_{-i}}(X_i, \Theta)}{\partial \Theta_s} = \frac{P(X_i|\theta = s)P(X_{-i}|\theta = s)}{\sum_\theta P(\theta)P(X_i|\theta)} - \frac{\sum_\theta P(\theta)P(X_i|\theta)P(X_{-i}|\theta) \cdot P(X_i|\theta = s)}{(\sum_\theta P(\theta)P(X_i|\theta))^2} \tag{D.99}$$

$$= \frac{P(X_i|\theta = s)P(X_{-i}|\theta = s)}{P(X_i)} - \frac{P(X_i, X_{-i}) \cdot P(X_i|\theta = s)}{P^2(X_i)}. \tag{D.100}$$

Hence, for fixed X_i, it holds that

$$\sum_{X_{-i}\in S} \left| \frac{\partial \mathbf{Q}_{X_{-i}}(X_i, \Theta)}{\partial \Theta_s} \right| \le \sum_{X_{-i}\in S} \left| \frac{P(X_i|\theta = s)P(X_{-i}|\theta = s)}{P(X_i)} \right| + \sum_{X_{-i}\in S} \left| \frac{P(X_i, X_{-i}) \cdot P(X_i|\theta = s)}{P^2(X_i)} \right| \tag{D.101}$$

$$= \frac{P(X_i|\theta = s)}{P(X_i)} + \frac{P(X_i) \cdot P(X_i|\theta = s)}{P^2(X_i)} \tag{D.102}$$

$$= \frac{2P(X_i|\theta = s)}{P(X_i)} \tag{D.103}$$

$$\le \frac{2}{P(X_i)}. \tag{D.104}$$

Therefore, we deduce that $\mathbf{Q}(X_i, \Theta)$ is $\frac{2}{P(X_i)}$-ℓ_1-Lipschitz at point Θ, in which $\frac{2}{P(X_i)}$ is a function of Θ because different priors result in different marginal probabilities of observations.

Now we denote that the priors in $\phi, \hat{\phi}$ as Θ_1, Θ_2, and consider the path $\omega : [0, 1] \to \mathbb{R}^S$

from Θ_1 to Θ_2 as

$$\omega(t) = (1-t)\Theta_1 + t\Theta_2,$$

and denote the environmental variable corresponding to $\omega(t)$ as φ_t. Then, because $P(X_i)$ is a linear function of $\{P(\theta)\}$, it holds that:

$$P(X_i|\varphi_t) = (1-t)P(X_i|\phi) + tP(X_i|\hat{\phi}) \tag{D.105}$$

$$\geq \min\{P(X_i|\phi), P(X_i|\hat{\phi})\}. \tag{D.106}$$

Hence, we deduce that $\mathbf{Q}(X_i, \cdot)$ is $\max\{\frac{2}{P(X_i|\phi)}, \frac{2}{P(X_i|\hat{\phi})}\}$-$\ell_1$-Lipschitz on ω.

For $X_i \in S^*$, from the assumption that $TV_\theta(P(\theta|\phi), P(\theta|\hat{\phi})) \leq \frac{\delta}{4N} \cdot \min_{X_i \in S^*, \varphi \in \{\phi, \hat{\phi}\}} \{P(X_i|\varphi)\}$, we see that the ℓ_1 length of ω is at most $\frac{\delta}{2N} \cdot \min_{X_i \in S^*, \varphi \in \{\phi, \hat{\phi}\}} \{P(X_i|\varphi)\}$. From the Lipschitz properties, we have

$$\|\mathbf{Q}(X_i, \Theta_1) - \mathbf{Q}(X_i, \Theta_2)\|_1 \leq \frac{\delta}{N}. \tag{D.107}$$

Notice that the TV distance between two distributions is $\frac{1}{2}$ times the ℓ_1 distance between the corresponding probability vectors, hence

$$TV_{X_{-i}}(P(X_{-i}|X_i, \phi), P(X_{-i}|X_i, \hat{\phi})) \leq \frac{\delta}{2N}. \tag{D.108}$$

For $X_i = 1$, i.e. the observation is "Dishonest", from the assumption that $P(X_i = 1|\theta \neq 1) = 0$ we deduce that $X_i = 1$ implies $\theta = 1$. Hence, $P(X_{-i}|P(X_i = 1)) = P(X_{-i}|\theta = 1)$ is a known constant (by the modeling in Section 6.3) and is not affected by the inaccurate prior distributions of θ, so Eq. (D.108) also holds for $X_i = 1$.

According to Theorem 6.9, Eq. (D.108) implies that the mechanism is 0-IA for environment ϕ.

D.6.10 Proof of Theorem D.1

(1) From Eq. (D.1) we have

$$\sum_{i\in\mathscr{C}} R_i(Z_i, \mathbf{Z}_{-i}) = \sum_{i\in\mathscr{C}} \frac{1}{n-1}\left(\sum_{j\neq i} Z_i' T Z_j\right) \tag{D.109}$$

$$= \frac{1}{n-1}\sum_{i\in\mathscr{C}}\left(\sum_{j\notin\mathscr{C}} Z_i' T Z_j + \sum_{j\in\mathscr{C}\setminus\{i\}} Z_i' T Z_j\right) \tag{D.110}$$

$$= \frac{c(n-c)}{n-1}\overline{\mathbf{Z}_{\mathscr{C}}}' T \overline{\mathbf{Z}_{-\mathscr{C}}} + \frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} Z_i' T Z_j, \tag{D.111}$$

in which $\overline{\mathbf{Z}_{\mathscr{C}}}, \overline{\mathbf{Z}_{-\mathscr{C}}}$ are the average reports of players inside and outside the colluding party $\mathscr{C}$, respectively. Hence, we have

$$\mathbb{E}_{\mathbf{Z}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[\sum_{i\in\mathscr{C}} R_i(Z_i, \mathbf{Z}_{-i})\right] = \overline{\mathbf{Z}_{\mathscr{C}}}' \cdot \frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{Z}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{Z}_{-\mathscr{C}}}\right] + \frac{1}{n-1}\sum_{j\in\mathscr{C}\setminus\{i\}} Z_i' T Z_j. \tag{D.112}$$

We notice that for fixed n, c, scoring matrix T, and shared belief $P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})$, the term

$$\frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{Z}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{Z}_{-\mathscr{C}}}\right]$$

is a constant and $\overline{\mathbf{Z}_{\mathscr{C}}}' \cdot \frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{Z}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{Z}_{-\mathscr{C}}}\right]$ is linear to $\overline{\mathbf{Z}_{\mathscr{C}}}$, the *average* of all players' reports in $\mathscr{C}$. Hence, we have

$$\overline{\mathbf{Z}_{\mathscr{C}}}' \cdot \frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{Z}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{Z}_{-\mathscr{C}}}\right] \leq \max_{i\in\mathscr{C}}\left\{\mathbf{Z}_i' \cdot \frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{Z}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{Z}_{-\mathscr{C}}}\right]\right\}. \tag{D.113}$$

Let i^* be the $i \in \mathscr{C}$ that yields the maximum in RHS. On the other hand, we have

$$\frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} Z_i' T Z_j = \frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} T_{Z_i Z_j}. \tag{D.114}$$

Since $\forall T_{Z_iZ_j} \in [-N, N]$,

$$\frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} Z_i' T Z_j \in \left[-\frac{c(c-1)}{n-1}N, \frac{c(c-1)}{n-1}N\right]. \tag{D.115}$$

Let $\forall Z_i = X_{i^*}$ and $h = \frac{2c(c-1)}{n-1}N$, we can see that Eq. (D.2) holds.

(2) We let $\forall Z_i = X_i$. Since for any $X_i \neq X_j$ it holds that $T_{X_iX_i} \geq T_{X_iX_j}$, and we first adopt the assumption that $\exists i, j \in \mathscr{C}$ s.t. $T_{X_iX_i} > T_{X_iX_j}$, we have

$$\begin{aligned}
\frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} Z_i' T Z_j &= \frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} T_{X_iX_j} && \text{(D.116)}\\
&< \frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} T_{X_iX_i} && \text{(D.117)}\\
&= \frac{c-1}{n-1}\sum_{i\in\mathscr{C}} T_{X_iX_i} && \text{(D.118)}\\
&= \frac{c-1}{n-1}\sum_{i\in\mathscr{C}} X_i' \cdot diag(T) && \text{(D.119)}\\
&= \overline{\mathbf{X}_{\mathscr{C}}}' \cdot \frac{c(c-1)}{n-1} diag(T). && \text{(D.120)}
\end{aligned}$$

Conditioned on the players in $\mathscr{C}$ all actively verify and observe $X_{\mathscr{C}}$, their observation costs are fixed, and their expected total reward is:

$$\begin{aligned}
r(\mathbf{X}_{\mathscr{C}}) = \overline{\mathbf{X}_{\mathscr{C}}}' \cdot \frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{X}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{X}_{-\mathscr{C}}}\right]\\
+ \frac{1}{n-1}\sum_{i\in\mathscr{C}}\sum_{j\in\mathscr{C}\setminus\{i\}} T_{X_iX_j}. \qquad \text{(D.121)}
\end{aligned}$$

On the other hand, we let $k = |S|$ be the number of types and define $f : \mathbb{R}^k \to \mathbb{R}$ s.t.

$$\begin{aligned}
f(Y) = Y' \cdot \frac{c(n-c)}{n-1}\mathbb{E}_{\mathbf{X}_{-\mathscr{C}}\sim P(\mathbf{X}_{-\mathscr{C}}|\mathbf{X}_{\mathscr{C}})}\left[T\overline{\mathbf{X}_{-\mathscr{C}}}\right]\\
+ Y' \cdot \frac{c(c-1)}{n-1} diag(T). \qquad \text{(D.122)}
\end{aligned}$$

From the arguments above, it holds that

$$r(\mathbf{X}_{\mathscr{C}}) < f(\overline{\mathbf{X}_{\mathscr{C}}}). \tag{D.123}$$

Since $f(\cdot)$ is a linear function, we have

$$f(\overline{\mathbf{X}_{\mathscr{C}}}) \leq \max_{i \in \mathscr{C}} f(X_i). \tag{D.124}$$

Let $i^* = \arg\max_{i \in \mathscr{C}} f(X_i)$, then we deduce that

$$r(\mathbf{X}_{\mathscr{C}}) < f(X_{i^*}). \tag{D.125}$$

Therefore, for the all-same report $\mathbf{Z}^*_{\mathscr{C}} = \{X_{i^*}, \cdots, X_{i^*}\}$, we have

$$\begin{aligned}
r(\mathbf{Z}^*_{\mathscr{C}}) &= \overline{\mathbf{Z}^*_{\mathscr{C}}}' \cdot \frac{c(n-c)}{n-1} \mathbb{E}_{\mathbf{X}_{-\mathscr{C}} \sim P(\mathbf{X}_{-\mathscr{C}} | \mathbf{X}_{\mathscr{C}})} \left[T \overline{\mathbf{X}_{-\mathscr{C}}} \right] + \frac{1}{n-1} \sum_{i \in \mathscr{C}} \sum_{j \in \mathscr{C} \setminus \{i\}} T_{Z^*_i Z^*_j} && \text{(D.126)} \\
&= X'_{i^*} \cdot \frac{c(n-c)}{n-1} \mathbb{E}_{\mathbf{X}_{-\mathscr{C}} \sim P(\mathbf{X}_{-\mathscr{C}} | \mathbf{X}_{\mathscr{C}})} \left[T \overline{\mathbf{X}_{-\mathscr{C}}} \right] + \frac{1}{n-1} \sum_{i \in \mathscr{C}} \sum_{j \in \mathscr{C} \setminus \{i\}} T_{X_{i^*} X_{i^*}} && \text{(D.127)} \\
&= X'_{i^*} \cdot \frac{c(n-c)}{n-1} \mathbb{E}_{\mathbf{X}_{-\mathscr{C}} \sim P(\mathbf{X}_{-\mathscr{C}} | \mathbf{X}_{\mathscr{C}})} \left[T \overline{\mathbf{X}_{-\mathscr{C}}} \right] + X'_{i^*} \cdot \frac{c(c-1)}{n-1} diag(T) && \text{(D.128)} \\
&= f(X_{i^*}) && \text{(D.129)} \\
&> r(\mathbf{X}_{\mathscr{C}}). && \text{(D.130)}
\end{aligned}$$

Hence we show that such collusion is strictly profitable in the shared-belief setting, implying that the mechanism is not strong-SCP.

If we do not adopt the assumption that $\exists i, j \in \mathscr{C}$ s.t. $T_{X_i X_i} > T_{X_i X_j}$, then Eq. (D.117) still holds with "$\leq$", and all the following strict inequalities become non-strict, implying that Eq. (D.2) holds with $h \leq 0$.

REFERENCES

[1] Robert A Heinlein. *Time enough for love*. 1973.

[2] Zishuo Zhao, Xi Chen, Xuefeng Zhang, and Yuan Zhou. Dynamic car dispatching and pricing: Revenue and fairness for ridesharing platforms. *arXiv preprint arXiv:2207.06318*, 2022.

[3] Xi Chen, David Simchi-Levi, Zishuo Zhao, and Yuan Zhou. Bayesian mechanism design for blockchain transaction fee allocation. *Operations Research*, 2025.

[4] Zishuo Zhao, Zhixuan Fang, Xuechao Wang, Xi Chen, Hongxu Su, Haibo Xiao, and Yuan Zhou. Proof-of-learning with incentive security. *arXiv preprint arXiv:2404.09005*, 2024.

[5] Zishuo Zhao, Xi Chen, and Yuan Zhou. It takes two: A peer-prediction solution for blockchain verifier's dilemma. *arXiv preprint arXiv:2406.01794*, 2024.

[6] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961. doi: https://doi.org/10.1111/j.1540-6261.1961.tb02789.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1961.tb02789.x`.

[7] Alex Gershkov, Jacob K Goeree, Alexey Kushnir, Benny Moldovanu, and Xianwen Shi. On the equivalence of bayesian and dominant strategy implementation. *Econometrica*, 81(1):197–220, 2013.

[8] Yuqing Kong and Grant Schoenebeck. An information theoretic framework for designing information elicitation mechanisms that reward truth-telling. *ACM Transactions on Economics and Computation (TEAC)*, 7(1):1–33, 2019.

[9] Yiling Chen, Yiheng Shen, and Shuran Zheng. Truthful data acquisition via peer prediction. *Advances in Neural Information Processing Systems*, 33:18194–18204, 2020.

[10] Yuqing Kong. Dominantly truthful peer prediction mechanisms with a finite number of tasks. *Journal of the ACM*, 2023.

[11] Xiaohui Bei and Shengyu Zhang. Algorithms for trip-vehicle assignment in ride-sharing. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[12] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. WWW '19, 2019.

[13] T. Qin, X. Tang, Y. Jiao, F. Zhang, and J. Ye. Ride-hailing order dispatching at didi via reinforcement learning. *Interface*, 50(5):272–286, 2020.

[14] Zhaodong Wang, Zhiwei Qin, Xiaocheng Tang, Jieping Ye, and Hongtu Zhu. Deep reinforcement learning with knowledge transfer for online rides order dispatching. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 617–626. IEEE, 2018.

[15] Jan Hrncir, Michael Rovatsos, and Michal Jakob. Ridesharing on timetabled transport services: A multiagent planning approach. *Journal of Intelligent Transportation Systems*, 19(1):89–105, 2015.

[16] Carlos Riquelme, Siddhartha Banerjee, and Ramesh Johari. Pricing in ride-sharing platforms: A queueing-theoretic approach. In *ACM Conference on Economics and Computation*, page 639, 2015.

[17] Juan Camilo Castillo, Dan Knoepfle, and Glen Weyl. Surge pricing solves the wild goose chase. In *ACM Conference on Economics and Computation*, pages 241–242, 2017.

[18] K. Bimpikis, O. Candogan, and D. Saban. Spatial pricing in ride-sharing networks. *Operations Research*, 67, 2019.

[19] Chiwei Yan, Helin Zhu, Nikita Korolko, and Dawn Woodard. Dynamic pricing and matching in ride-hailing platforms. *Naval Research Logistics, Forthcoming*, 2018.

[20] Hongyao Ma, Fei Fang, and David C. Parkes. Spatio-temporal pricing for ridesharing platforms. *SIGecom Exch.*, 18(2):53–57, December 2020.

[21] Éva Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.

[22] Maiwenn J. Al, Talitha L. Feenstra, and Ben A. van Hout. Optimal allocation of resources over health care programmes: dealing with decreasing marginal utility and uncertainty. *Health Economics*, 14(7):655–667, 2005.

[23] Tongtiegang Zhao, Jianshi Zhao, Pan Liu, and Xiaohui Lei. Evaluating the marginal utility principle for long-term hydropower scheduling. *Energy Conversion and Management*, 106:213–223, 2015.

[24] Jian Wang and Yi Zhang. Utilizing marginal net utility for recommendation in e-commerce. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1003–1012, 2011.

[25] Shuchi Chawla, Jason D Hartline, and Robert Kleinberg. Algorithmic pricing via virtual valuations. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 243–251, 2007.

[26] Jacob Glazer and Ariel Rubinstein. An extensive game as a guide for solving a normal game. *journal of economic theory*, 70(1):32–42, 1996.

[27] Didi Chuxing. The GAIA Public Dataset, 2021. data retrieved from `https://outreach.didichuxing.com/research/opendata/en/`.

[28] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, page 938–942, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi: 10.1145/3313276.3316303. URL `https://doi.org/10.1145/3313276.3316303`.

[29] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 910–918. IEEE, 2020.

[30] Tetiana Davydiuk, Deeksha Gupta, and Samuel Rosen. De-crypto-ing signals in initial coin offerings: Evidence of rational token retention. *Management Science*, 2023.

[31] Garud Iyengar, Fahad Saleh, Jay Sethuraman, and Wenjun Wang. Economics of permissioned blockchain adoption. *Management Science*, 2022.

[32] Rizwan Manzoor, BS Sahay, and Sujeet Kumar Singh. Blockchain technology in supply chain management: an organizational theoretic overview and research agenda. *Annals of Operations Research*, pages 1–48, 2022.

[33] Amy Whitaker and Roman Kräussl. Fractional equity, blockchain, and the future of creative work. *Management Science*, 66(10):4594–4611, 2020.

[34] N Bora Keskin, Chenghuai Li, and Jing-Sheng Jeannette Song. The blockchain newsvendor: Value of freshness transparency and smart contracts. *Available at SSRN 3915358*, 2023.

[35] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. A survey on applications of game theory in blockchain. *arXiv preprint arXiv:1902.10865*, 2019.

[36] Guangju Wang, Jiheng Zhang, Guangyuan Zhang, and Jiahao He. Consensus mechanism design based on structured directed acyclic graphs. *arXiv*, 2019.

[37] Tim Roughgarden. Transaction fee mechanism design. *ACM SIGecom Exchanges*, 19 (1):52–55, 2021.

[38] Jiahao He, Guangyuan Zhang, Jiheng Zhang, and Rachel Q Zhang. Blockchain operations in the presence of security concerns. *Manufacturing & Service Operations Management*, 25(3):1117–1135, 2023.

[39] Matheus V. X. Ferreira and S. Matthew Weinberg. Credible, truthful, and two-round (optimal) auctions via cryptographic commitments. In *Proceedings of the 21st ACM Conference on Economics and Computation*. ACM, jul 2020. doi: 10.1145/3391403.3399495. URL `https://doi.org/10.1145%2F3391403.3399495`.

[40] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE symposium on security and privacy (SP)*, pages 910–927. IEEE, 2020.

[41] Hao Chung and Elaine Shi. Foundations of transaction fee mechanism design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3856–3899. SIAM, 2023.

[42] Tim Roughgarden. Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559. *arXiv preprint arXiv:2012.00854*, 2020.

[43] Yotam Gafni and Aviv Yaish. Greedy transaction fee mechanisms for (non-)myopic miners. *CoRR*, abs/2210.07793, 2022. doi: 10.48550/arXiv.2210.07793. URL `https://doi.org/10.48550/arXiv.2210.07793`.

[44] Elaine Shi, Hao Chung, and Ke Wu. What can cryptography do for decentralized mechanism design? Cryptology ePrint Archive, Paper 2022/1294, 2022. URL `https://eprint.iacr.org/2022/1294`. `https://eprint.iacr.org/2022/1294`.

[45] Ke Wu, Elaine Shi, and Hao Chung. Maximizing miner revenue in transaction fee mechanism design. *Cryptology ePrint Archive*, 2023.

[46] Sankarshan Damle, Varul Srivastava, and Sujit Gujar. No transaction fees? no problem! achieving fairness in transaction fee mechanism design. *arXiv preprint arXiv:2402.04634*, 2024.

[47] Truman Bewley. The optimum quantity of money. Technical report, Discussion Paper, 1979.

[48] Mr Taimur Baig, Mr Jörg Decressin, Mr Tarhan Feyzioglu, Mr Manmohan S Kumar, and Mr Chris Faulkner-MacDonagh. *Deflation: determinants, risks, and policy options*. International Monetary Fund, 2003.

[49] Alejandro M Manelli and Daniel R Vincent. Bayesian and dominant-strategy implementation in the independent private-values model. *Econometrica*, 78(6):1905–1938, 2010.

[50] Andrew Chi-Chih Yao. An incentive analysis of some bitcoin fee designs. *arXiv preprint arXiv:1811.02351*, 2018.

[51] Ron Lavi, Or Sattath, and Aviv Zohar. Redesigning bitcoin's fee market. *ACM Transactions on Economics and Computation*, 10(1):1–31, 2022.

[52] Hao Chung, Tim Roughgarden, and Elaine Shi. Collusion-resilience in transaction fee mechanism design. *arXiv preprint arXiv:2402.09321*, 2024.

[53] Vilfredo Pareto. *Manuale di economia politica con una introduzione alla scienza sociale*, volume 13. Società editrice libraria, 1919.

[54] Matheus VX Ferreira, Daniel J Moroz, David C Parkes, and Mitchell Stern. Dynamic posted-price mechanisms for the blockchain transaction-fee market. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pages 86–99, 2021.

[55] Robert G Hammond. Comparing revenue from auctions and posted prices. *International Journal of Industrial Organization*, 28(1):1–9, 2010.

[56] Sebastien Bubeck, Nikhil R Devanur, Zhiyi Huang, and Rad Niazadeh. Online auctions and multi-scale online learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 497–514, 2017.

[57] Rose Fiamohe, Tebila Nakelse, Aliou Diagne, and Papa A Seck. Assessing the effect of consumer purchasing criteria for types of rice in togo: A choice modeling approach. *Agribusiness*, 31(3):433–452, 2015.

[58] Gabriel Bitran and René Caldentey. An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203–229, 2003.

[59] G. van Ryzin and S. Mahajan. On the relationships between inventory costs and variety benefits in retail assortments. *Management Science*, 45(11):1496–1509, 1999.

[60] S. Mahajan and G.J. van Ryzin. Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 49:334–351, 2001.

[61] Qian Liu and G van Ryzin. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10(2):288–310, 2008.

[62] Paat Rusmevichientong, Zuo-Jun Max Shen, and David B Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations research*, 58(6):1666–1680, 2010.

[63] James M Davis, Guillermo Gallego, and Huseyin Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014.

[64] Guang Li and Paat Rusmevichientong. A greedy algorithm for the two-level nested logit model. *Operations Research Letters*, 42(5):319–324, July 2014.

[65] Vivek F Farias, Srikanth Jagabathula, and Devavrat Shah. A Nonparametric Approach to Modeling Choice with Limited Data. *Management Science*, 59(2):305–322, February 2013.

[66] Jose Blanchet, Guillermo Gallego, and Vineet Goyal. A markov chain approximation to choice modeling. *Operations Research*, 64(4):886–905, 2016.

[67] Antoine Désir, Vineet Goyal, Danny Segev, and Chun Ye. Constrained assortment optimization under the markov chain–based choice model. *Management Science*, 66 (2):698–721, 2020.

[68] K. Train. *Discrete choice methods with simulation*. Cambridge University Press, 2nd edition, 2009.

[69] Zhiyi Huang and Sampath Kannan. The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 140–149. IEEE, 2012.

[70] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1): 58–73, 1981.

[71] Roger B Myerson. Incentive compatibility and the bargaining problem. *Econometrica: journal of the Econometric Society*, pages 61–73, 1979.

[72] Péter Esö and Gabor Futo. Auction design with a risk averse seller. *Economics Letters*, 65(1):71–74, 1999.

[73] Chinmay Maheshwari, Kshitij Kulkarni, Manxi Wu, and S Shankar Sastry. Inducing social optimality in games via adaptive incentive design. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2864–2869. IEEE, 2022.

[74] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383, 2014.

[75] Anna Ben-Hamou, Yuval Peres, and Justin Salez. Weighted sampling without replacement. *Brazilian Journal of Probability and Statistics*, 32(3):657–669, 2018.

[76] Gagan Aggarwal and Jason D. Hartline. Knapsack auctions. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1083–1092, 2006. URL `http://doi.acm.org/10.1145/1109557.1109677`.

[77] Alon Eden, Kira Goldner, and Shuran Zheng. Private interdependent valuations. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2920–2939. SIAM, 2022.

[78] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 2008.

[79] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[80] Markus Jakobsson and Ari Juels. Proofs of work and bread pudding protocols. In *Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium*, pages 258–272. Springer, 1999.

[81] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 3–16, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978341. URL `https://doi.org/10.1145/2976749.2978341`.

[82] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16, 2016.

[83] Aggelos Kiayias, Andrew Miller, and Dionysis Zindros. Non-interactive proofs of proof-of-work. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10–14, 2020 Revised Selected Papers 24*, pages 505–522. Springer, 2020.

[84] Harald Vranken. Sustainability of bitcoin and blockchains. *Current opinion in environmental sustainability*, 28:1–9, 2017.

[85] Christian Stoll, Lena Klaas,s'en, and Ulrich Gallersdörfer. The carbon footprint of bitcoin. *Joule*, 3(7):1647–1661, 2019.

[86] John Riley. The current status of cryptocurrency regulation in china and its effect around the world. *China and WTO Review*, 7(1):135–152, 2021.

[87] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual international cryptology conference*, pages 357–388. Springer, 2017.

[88] Chaya Ganesh, Claudio Orlandi, and Daniel Tschudi. Proof-of-stake protocols for privacy-aware blockchains. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 690–719, Cham, 2019. Springer International Publishing. ISBN 978-3-030-17653-2.

[89] Fahad Saleh. Blockchain without Waste: Proof-of-Stake. *The Review of Financial Studies*, 34(3):1156–1190, 07 2020. ISSN 0893-9454. doi: 10.1093/rfs/hhaa075. URL `https://doi.org/10.1093/rfs/hhaa075`.

[90] Vivek Bagaria, Amir Dembo, Sreeram Kannan, Sewoong Oh, David Tse, Pramod Viswanath, Xuechao Wang, and Ofer Zeitouni. Proof-of-stake longest chain protocols: Security vs predictability. In *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, pages 29–42, 2022.

[91] Thomas Piketty. *Capital in the twenty-first century*. Harvard University Press, 2014.

[92] Felix Hoffmann. Challenges of proof-of-useful-work (pouw). In *2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & Beyond (iGETblockchain)*, pages 1–5. IEEE, 2022.

[93] Ambre Toulemonde, Loic Besson, Louis Goubin, and Jacques Patarin. Useful work: a new protocol to ensure usefulness of pow-based consensus for blockchain. In *Proceedings of the 2022 ACM Conference on Information Technology for Social Good*, GoodIT '22, page 308–314, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392846. doi: 10.1145/3524458.3547248. URL `https://doi.org/10.1145/3524458.3547248`.

[94] Juan Ignacio Ibañez and Alexander Freier. Bitcoin's carbon footprint revisited: Proof of work mining for renewable energy expansion. *Challenges*, 14(3):35, 2023.

[95] Kaiwen Zheng, Shulai Zhang, and Xiaoli Ma. Difficulty prediction for proof-of-work based blockchains. In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2020.

[96] Hengrui Jia, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1039–1056. IEEE, 2021.

[97] Alejandro Baldominos and Yago Saez. Coin. ai: A proof-of-useful-work scheme for blockchain-based distributed deep learning. *Entropy*, 21(8):723, 2019.

[98] Yuan Liu, Yixiao Lan, Boyang Li, Chunyan Miao, and Zhihong Tian. Proof of learning (pole): empowering neural network training with consensus building on blockchains. *Computer Networks*, 201:108594, 2021.

[99] Felipe Bravo-Marquez, Steve Reeves, and Martin Ugarte. Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions. In *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pages 119–124. IEEE, 2019.

[100] Congyu Fang, Hengrui Jia, Anvith Thudi, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning is currently more broken than you think. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 797–816. IEEE, 2023.

[101] Rui Zhang, Jian Liu, Yuan Ding, Zhibo Wang, Qingbiao Wu, and Kui Ren. "adversarial examples" for proof-of-learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1408–1422. IEEE, 2022.

[102] Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, et al. Managing extreme ai risks amid rapid progress. *Science*, 384(6698):842–845, 2024.

[103] Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Danielle Goldfarb, Hoda Heidari, Leila Khalatbari, et al. International scientific report on the safety of advanced ai (interim report). *arXiv preprint arXiv:2412.05282*, 2024.

[104] Dan Hendrycks. *Introduction to AI safety, ethics and society*. Dan Hendrycks, 2024.

[105] Richard Ren, Steven Basart, Adam Khoja, Alice Gatti, Long Phan, Xuwang Yin, Mantas Mazeika, Alexander Pan, Gabriel Mukobi, Ryan H Kim, et al. Safetywashing: Do ai safety benchmarks actually measure safety progress? *arXiv preprint arXiv:2407.21792*, 2024.

[106] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

[107] Erez Firt. Calibrating machine behavior: a challenge for ai alignment. *Ethics and Information Technology*, 25(3):42, 2023.

[108] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *arXiv preprint arXiv:2402.18540*, 2024.

[109] Reuters. Bytedance seeks $1.1 mln damages from intern in ai breach case, report says. 2024. URL `https://www.reuters.com/technology/artificial-intelligence/bytedance-seeks-11-mln-damages-intern-ai-breach-case-report-says-2024-11-28/`. November 28, 2024.

[110] Tiffany Wenting Li, Silas Hsu, Max Fowler, Zhilin Zhang, Craig Zilles, and Karrie Karahalios. Am i wrong, or is the autograder wrong? effects of ai grading mistakes on learning. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*, pages 159–176, 2023.

[111] Bojan B Tomić, Anisja D Kijevcanin, Zoran V Sevarac, and Jelena M Jovanović. An ai-based approach for grading students' collaboration. *IEEE Transactions on Learning Technologies*, 16(3):292–305, 2022.

[112] Siddharth Bhatore, Lalit Mohan, and Y Raghu Reddy. Machine learning techniques for credit risk evaluation: a systematic literature review. *Journal of Banking and Financial Technology*, 4(1):111–138, 2020.

[113] Katja Langenbucher. Ai credit scoring and evaluation of creditworthiness–a test case for the eu proposal for an ai act. *how the challenges of today prepare the ground for tomorrow*, page 362, 2022.

[114] Canhui Chen, Zerui Cheng, Shutong Qu, and Zhixuan Fang. Crowdsourcing work as mining: A decentralized computation and storage paradigm. *arXiv preprint arXiv:2211.06669*, 2022.

[115] Sunny King. Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th*, 1(6), 2013.

[116] Atalay M Ileri, Halil I Ozercan, Alper Gundogdu, Ahmet K Senol, M Yusuf Ozkaya, and Can Alkan. Coinami: a cryptocurrency with dna sequence alignment as proof-of-work. *arXiv preprint arXiv:1602.03031*, 2016.

[117] Charles AE Goodhart. *Problems of monetary management: the UK experience*. Springer, 1984.

[118] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of useful work. *Cryptology ePrint Archive*, 2017.

[119] Elisa Bertino, Murat Kantarcioglu, Cuneyt Gurcan Akcora, Sagar Samtani, Sudip Mittal, and Maanak Gupta. Ai for security and security for ai. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 333–334, 2021.

[120] Yupeng Hu, Wenxin Kuang, Zheng Qin, Kenli Li, Jiliang Zhang, Yansong Gao, Wenjia Li, and Keqin Li. Artificial intelligence security: Threats and countermeasures. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[121] Jayden Khakurel, Birgit Penzenstadler, Jari Porras, Antti Knutas, and Wenlu Zhang. The rise of artificial intelligence under the lens of sustainability. *Technologies*, 6(4): 100, 2018.

[122] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 896–902. IEEE, 2015.

[123] Nuno Antunes, Leandro Balby, Flavio Figueiredo, Nuno Lourenco, Wagner Meira, and Walter Santos. Fairness and transparency of machine learning for trustworthy cloud services. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 188–193. IEEE, 2018.

[124] Warren J Von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622, 2021.

[125] Adnan Qayyum, Aneeqa Ijaz, Muhammad Usama, Waleed Iqbal, Junaid Qadir, Yehia Elkhatib, and Ala Al-Fuqaha. Securing machine learning in the cloud: A systematic review of cloud machine learning security. *Frontiers in big Data*, 3:587139, 2020.

[126] Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durresi. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)*, 55(2):1–38, 2022.

[127] Mohamed Nassar, Khaled Salah, Muhammad Habib ur Rehman, and Davor Svetinovic. Blockchain for explainable and trustworthy artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(1):e1340, 2020.

[128] Jei Young Lee. A decentralized token economy: How blockchain and cryptocurrency can revolutionize business. *Business Horizons*, 62(6):773–784, 2019.

[129] Tao Lu, Haoyu Wang, Wenjie Qu, Zonghui Wang, Jinye He, Tianyang Tao, Wenzhi Chen, and Jiaheng Zhang. An efficient and extensible zero-knowledge proof framework for neural networks. *Cryptology ePrint Archive*, 2024.

[130] KD Conway, Cathie So, Xiaohang Yu, and Kartin Wong. opml: Optimistic machine learning on blockchain. *arXiv preprint arXiv:2401.17555*, 2024.

[131] Yue Zhang, Shouqiao Wang, Xiaoyuan Liu, Sijun Tan, Raluca Ada Popa, and Ciamac C Moallemi. Proof of sampling: A nash equilibrium-secured verification protocol for decentralized systems. *arXiv preprint arXiv:2405.00295*, 2024.

[132] John Kelsey, Bruce Schneier, David Wagner, and Chris Hall. Cryptanalytic attacks on pseudorandom number generators. In *International workshop on fast software encryption*, pages 168–188. Springer, 1998.

[133] Daria Smuseva, Ivan Malakhov, Andrea Marin, Aad van Moorsel, and Sabina Rossi. Verifier's dilemma in ethereum blockchain: A quantitative analysis. In *International Conference on Quantitative Evaluation of Systems*, pages 317–336. Springer, 2022.

[134] Beltrán Borja Fiz Pontiveros, Christof Ferreira Torres, and Radu State. Sluggish mining: Profiting from the verifier's dilemma. In *Financial Cryptography and Data Security: FC 2019 International Workshops, VOTING and WTSC, St. Kitts, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23*, pages 67–81. Springer, 2020.

[135] Maxime Reynouard, Rida Laraki, and Olga Gorelkina. Bar nash equilibrium and application to blockchain design. *arXiv preprint arXiv:2401.16856*, 2024.

[136] Jason Teutsch and Christian Reitwies,s'ner. A scalable verification solution for blockchains. In *ASPECTS OF COMPUTATION AND AUTOMATA THEORY WITH APPLICATIONS*, pages 377–424. World Scientific, 2024.

[137] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying incentives in the consensus computer. In *Proceedings of the 22Nd acm sigsac conference on computer and communications security*, pages 706–719, 2015.

[138] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[139] Yiling Chen, Yiheng Shen, and Shuran Zheng. Truthful data acquisition via peer prediction. *Advances in Neural Information Processing Systems*, 33:18194–18204, 2020.

[140] Shengling Wang, Xidi Qu, Qin Hu, Xia Wang, and Xiuzhen Cheng. An uncertainty-and collusion-proof voting consensus mechanism in blockchain. *IEEE/ACM Transactions on Networking*, 31(5):2376–2388, 2023.

[141] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

[142] Zhenyu Zhang, Ajay Jaiswal, Lu Yin, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. Q-galore: Quantized galore with int4 projection and layer-adaptive low-rank gradients. *arXiv preprint arXiv:2407.08296*, 2024.

[143] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[144] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. Layer 2 blockchain scaling: a survey, 2021.

[145] Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041*, 2018.

[146] Xueping Gong, Qing Zhang, Huizhong Li, and Jiheng Zhang. Improving blockchain consistency bound by assigning weights to random blocks. *Operations Research*, 0 (0):null, 2024. doi: 10.1287/opre.2022.0463. URL `https://doi.org/10.1287/opre.2022.0463`.

[147] Rosanna Cole, Mark Stevenson, and James Aitken. Blockchain technology: implications for operations and supply chain management. *Supply chain management: An international journal*, 24(4):469–483, 2019.

[148] Yao Cui, Vishal Gaur, and Jingchen Liu. Supply chain transparency and blockchain design. *Management Science*, 70(5):3245–3263, 2024.

[149] Zhipeng Wang, Rui Sun, Elizabeth Lui, Vatsal Shah, Xihan Xiong, Jiahao Sun, Davide Crapis, and William Knottenbelt. Sok: Decentralized ai (deai). *arXiv preprint arXiv:2411.17461*, 2024.

[150] Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. Zkml: An optimizing system for ml inference in zero-knowledge proofs. In *Proceedings of the Nineteenth European Conference on Computer Systems*, pages 560–574, 2024.

[151] Albrecher Hansjoerg and Goffard Pierre-Olivier. On the profitability of selfish blockchain mining under consideration of ruin. *Operations Research*, 70(1):179–200, 2022.

[152] Chien-Chih Chen and Wojciech Golab. A game theoretic analysis of validator strategies in ethereum 2.0. In *Proceedings of the 6th ACM International Symposium on Blockchain and Secure Critical Infrastructure*, pages 1–14, 2024.

[153] Mikhail Kalinin, Danny Ryan, and Vitalik Buterin. Eip-3675: Upgrade consensus to proof-of-stake, Jul 2021. URL `https://eips.ethereum.org/EIPS/eip-3675`.

[154] Daria Smuseva, Andrea Marin, Sabina Rossi, and Aad Van Moorsel. Verifier's dilemma in proof-of-work public blockchains: A quantitative analysis. *ACM Transactions on Modeling and Computer Simulation*, 35(2):1–24, 2025.

[155] Peiyao Sheng, Ranvir Rana, Senthil Bala, Himanshu Tyagi, and Pramod Viswanath. Proof of diligence: Cryptoeconomic security for rollups. *arXiv preprint arXiv:2402.07241*, 2024.

[156] Mengfan Xu and Diego Klabjan. Decentralized blockchain-based robust multi-agent multi-armed bandit. *arXiv preprint arXiv:2402.04417*, 2024.

[157] Shahinaz Kamal Ezzat, Yasmine NM Saleh, and Ayman A Abdel-Hamid. Blockchain oracles: State-of-the-art and research directions. *IEEE Access*, 10:67551–67572, 2022.

[158] Behkish Nassirzadeh, Stefanos Leonardos, Albert Heinle, Anwar Hasan, and Vijay Ganesh. Countchain: A decentralized oracle network for counting systems. *arXiv preprint arXiv:2409.11592*, 2024.

[159] Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, pages 319–330, 2013.

[160] Yuxi Cai, Georgios Fragkos, Eirini Eleni Tsiropoulou, and Andreas Veneris. A truth-inducing sybil resistant decentralized blockchain oracle. In *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 128–135. IEEE, 2020.

[161] Shuran Zheng, Xuan Qi, Rui Ray Chen, Yongchan Kwon, and James Zou. Proper dataset valuation by pointwise mutual information. *arXiv preprint arXiv:2405.18253*, 2024.

[162] Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C Parkes. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 179–196, 2016.

[163] Jacques Crémer and Richard P McLean. Full extraction of the surplus in bayesian and dominant strategy auctions. *Econometrica: Journal of the Econometric Society*, pages 1247–1257, 1988.

[164] Tong Cao, Jérémie Decouchant, and Jiangshan Yu. Leveraging the verifier's dilemma to double spend in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 149–165. Springer, 2023.

[165] Maher Alharby, Roben Castagna Lunardi, Amjad Aldweesh, and Aad Van Moorsel. Data-driven model-based analysis of the ethereum verifier's dilemma. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 209–220. IEEE, 2020.

[166] Shafaq Naheed Khan, Faiza Loukil, Chirine Ghedira-Guegan, Elhadj Benkhelifa, and Anoud Bani-Hani. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications*, 14:2901–2925, 2021.

[167] Lennart Ante. Smart contracts on the blockchain–a bibliometric analysis and review. *Telematics and Informatics*, 57:101519, 2021.

[168] Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, and Fei-Yue Wang. Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2266–2277, 2019.

[169] Hanzaleh Akbari Nodehi, Viveck R Cadambe, and Mohammad Ali Maddah-Ali. Game of coding: Beyond trusted majorities. *arXiv preprint arXiv:2401.16643*, 2024.

[170] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pages 5650–5659. Pmlr, 2018.

[171] Zhaoxian Wu, Tianyi Chen, and Qing Ling. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. *IEEE transactions on signal processing*, 2023.

[172] Ruiliang Chen, Jung-Min Jerry Park, and Kaigui Bian. Robustness against byzantine failures in distributed spectrum sensing. *Computer Communications*, 35(17):2115–2124, 2012.

[173] Grant Schoenebeck, Fang-Yi Yu, and Yichi Zhang. Information elicitation from rowdy crowds. In *Proceedings of the Web Conference 2021*, pages 3974–3986, 2021.

[174] Farhad Soleimanian Gharehchopogh and Hassan Arjang. A survey and taxonomy of leader election algorithms in distributed systems. *Indian journal of science and technology*, 7(6):815, 2014.

[175] Rafael Frongillo and Jens Witkowski. A geometric perspective on minimal peer prediction. *ACM Transactions on Economics and Computation (TEAC)*, 5(3):1–27, 2017.

[176] Daniel Kuhn, Soroosh Shafiee, and Wolfram Wiesemann. Distributionally robust optimization, 2025. URL `https://arxiv.org/abs/2411.02549`.

[177] Shuran Zheng, Fang-Yi Yu, and Yiling Chen. The limits of multi-task peer prediction. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 907–926, 2021.

[178] Raja Siddharth Raju, Sandeep Gurung, and Prativa Rai. An overview of 51% attack over bitcoin network. *Contemporary Issues in Communication, Cloud and Big Data Analytics: Proceedings of CCB 2020*, pages 39–55, 2022.

[179] Shengwei Xu, Yichi Zhang, Paul Resnick, and Grant Schoenebeck. Spot check equivalence: An interpretable metric for information elicitation mechanisms. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 276–287, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645679. URL `https://doi.org/10.1145/3589334.3645679`.

[180] Yichi Zhang, Shengwei Xu, David Pennock, and Grant Schoenebeck. Stochastically dominant peer prediction. *arXiv preprint arXiv:2506.02259*, 2025.

[181] Ke Wang, Zishuo Zhao, Xinyuan Song, Bill Shi, Libin Xia, Chris Tong, Lynn Ai, Felix Qu, and Eric Yang. Verillm: A lightweight framework for publicly verifiable decentralized inference. *arXiv preprint arXiv:2509.24257*, 2025.

[182] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. 1972.

[183] Albert Xin Jiang and Kevin Leyton-Brown. Estimating bidders' valuation distributions in online auctions. In *Proceedings of IJCAI-05 workshop on game theoretic and decision theoretic agents*, pages 98–107, 2005.

[184] Wen Zhao and Yu-Sheng Zheng. Optimal dynamic pricing for perishable assets with nonhomogenous demand. *Management Science*, 46:375–388, 03 2000.

[185] G. Gallego and G. Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40:999–1020, 1994.

[186] Omar Besbes and Assaf Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57:1407–1420, 12 2009.

[187] Ravi Kashyap. Auction theory adaptations for real life applications. *Research in Economics*, 72(4):452–481, 2018.

[188] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[189] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

[190] Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in neural information processing systems*, pages 1297–1305, 2015.

[191] Eric M Schwartz, Eric T Bradlow, and Peter S Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

[192] Jian Wang, William M Wells, Polina Golland, and Miaomiao Zhang. Efficient Laplace approximation for bayesian registration uncertainty quantification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 880–888. Springer, 2018.

[193] Peter D Lax and Maria Shea Terrell. *Calculus with applications*. Springer, 2014.

[194] Elisa Bertino, Ahish Kundu, and Zehra Sura. Data transparency with blockchain and ai ethics. *Journal of Data and Information Quality (JDIQ)*, 11(4):1–8, 2019.

[195] Andriy Luntovskyy and Dietbert Guetter. Cryptographic technology blockchain and its applications. In *The International Conference on Information and Telecommunication Technologies and Radio Electronics*, pages 14–33. Springer, 2018.

[196] Franck Cassez, Joanne Fuller, and Aditya Asgaonkar. Formal verification of the ethereum 2.0 beacon chain. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 167–182. Springer, 2022.

[197] Merve Can Kus Khalilov and Albert Levi. A survey on anonymity and privacy in bitcoin-like digital cash systems. *IEEE Communications Surveys & Tutorials*, 20(3): 2543–2585, 2018.

[198] E Connell and J Drost. Conservative and divergence free algebraic vector fields. *Proceedings of the American Mathematical Society*, 87(4):607–612, 1983.

[199] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[200] Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024.

[201] Jaymari Chua, Yun Li, Shiyi Yang, Chen Wang, and Lina Yao. Ai safety in generative ai large language models: A survey. *arXiv preprint arXiv:2407.18369*, 2024.

[202] Ahmed M Shamsan Saleh. Blockchain for secure and decentralized artificial intelligence in cybersecurity: A comprehensive review. *Blockchain: Research and Applications*, 5 (3):100193, 2024.

[203] Matthew Armstrong. Ethereum, smart contracts and the optimistic roll-up, 2021.