It Takes Two: A Peer-Prediction Solution for Blockchain Verifier's Dilemma

ZISHUO ZHAO, University of Illinois Urbana-Champaign, USA
XI CHEN, New York University, USA
YUAN ZHOU, Tsinghua University, China

The security of blockchain systems is fundamentally based on the decentralized consensus in which the majority of parties behave honestly, and the content verification process is essential to maintaining the robustness of blockchain systems. However, the phenomenon that a secure blockchain system with few or no cheaters could not provide sufficient incentive for verifiers to honestly perform the costly verification, referred to as the Verifier's Dilemma, could severely undermine the fundamental security of blockchain systems. While existing works have attempted to insert deliberate errors to disincentivize lazy verification, the decentralized environment renders it impossible to judge the correctness of verification or detect malicious verifiers directly without additional layers of procedures, e.g., reputation systems or additional committee voting.

In this paper, we initiate the research that leverages the peer prediction approach towards the design of one-phase Bayesian truthful mechanisms for the *decentralized verification games* among multiple verifiers, incentivizing all verifiers to perform honest verification without access to the ground truth even in the presence of noisy observations in the verification process. Furthermore, we optimize our mechanism to realize provable robustness against collusions and other malicious behavior of the verifiers. With the theoretically guaranteed Byzantine-robust incentive properties of our mechanism, our study provides a framework of decentralized verification protocols that enhances the security and robustness of the blockchain and potentially other decentralized systems.

Contents

1 INTRODUCTION

Blockchain, with prevailing examples as Bitcoin [Nakamoto, 2008] and Ethereum [Buterin et al., 2014], is an emerging technology that maintains decentralized consensus via a distributed ledger that utilizes cryptographic techniques to achieve trust and security. To ensure that the blockchain runs correctly, certain contents on the blockchain need to be "proven" and verified by other parties. The "proofs" can either be developed cryptographically, e.g., via zero-knowledge proofs [Goldreich and Oren, 1994], or in a game-theoretical way, e.g., via *verification games.* While the method of zero-knowledge proofs achieves cryptographical security, its high computational overhead limits its application in large-scale tasks, e.g. machine learning [Chen et al., 2024].

In several previous studies on blockchain verification (e.g. Ileri et al. [2016], Jia et al. [2021]), it is typical that the system only prevents the provers from cheating while assuming that verifiers are honest. However, in a fully decentralized and permissionless blockchain system, this is not necessarily true, especially in verification games in which the incentive compatibility is not theoretically guaranteed.

When considering game-theoretic ways to incentivize verifiers to verify honestly, a straightforward way is to introduce a penalty ("slash") when a verifier is detected to be dishonest, as in Ethereum [Cassez et al., 2022], opML [Conway et al., 2024], etc. Nevertheless, a simple "slashing" mechanism, in which a verifier is penalized for a constant amount when the verifier is detected to have passed the verification of an invalid proof, has a two-fold challenge as follows. The first one is that in a decentralized environment, there is not a trusted "root" that makes the decision, so the decision needs to be performed in a voting protocol, but it is hard to ensure the honesty of voters with theoretical guarantees; the second one is the Verifier's Dilemma [Luu et al., 2015]:

Verifier's Dilemma:

- If a mechanism is (incentive-)secure against strategic provers, 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 report "Success" without actual verification.
- If all verifiers are rational and would not actually verify, then the security properties no longer hold.

Formally, the Verifier's Dilemma can be formulated as the following theorem. The proof is deferred to Appendix A.

Theorem 1.1 (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 a verification mechanism with a pure-strategy Nash equilibrium that the prover and verifier(s) simultaneously act honestly.

The Verifier's Dilemma only occurs in the scenario of ϵ, the probability of cheating, tends to zero, so that any bounded reward for catching a cheat cannot uniformly compensate for the verification cost in expectation. Hence, a conventional penalty-based approach can only lower, but not eliminate, the rate of cheating in the pool of proofs. For example, the study of [Conway et al., 2024] shows that the opML mechanism achieves a mixed-strategy Nash equilibrium that the prover has a constant probability to cheat and the verifier has a constant probability to honestly verify.

To achieve an honest pure-strategy Nash equilibrium, a natural idea is to introduce "attention challenges" that contains extra information or deliberate errors, e.g. insert deliberate objects (which can be valid or invalid), as so-called "flags" or to incentivize verifiers to find and report, as in the works of [Luu et al., 2015, Reynouard et al., 2024, Teutsch and Reitwießner, 2024, Zhao et al., 2024]. Besides, the flags may also appear from the internal uncertainties of the verification process, e.g., in ML inference [Nodehi et al., 2024]. The simple flag-insertion mechanism works well for "honest-but-lazy" verifiers, as long as the expected reward of finding the flags can cover the verification cost. However, in the scenario where the validity of the verification result is also costly to verify, e.g., Proof-of-Learning [Jia et al., 2021, Zhao et al., 2024], we generally need a sort of *authentication* or higher-level verifiers, or need additional phases of "committee voting" [Zhang et al., 2024] to decide on their validity and prevent dishonest verifiers from harming the system. As long as we want to design a fully decentralized blockchain with no trusted "root" parties, there is no way to access the "ground truth" and the validity and payments can only be decided by consensus among the voting parties.

In the field of computational economics, the concept of *peer prediction* [Miller et al., 2005] refers to a wide scope of mechanisms to elicit honest information when direct verification is unavailable, which is widely adopted in the applications of dataset acquisition [Chen et al., 2020], peer grading [Dasgupta and Ghosh, 2013], etc. Particularly, a recent work [Wang et al., 2023] also utilized the peer prediction approach for the voting consensus on blockchain platforms. A widely used practice for peer prediction is to design a mutual-information-based payment rule [Kong and Schoenebeck, 2019] and leverage the data processing inequality to make sure that any manipulation cannot increase the expected utility for an agent. While these types of scoring rules are convenient for usage in scenarios with known prior, this family of peer prediction mechanisms is still unable to directly resolve the Verifier's Dilemma because the infinitesimal cheating probability will make the logarithmic scoring rule diverge. In the practice of blockchain systems, as the Verifier's Dilemma leads to an arbitrarily low cheating probability, its empirical value becomes difficult to estimate and the scoring rule's sensitivity to small probabilities will severely undermine the robustness of the verification mechanism.

In contrast to most peer prediction mechanisms that guarantee Bayesian Nash equilibria requiring a known prior, a recent work [Kong, 2023] utilizes determinant-based mutual information (DMI) to design a dominantly-truthful finite-task peer prediction mechanism which has the potential to tackle with the scenario of unbalanced and unknown priors. However, although it does not need an estimation of ϵ, its need and assumption of multiple homogeneous tasks, i.e. at least 4 tasks sharing a common ϵ, may still lead to complication in the analysis as it has assumptions on and is still sensitive to the latent value of ϵ. Furthermore, the lack of permutation-proofness also renders the DMI mechanism impractical for the verification games: a verifier who flips all her reports, i.e., reporting "Dishonest" when observing "Honest" and vice versa, is genuinely malicious but still gets optimal utility. While we acknowledge the potential of the DMI mechanism to be adapted as a solution to the problem, possibly after well-designed modifications, we would still be motivated to design a conceptually "more robust" mechanism that is inherently permutation-proof and insensitive to ϵ as long as it is small enough.

In a game-theoretic interpretation, it is worth noting that the longest-chain rule proposed by Nakamoto [2008] essentially implements an *implicit* peer prediction mechanism for decentralized consensus. In the presence of disagreement ("forks"), miners follow and *vote* on the branch that they believe would be the main chain, and get block rewards if and only if they are agreed by the most resource (work, stake, etc.) While the protocol is designed for incentivizing honest actions via consensus, a miner's reward is indeed solely depending on other miners' votes without reference to the "ground truth" that a previous block is honest or not. Through the testimony of history, the success of longest-chain rule for blockchain has shown the practical value of peer prediction mechanisms in decentralized consensus. Besides, several current studies on incentivized consensus relying on votes to decide on the results and rewards for voters (e.g., Nassirzadeh et al. [2024], Zhang et al. [2024]), also represent the high-level principles of peer prediction; while they are likely to work in practice, these studies do not delve into rigorous economic analyses, or rely on ad-hoc assumptions (e.g., "the majority is the ground truth"), which may lack theoretical guarantees especially for the scenario when the verification process is stochastic, e.g., for ML training and inference. Hence, the study on the framework of peer prediction mechanisms, beyond the current paradigm of simple majority voting, is indeed essential to the theoretical reliability of incentivized consensus mechanisms for decentralized systems. On the other hand, the closed-form incentive guarantees can in turn back up the canonical assumption on honest majority via the rationality of agents.

While the longest-chain rule works as promised for both traditional Proof-of-Work (PoW) and Proof-of-Stake (PoS) mechanisms, as in Bitcoin and Ethereum respectively, it is based on the fact of low observation costs in the verification of PoW and PoS. In the scenarios in which verification is costly, particularly for the current demand for verification of machine learning models [Conway et al., 2024, Nodehi et al., 2024, Zhao et al., 2024], the Verifier's Dilemma motivates us to design a more robust **explicit** peer prediction mechanism for the design of reliable decentralized consensus mechanisms.

Compared to traditional peer prediction mechanisms which mainly consider incentive compatibility (IC), in our setting, we are additionally required to explicitly consider individual rationality (IR) so that an honest verifier would get an expected reward no less than the verification cost, and the verification cost may be dependent on the verification result. On the other hand, due to the permissionless property of the blockchain system, we also need a no-free-lunch (NFL) property, which is a stronger form of *informed truthfulness* [Shnayder et al., 2016], that any verifier who does not actually perform the verification cannot get a positive expected reward from the mechanism.

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 simultaneously satisfies the following properties:

- Interim *unique* incentive compatibility (interim UniIC): Given all other verifiers act honestly, 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): Given all other verifiers act honestly, a verifier, after performing the verification, gets a non-negative expected utility when she acts honestly.
- Interim no-free-lunch (interim NFL): Given all other verifiers act honestly, a verifier cannot get a positive expected utility via any *uninformed strategy* [Shnayder et al., 2016], i.e., without doing the verification.

Combining all the desirable 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 deterministic strategy gains a positive interim utility if and only if it is honest, with a margin of δ (details in Section 3.3). 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.

In this study, we first consider a simple case of 2-verifier DVG, modeling the design of a δ-IA mechanism as a linear program, and show the general feasibility of this problem for all non-degenerate cases. Furthermore, we extend our methodology for general n-verifier DVG and derive the Byzantine robustness (e.g., collusion-proofness) properties of our methods. Finally, by computing a numerical solution to a typical set of parameters that applies to the model in the Incentive-Secure PoL [Zhao et al., 2024], we show the potential of our approach as a theoretically guaranteed solution that breaks the critical barrier of Verifier's Dilemma in the design of blockchain (and more generally, decentralized) verification mechanisms.

In the rest of this paper, we omit the term "interim" as all the properties are considered in the interim setting.

2 BACKGROUND AND RELATED WORK

2.1 Blockchain Content Verification and Verifier's Dilemma

Since the emergence of Bitcoin [Nakamoto, 2008], 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 create the certificate that gains her the access to produce the block. When a miner attach 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 agents that are responsible to verify the contents, usually called as *verifiers* or validators, may be economically rational or selfish. In this context, Verifier's Dilemma [Luu et al., 2015] occurs as a phenomenon that in a decentralized verification protocol, the verifiers may behave lazily when there are no sufficient rewards to incentivize honest verification, especially when all or overwhelming majority of the contents are valid. For example, [Cao et al., 2023] proposes an attack that leverages the Verifier's Dilemma to double spend in Bitcoin. Besides, the studies of Alharby et al. [2020], Smuseva et al. [2022] 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 economically 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 [Ante, 2021, Khan et al., 2021, Wang et al., 2019]. Furthermore, with the recent rapid development of artificial intelligence (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 [Chen et al., 2024, Conway et al., 2024, Jia et al., 2021, Nodehi et al., 2024, Zhao et al., 2024], 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. [2023] 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. [2024] 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 [Conway et al., 2024], 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.

Since the Verifier's Dilemma appears as a fundamental vulnerability in the incentive structure of blockchains that may severely undermine the reliability of blockchain systems, the studies of Teutsch and Reitwießner [2024], Zhang et al. [2024] 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 an one-phase general-purposed 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.

2.2 Incentivized Consensus for Blockchain and Decentralized Systems

[TODO]

2.3 Peer Prediction Mechanisms in Literature

[TODO]

3 MODELING OF DECENTRALIZED VERIFICATION GAMES

To initiate the study, we first consider the modeling of decentralized verification games (DVG). In a n-verifier DVG, there are n homogeneous verifiers $i = 1, \cdots, n$ independently verifying an on-chain "proof", which has an underlying type $\theta \in S$ that may be "Honest" ($\theta = 0$), "Flag j" ($\theta = F_j$, $j = 1, 2, \cdots, m$) or "Dishonest" ($\theta = 1$), and we denote $S_* = \{0, F_1, \cdots, F_m\}$ as the set of all non-dishonest types. 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. Since the system can insert flags to maintain a pre-set flag rate that robustly incentivizes verification, we have the prior probabilities $P(\theta = F_i)$ as publicly known information; on the other hand, 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. Hence, we have $P(\theta = 0) = 1 - \sum P(\theta = F_i) - \epsilon$.

Similar to [Zhao et al., 2024], we begin with the verification process in a *lossy-channel* model as follows, and then study the general case (as shown in Theorem 4.1). 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 μ_j 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 $P(X_i = 0|\theta = F_j) = 1 - \mu_j$.

For each verifier i, she first makes a decision to follow one of the following strategies:

(1) Informed 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 belief $\mathcal{B}(\boldsymbol{X}_{-i})$ on $\boldsymbol{Z}_{-i}$ is the conditional distribution of $P(\boldsymbol{X}_{-i}|X_i)$ for the informed strategy, or the marginal distribution $P(\boldsymbol{X}_{-i}|\perp) = P(\boldsymbol{X}_{-i})$ 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, and we desire to design a mechanism that uniformly satisfies the incentive guarantees for arbitrary $\{\epsilon_i'\} \in [0, \epsilon_0]^n$.

Then, verifier i reports a Z_i that maximizes $E_{\tilde{\boldsymbol{Z}}_{-i}\sim\mathcal{B}}[R_i(Z_i, \tilde{\boldsymbol{Z}}_{-i})]$ in which $\mathcal{B}$ is her belief of $\boldsymbol{Z}_{-i}$, and claims that Z_i is her observation. After each verifier i independently reports Z_i without seeing $\boldsymbol{Z}_{-i}$[1], the system has the information of $Z_1, \cdots, Z_n$, but not θ, and rewards each prover i based on a *scoring rule* from the reports, denoted as $R_i(Z_i, \boldsymbol{Z}_{-i})$. Then, verifier i's net utility is $R_i(Z_i, \boldsymbol{Z}_{-i}) - c(X_i)$. In the rest of this section, we consider the informed and uninformed strategies separately.

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

$$\begin{aligned} r_{X_i}(Z_i) &= \sum_{\boldsymbol{X}_{-i}\in S} R_i(Z_i, \boldsymbol{X}_{-i}) P(\boldsymbol{X}_{-i}|X_i) && (1)\\ &= \sum_{\boldsymbol{X}_{-i}\in S} R_i(Z_i, \boldsymbol{X}_{-i}) \frac{P(X_i, \boldsymbol{X}_{-i})}{P(X_i)} && (2)\\ &= \sum_{\boldsymbol{X}_{-i}\in S} R_i(Z_i, \boldsymbol{X}_{-i}) \frac{\sum_{\theta\in S} P(\theta)P(X_i|\theta)P(\boldsymbol{X}_{-i}|\theta)}{\sum_{\theta\in S} P(\theta)P(X_i|\theta)}. && (3) \end{aligned}$$

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

$$\sum_{\boldsymbol{X}_{-i}\in S} R_i(X_i, \boldsymbol{X}_{-i}) \frac{\sum_{\theta\in S} \tilde{P}(\theta)\tilde{P}(X_i|\theta)\tilde{P}(\boldsymbol{X}_{-i}|\theta)}{\sum_{\theta\in S} \tilde{P}(\theta)\tilde{P}(X_i|\theta)} \geq c(X_i) + \delta, \qquad \forall X_i \in S_*, \tag{4}$$

in which $\tilde{P}$ denotes the probability assuming $\epsilon = 0$, and for any $\delta > 0$, the constraints hold 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.

For the case of $X_i = 1$, from the DVG model, we know that it must hold that $\theta = 1$. Therefore, we have

$$r_1(1) = \sum_{X_{-i}\in S} R_i(1, \boldsymbol{X}_{-i}) P(\boldsymbol{X}_{-i}|\theta = 1) \geq c(1) + \delta. \tag{5}$$

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

So Eqs. (4- 5) are sufficient and almost necessary conditions that a CTF-PP mechanism is IR.

For the IC constraint, we need and only need $r_{X_i}(X_i) = \max_{Z_i \in S}\{r_{X_i}(X_i)\}$. 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 has already 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:

$$r_{X_i}(Z_i) \leq c(X_i) - \delta, \qquad \forall X_i \in S,\ Z_i \neq X_i. \tag{6}$$

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 $\boldsymbol{Z}_{-i} = \boldsymbol{X}_{-i}$. 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:

$$r_\perp(Z_i) = \sum_{\boldsymbol{X}_{-i} \in S} R_i(Z_i, \boldsymbol{X}_{-i}) P(\boldsymbol{X}_{-i}). \tag{7}$$

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

$$\sum_{\boldsymbol{X}_{-i} \in S} R_i(Z_i, \boldsymbol{X}_{-i}) P(\boldsymbol{X}_{-i}) \leq -\delta, \qquad \forall Z_i \in S. \tag{8}$$

3.3 Incentive Alignment for Decentralized Verification Games

From the discussion in Section 3.1-3.2, 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 3.1. *A CTF-PP mechanism is δ-incentive-aligned (δ-IA) if and only if*

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

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

4 THEORETICAL GUARANTEE FOR 2-VERIFIER 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.

Assume $\epsilon = 0$, and define the cheat-free belief matrix $B : S^2 \longrightarrow \mathbb{R}$ as $B_{xy} = P(X_{-i} = y | X_i = x)$.[2] Besides, we define $B_\perp$ as the blind-belief (row) vector as $B_{\perp y} = P(X_{-i} = y)$ that describes the belief

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

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 $T : S^2 \longrightarrow \mathbb{R}$ with $T_{xy} = R_i(x, y)$, and denote $W = BT'$. 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:

$$W_{xx} \geq c(x) + \delta, \qquad \forall x \in S; \tag{9}$$

$$W_{xy} \leq c(x) - \delta, \qquad \forall x \in S, \quad y \in S - \{x\}. \tag{10}$$

Similarly, we denote $W_\perp = B_\perp T'$, 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:

$$W_\perp \leq -\delta. \tag{11}$$

Hence, we only need to find a feasible solution of the linear system (9)-(11). Here, we propose our main theorem, with the proof deferred to Appendix B:

Theorem 4.1 (Main 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 $\delta \geq 0$, there exists a δ-IA mechanism for the 2-verifier DVG.*

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 3 with the following proposition. The proof is omitted.

Proposition 4.2. *In the lossy-channel model defined in Section 3, the cheat-free belief matrix B is invertible.*

5 GENERAL MECHANISM DESIGN FOR n-VERIFIER DVG

In this section, we leverage the result for the 2-verifier DVG to design a CTF-PP mechanism for general n-verifier DVG, with additional robustness properties against collusion and malicious verifiers (to be shown in the next section).

Conceptually, in Section 4 we have shown that under mild assumptions there always exists an incentive-aligned mechanism for any 2-verifier DVG. In this section, we invoke the 2-verifier mechanism as a building block and construct our mechanism for the general setting of n verifiers.

5.1 Warmup: Random Pairing Mechanism

Imagine a simple scenario of 4 verifiers, in which each verifier $i \in \{1, 2, 3, 4\}$ independently verifies the block content θ with their observations $\{X_i\}$ as modeled in Section 3. To incentivize all verifiers to behave honestly, a simple mechanism works as follows:

(1) Divide the verifiers into two pairs $\{1, 2\}$ and $\{3, 4\}$.
(2) Run the 2-verifier mechanism in each pair.
(3) Each verifier i's report Z_i is compared to another verifier j's report Z_j, getting an ex-post reward of $T_{Z_i Z_j}$ as defined in Section 4.

In this simple mechanism, from the perspective of each verifier i, she is playing the 2-verifier DVG with another verifier, with mechanism described as the scoring matrix T. Hence, as long as the 2-verifier mechanism is δ-IA, the proposed 4-verifier mechanism also satisfies the desired δ-IA property.

In actual decentralized environments, as the verifiers are anonymous and can arrive in arbitrary orders, their indices are essentially randomly shuffled. Hence, the pairing result is random and uniformly distributed among the following 3 cases, each with $\frac{1}{3}$ probability:

- $\{1,2\},\{3,4\}$;
- $\{1,3\},\{2,4\}$;
- $\{1,4\},\{2,3\}$.

As we can observe, each verifier i may be paired with any of the remaining 3 verifiers with equal probabilities. Hence among the distribution of random pairing, the verifier i gets an expected reward of

$$R_i(Z_i, Z_{-i}) = \frac{1}{3}\sum_{j\neq i} T_{Z_iZ_j}. \tag{12}$$

In fact, the random pairing mechanism can be generalized for any even n. From the symmetry of the random shuffling, the verifier i will be paired with any verifier $j \neq i$ with probability $\frac{1}{n-1}$, and i's expected reward is:

$$R_i(Z_i, Z_{-i}) = \frac{1}{n-1}\sum_{j\neq i} T_{Z_iZ_j}. \tag{13}$$

From the argument above, the random pairing mechanism for any even n, formulated as Eq. (13), is δ-IA as long as the 2-verifier mechanism with the scoring rule described as T is δ-IA. The formal results are shown in Theorem 5.1, Section 5.2.

5.2 Revisit of Random Pairing: Linear Average Mechanism

While the random pairing mechanism only works for even n, in practical usage of decentralized consensus, e.g., the common practice of majority voting, the case of odd n is also (if not more) essential. On the other hand, by looking at Eq. (13), the scoring rule is not "incompatible" to odd n. Actually, there can be an alternative interpretation to the scoring rule:

(1) For fixed verifier i, collect her report Z_i and all other verifiers' reports $\{Z_j : j \neq i\}$.
(2) For each $j \neq i$, compare Z_i and Z_j and get a sub-score $T_{Z_iZ_j}$ invoking the 2-verifier mechanism.
(3) Take the average of all sub-scores as the final reward to i.

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 column vector in the corresponding dimension, we can get $T_{Z_iZ_j} = Z_i' T Z_j$. Hence, from the inspiration of Section 5.1, our linear average mechanism can be formulated as:

$$R(Z_i, Z_{-i}) = Z_i T \overline{Z_{-i}}, \tag{14}$$

in which we denote

$$\overline{Z_{-i}} = \frac{1}{n-1}\sum_{j\neq i} Z_j. \tag{15}$$

Then, we can show that the linear average mechanism as described as Eq. (14) has equivalent incentive structures as the 2-verifier mechanism characterized as T. Formally, we have:

Theorem 5.1. *If the scoring matrix T satisfies the δ-IA property for the 2-verifier DVG, then the scoring rule as Eq.* (14) *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. (14) *also satisfies the same properties.*

The proof of Theorem 5.1 is deferred to Appendix C.

6 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 and permissionless 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. [2022], Chung and Shi [2023], Roughgarden [2020, 2021], Wu et al. [2023b]), blockchain players may also potentially collude with each other or create fake identities (aka. *Sybil attack*) to increase their utilities.

While 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., Chen et al. [2012], Wu et al. [2023a], Yin et al. [2018]), also called as *Byzantine fault tolerance*, 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 in the correct way. In this notion, the types of players can be generally classified into the following categories:

(1) Altruistic ($\mathcal{A}$): Acting honestly without consideration of utilities;
(2) Selfish ($\mathcal{S}$): Acting in the way that maximizes their own utilities;
(3) Colluding ($\mathcal{C}$): Conducting collusions with other players (within $\mathcal{C}$) to maximize their joint utilities;
(4) Malicious ($\mathcal{M}$): Acting arbitrarily without any assumption.

Traditional game theory primarily focuses on the behavior of $\mathcal{A}$ and $\mathcal{S}$ players, while $\mathcal{C}$ and $\mathcal{M}$ players typically fall outside the scope of its standard models. Therefore, we call $\mathcal{A}, \mathcal{S}$ players as **benign** and $\mathcal{C}, \mathcal{M}$ 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 report honestly. However, we rule out the malicious verifiers from incentive constraints for the following reasons:

- Psychologically, a malicious player would act in an arbitrary way. Even the incentive mechanism may not effectively prevent them from behaving maliciously.
- Systematically, to make the category $\mathcal{M}$ general enough, we allow they to have extra resources (i.e., the information of others' reports) that may bypass the Bayesian game assumption, so the Bayesian incentive guarantees may not work on them in practice.
- Technically, if malicious players without such additional resources still manage to benefit from deviation, it would imply that selfish players ($\mathcal{S}$), who are solely motivated by their utilities, would also find deviation beneficial — a scenario we have already addressed and prevented.

From the argument above, we define the notion of $f(n)$-Byzantine-robustness ($f(n)$-BR) as follows:

Definition 6.1 (Byzantine Robustness). *For a DVG with n verifiers, we call a mechanism $f(n)$-BR if and only if: as long as the total number of rogue ($\mathcal{C}$ and $\mathcal{M}$) verifiers does not exceed $f(n)$, there is a Bayesian Nash equilibrium that all $\mathcal{A}, \mathcal{S}, \mathcal{C}$ verifiers act honestly.*

In the rest of this section, we show that under mild assumptions, the design in Section 5, as long as the scoring matrix T comes from a "good" solution of the linear system Eqs. (9-11), is $\Theta(n)$-BR, i.e., resilient against a constant portion of rogue verifiers.

6.1 Criteria and Optimization of Byzantine Robustness

When we look at the linear average mechanism Eq. (14), the reward of each verifier i is essentially based on the comparison of her report Z_i and the *average* of other verifiers' reports $\overline{Z_{-i}}$. Intuitively, if only a small portion of other verifiers may act dishonestly, since their contribution to $\overline{Z_{-i}}$ is not significant, the actual expectation of $\overline{Z_{-i}}$ conditioned on X_i would not deviate significantly from $\mathbb{E}\left[\overline{X_{-i}}\middle|X_i\right]$, 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 $\overline{Z_{-i}}$.

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 6.2 below. In later sections we will show that collusions can also be reduced to the canonical Byzantine setting. For Sybil attacks, while no voting-based protocols can effectively prevent them if the attacker has unlimited resources (e.g., 51% attack [Raju et al., 2022]), our study also show that each Sybil identity can also be reduced to 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 significant advantage via Sybil attacks compared to honest behavior. The detailed discussion on the incentives of Sybil attacks is deferred to Section [??].

Definition 6.2 (Canonical Byzantine Setting). *In a canonical Byzantine setting, each verifier acts in one of the following strategies:*

- *No-Sybil selfish ($\mathcal{S}_*$): Acting in a way that maximizes their own utilities, but unable to create fake identities.*
- *No-Sybil malicious ($\mathcal{M}_*$): Reporting arbitrarily, but unable to create fake identities.*

From the linear average mechanism, we can see that conditioned on the verifier i observing X_i, the expected utility of reporting Z_i is

$$r_{X_i}(Z_i) = \frac{1}{n-1}\sum_{j\neq i}\left(\sum_{X_j\in S} P(Z_j|X_i)T_{Z_iZ_j} - c(X_i)\right). \tag{16}$$

When verifier j is honest, we see that $Z_j = X_j$ and $\sum_{X_j\in S} P(Z_j|X_i)T_{Z_iZ_j} - c(X_i) = (BT')_{X_iZ_i} - c(X_i) = W_{X_iZ_i}$, 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 $\sum_{X_j\in S} P(Z_j|X_i)T_{Z_iZ_j}$ and leading to a perturbation to $r_{X_i}(Z_i)$.

Intuitively, if the summation of all these perturbations is bounded below δ, then $\{r_{X_i}(Z_i)\}$ still has positive diagonal entries and negative non-diagonal entries, satisfying the incentive alignment notion with a smaller margin. On the other hand, we notice that $\sum_{X_j\in S} P(Z_j|X_i)T_{Z_iZ_j}$ is a convex combination of $\{T_{Z_iZ_j} : Z_j \in S\}$. If we have

$$|T_{Z_iZ_j}| \leq N, \quad \forall Z_i, Z_j \in S, \tag{17}$$

then we can deduce that

$$\sum_{X_j\in S} P(Z_j|X_i)T_{Z_iZ_j} \in [-N, N]. \tag{18}$$

Hence, each dishonest verifier j can perturb the value of $r_{X_i}(Z_i)$ by at most $\frac{2N}{n-1}$. In this sense, we define (δ, N)-compactness as:

Definition 6.3 ((δ, N)-compactness). *For fixed observation costs $c(\cdot)$ and cheat-free belief matrix B (denoted as the (c, B)-environment), a scoring matrix T is called (δ, N)-compact if and only if its entries are bounded within $[-N, N]$ and the corresponding mechanism is δ-IA.*

For convenience, we also call a mechanism (or a scoring matrix) $\frac{\delta}{N}$-compact if it is (δ, N)-compact for some (δ, N).

Then, we immediately derive the following lemma:

Lemma 6.4. *If a CTF-PP mechanism has a (δ, N)-compact scoring matrix, then it is $\frac{\delta}{2N}(n-1)$-BR in the canonical Byzantine setting.*

As we will also show that (δ, N)-compactness implies $\Omega(\frac{\delta n}{N})$-BR for the general setting in Section [???], in the following parts we focus on the construction of (δ, N)-compact scoring matrices with optimized $\frac{\delta}{N}$.

6.2 LP Modeling for Byzantine Robustness

In Section 4, we formulated the δ-IA condition for the 2-verifier DVG as the linear system of Eqs. (9-11), and showed that the linear system is generally feasible, so that a desirable mechanism can be found via linear programming; in Section 5 we 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.

Besides the linear constraints, 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 6.4 we would like to construct a (δ, N)-compact scoring matrix with a large $\frac{\delta}{N}$. Hence, an intuitive idea is to fix δ and minimize N. For fixed cheat-free belief matrix B, prior distribution vector $B_\perp$[3] and observation costs $c(\cdot)$, we can formulate the LP problem $LP_1(B, B_\perp, c, \delta)$ with decision variable $\left(N \in \mathbb{R}, T \in \mathbb{R}^{S^2}\right)$ as:

$$\begin{aligned}
LP_1(B, B_\perp, c, \delta): \quad & \textit{minimize} & & N & & \qquad (19)\\
& \textit{s.t.} & |T| &\le N, & & \qquad (20)\\
& & (BT')_{xx} &\ge c(x) + \delta, & \forall x \in S & \qquad (21)\\
& & (BT')_{xy} &\le c(x) - \delta, & \forall x \in S, \quad y \in S - \{x\} & \qquad (22)\\
& & B_\perp T' &\le -\delta. & & \qquad (23)
\end{aligned}$$

In fact, denoting $N_* = (B, B_\perp, c, \delta)$ as the optimal objective value of $LP_1(B, B_\perp, c, \delta)$, we can show an upper bound on $N_*(B, B_\perp, c, \delta)$ 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*

$$N_*(B, B_\perp, c, \delta) \le \|B^{-1}\|_2 \cdot (c_1 + \delta \cdot f(k, p_1)), \qquad (24)$$

in which

$$f(k, p_1) = \sqrt{\left(k + (2k-2)\frac{p_1}{1-p_1}\right)\left(k + \frac{2}{1-p_1}\right)} \qquad (25)$$

is only dependent on k, p_1 but independent to n.

Additionally, there exists a feasible solution satisfying (24) *that makes the equality hold in* (21).

[3] $B'_\perp$ can be actually computed from B as the eigenvector of B' with eigenvalue 1 (See in Lemma D.1, Appendix D), but here we just formulate it as an extra constraint parameter.

The proof of Theorem 6.5 is deferred to Appendix D.

6.3 Reduction of Colluding and Sybil Identities

7 EXPERIMENTAL EVALUATION: VERIFICATION OF INCENTIVE-SECURE POL

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.

7.1 Construction of the Scoring Rule

We consider the 2-verifier DVG which captures the case of one stage in [Zhao et al., 2024]. 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 [4] 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 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 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 cheat-free belief matrix B as Table 3.

	$X_i = 0$	$X_i = F_1$	$X_i = F_2$	$X_i = 1$
$\theta = 0$	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 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 2. Joint probabilities $P(X_1, X_2)$ ($\epsilon = 0$).

	$X_2 = 0$	$X_2 = F_1$	$X_2 = F_2$	$X_2 = 1$
$X_1 = 0$	$\frac{5}{6}$	$\frac{1}{12}$	$\frac{1}{12}$	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 3. The cheat-free belief matrix B.

[4] It is significantly more than needed, but does work.

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. (9)-(11) 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:

$$\begin{aligned} &\textit{minimize} \;\; N \\ &\qquad s.t. \;\; \text{(9)-(11)}, \qquad -N \leq T \leq N. \end{aligned}$$

We set a margin of $\delta = 0.2$, and compute a numerical solution to this LP, getting a scoring rule as shown in Table 4.

	$Z_{-i} = 0$	$Z_{-i} = F_1$	$Z_{-i} = F_2$	$Z_{-i} = 1$
$Z_i = 0$	+2.0959	−7.1643	−7.1643	−1.0764
$Z_i = F_1$	−1.5421	+6.4732	−4.4574	−1.2378
$Z_i = F_2$	−1.5421	−4.4574	+6.4732	−1.2378
$Z_i = 1$	−2.2000	+5.8000	+5.8000	+7.4000

Table 4. $T_{Z_i Z_{-i}} = R_i(Z_i, Z_{-i})$ as a numerical solution.

7.2 Evaluation

With this scoring rule, given verifier $-i$ acts honestly, we report the expected utility of verifier i is in Table 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 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	**+0.2192**	−1.4504	−1.4504	−1.2000
Observing F_1	−4.5342	**+0.4655**	−4.9998	−0.2000
Observing F_2	−4.5342	−4.9998	**+0.4655**	−0.2000
Observing 1	−3.0122	−2.8285	−2.8285	**+0.2000**
Does not verify	−0.2192	−0.9046	−0.9046	−0.2000

Table 5. Verifier's expected utility, $\epsilon = 0$.

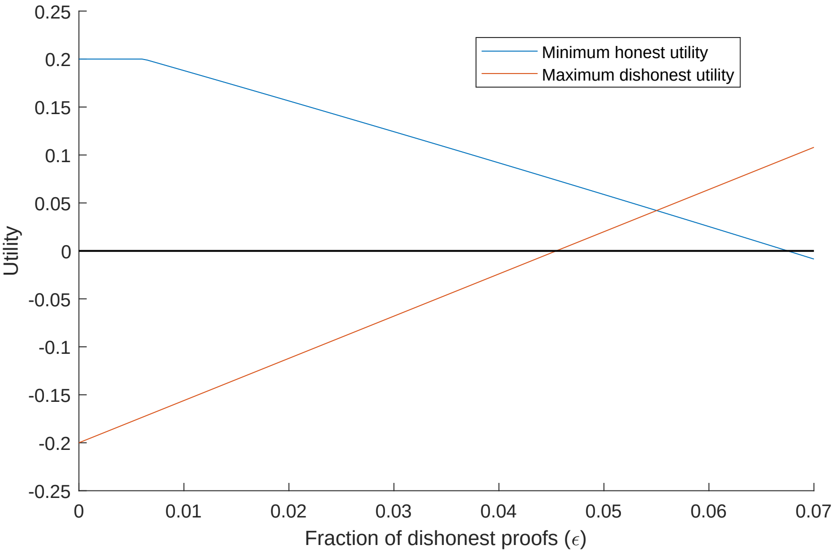

Fig. 1. Verifier's expected utility, $\epsilon > 0$.

8 DISCUSSION

In this paper, we develop a theoretical framework for the decentralized verification game on blockchain proving protocols and get preliminary results that the combination of the Capture-The-Flag design (as in [Zhao et al., 2024]) and peer prediction techniques has the potential to resolve the Verifier's Dilemma in a fully decentralized environment. On the other hand, we also explore the design of peer prediction mechanisms with broader agent strategy spaces and more general settings (e.g. observation costs dependent on observation results). In future work, we will improve and broaden the study in the following aspects:

(1) While this paper is mainly on the elicitation of truthful verification results, we will also develop voting/aggregation mechanisms that (optimally) make decisions on whether to accept the proof.

Additionally, while our mechanism is conceptually designed for $\epsilon \approx 0$ and does not work for a relatively large ϵ (e.g. > 0.045 in the example), in real-world cases where ϵ is (unexpectedly) large but estimable, the system can be augmented to adjust the belief matrix B based on an (automated) estimation of ϵ, denoted as $\hat{\epsilon}$, and compute an adjusted scoring rule accordingly. From the same argument on the margin δ, the adjusted scoring rule would incentivize honest verification as long as $|\epsilon - \hat{\epsilon}|$ is small.

[TODO: More general applications beyond blockchain...]

REFERENCES

Maher Alharby, Roben Castagna Lunardi, Amjad Aldweesh, and Aad Van Moorsel. 2020. 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)*. IEEE, 209–220.

Lennart Ante. 2021. Smart contracts on the blockchain–A bibliometric analysis and review. *Telematics and Informatics* 57 (2021), 101519.

Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. *white paper* 3, 37 (2014), 2–1.

Tong Cao, Jérémie Decouchant, and Jiangshan Yu. 2023. Leveraging the verifier's dilemma to double spend in Bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 149–165.

Franck Cassez, Joanne Fuller, and Aditya Asgaonkar. 2022. Formal Verification of the Ethereum 2.0 Beacon Chain. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 167–182.

Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. 2024. ZKML: An Optimizing System for ML Inference in Zero-Knowledge Proofs. In *Proceedings of the Nineteenth European Conference on Computer Systems*. 560–574.

Ruiliang Chen, Jung-Min Jerry Park, and Kaigui Bian. 2012. Robustness against Byzantine failures in distributed spectrum sensing. *Computer Communications* 35, 17 (2012), 2115–2124.

Xi Chen, David Simchi-Levi, Zishuo Zhao, and Yuan Zhou. 2022. Bayesian mechanism design for blockchain transaction fee allocation. *arXiv preprint arXiv:2209.13099* (2022).

Yiling Chen, Yiheng Shen, and Shuran Zheng. 2020. Truthful data acquisition via peer prediction. *Advances in Neural Information Processing Systems* 33 (2020), 18194–18204.

Hao Chung and Elaine Shi. 2023. Foundations of transaction fee mechanism design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 3856–3899.

KD Conway, Cathie So, Xiaohang Yu, and Kartin Wong. 2024. opML: Optimistic Machine Learning on Blockchain. *arXiv preprint arXiv:2401.17555* (2024).

Anirban Dasgupta and Arpita Ghosh. 2013. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*. 319–330.

Congyu Fang, Hengrui Jia, Anvith Thudi, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Varun Chandrasekaran, and Nicolas Papernot. 2023. Proof-of-Learning is Currently More Broken Than You Think. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 797–816.

Oded Goldreich and Yair Oren. 1994. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology* 7, 1 (1994), 1–32.

Atalay M Ileri, Halil I Ozercan, Alper Gundogdu, Ahmet K Senol, M Yusuf Ozkaya, and Can Alkan. 2016. Coinami: a cryptocurrency with DNA sequence alignment as proof-of-work. *arXiv preprint arXiv:1602.03031* (2016).

Hengrui Jia, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. 2021. Proof-of-learning: Definitions and practice. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1039–1056.

Shafaq Naheed Khan, Faiza Loukil, Chirine Ghedira-Guegan, Elhadj Benkhelifa, and Anoud Bani-Hani. 2021. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications* 14 (2021), 2901–2925.

Yuqing Kong. 2023. Dominantly Truthful Peer Prediction Mechanisms with a Finite Number of Tasks. *J. ACM* (2023).

Yuqing Kong and Grant Schoenebeck. 2019. An information theoretic framework for designing information elicitation mechanisms that reward truth-telling. *ACM Transactions on Economics and Computation (TEAC)* 7, 1 (2019), 1–33.

Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. 2015. Demystifying incentives in the consensus computer. In *Proceedings of the 22Nd acm sigsac conference on computer and communications security*. 706–719.

Nolan Miller, Paul Resnick, and Richard Zeckhauser. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51, 9 (2005), 1359–1373.

Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review* (2008).

Behkish Nassirzadeh, Stefanos Leonardos, Albert Heinle, Anwar Hasan, and Vijay Ganesh. 2024. CountChain: A Decentralized Oracle Network for Counting Systems. *arXiv preprint arXiv:2409.11592* (2024).

Hanzaleh Akbari Nodehi, Viveck R Cadambe, and Mohammad Ali Maddah-Ali. 2024. Game of Coding: Beyond Trusted Majorities. *arXiv preprint arXiv:2401.16643* (2024).

Raja Siddharth Raju, Sandeep Gurung, and Prativa Rai. 2022. An overview of 51% attack over Bitcoin network. *Contemporary Issues in Communication, Cloud and Big Data Analytics: Proceedings of CCB 2020* (2022), 39–55.

Maxime Reynouard, Rida Laraki, and Olga Gorelkina. 2024. BAR Nash Equilibrium and Application to Blockchain Design. *arXiv preprint arXiv:2401.16856* (2024).

Tim Roughgarden. 2020. Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559. *arXiv preprint arXiv:2012.00854* (2020).

Tim Roughgarden. 2021. Transaction fee mechanism design. *ACM SIGecom Exchanges* 19, 1 (2021), 52–55.

Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C Parkes. 2016. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. 179–196.

Daria Smuseva, Ivan Malakhov, Andrea Marin, Aad van Moorsel, and Sabina Rossi. 2022. Verifier's dilemma in ethereum blockchain: A quantitative analysis. In *International Conference on Quantitative Evaluation of Systems*. Springer, 317–336.

Jason Teutsch and Christian Reitwießner. 2024. A scalable verification solution for blockchains. In *ASPECTS OF COMPUTATION AND AUTOMATA THEORY WITH APPLICATIONS*. World Scientific, 377–424.

Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, and Fei-Yue Wang. 2019. Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49, 11 (2019), 2266–2277.

Shengling Wang, Xidi Qu, Qin Hu, Xia Wang, and Xiuzhen Cheng. 2023. An uncertainty-and collusion-proof voting consensus mechanism in blockchain. *IEEE/ACM Transactions on Networking* 31, 5 (2023), 2376–2388.

Ke Wu, Elaine Shi, and Hao Chung. 2023b. Maximizing Miner Revenue in Transaction Fee Mechanism Design. *Cryptology ePrint Archive* (2023).

Zhaoxian Wu, Tianyi Chen, and Qing Ling. 2023a. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. *IEEE transactions on signal processing* (2023).

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*. Pmlr, 5650–5659.

Yue Zhang, Shouqiao Wang, Xiaoyuan Liu, Sijun Tan, Raluca Ada Popa, and Ciamac C Moallemi. 2024. Proof of Sampling: A Nash Equilibrium-Secured Verification Protocol for Decentralized Systems. *arXiv preprint arXiv:2405.00295* (2024).

Zishuo Zhao, Zhixuan Fang, Xuechao Wang, and Yuan Zhou. 2024. Proof-of-Learning with Incentive Security. *arXiv preprint arXiv:2404.09005* (2024).

Appendix

A PROOF OF THEOREM 1.1

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.

B PROOF OF THEOREM 4.1

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

$$\begin{aligned} B_{\perp y} &= P(X_{-i} = y) && (26)\\ &= \sum_{x \in S} P(X_i = x) \cdot P(X_{-i} = y | X_i = x) && (27)\\ &= \sum_{x \in S} B_{\perp x} \cdot B_{xy}, && (28) \end{aligned}$$

It holds that $B_\perp = B_\perp B$. Hence, $W_\perp = B_\perp T' = B_\perp BT' = B_\perp W$. 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. 9 with non-diagonal entries small enough.

Here, for a constant $M > 0$ large enough, we construct W as:

$$\begin{aligned} W_{xx} &= c(x) + \delta, \qquad \forall x \in S; && (29)\\ W_{xy} &= -M, \qquad \forall x \in S, \quad y \in S - \{x\}. && (30) \end{aligned}$$

Then,

$$\begin{aligned} W_{\perp y} &= \sum_{x \in S} B_{\perp x} W_{xy} && (31)\\ &= B_{\perp y} W_{yy} + \sum_{x \in S - \{y\}} B_{\perp x} W_{xy} && (32)\\ &= B_{\perp y}(c(y) + \delta) - (1 - B_{\perp y})M. && (33) \end{aligned}$$

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

$$M \geq \frac{B_\perp^* \cdot (c(y) + \delta) + \delta}{1 - B_\perp^*}, \qquad (34)$$

Then the required constraints of Eqs. (9)-(11) 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 $\theta \neq 1 \implies X_i \neq 1$, Eq. (5) 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.

C PROOF OF THEOREM 5.1

In the context of Bayesian Nash equilibrium, we can assume each verifier $j \neq i$ is honest, i.e., $Z_{-i} = \boldsymbol{X}_{-i}$. Hence, given that verifier i observes $X_i \in S \cup \{\perp\}$ and reports $Z_i \in S$, the interim expected reward is:

$$\begin{aligned} r_{X_i}(Z_i) &= \mathbb{E}\left[Z_i T\overline{\boldsymbol{X}_{-i}}\middle|X_i\right] && (35)\\ &= Z_i T \cdot \mathbb{E}\left[\overline{\boldsymbol{X}_{-i}}\middle|X_i\right] && (36)\\ &= Z_i T \cdot \mathbb{E}\left[\frac{1}{n-1}\sum_{j\neq i} X_j\middle|X_i\right] && (37)\\ &= \frac{1}{n-1}\sum_{j\neq i} Z_i T\mathbb{E}\left[X_j\middle|X_i\right] && (38)\\ &= \frac{1}{n-1}\sum_{j\neq i} Z_i T \sum_{X_j\in S} P(X_j|X_i)X_j && (39)\\ &= \frac{1}{n-1}\sum_{j\neq i}\sum_{X_j\in S} P(X_j|X_i)T_{Z_iX_j}. && (40) \end{aligned}$$

With similar arguments as Section 4, we assume $\epsilon = 0$, and $P(X_j|X_i)$ is the (i, j)-th entry of the cheat-free belief matrix B for any $j \neq i$. Hence, we have

$$\begin{aligned} r_{X_i}(Z_i) &= \frac{1}{n-1}\sum_{j\neq i}\sum_{X_j\in S} P(X_j|X_i)T_{Z_iX_j} && (41)\\ &= \sum_{X_j\in S} B_{X_iX_j}T_{Z_iX_j} && (42)\\ &= (BT')_{X_iZ_i}. && (43) \end{aligned}$$

Hence, the linear program of Eqs. (9-11) works equivalently for the n-verifier DVG when we use the linear average mechanism as Eq. (14) 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. (14).

D 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 1. *If (N_c, T_c) is a feasible solution of $LP_1(B, B_\perp, c, 0)$ and (N_δ, T_δ) is a feasible solution of $LP_1(B, B_\perp, 0, 1)$, then $(N_c + \delta N_\delta, T_c + \delta T_\delta)$ is a feasible solution of $LP_1(B, B_\perp, c, \delta)$.*

Hence, we can estimate upper bounds of optimal N_c and N_δ separately. Here we denote $\|\cdot\|_2$ as the matrix ℓ_2-norm, and denote $W = BT'$. From the assumption in Theorem 4.1 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 $W = \text{diag}(c)$, then the corresponding $T_c = (B^{-1}W)'$ satisfies conditions (21-23). Because $\|W\|_2 = \|\text{diag}(c)\|_2 = \max\{c\}$, we have

$$|T_c| \leq \|B^{-1}\|_2 \cdot \max\{c\}. \tag{44}$$

Hence, $N_c = \|B^{-1}\|_2 \cdot \max\{c\}$ is feasible for $LP_1(B, B_\perp, c, 0)$.

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. Let j be an arbitrary verifier other than j. From the discussion of the uninformed strategy (in Section 3), we have

$$\begin{aligned} B_{\perp y} &= P(X_j = y | X_i = \perp) && (45)\\ &= P(X_j = y). && (46) \end{aligned}$$

On the other hand,

$$\begin{aligned} (B_\perp B)_y &= \sum_{x \in S} B_{\perp x} B_{xy} && (47)\\ &= \sum_{x \in S} P(X_i = x) \cdot P(X_j = y | X_i = x) && (48)\\ &= \sum_{x \in S} P(X_i = x, X_j = y) && (49)\\ &= P(X_j = y). && (50) \end{aligned}$$

Hence we have $B_{\perp y} = (B_\perp B)_y$ for $\forall y \in S$, so $B_\perp B = B_\perp$. □

From Lemma D.1, the $LP_1(B, B_\perp, 0, 1)$ can be reformulated as:

$$\begin{aligned} LP_2(B, B_\perp, 0, 1) : \quad & \textit{minimize} & N & & (51)\\ & s.t. & |B^{-1}W| \leq N, & & (52)\\ & & W_{xx} \geq 1, & \quad \forall x \in S & (53)\\ & & W_{xy} \leq -1, & \quad \forall x \in S, \quad y \in S - \{x\} & (54)\\ & & B_\perp W \leq -1. & & (55) \end{aligned}$$

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 (53-54) are satisfied. For condition (55), we have

$$\begin{aligned} (B_\perp W)_y &= \sum_{x \in S} B_{\perp x} W_{xy} && (56)\\ &= B_{\perp y} W_{yy} + \sum_{x \in S - \{y\}} B_{\perp x} W_{xy} && (57)\\ &= B_{\perp y} \cdot 1 + (1 - B_{\perp y}) \cdot \left(- \frac{1 + B_{\perp y}}{1 - B_{\perp y}} \right) && (58)\\ &= -1. && (59) \end{aligned}$$

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,*

$$\|A\|_2 \le \sqrt{\|A\|_1 \|A\|_\infty}. \tag{60}$$

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

$$\|A\|_2 = \sqrt{\lambda_{max}(A^*A)}. \tag{61}$$

Because the maximum eigenvalue is a lower bound on the ℓ_∞ norm, we have

$$\lambda_{max}(A^*A) \le \|A^*A\|_\infty \tag{62}$$

$$\le \|A^*\|_\infty \|A\|_\infty \tag{63}$$

$$= \|A\|_1 \|A\|_\infty. \tag{64}$$

Hence we prove $\|A\|_2 \le \sqrt{\|A\|_1 \|A\|_\infty}$.

□

Now we sort $\{B_{\perp y} : y \in S\}$ as $p_1 \ge p_2 \ge \cdots \ge p_k$, in which $k = |S| = m + 2$. Then we have

$$\|W\|_1 = \max_{y \in S} \sum_{x \in S} |W_{xy}| \tag{65}$$

$$= \max_{1 \le j \le k} \left\{ 1 + (k-1) \frac{1 + p_j}{1 - p_j} \right\} \tag{66}$$

$$= 1 + (k-1) \frac{1 + p_1}{1 - p_1} \tag{67}$$

$$= k + (2k - 2) \frac{p_1}{1 - p_1}, \tag{68}$$

and

$$\|W\|_\infty = \max_{x \in S} \sum_{y \in S} |W_{xy}| \tag{69}$$

$$= \max_{1 \le i \le k} \left\{ 1 + \sum_{j \ne i} \frac{1 + p_j}{1 - p_j} \right\} \tag{70}$$

$$\le \sum_{i=1}^{k} \frac{1 + p_j}{1 - p_j} \tag{71}$$

$$= k + 2 \sum_{i=1}^{k} \frac{p_j}{1 - p_j} \tag{72}$$

$$\le k + 2 \sum_{i=1}^{k} \frac{p_j}{1 - p_1} \tag{73}$$

$$= k + \frac{2}{1 - p_1}. \tag{74}$$

Therefore, we have

$$\|W_2\| \le \sqrt{\left(k + (2k-2)\frac{p_1}{1 - p_1}\right)\left(k + \frac{2}{1 - p_1}\right)} \tag{75}$$

and

$$N_\delta = \|B^{-1}\|_2 \cdot \sqrt{\left(k + (2k-2)\frac{p_1}{1-p_1}\right)\left(k + \frac{2}{1-p_1}\right)} \tag{76}$$

is feasible for $LP_2(B, B_\perp, 0, 1)$.

Hence, $N_c + \delta N_\delta$ is feasible for $LP_1(B, B_\perp, c, \delta)$. Because our constructions for both parts make the equality hold in (21), the final construction makes the equality hold naturally.