Discussion Paper Series – CRC TR 224

Discussion Paper No. 685
Project C 03

Achieving Consensus on Blockchains

Zahra Ebrahimi[1]
Maxi Guennewig[2]
Bryan Routledge[3]
Ariel Zetlin-Jones[4]

April 2025

[1]Carnegie Mellon University
[2]University of Bonnn,
[3]Carnegie Mellon University
[4]Carnegie Mellon University

Support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through CRC TR 224 is gratefully acknowledged.

Collaborative Research Center Transregio 224 - www.crctr224.de
Rheinische Friedrich-Wilhelms-Universität Bonn - Universität Mannheim

Achieving Consensus on Blockchains*

Zahra Ebrahimi† Maxi Guennewig‡ Bryan Routledge†
Ariel Zetlin-Jones†

April 4, 2025

Abstract

Blockchain is a database technology that enables a group of self-interested users to maintain a distributed ledger without relying on a trusted third party, such as a bank. In this paper, we develop a new game-theoretic framework for analyzing blockchain systems, wherein each user determines how to update the distributed ledger. The usefulness of blockchains depends on whether users' updating strategies achieve consensus—meaning that they agree on the correct version of the ledger and have no incentive to omit or alter past data. We show that the currently implemented strategy—the longest chain rule—fails to achieve consensus when users are sufficiently heterogeneous. We then establish the existence of new equilibrium strategies, which are slight modifications of the longest chain rule and ensure consensus regardless of the degree of heterogeneity. In practice, these equilibrium strategies enhance the resilience of blockchain systems against threats such as double-spending and 51% attacks. Our findings underscore the critical role economic incentives play in determining the security and stability of blockchain ledgers.

Keywords: Blockchain, consensus, double-spending.

*Previously circulated as "Getting Blockchain Incentives Right." First version: February 2020. We thank Dimitry Orlov and seminar participants at the Virtual Finance Theory Seminar, The Joint Renmin University, Hong Kong Baptist University, and National Taiwan University Virtual Seminar; The 2nd Tokenomics Conference on Blockchain Economics, Security and Protocol, the Madison Money Workshop (2021), SED Meetings (Minneapolis), Michigan State University, and the University of Calgary for insightful comments and discussions. Zahra Ebrahimi, Bryan Routledge and Ariel Zetlin-Jones are grateful for the financial support of the PNC Center for Financial Services Innovation at Carnegie Mellon University and The Ripple Foundation. Maxi Guennewig gratefully acknowledges support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the CRC TR 224 (Project C03).

†Tepper School of Business, Carnegie Mellon University.

‡Department of Economics, University of Bonn.

1 Introduction

Blockchains are decentralized, distributed ledgers. They are records of (sequenced) data, possibly transactions as with bank ledgers, maintained by a disperse group of self-interested individuals or users. Unlike ledgers maintained by banks, governments, or other parties, in a blockchain setting, there is no single party responsible for maintenance and security of the ledger, nor is there a single party to resolve conflicts in the ledgers held by distinct individuals. For such decentralized, distributed ledgers to be socially valuable, the dispersed group of individuals who each maintain their own record of the blockchain's data must come to an agreement on what is the "correct" version of the ledger. In other words, establishing and maintaining *consensus* among the individuals who maintain the data is paramount for blockchain systems.

A long tradition from theoretical computer science studies various consensus protocols and their security properties (see Lynch (1996) for a textbook treatment of such protocols). The analysis of consensus protocols inspired by this literature typically proceeds by positing strong assumptions on the nature of agents' strategies, effectively imposing certain behavioral types. So-called honest agents have singleton strategy sets: they are required to follow the consensus protocol as proposed by the protocol designer. In the face of these honest types, models typically also include malicious or Byzantine agents who may adopt arbitrary strategies (with potentially unlimited costs) to disrupt the nature of consensus among the honest types. Classical results on consensus protocols explore the extent to which these protocols can achieve specific security properties under varying assumptions on the relative mix of honest and malicious agents.[1]

Nakamoto (2008) famously proposed a novel protocol that powers the Bitcoin blockchain. This protocol makes no underlying assumptions about the mix of honest or malicious types. Indeed, Nakamoto effectively treats protocol participants as "rational" and free to choose any record validation process they find individually optimal. In practice, the Bitcoin consensus protocol has been remarkably secure since its inception.

However, almost two decades after Nakamoto first published the Bitcoin protocol, we have a limited understanding of why the Bitcoin protocol is secure in the absence of guarantees on the number of honest miners in the system. What is well understood are the limitations of the specific mining strategies proposed in Nakamoto (2008). In Bitcoin, the most widely

[1] A leading example is Fischer et al. (1985) who show that single faulty node makes it impossible to reach consensus among deterministic asynchronous processes.

studied blockchain, the proposed consensus protocol (or strategy) is for users to agree that the longest chain—technically, the chain that represents the most computational work—is the correct chain. For example, Biais et al. (2019) and Budish (2025) have both shown that if the value of modifying the data on the blockchain is sufficiently large or the users' ability to write data to the blockchain is not evenly distributed, then the longest chain consensus protocol is not sufficient to prevent users from modifying past data. Such critiques call into question the economic viability and security of blockchain-based ledgers.

Our paper's main contribution is to develop a new theoretical model of blockchain consensus that permits us to study and develop new protocols—strategies for appending data to blockchain ledgers—that generate consensus. The key innovation in our model relative to the nascent literature on consensus with rational agents is that agents in our model have direct preferences over information in different versions of the blockchain ledger. This innovation allows us to derive the relevant incentive constraints for any candidate protocol as opposed to only that proposed in Nakamoto (2008). We use this new model to derive a new protocol that generates consensus as an equilibrium outcome even when an individual agent may have an outsize ability to append data to the blockchain (e.g. concentrated mining power under Proof-of-Work or concentrated stake under Proof-of-Stake). We demonstrate how a simple modification of Nakamoto (2008)'s longest chain strategy may resolve the double-spend attacks raised as a critical technological fault underlying the Bitcoin protocol by Budish (2025). Finally, we explore how technological features of the blockchain protocol interact with strategies that generate equilibrium consensus.

We begin by proposing a dynamic model of decentralized, distributed record-keeping. We interpret the records being kept in our model as records of transactions involving agents' balances of a unit of account—as in the Bitcoin blockchain. However, we may also interpret this data as computational code intended to be conducted by the network of validators as well—as with smart contracts in the Ethereum blockchain. In each period, rational, self-interested agents we call "miners" decide where to append a new block of transactions to a blockchain. Given this "locational" choice, the likelihood a miner's block is added to the blockchain depends on her (exogenous) mining power, which we model as a probability. If a miner's block of data is added, the block not only includes transactions but also a mining reward for that miner—the reward is an increment to the miner's balance of a unit of account on the blockchain. This process of mining implies that in any period, the blockchain resembles a graph (or tree) of blocks. Each possible path from the first, "genesis" block to any other block in the tree of blocks, which may be thought of as a fork from any chain in the tree, represents a distinct ledger with a possibly distinct value for each miners' aggregate balances.

We assume that miners value these ledger balances because positive net balances are in theory spendable for physical goods, services, or other currencies. However, these balances are only spendable if other miners agree that the balances are valid. To formalize a notion of agreement, we assume miners "vote with their feet" or, more aptly, their mining power. That is, we assume a given miner values balances on ledgers being mined by other miners more than balances on ledgers that are not being mined by other miners. Miners then have a direct preference for agreement or consensus. Whether consensus is achieved, however, is an equilibrium outcome.

To capture the idea that balances are spendable, we consider transactions involving negative balances, or spend transactions, as corresponding to some form of consumption off the blockchain. A spend transaction, once added to the blockchain graph, persists in the graph in perpetuity. We assume that the miner is compensated for this spend by the one-time receipt of "goods" from an external party. To the extent settlement of goods is delayed (as it is in practice with individuals typically waiting for at least six additional confirmed blocks before settlement in the case of the Bitcoin protocol), the settlement transaction may also involve a premium paid to the miner who endures the cost of the negative transaction once it is mined before settlement occurs.

In this environment, we then evaluate different strategies that may generate consensus on ledgers. We first show that Nakamoto's proposed longest chain consensus protocol features the same flaws in our model as found in earlier work: longest chain strategies fail to be equilibrium strategies when mining power is concentrated and an individual miner has sufficient (net) balances of unit of account on some chain that is not the longest chain. Miners therefore find it profitable to engage in double-spending attacks, a result obtained in Biais et al. (2019) and Budish (2025). In this sense, our more general model of shared record keeping features the same incentive problems raised in the earlier literature.

Critically, our more general model allows us to identify the key features of the underlying transaction data that cause incentive problems for Nakamoto's proposed strategies. First, miners may have incentives to remove "spend" transactions. Once a miner spends balances in order to obtain physical goods or other services, they have incentives to work on forks from the tree that omit this spend transaction. To the extent other miners are assumed to follow the longest chain protocol, it is straightforward to develop conditions where so-called double-spending or 51% attacks are profitable for a rational miner. The reason, as identified in both Biais et al. (2019) and Budish (2025), is that the costs of a double-spend attack do not scale with the potential benefits. Second, miners may have incentives to work on ledgers that are not the longest chain but involve large transactions that increase their balances in the hopes that these ledgers become the longest chain.

We then use our model to show how modifications to the proposed strategies of miners can resolve the problematic incentives that arise in these situations. To address the first concern—removing spend transactions—a modified version of longest-chain consensus, which features "checkpoints" is an equilibrium even when a single miner may have a large amount of mining power and the chain contains arbitrarily large spend transactions.

The checkpoint rule, embedded in miners' strategies, selects a single existing block in each period as a function of the history of the blockchain each miner observes. Miners strategies call on them to ignore any blockchain forks that begin behind the current checkpoint. These checkpoints ensure that if any miner attempts to omit data or blocks behind the current consensus checkpoint, then no miners will treat this new deviation chain as the correct chain regardless of its length. In this way, checkpoints are a strategic solution to double-spending attacks, the central problem identified in Budish (2025). This result suggests that the concerns raised in previous work may be a feature of a particular candidate equilibrium strategy and not a concern about the technology of blockchain itself.

An important implementation consideration of our checkpoint equilibrium is network latency. Since the entire network of miners does not see new blocks at the same time it is likely that accidental forks will occur.[2] If information on the checkpoint is also latent, miners may disagree on the checkpoint block, potentially causing a temporary fork due to latency to become permanent. Such disagreement would undermine the usefulness of the blockchain. Optimizing the checkpoint block—choosing the settlement lag—would require comparing the cost of a settlement lag with the likelihood of a permanent fork.

A more subtle issue to resolve with checkpoint strategies are miners' incentives to shift consensus to their most preferred chain ahead of the checkpoint. Here, we construct an equilibrium which relies on a form of mining-weighted approval voting. We term this strategy the approval weighted chain rule. The approval weighted chain rule calls on miners to add blocks to the chain where the total mining power (i.e., the sum of probabilities) of miners who have balances on the chain is largest. Recall that the longest chain rule failed because

[2] In Bitcoin, for example, it takes about 11 seconds for all nodes to hear of a new block. Average new-block arrival time on Bitcoin is designed to be 600 seconds. Solving a block is Poisson and so a second block will arrive before all nodes are informed that a new block has already been solved about 1.8% of the time (11 *seconds*/600 *seconds* $\approx$ 1.8%).

miners with large mining power preferred to work on a shorter chain on which they have large positive balances, because they expect to turn this into the longest chain with sufficiently high probability. Under the approval weighted chain rule miners are called to work on the shorter chain if it contains such balances for large miners. Neither small nor large miners then have incentives to deviate from the equilibrium strategy.

However, key to obtaining this result is a technological restriction: until all forks ahead of the checkpoint are resolved, miners can only add blocks to the blockchain that do not contain any transaction data. Otherwise, miners can "bribe" other miners to mine their preferred chain by including small positive transactions into the newly added block, which increases the chain's approval weights. On the equilibrium path, the approval weighted chain consensus then generates a graph with a single path of data and thus perfect consensus while disincentivizing deviations by miners with large mining power.

Contribution to the literature. We develop a dynamic, game-theoretic model of blockchain consensus in which rational, self-interested miners (validators) have well-defined preferences over the data recorded on the blockchain and strategically decide how to update it. Our paper is therefore most closely related to papers on the economics of blockchain consensus and security.

Biais et al. (2019) and Budish (2025) also present game-theoretic models of blockchain environments, showing that blockchains are susceptible to double-spending attacks if mining power is concentrated and transaction values are large. In their analyses, honest miners follow the longest chain rule and therefore switch to an attacker's fork once it becomes the longest chain. The expected cost of a successful attack is given by the expected cost of creating a fork and extending it to surpass the original chain—it is therefore fixed conditional on the honest miners' computing power. The expected benefit of an attack is increasing in the size of the spend transaction. Budish (2025) argues that, since the honest miners' computing power is increasing in the cost of using the blockchain (by a free entry logic), mining rewards must scale with transaction size on the blockchain.[3, 4]

[3] Biais et al. (2019) further highlight that consensus can be fragile: miners can coordinate on creating forks if mining strategies exhibit strategic complementarities. With limited competition among miners, forks can persist.

[4] Other papers take honest miners' strategies as given. Gans and Halaburda (2023) generalize and extend the analysis of the majority attack. They find that the cost of an attack may be lower when honest miners' endogenously response by adjusting their computer power. Saleh (2021) studies Proof-of-Stake consensus protocols and finds that attacking the blockchain is not profitable if the market capitalization of the blockchain-native coin is sufficiently large and the settlement lag sufficiently long. John et al. (2020) study security properties of Proof-of-Work and Proof-of-Stake consensus protocols when blockchain capacity

Our contribution is to demonstrate that incorporating a simple history-dependence in the form of checkpoints into the longest chain rule can effectively prevent double-spending attacks. The profitability of such attacks is therefore a feature of a particular consensus protocol (or mining strategy) and not a concern about the technology of blockchain itself. Our general model enables a deeper analysis of the incentive structures underlying different consensus mechanisms. We also propose a new protocol and explore technological features that interact with strategies to generate equilibrium consensus.

Garratt and van Oordt (2023) argue that a double-spending attack may not be profitable if the hardware required is specialized and cannot be repurposed, raising the fixed cost of the attack. Moroz et al. (2020) show that when the victim of a double-spending attack can counterattack in the same way as the attacker, then this results in a variant of the 'War of Attrition' game. The threat of a counterattack induces a subgame perfect equilibrium of this game in which no attack occurs in the first place. Chiu and Koeppl (2022) argue that more intensive miner competition (i.e., more widely distributed mining power) and long settlement lags, which increase in transaction size, can help render double-spending attacks unprofitable as the cost to create a competing, longer chain becomes excessive. We highlight that neither large fixed costs, the ability to counterattack, nor extensive settlement lags are necessary to achieve blockchain security.[5]

Our paper shares its objectives with Halaburda et al. (2022). They also develop a game-theoretic model of blockchain, although with reduced form payoffs, emphasizing Knightian uncertainty to capture the spirit of Byzantine Fault Tolerance (BFT) in the computer science literature. More recently, Leshno et al. (2024) highlight the 'community response' to override nodes with a corrupted ledger and present a protocol which formalizes this feature. One aspect of the protocol is that nodes finalize transactions once they are certain that the ledger has not been corrupted, a feature which bears similarities with the checkpoint equilibrium

constraint are alleviated.

[5] Pagnotta (2022) studies equilibrium multiplicities that arise if blockchain security depends on the real value of blockchain-native coins. Makarov and Schoar (2021) study the Bitcoin blockchain ecosystem. One of their findings is that mining power is highly concentrated among a small number of mining pools. Cong et al. (2021) show that the rise of centralized mining pools does not necessarily undermine decentralization, which, as frequently argued, needs to be sufficiently high for blockchains to be secure. Auer et al. (2021), Amoussou-Guenou et al. (2024), and Benhaim et al. (2023) study consensus on committee-based or permissioned blockchains. Bakos and Halaburda (2021) compare the security properties of permissioned and permissionless blockchains. Similar to Budish (2025), attacks on permissionless blockchains are profitable if the value of an attack is large relative to block rewards. See, among others, Li (2023) for a study on the security of blockchain scaling solutions. Kang (2023) studies a reputation-based mechanism to address double-spending attacks. Merchants delay the delivery of consumption goods if the payment is done using a wallet which has been found to have double-spent in the past.

described in this paper.[6] Our contribution relative to these papers lies in our framework, which allows us to study how the specific data recorded on the blockchain, such as transactions, impact incentives of rational nodes across different consensus protocols. We then present protocols (or mining strategies) which prevent double-spending attacks, and explore technological features that interact with strategies to generate equilibrium consensus.

2 A Model of Blockchain

In this section, we develop a model to analyze blockchain consensus. In this model, in each period, miners add a block of data to an existing graph of blockchain data. A block includes units of account on the blockchain ledger as well as, in principle, other data. This model features no latency in the sense that each individual in the model perfectly observes each addition to the blockchain.

Preliminaries. There are $N \in \mathbb{N}$ miners, each infinitely lived and with a rate of time preference $\delta \in (0, 1)$. Time is discrete. In each period t, each miner i proposes a location to add a *block*, $b_{i,t}$, of data. A block consists of three components: hash data, mining rewards, and data entries, e.g. transaction data (as in the Bitcoin blockchain) or computations to be conducted (as in the Ethereum blockchain).[7] The hash data is determined technologically and and is not relevant for our model beyond the fact that it implies a chained data structure. We let $R_{j,b_{i,t}}$ denote the mining rewards in block $b_{i,t}$ for miner j. We assume that mining rewards have the property that $R_{j,b_{i,t}} = \bar{R} \in (0, \infty)$ if $i = j$ and $R_{j,b_{i,t}} = 0$ for $j \neq i$. This implies that only miner i earns a reward if block $b_{i,t}$ is added to the blockchain. In addition, we represent the data for each miner j in any block $b_{i,t}$ proposed by miner i by $Y_{j,b_{i,t}} \in \mathbb{R}$. We assume that the data in a given time period t are exogenous and identical across all miners' blocks, and hence we write $Y_{j,b_{i,t}} = Y_{j,b_t}$. Miners' blocks therefore only differ in terms of block rewards.

A blockchain, in the language of graph theory, is an arborescence. It is a directed graph in which from the genesis block b_0 to any other block b there is exactly one directed path

[6] Other papers follow the approach typically taken in the computer science literature to study protocols involving checkpoints, e.g. Buterin and Griffith (2017) and Neu et al. (2021). Karakostas and Kiayias (2021) consider a consensus protocol with checkpoints set by an external party and discuss how to decentralize the process on an external blockchain. Sankagiri et al. (2021) develop a protocol which incorporates checkpoints into the longest chain rule.

[7] Technically, mining rewards are simply a transaction, but it is useful for us to separate them.

from b_0 to b. Let $\mathcal{B}(G_t)$ denote the set of all blocks in the graph G_t. Let (b', b) denote the edge from block b to block b', leading away from the genesis block. Denote by $\mathcal{E}(G_t)$ the set of all edges that link the blocks in graph G_t. Let $\mathcal{G}_t$ represent the set of all possible graphs with t blocks and $\mathcal{G} = \bigcup_{t=0}^{\infty} \mathcal{G}_t$. Let $H_t \in \mathcal{H}_t = \bigcup_{\tau=0}^{t} \mathcal{G}_\tau$ denote the history of the graph at time t, and $\mathcal{H}_t$ the set of all possible histories at time t.

Each miner's action in period t is to choose a location to attempt to add block $b_{i,t}$. A location choice of miner i in period t is a mapping $a_{i,t} : G_t \to \mathcal{B}(G_t)$. Miners' location choices stochastically determine the state of the graph in the subsequent period. Specifically, we assume that each miner's block is added (in the location of choice chosen by miner i) probabilistically with at most one miner adding a block in a given period. Let $p_i \in (0, 1)$ denote the probability that miner i successfully adds a block to the existing graph with $\sum_{i=1}^{N} p_i = 1$. This probability represents the mining power of miner i and we treat it as exogenous.

Given a graph G_t and the location choices of miners $(a_{i,t})_{i=1}^{N}$, the graph in the subsequent period is $G_{t+1} = G_t \bigcup (b_{i,t}, (b_{i,t}, a_{i,t}))$ with probability p_i. In words, the graph G_{t+1} is the same as the graph G_t but includes a new node $b_{i,t}$ and a new edge from $b_{i,t}$ to $a_{i,t}$.

Chains. Before turning to the structure of preferences and payoffs, it is useful to create notation to describe the various databases that are represented in a graph, G_t. We interpret each path through the graph, from the origin node to any other node, as a *chain*. Note that each chain may represent a different database than any other chain. Furthermore, recall that the blockchain protocol imposes that every block has a unique parent block (although it may have more child blocks). Hence, the path backwards from any block to the genesis block is unique.

For any graph G_t and block $b \in \mathcal{B}(G_t)$, define the chain $C(b, G_t)$ as the unique path from block b back to the genesis block b_0. Let $\mathcal{C}(b, G_t) \subseteq \mathcal{B}(G_t)$ denote the set which contains the blocks on the chain from b to b_0:

$$\mathcal{C}(b, G_t) = \left\{ \{b, b_n, \ldots, b_1, b_0\} \in \mathcal{B}(G_t) \,\middle|\, (b, b_n), (b_n, b_{n-1}), ..., (b_1, b_0) \in \mathcal{E}(G_t) \right\} \qquad (1)$$

We say that block b_n is on the chain $C(b, G_t)$ if $b_n \in \mathcal{C}(b, G_t)$. Furthermore, define $\#C(b, G_t)$ as the number of blocks in the chain. We refer to this number as the *length* of the chain.

Finally, we refer to blocks that have no edges leading away from the genesis block as *terminal blocks*, and define $\mathcal{T}(G_t)$ as the set of terminal blocks in graph G_t.

Preferences. We now propose a specific functional form for the period payoff that has two components: first, a flow payoff derived from the data contained on the blockchain; and second, a flow payoff derived from consumption of real goods.

We assume that miners derive a linear flow utility from the weighted sum of their data entries on the blockchain, given by

$$(1-\delta)\sum_{b\in\mathcal{B}(G_t)} q_{i,b,t}\left(Y_{i,b}+R_{i,b}\right). \tag{2}$$

where $q_{i,b,t}$ denotes the weight of block b for miner i at time t. We link these weights to miners' actions and computing power: if a miner chooses a location in $\mathcal{B}(G_t)$, we say that the miner works on the chain from the origin to that existing block. If more miners work on the same chain, then the data on that chain have a larger weight. In particular, we assume that blocks are weighted according to the other miners' computational mining power allocated to those blocks:

$$q_{i,b,t} \;=\; \frac{\sum_{\{j\neq i: b\in\mathcal{C}(a_{j,t},G_t)\}} p_j}{\sum_{\{j\neq i\}} p_j}. \tag{3}$$

Equation (3) captures the notion that data are more valuable if they are written in blocks on which more miners agree. Miner i receives value for any data written in blocks that are on the blockchain associated with some other miner's location choice $a_{j,t}$. To the extent there is disagreement, miners obtain value from their data as long as some other miners apply their mining power to these blocks. When there is full consensus and all miners choose the same location, then all data entries in blocks on that chain receive their full value of 1.

It is natural to think of the data Y and R as representing units of account held on the graph G_t. These transaction data could be positive or negative with the interpretation that positive data represent payments received while negative data represent payments sent. Going forward, we refer to these units of account as *coin balances.* Then $Y_{i,b}+R_{i,b}$ represents miner i's coin balances in block b. The miner's flow utility from her coin balances in this block increases as more miners recognize the block as valid.

Given this interpretation, the second component of miners' period payoffs explicitly links consumption to negative data entries, or *spend transactions.* We assume consumption goods are fairly priced and that settlement occurs with a delay, reflecting existing blockchain norms. For instance, Bitcoin recommends finalizing spend transactions after waiting for six additional confirmed blocks (i.e., six blocks appended on a single chain which achieves full consensus). For simplicity, we assume a one-block delay (on a single chain). Miners derive linear flow utility from consumption. Fair pricing then requires that each unit of coin balances purchases $1/\delta$ units of consumption. Let the absolute value of spend transactions be denoted by $Y^-_{i,b} = |Y_{i,b}| \cdot \mathbb{1}\left\{Y_{i,b} < 0\right\}$. Consumption as well as the flow utility derived from consumption at time t are then given by

$$\sum_{b \in \mathcal{B}(G_t)} \frac{Y^-_{i,b}}{\delta} \cdot \lambda_t(b, H_t). \tag{4}$$

where $\lambda_t(b, H_t)$ is an indicator that takes the value of one if settlement takes place in time period t. We provide the precise definitions of $\lambda_t(b, H_t)$ in Sections 3 and 4 below.

Miner i's flow utility at time t can then be represented by

$$u^i(\boldsymbol{q}_{i,t}; H_t) \;=\; \sum_{b \in \mathcal{B}(G_t)} \left[(1-\delta) q_{i,b,t} \left(Y_{i,b} + R_{i,b}\right) \;+\; \frac{Y^-_{i,b}}{\delta} \cdot \lambda_t(b, H_t) \right]. \tag{5}$$

where $\boldsymbol{q}_{i,t} \;=\; \left(q_{i,b,t}\right)_{b \in \mathcal{B}(G_t)}$. Note that in this formulation, preferences are a function of the value of coin balances (given the current actions of miners) and the history of the graph.[8]

As an example, consider one miner m called 'Satoshi.' Suppose Satoshi has 1 unit of account on the genesis block $R_{m,b_0} = 1$. Satoshi also has a spend transaction $Y_{m,b_1} = -1$ in the second block b_1, which is added to the blockchain at the end of period 1. Suppose further that there is a single chain in the graph in every period. In period 2, Satoshi's flow utility over the graph is 0 because her balance aggregated over the two blocks is zero. Unless future blocks contain more transactions for Satoshi, her utility over the graph (excluduing consumption of real goods) will continue to be zero in all future periods. However, when a

[8] Our specification of preferences can be microfounded within a monetarist framework, following the approach of Lagos and Wright (2005) and Rocheteau and Wright (2005). In this setting, money trades at a premium. The reason is that money (or coin) balances represent past "work," creating an option value tied to exchanging goods for money in frictional goods markets. In our framework, this premium is reflected in the flow utility derived from coin balances, and it is larger the more miners consider these balances as valid. Meanwhile, the consumption flow utility reflects the utility derived from realized consumption purchases in both frictional and frictionless goods markets using coin balances.

In frameworks such as Lagos-Wright, agents only value money if there is a strictly positive probability of spending it in frictional markets; otherwise, they immediately exchange their balances for consumption goods in frictionless markets. This is not true in our model. However, our specification indirectly captures this behavior. For simplicity, we take the transaction process as given. In our model, miners then derive the flow utility from holding coin balances, even if they never spend them with probability one. In other words, miners receive the same lifetime payoff whether they never spend their balances or spend them entirely at once. If miners did not derive utility simply from holding balances and instead chose their transactions optimally, they should find it optimal to immediately spend them, as in the Lagos-Wright framework.

third block is appended to block b_1 at the end of period 2, satoshi's spend transaction in the second block vests. She then earns the consumption flow utility equal to $1/\delta$ in period 3. Aggregating and discounting these payoffs from the perspective of period 1 or of period 2, her lifetime utility is 1. Of course, once Satoshi has derived the consumption flow utility in period 3, her discounted lifetime utility is 0. She would then benefit from the construction of an alternative path through the blockchain where her total balance on the consensus chain is 1 instead of 0.

Strategies and Equilibrium. Since miners take transactions as given, they only choose the location of their new block. We focus on *public strategies*, which only depend on the publicly observed sequence of graphs. Formally, a public strategy for miner i, σ_i, is a sequence of mappings from the set of all possible public histories into a set of pure actions,

$$\sigma_i = (\sigma_{i,t})_{t=0}^{\infty}, \text{ where } \quad \sigma_{i,t} : \mathcal{H}_t \to \mathcal{B}(G_t). \tag{6}$$

Our equilibrium concept is *perfect public equilibrium*, that is, subgame perfect equilibrium in public strategies. We therefore insist that a strategy profile $\sigma = (\sigma_i)_{i=1}^{N}$ is an equilibrium if and only if each miner's strategy is a best response to other miners' strategies for each history $H_t \in \mathcal{H}_t$ at each date $t \geq 0$.

Due to their distributed nature, blockchain databases occasionally generate conflicting chains accidentally. (For example, it is well understood that with Bitcoin, two miners may occasionally find a valid block at roughly the same time generating an accidental fork from the perspective of other miners.) For this reason, we view the robustness of strategies that are subgame perfect as an important feature of equilibrium analysis of blockchains.[9]

An advantage of studying public perfect equilibria with discounting in our environment is that we may apply and use the one-shot deviation principle. The literature on Nakamoto consensus has routinely studied complex, multi-period deviations and the incentives individuals miners may have to pursue these (see Carlsten et al. (2016) and Eyal and Sirer (2018) for leading examples). The one-shot deviation principle allows us to study these complex strategies as one-shot deviations from particular subgames. But more powerfully, we need not worry about more complex deviation strategies once we construct a strategy that is

[9] Our focus on public equilibria is natural given the assumptions we have made that mining locations are public information. In practice, at least for short periods of time, miners may be able to hide their mining activity. In such a case, one would want to also permit private actions and study equilibria with private monitoring in our environment.

immune to one-shot deviations from all histories.

3 Longest Chain Rule

In this section, we analyze Bitcoin's proposed equilibrium strategy, the *longest chain rule* (Nakamoto, 2008). The gist of the longest chain rule is that miners choose the block that defines the longest chain as the predecessor for their potential block. This is a simple coordination mechanism in that it depends only on the current graph G_t. To ease notation, let

$$\mathcal{B}^{LC}(G_t) \;=\; \underset{b\in\mathcal{B}(G_t)}{\operatorname{argmax}} \;\#C(b,G_t) \tag{7}$$

denote the set of (terminal) blocks in the graph G_t such that the chain to these blocks has the largest number of blocks. We now describe the settlement of consumption goods and miner strategies under the longest chain rule.

Settlement. We assume that merchants deliver consumption goods that correspond to spend transactions contained in block b at time t if two conditions are met. First, at least one block has been appended to block b. Second, block b is contained in the unique longest chain for the first time. More formally, we write

$$\lambda_t(b,H_t) \;=\; \begin{cases} 1 & \text{if} \quad t=\inf\left\{\tau\geq 0:\ \exists\, b'\neq b \ \text{ s.t. }\ b\in\mathcal{C}(b',G_t),\ b'\in\mathcal{B}^{LC}(G_t),\ |\mathcal{B}^{LC}(G_t)|=1\right\}, \\ 0 & \text{otherwise.}\end{cases} \tag{8}$$

Let $\Lambda_t(b,H_t)=\prod_{s=0}^{t}(1-\lambda_s(b,H_s))$ denote an indicator function that takes the value of 1 if a spend transaction has not vested yet, and 0 otherwise.

Strategies. We now formalize the notion that the longest chain rule calls for miners to extend the longest chain. If $\mathcal{B}^{LC}(G_t)$ is a singleton, miners are called choose the only block in this set as predecessor.

If the graph features multiple longest chains so that $\mathcal{B}^{LC}(G_t)$ is not a singleton, the necessary tie-breaking rule for the longest chain rule to be a candidate equilibrium strategy is intuitive. By the coin value equation (3), miner i's location decision does not influence her own static payoffs but only the static payoff of other miners. However, starting from a graph with multiple longest chains and only positive transactions (and block rewards), if the miner has strictly higher coin balances on one of the longest chains, than by mining in that location she strictly increases the likelihood that this chain becomes the single longest chain and thus the consensus chain in next period. She then earns the flow utility associated with her balances in perpetuity.

A similar logic applies to unvested spend transactions. Once a block is added to one of the longest chains and consensus is achieved, the spend transactions on that chain vest immediately and generate a utility of $1/\delta > 1$ per unit of spending. Miners therefore earn a net benefit when spend transactions vest.

Hence, the only tie-breaking rule that is immune to one-shot deviations is the rule that prescribes miners choose the terminal block on their most preferred longest chain, which contains the largest sum of positive transactions, block rewards, and the net benefit from unvested spend transactions. We summarize this observation in the following Lemma:

Lemma 1. *If the longest chain rule is a (perfect public) equilibrium, then for any graph G_t such that $\mathcal{B}^{LC}(G_t)$ is not a singleton, the longest chain rule must satisfy*

$$\sigma^{LC}_{i,t}(H_t) \;=\; b^*_{i,t} \;\equiv\; \underset{b\in\mathcal{B}^{LC}(G_t)}{\operatorname{argmax}} \sum_{b'\in\mathcal{C}(b,G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b,H_t) \right). \tag{9}$$

The proof is in Appendix A.1. With this lemma in hand, we now proceed by obtaining conditions such that the longest chain rule is a (perfect public) equilibrium. We therefore examine conditions under which no profitable one-shot deviations exist from any possible graph of data. Developing these conditions will help to better understand the situations when deviations like double-spend attacks may arise.

Equilibrium. It is useful to define a set of blocks for which one of two things is true. First, a block is a terminal block on a fork which is one block shorter than the longest chain. Or second, a block is the parent block of the terminal block of a longest chain. More formally, the set is defined as:

$$\mathcal{B}^{-1}(G_t) \;=\; \{b' \in \mathcal{B}(G_t) : \#C(b',G_t) \;=\; \max_{b\in\mathcal{B}(G_t)} \#C(b,G_t) - 1\}. \tag{10}$$

Under the tie-breaking rule implied by (9), the only relevant one-shot deviations are those to some block $b \in \mathcal{B}^{-1}(G_t)$. For any other deviation, if the miner successfully adds her block and then reverts to the candidate equilibrium strategy, she immediately abandons her block which has not become part of a longest chain. Since no other miner is working on the fork either, she thus forgoes the opportunity to have earned the rewards and transactions associated with mining that block to a longest chain.

To compare incentives to append a new block to the end of the longest chain or to some block in $\mathcal{B}^{-1}(G_t)$, consider a thought experiment where miner i adds block $b_{i,t}$ to the graph for sure in period t.

If she adds her block to her preferred longest chain, $b^*_{i,t}$, then that chain becomes the single longest chain and thus the consensus chain in the subsequent period. Consensus is achieved from time $t+1$ onwards. The miner earns the balances in block $b_{i,t}$ as well as all her balances on preferred longest chain in perpetuity, starting in period $t+1$. Furthermore, she derives the consumption flow utility $Y^-_{i,b}/\delta$ due to unvested spend transactions contained in all blocks b which lie on the chain running to $b^*_{i,t}$ at time $t+1$. Miner i also derives the consumption flow utility $Y^-_{i,b_t}/\delta$ once any spend transaction in block $b_{i,t}$ vests at time $t+2$. Thus, miner i enjoys the following *continuation utility* from time $t+1$ onwards based on the data present on the blockchain at the end of time t (after miner i has added block $b_{i,t}$):

$$U^i_{t+1}(b_{i,t}, b^*_{i,t}; H_t) \;=\; Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y^-_{i,b_t}}{\delta} + \sum_{b' \in \mathcal{C}(b^*_{i,t}, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right). \quad (11)$$

Suppose instead she adds her block to a chain which is one block shorter. This deviation extends a lack of consensus into period $t+1$ as the number of longest chains in that period increases by one. Of course, since all miners including miner i will then follow the longest chain rule from time $t+1$ onwards and only one miner will successfully append a block at time $t+1$, consensus will be achieved from time $t+2$ onwards.

At time $t+1$, miner i derives flow utility from the value of balances on each longest chain on which at least one miner $j \neq i$ is working. This value is given by

$$\begin{aligned} F^i_{t+1}\left(b_{i,t}, \hat{b}; H_t\right) \;=\; & \sum_{\{j \neq i:\, b^*_{j,t+1} = b_{i,t}\}} \frac{p_j}{1-p_i} \cdot \left(Y_{i,b_t} + \bar{R} + \sum_{b' \in \mathcal{C}(\hat{b}, G_t)} (Y_{i,b'} + R_{i,b'}) \right) \quad (12) \\ & + \sum_{b \in \mathcal{B}^{LC}(G_t)} \sum_{\{j \neq i:\, b^*_{j,t+1} = b\}} \frac{p_j}{1-p_i} \cdot \sum_{b' \in \mathcal{C}(b, G_t)} (Y_{i,b'} + R_{i,b'}). \end{aligned}$$

Recall that the value of miner i's balances is proportional to the computing power of (other) miners working on each chain. When other miners work to extend miner i's preferred longest chain, the block she mined in period t as well as all other transactions and block rewards along that chain have value given by the first line in (12). When other miners work on other chains, transactions in those chains—which necessarily exclude the block miner i added in period t—have value given by the second line in (12).

As discussed above, the miners achieve consensus in period $t+2$. Miner i's expectation over the continuation utility derived from time $t+2$ onwards based on data present in the blockchain at the end of period t is given by

$$\begin{aligned}\mathbb{E}_t\left[U^i_{t+2}\left(b_{i,t},\hat{b};H_t\right)\right] = &\left\{p_i + \sum_{\{j\neq i:\, b^*_{j,t+1}=b_{i,t}\}} p_j\right\}\left(Y_{i,b_t} + \bar{R} + \frac{Y^-_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(\hat{b},G_t)}\left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right)\right)\\ &+ \sum_{b\in\mathcal{B}^{LC}(G_t)}\ \sum_{\{j\neq i:\, b^*_{j,t+1}=b\}} p_j \cdot \sum_{b'\in\mathcal{C}(b,G_t)}\left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right)\end{aligned} \tag{13}$$

With probability $\left\{p_i + \sum_{\{j\neq i:\, b^*_{j,t+1}=b_{i,t}\}} p_j\right\}$, miner i's preferred chain is the consensus chain. The miner then enjoys the value of rewards and transactions in the blocks on this chain as well as the value of newly vested spend transactions, reflected in the first line of (13). Instead, each other chain to $b \in \mathcal{B}^{LC}(G_t)$ becomes the new consensus chain with probability $\sum_{\{j\neq i:\, b^*_{j,t+1}=b\}} p_j$ yielding similar payoffs, captured by the second line of (13).

We are now ready to state our first main result. Proposition 1 shows that miners are incentivized to mine the longest chain only for a set of restrictions on transactions and mining power.

Proposition 1 (Longest Chain Rule is a Perfect Public Equilibrium). *The longest chain rule is a perfect public equilibrium if for every history H_t, for every block $\hat{b} \in \mathcal{B}^{-1}(G_t)$, and for every miner i:*

$$U^i_{t+1}\left(b_{i,t}, b^*_{i,t}; H_t\right) \ \geq\ (1-\delta)\cdot F^i_{t+1}\left(b_{i,t},\hat{b};H_t\right) \ +\ \delta\cdot\mathbb{E}_t\left[U^i_{t+2}\left(b_{i,t},\hat{b};H_t\right)\right] \tag{14}$$

The proof is in Appendix A.2. Miner i only finds it profitable to follow the longest chain rule if the benefit from doing so—turning the preferred longest chain into the consensus chain immediately and vesting the new transactions and block rewards for sure—outweighs the benefit from deviating. That is, it outweighs the benefit associated with creating a new preferred longest chain which yields some flow utility and becomes the consensus chain in the subsequent period with some probability. Note that, since there is consensus both after following the longest chain rule and after a one-shot deviation once a block has been appended at the end of time $t+1$, all data in blocks added from time $t+1$ onwards do not affect incentives at time t.

It is useful to study a simpler case to help understand the condition in (14). Consider some graph G_t such that there is one longest chain with terminal block b_{l_2} and one fork, which is one block shorter than the longest chain, with terminal block b_f. Figure 1 depicts the blockchain in this scenario.

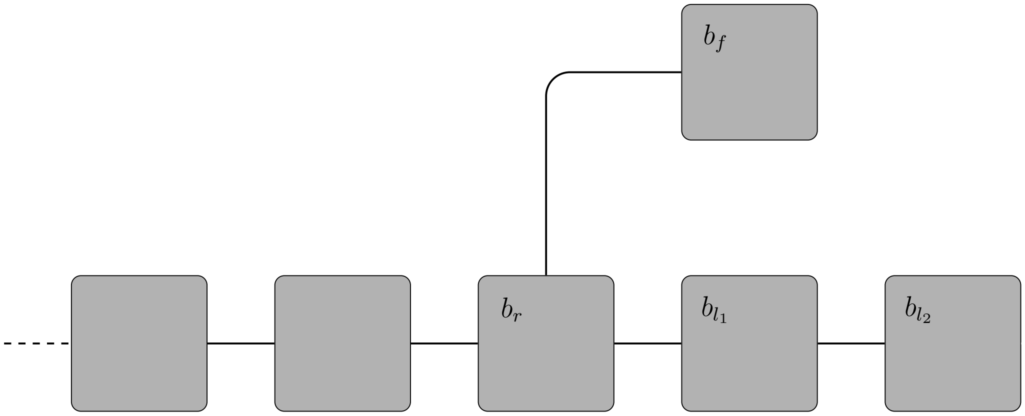

Figure 1: An illustration of the condition in (14).

Suppose miner i is mining some block $b_{i,t}$ such that, should miner i append it to b_f, all other miners find it optimal to work on the previously longest chain: $b^*_{j,t+1} = b_{l_2}$ for all $j \neq i$. Then, miner i does not face a profitable one-shot-deviation from the longest chain rule if

$$\begin{aligned} & Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y^-_{i,b_t}}{\delta} \qquad (15) \\ & \geq \; \delta p_i \cdot \left(\sum_{b' \in \{b_f, b_t\}} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \right) - \left(\sum_{b' \in \{b_{l_1}, b_{l_2}\}} \left(Y_{i,b'} + R_{i,b'} \right) + \frac{Y^-_{i,b_{l_2}}}{\delta} \right) \right) \end{aligned}$$

If miner i appends block $b_{i,t}$ to the only longest chain ending at b_{l_2}, then the longest chain remains the consensus chain. The value of balances on the consensus chain are 1 for all time periods going forward, and the miner earns the transactions and block rewards in $b_{i,t}$ from time $t+1$ onwards. If there is a spend transaction in block b_{l_2}, it vests at time $t+1$ and miner i derives the corresponding flow consumption utility. Similarly, if there is a spend transaction in block $b_{i,t}$, it vests at time $t+2$.

If miner i appends block $b_{i,t}$ to b_f, then period $t+1$ features two longest chains. Since no other miner is working on the now second longest chain, the transactions and block rewards in $b_{i,t}$ have a zero value to miner i at time $t+1$. With probability p_i, the fork becomes the consensus chain in the subsequent period. In that case, the miner earns the transactions and block rewards contained in $b_{i,t}$. She also earns the difference in balances between the fork and the previously longest chain, which is then abandoned. Spend transactions in block b_f and $b_{i,t}$ vest at times $t+2$ and $t+3$, respectively, whereas a spend transaction in block b_{l_2} never vests. With probability $(1-p_i)$, the initially longest chain becomes the consensus chain again. The miner obtains the same payoff going forward as under the longest chain rule, with one important difference: the block $b_{i,t}$ is orphaned. As a consequence, the transactions and block rewards in $b_{i,t}$ have zero value to miner i going forward, and the spend transactions never vest.

We proceed by explaining the three distinct manners in which the condition in (14) fails.

Negative transaction data and double-spending attacks. We now illustrate that the longest chain rule is not robust to negative transaction data and is therefore vulnerable to double-spending attacks. Our argument proceeds in two steps. First, we show that the condition in (14) fails if a miner has a large, vested spend transaction on the longest chain and a competing fork exists that is only one block shorter. Since (14) provides a sufficient condition for the longest chain rule to be a perfect public equilibrium, its failure is a necessary condition for a profitable double-spending attack. Otherwise, the attacker never finds it profitable to extend the fork they created during the attack beyond the longest chain. We then construct a double-spending attack using a two-period strategy. In the first period, the attacker tries to create a fork and, if successful, tries to extend it in the second period. We then identify the conditions under which this more complex strategy is a strictly profitable deviation, i.e., under which a double-spending attack is profitable.

Consider Figure 2. This is the same blockchain depicted in the previous figure, now containing a spend transaction for miner i in block b_{l_1}. Suppose that all blocks b_{l_1}, b_{l_2} and b_f have been appended by miner i and thus contain block rewards $\bar{R}$ for her. Assume that block b_f was appended after block b_{l_2}. Hence, the spend transaction in b_{l_1} has vested and miner i has derived the associated consumption flow utility. For simplicity, further suppose that none of these blocks contain any data for any other miner who all continue working on

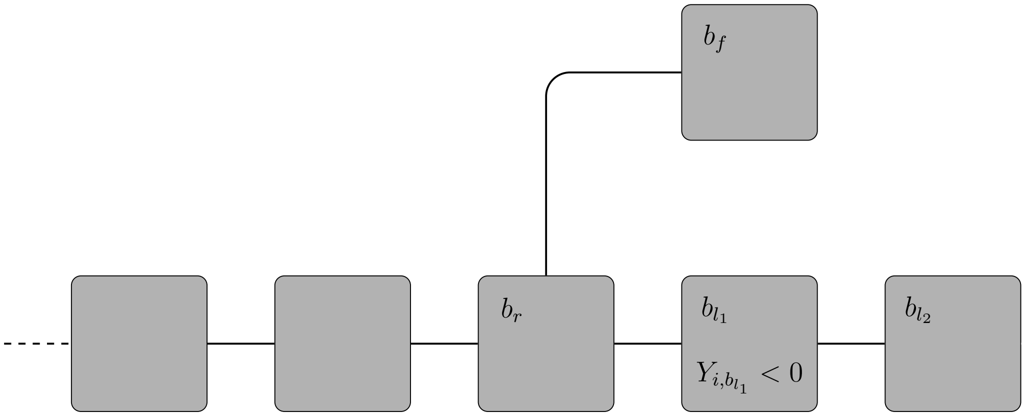

Figure 2: An illustration of incentives to deviate from the longest chain strategy.

the previously longest chain should miner i successfully extend the fork. Equation (15) then simplifies to

$$\bar{R} \;\geq\; p_i \delta \cdot |Y_{i,b_{l_1}}|. \tag{16}$$

In this example, miner i prefers to deviate from the longest chain rule and extend the fork if the spend transaction is sufficiently large relative to the level of block rewards, given her mining power p_i. This is especially true if the longest chain also contains vested spend transactions or the fork contains balances for other miners, who may then also work on the fork in the subsequent period and help miner i establish it as consensus chain. In other words, forking may be profitable even if the inequality above holds. Importantly, the longest chain rule fails in this case because miners may seek to *remove* transaction data from the consensus chain by working on a shorter fork.

We now describe the two-period strategy that amounts to a double-spending attack. Recall that block b_{l_2} was appended before the fork b_f was created in the example above. Now consider the time period $t-1$, i.e., the period before b_f was added and the blockchain contains one single chain (b_r—b_{l_2}). Suppose miner i is mining block b_f. Then, consider the following two-period strategy for miner i:

$$\sigma'_{i,t-1}(H_{t-1}) \;=\; b_r \qquad \text{and} \qquad \sigma'_{i,t}(H_t) \;=\; \begin{cases} b_f & \text{if} \quad b_f \in \mathcal{B}(G_t), \\ b^*_{i,t} & \text{otherwise.} \end{cases} \tag{17}$$

Under this strategy, miner i works on block b_r at time $t-1$ and thus attempts to create a fork. If successful, she continues working on the fork at time t. Miner i reverts to following the longest chain rule at time t if she is unsuccessful at time $t-1$. Miner i also follows the longest chain rule from time $t+1$ onwards. Denote this strategy profile by σ'. Note that miner i's strategy induces the blockchain of Figure 2 if she is successful in appending her block b_f at time $t-1$.

We now provide an example in which this strategy constitutes a strictly profitable deviation from the longest chain rule. In this example, suppose that all transactions for all miners in all time periods are zero other than miner i's spend transaction in block b_{l_1}. Suppose further that miner i has solved blocks b_{l_1} and b_{l_2} (but no other blocks on the chain running to b_r) and that all other miners continue working on b_{l_2} at time $t+1$ if miner i successfully creates and extends the fork at times $t-1$ and t.

Proposition 2 (Profitable double-spending attack). *Suppose $Y_{j,b} = 0$ for all miners $j \in \{1, 2, ..., N\}$ and all blocks b except for $Y_{i,b_{l_1}} < 0$. The strategy profile σ' constitutes a strictly profitable deviation from the longest chain rule if*

$$(1 + p_i\delta) \cdot \bar{R} \;<\; p_i^2\delta^2 \cdot |Y_{i,b_{l_1}}|. \tag{18}$$

The proof is in Appendix A.3. To form an intuition, consider the thought experiment where miner i adds a block at time $t-1$ for sure. She now trades off the opportunity cost of an attack against the expected benefits from a successful attack. The benefit is the discounted absolute value of the spend transaction in block b_{l_1} which the attack would undo at time $t+1$. Her probability of success is given by p_i^2, i.e., the probability of both extending the fork and turning into the unique longest chain. The expected cost are the block rewards in block b_f—which she would earn with probability 1 if she extended the longest chain at time $t-1$—and in the subsequent block, which she appends with probability p_i at time t and would earn with probability 1 on the longest chain. Note that the inequality in (18) implies that the equality in (16) fails, i.e., that is profitable to extend the fork which was created as part of the attack.

Importantly, double-spending attacks may be profitable in practice even if (18) is not satisfied. In our example above, deviating becomes more profitable if other miners also work on the fork and if miner i does not have block rewards on the longest chain. Furthermore, if creating a fork is profitable in the first place, it may well be optimal to continue working on the fork even if miner i is unsuccessful in extending the fork at time t. This is especially true since she holds strictly positive balances in the form of block rewards on the fork.

Our findings align with Budish (2025), who shows that blockchains become susceptible to double-spending attacks under the longest chain rule unless transactions are sufficiently small relative to block rewards. In the language of Budish (2025), the block rewards on the LHS of (18) are the *flow* benefit of following the longest chain rule. They describe the opportunity cost of a double spending attack. The RHS of (18) captures miner i's expected *stock* benefit from attacking the blockchain by attempting to omit a negative transaction. Proposition 2 suggests that double-spending attacks become profitable if the absolute value of spend transactions is large. Since the size of transactions in the real world is endogenous to blockchain security, this result calls the usability of blockchain technology into question. More precisely, under the longest chain rule, either transactions need to be small, or the transfer to miners in the form of block rewards—and thus the cost of using a blockchain—need to scale with transactions. Otherwise, the blockchain becomes susceptible to double-spending attacks.

Positive transaction data and miner heterogeneity. We now explain a second, distinct reason why miners may find it optimal to deviate from the longest chain rule. Suppose that transaction data are positive: $Y_{i,b} \geq 0$. To visualize the constraints that arise from (15), consider the graph displayed in Figure 3 which features $Y_{i,b} = 0$ and thus only contains block rewards. The graph exhibits a fork where a parent block b_r has two subsequent edges, one leading to block b_{l_1} and one leading to block b_{f_1}. Since $\mathcal{B}^{LC}(G_t) = \{b_{l_2}\}$, the longest chain strategy calls for all miners to choose location b_{l_2}. Suppose now that miner 1 has earned the mining rewards in block b_{f_1} but miners 2 and 3 have the mining rewards on blocks b_{l_1} and b_{l_2}, respectively.

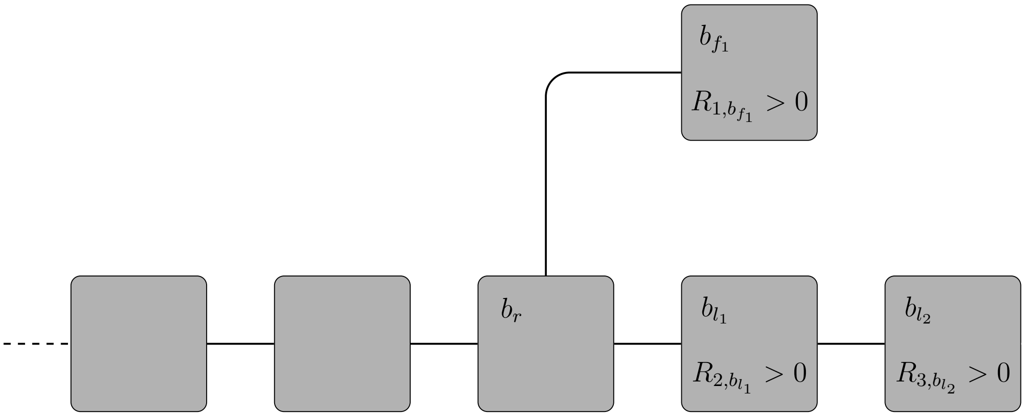

Figure 3: An illustration of incentives to deviate from the longest chain strategy.

Consider the net benefit to miner 1 of deviating from longest chain and choosing block b_{f_1}. Given the mining rewards illustrated in Figure 3, the weight of miners who would like to see fork b_{f_1} extended is simply p_1. Hence, the condition in (15) requires $\bar{R} \geq \delta p_1 2\bar{R}$ or $p_1 \leq 1/(2\delta)$. In other words, should forks appear and miner 1 have too much weight (say if $\delta \to 1$ and $p_1 > 0.5$), then she can likely direct consensus to her most preferred chain. And, when her most preferred chain does not coincide with the longest chain, she has incentives to deviate from the longest chain.

More generally, we argue that Proposition 1 likely imposes stringent limits on the distribution of mining power and these limits are likely to be violated (or provide miners with incentives to acquire mining power such that they are violated). Indeed, the condition in (15) imposes an upper bound on p_i for miner i should she be the only miner with larger balances on a fork to some $\hat{b} \in \mathcal{B}^{-1}(G_t)$ than on the longest chain. Of course, this upper bound may not suffice should other miners have positive transaction data on the fork, suggesting that the condition in (15) is likely to fail in general. Thus, Proposition 1 reveals that even when a blockchain only features mining rewards, the longest chain rule may not be robust as an equilibrium (in the perfect public sense) to general distributions of mining power. This is particularly true if we also consider positive transaction data. Importantly, mining power in the Bitcoin network is highly concentrated in practice (Makarov and Schoar, 2021).

Interestingly, and in contrast to the case of negative transaction data where miners may find it optimal to create forks to remove transaction data, the longest chain rule fails because miners want to *add* transaction data to the consensus chain. In Figure 3 above, miner 1's balances are not contained in the longest chain and this is precisely the reason why she might find it profitable to deviate from the longest chain rule and *extend* their preferred fork instead. However, it is not profitable to create a fork if transaction data are (weakly) positive:

Lemma 2. *Suppose $\#C(b, G_t) = \#C(b', G_t)$ for every $b, b' \in \mathcal{T}(G_t)$ and $Y_{i,b_t} \geq 0$. Then the condition in* (14) *is satisfied for miner i.*

The proof is in Appendix A.4. The lemma says that if only longest chains exist (and thus no fork which is one block short), and the transaction data for miner i are weakly positive, then she has no strictly profitable deviation from the longest chain rule. Instead, she prefers to follow the longest chain rule, ensuring that the balances on her preferred longest chain as well as her new payments received and block rewards become part of consensus. This contrasts with the case of negative transaction data and double-spending attacks, where miners may find it optimal to create forks to remove transaction data.

Negative transaction data and saving on consensus. We now explain the third distinct reason why miners may find it optimal to deviate from the longest chain rule. Suppose that miner i is mining block b_t which contains a spend transaction for her, $Y_{i,b_t} < 0$. Miner i may now face a profitable deviation to *create* a fork by working on the parent block of the terminal block of one of the longest chains.

The reason is the delay with which transaction goods are delivered. The spend transaction is associated with a flow disutility once it is included in the blockchain—and thus before the miner receives the corresponding goods. This disutility is larger the more other miners work on the chain that includes the spend transaction. If miner i creates a fork and few other miners work on it, then the flow disutility is reduced. If miner i also holds a large amount of mining power, she has a good chance of turning this newly created fork into the consensus chain in the following time period. She then earns the full transaction benefit $Y^{-}_{i,b_t}/\delta$ but has reduced the associated cost when the transaction is first included in the blockchain. In sum, miners may want to create forks if they have a lot of mining power and initially other miners will not work on the newly created fork.

To illustrate, suppose there is a single chain and miner i is mining some block $b_{i,t}$, which does not contain any data for any miner other than the spend transaction for miner i. The terminal block of the longest chain does contain data for all other miners, and hence they continue working on this chain at time $t+1$ even after miner i has successfully created a fork of equal length at time t. Suppose further that miner i has no data in any other block. Equation (15) then simplifies to

$$\bar{R} \;\geq\; \delta p_i \cdot \left(Y_{i,b_t} + \bar{R} + \frac{|Y_{i,b_t}|}{\delta} \right). \tag{19}$$

Miner i trades off earning the block reward for sure (as well as deriving zero net utility from the spend transaction) against earning the block rewards and a strictly positive net utility from the spend transaction with probability p_i in the following period. Rearranging, we find that this deviation is not profitable if (19) is satisfied. Reversely, if the inequality in (19) fails, the deviation does become profitable in this example.

In practice, the deviation may not profitable even if the inequality in (19) fails. If all other miners work on miner i's fork, then the fork becomes the consensus chain with probability 1 in the following time period. Since miner i cannot avoid the flow disutility when including the transaction on the blockchain, she is better off working on her preferred longest chain in the first place. More generally, the deviation becomes less profitable the more other miners work on the fork, as the immediate cost of the spend transaction increases. Furthermore, if miner i has balances in the terminal block of her preferred longest chain, this also reduced the profitability of this deviation.[10] The following lemma, which we prove in Appendix A.5, formalizes this discussion:

Lemma 3. *Suppose $\#C(b, G_t) = \#C(b', G_t)$ for every $b, b' \in \mathcal{T}(G_t)$ and $Y_{i,b_t} < 0$. Then the condition in* (14) *is satisfied for miner i if* (19) *is satisfied.*

Intuitively, the equality in (19) is a sufficient condition such that this deviation is not profitable. As an example, if $p_i = 0.5$ and $\delta = 0.99$, then (19) becomes $\bar{R} \geq |Y_{i,b_t}|/101$. Given the block rewards of 3.125 Bitcoins (as of January 2025, ignoring transaction fees), creating a fork is not profitable for all spend transactions below 300 Bitcoins in this numerical example.

Note that if the benefit of delaying consensus is low ($\delta \to 1$), this deviation becomes unprofitable. Indeed, if the discount factor is sufficiently large, then creating this short-lived disagreement is no longer profitable as miner i values the present relatively less.

Interestingly, relative to the previous motives to deviate from the longest chain rule, this third type of profitable deviation arises because miners disagree on how to *add new* data to the blockchain. In particular, miners disagree on how to include spend transactions before they have vested.

Given these results, it is natural to explore strategies other than the longest chain rule. Section 4 extends the longest chain rule to include checkpoints, which renders it robust to double-spending attacks and thus addresses the first type of deviation. Section 5 presents a modification of the longest chain rule with checkpoints to address the other two types.

4 Checkpoint Strategies

We now show that a simple modification of the longest chain rule is not susceptible to double-spending attacks. Our proposed resolution to miners' incentives to omit data from the chain is to introduce *checkpoints*, a form of history dependence, into the candidate equilibrium strategies. The basic idea is that for every graph, agents determine a reference block—the checkpoint—and restrict attention to all chains containing this block. All other chains are ignored regardless of their length.

[10] If the consumption goods are unfairly priced in the sense that miner i derives a higher marginal flow utility than one for each unit spent, then risking that the spend transaction never vests also becomes less appealing to miner i, decreasing the profitability of such a deviation.

One way to use checkpoints to rule out double-spend behavior is to simply impose that the last block added is the new checkpoint. In essence, this proposal rules out all possible forks in the blockchain. If no forks are permitted, then it is impossible for any one agent to omit data from the blockchain. We find this resolution to the double spend problem implausible for real-world implementations. In reality, some forks are non-malicious and occur due to latency—within the unit of time agents observe updates to the blockchain, it is possible to observe multiple blocks being added in the same period.

We therefore proceed by assuming such strategies are infeasible and looking for checkpoint rules that admit the possibility of forks. More formally, for any history H_t, let denote $b^{CP}(H_t)$ the checkpoint which selects a specific block on the current graph, G_t. We assume the following:

Assumption 1. *$b^{CP}(H_t) \notin \mathcal{T}(G_t)$ for all histories H_t.*

Assumption 1 states that checkpoint rules may not select a terminal block in the graph for any history. Such a restriction ensures that forks of at least length one are always feasible.

Checkpoint Settlement. We now explain settlement with checkpoints. We continue assuming that consumption goods are fairly priced. Under the checkpoint strategies we suggest below, blocks become checkpoints and thus part of consensus one period after they have been appended to the blockchain. The price of consumption goods therefore remains at $1/\delta$. Settlement of spend transactions contained in block b takes place at time t if this is the first time period that block b lies on a chain to the checkpoint. This includes the possibility that the block b itself is the checkpoint. More formally, we redefine

$$\lambda_t(b, H_t) \;=\; \begin{cases} 1 & \text{if} \quad t = \inf\left\{\tau \geq 0 : \, b \in \mathcal{C}(b^{CP}(H_\tau), G_\tau)\right\}, \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

We again define $\Lambda_t(b, H_t)$ to indicate whether the spend transactions contained in block b have yet to vest at time t.

Checkpoint Strategies. To specify a candidate equilibrium strategy with checkpoints, it is helpful to introduce two pieces of notation. First, let $J(b', G_t) \subseteq G_t$ denote the subgraph associated with some root block $b' \in \mathcal{B}(G_t)$. This subgraph contains the block b', all its child blocks, the child blocks' child blocks, and so on. It also contains all edges connecting these blocks. Let $\mathcal{J}(b', G_t)$ denote the set of blocks on the subgraph.[11]

As an example, consider Figure 4. The blocks are labeled according to whether they are on the longest chain (b_{l_1}—b_{l_4}) or on one of two forks (b_{f_1}—b_{f_2}) and (b_{k_1}). Consider then the subgraph $J(b_{l_2}, G_t)$. It contains the blocks (b_{l_2}—b_{l_4}) and b_{k_1} as well as the edges (b_{l_3}, b_{l_2}), (b_{l_4}, b_{l_3}) and (b_{k_1}, b_{l_2}). All other blocks and edges in the figure are not part of the subgraph.

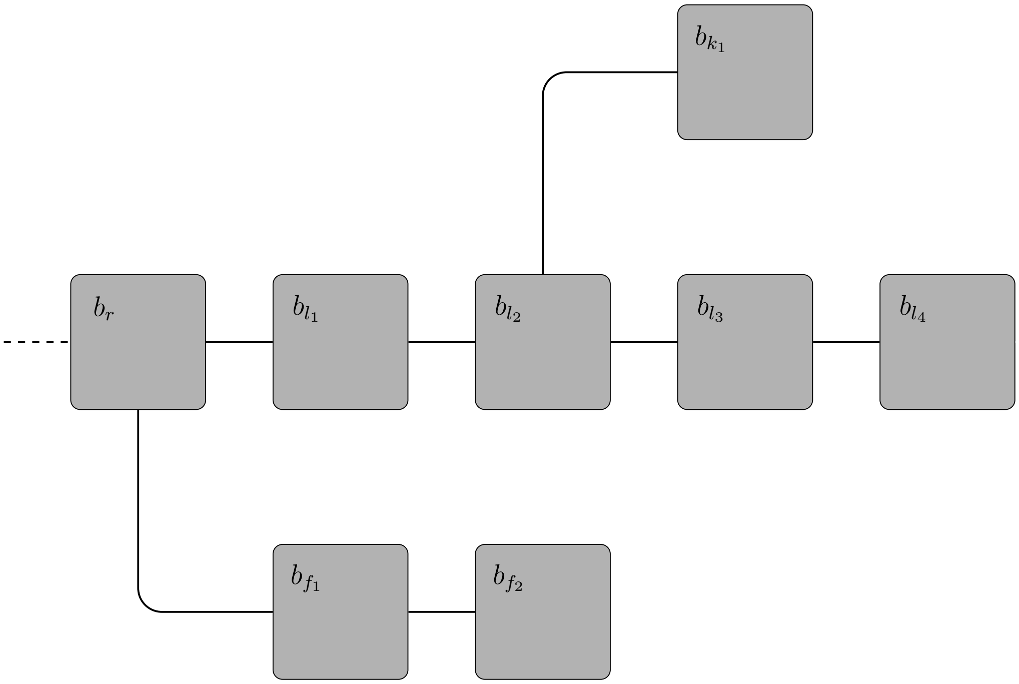

Figure 4: An illustration of subgraphs and checkpoints.

Second, let $M(b', G_t)$ denote the parent block of block b'. To illustrate, consider again Figure 4. The parent block of b_{l_4} is then given by $M(b_{l_4}, G_t) = b_{l_3}$.

We now define checkpoint blocks as follows. In any period, we consider the subgraph with the checkpoint block from the previous period as the common root, $J(b^{CP}(H_{t-1}), G_t)$. Next, we find the set of terminal blocks of the longest chains on this subgraph,

$$\mathcal{B}^{CP}\big(b^{CP}(H_{t-1}), G_t\big) \;=\; \underset{b \in \mathcal{J}(b^{CP}(H_{t-1}), G_t)}{\operatorname{argmax}} \#C(b, G_t). \tag{21}$$

[11] Formally, consider the set $\mathcal{J}(b', G_t) \subseteq \mathcal{B}(G_t)$ satisfying

$$\mathcal{J}(b', G_t) \;=\; \{b \in \mathcal{B}(G_t) : \#C(b, G_t) \geq \#C(b', G_t) \text{ and } b' \in \mathcal{C}(b, G_t)\}.$$

In the language of graph theory (Bondy and Murty, 1976), we say that the set of blocks $\mathcal{J}(b', G_t)$ induces the subgraph $J(b', G_t) = G[\mathcal{J}(b', G_t)]$ in G_t. Note that $J(b', G_t) = G_t$ if only if b' is the genesis block b_0.

If this set is a singleton, we choose the new checkpoint to be the parent block of the terminal block on the longest chain:

$$b^{CP}(H_t) \;=\; M\Big(\mathcal{B}^{CP}\big(b^{CP}(H_{t-1}), G_t\big), G_t\Big). \tag{22}$$

If there are multiple longest chains ahead of the checkpoint, then the checkpoint randomly updates to the parent of a terminal block of one of these longest chains. More formally, if the set $\mathcal{B}^{CP}\big(b^{CP}(H_{t-1}), G_t\big)$ is not a singleton, we say that $b^{CP}(H_t) = M\big(b, G_t\big)$ with probability $\pi_b \in [0,1]$ for every $b \in \mathcal{B}^{CP}\big(b^{CP}(H_{t-1}), G_t\big)$ and insist that $\sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_{t-1}),G_t)} \pi_b = 1$.[12] Given this new checkpoint, the set of terminal blocks of the longest chains on the new subgraph is then given by $\mathcal{B}^{CP}\big(b^{CP}(H_t), G_t\big)$.

Note that it is not difficult to randomly select blocks in blockchain environments. Recall that each block contains hash data, which come in the form of fixed-size values. One method of randomization is to choose the checkpoint candidate block with the lowest (or highest) hash.[13]

To illustrate the checkpoint selection, consider once again Figure 4. Suppose that up to b_r there had been no forks in the chain so the parent of b_r is the initial checkpoint. Then, Table 1 illustrates a possible sequence of added blocks and the resultant checkpoints:

Period	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$
Block Added	b_{l_1}	b_{f_1}	b_{l_2}	b_{k_1}	b_{l_3}	b_{f_2}	b_{l_4}
Checkpoint	b_r	b_r	b_{l_1}	b_{l_2}	b_{l_2}	b_{l_2}	b_{l_3}

Table 1: A Sequence of Blocks and Checkpoints.

Table 1 highlights that the checkpoint updates whenever there is a single longest chain following the checkpoint. This, for example, occurs when block b_{l_2} is appended to block b_{l_1}, which then becomes the new checkpoint. The checkpoint does not update when b_{f_1} is appended to the checkpoint b_r which already has one child block in b_{l_1}.

[12] Our formulation includes the possibility that the checkpoint does not update, $b^{CP}(H_t) = b^{CP}(H_{t-1})$, if all longest chains ahead of the checkpoint consist of only one block.

[13] We do not need to specify the probabilities with which checkpoints are selected. On the Ethereum blockchain, a subset of validators is called to attest to new blocks. The blocks with the largest stake-weighted attestations become finalized, which, in the language of our paper, corresponds to the checkpoint updating such that these blocks are now 'behind the checkpoint.' The probability of selecting a particular block as the new checkpoint is then a function of the data contained in different blocks of the subgraph following the current checkpoint and the miners' probability of proposing the next block, $(p_1, p_2, ..., p_N)$. Of course, on a Proof-of-Stake blockchain, these probabilities themselves can be a function of coin balances.

Since the checkpoint always updates when possible, the subgraph with the new checkpoint as common root only contains longest chains. This is illustrated in Figure 5. Suppose that b_r is the previous checkpoint. The subgraph ahead of the checkpoint features three longest chains (b_r—b_{l_2}), (b_r—b_{k_2}), and (b_r—b_{n_1}), respectively. The checkpoint then updates to either b_{l_1} as the parent block of b_{l_2}, or to b_{k_1} as parent of both b_{k_2} and b_{n_1}. While miners may disagree on their preferred longest chain if the checkpoint updates to b_{k_1} and miners are called to work on either b_{k_2} or b_{n_1}, the figure highlights that only longest chains remain once the checkpoint updates.

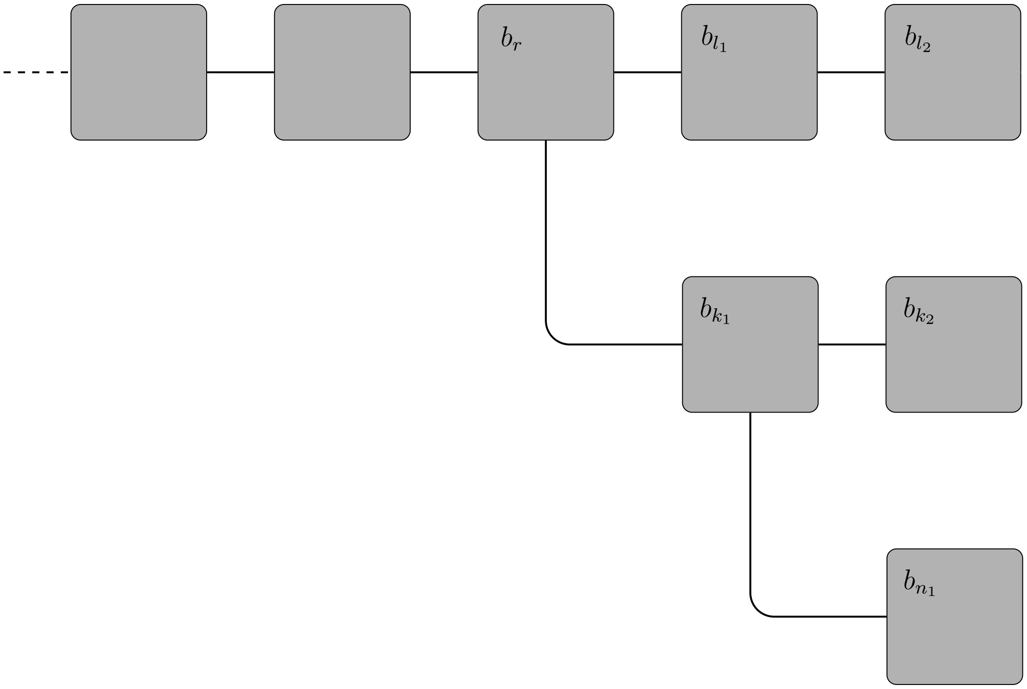

Figure 5: An illustration of checkpoints.

Given the checkpoint selection, the checkpoint rule satisfies

$$\sigma_{i,t}^{CP}(H_t) \; = \; b_{i,t}^* \; \equiv \underset{b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_t)}{\operatorname{argmax}} \left(Y_{i,b} + R_{i,b} \; + \; \frac{Y_{i,b}^-}{\delta} \right), \tag{23}$$

where we have abused notation by redefining $b_{i,t}^*$. The strategy calls miners to work on their preferred longest chain ahead of the checkpoint, which has the highest sum of positive transactions, block rewards, and unvested spend transactions. Notice that this checkpoint rule corresponds closely to the longest chain rule of equation (9), but is limited to the subgraph following the new checkpoint $b^{CP}(H_t)$. Hence, miners only consider blocks in $\mathcal{B}^{CP}\big(b^{CP}(H_t), G_t\big)$.

Checkpoint Equilibrium. The checkpoint strategy has one key feature with regards to double-spending, which we describe in the following proposition:

Proposition 3 (No Double-Spending under the Checkpoint Rule). *For any history H_t and for any miner i, there exists no (weakly) profitable one-shot deviations from the checkpoint rule to any block $b \notin \mathcal{J}(b^{CP}(H_t), G_t)$.*

The proposition implies that it is never profitable to append new blocks to any block *behind* the checkpoint—neither to remove transactions in order to double-spend, nor to add transactions. The proof is in Appendix B.1.

Intuitively, miners ignore all blocks which lie on chains that do not include the checkpoint block. Thus, a newly mined block is immediately abandoned if it is appended to the parent of the checkpoint block. The same is true for a block appended to any chain which branches off behind the checkpoint. In the illustration of double-spending attacks of Figure 2, any blocks appended to b_r or b_f are ignored by all miners once block b_{l_1} has become the checkpoint. As a consequence, spend transactions in block b_{l_1} cannot be removed after this block has become the checkpoint block and the corresponding consumption goods have been delivered. Since the first spend only occurs once blocks with spend transactions are behind the checkpoint, double-spending cannot occur under checkpoints strategies.

Furthermore, it is costly to append a block behind the checkpoint. Any positive transaction data and block rewards contained in the new block are then lost, and any spend transactions never vest. It follows that deviations to blocks behind the checkpoint are strictly unprofitable.

With this result in hand, we are now ready to characterize conditions such that the checkpoint rule is indeed a perfect public equilibrium. We proceed in analogoulsy to Proposition 1 for the longest chain rule, and so again consider a thought experiment in which miner i appends block $b_{i,t}$ for sure. Given our checkpoint selection rule, only longest chains remain ahead of the checkpoint. The continuation utility from following the checkpoint rule is then given by

$$U^i_{t+1}\left(b_{i,t}, b^*_{i,t}; H_t\right) \;=\; Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y^-_{i,b_t}}{\delta} + \left(Y_{i,b^*_{i,t}} + R_{i,b^*_{i,t}} + \frac{Y^-_{i,b^*_{i,t}}}{\delta}\right). \tag{24}$$

By following the proposed strategy, the utility derived from time $t+1$ onwards based on data present on the blockchain at the end of time t is given by the balances in the preferred block ahead of the checkpoint (and the corresponding consumption flow utility derived at time $t+1$) plus the balances in the new block (and the corresponding consumption flow utility derived at time $t+2$).

Suppose next that miner i deviates from the checkpoint rule and instead works on the checkpoint itself. Analogous to (12), the total flow value derived from the balances on the new longest chain as well as on all previous longest chains—all ahead of the checkpoint—at time $t+1$ is given by:

$$\begin{aligned} F^i_{t+1}\left(b_{i,t}, b^{CP}(H_t); H_t\right) \;&=\; \sum_{\{j \neq i \colon b^*_{j,t+1} = b_{i,t}\}} \frac{p_j}{1-p_i} \cdot \left(Y_{i,b_t} + \bar{R}\right) \\ &+ \sum_{b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_t)} \; \sum_{\{j \neq i \colon b^*_{j,t+1} = b\}} \frac{p_j}{1-p_i} \cdot \left(Y_{i,b} + R_{i,b}\right). \end{aligned} \tag{25}$$

As before, the flow value is proportional to the computing power of miners working on the respective chains.

Analogous to (13), miner i's expectation over the continuation utility derived from time $t+2$ onwards based on data present in the blockchain at the end of period t (after miner i has appended block $b_{i,t}$) is given by

$$\begin{aligned} \mathbb{E}_t\left[U^i_{t+2}\left(b_{i,t}, b^{CP}(H_t); H_t\right)\right] \;&=\; \left\{ \sum_{\{j \neq i \colon b^*_{j,t+1} = b_{i,t}\}} p_j \;+ p_i \right\} \cdot \left(Y_{i,b_t} + \bar{R} + \frac{Y^-_{i,b_t}}{\delta} \right) \\ &+ \sum_{b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_t)} \; \sum_{\{j \neq i \colon b^*_{j,t+1} = b\}} p_j \cdot \left(Y_{i,b} + R_{i,b} + \frac{Y^-_{i,b}}{\delta} \right). \end{aligned} \tag{26}$$

As before, miner i's new chain becomes the consensus chain from time $t+2$ onwards with probability $\left\{\sum_{\{j \neq i \colon b^*_{j,t+1} = b_{i,t}\}} p_j + p_i\right\}$. The miner then derives the corresponding flow utility in perpetuity, and the the spend transactions vest. Every other chain to some $b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_t)$ becomes the consensus chain with probability $\sum_{\{j \neq i \colon b^*_{j,t+1} = b\}} p_j$.

We are now ready to state our next main result.

Proposition 4 (Checkpoint Rule is a Perfect Public Equilibrium). *The checkpoint* *rule is a perfect public equilibrium if for every history H_t and for every miner i:*

$$U^i_{t+1}\left(b_{i,t}, b^*_{i,t}; H_t\right) \;\geq\; (1-\delta)\cdot F^i_{t+1}\left(b_{i,t}, b^{CP}(H_t); H_t\right) \;+\; \delta \cdot \mathbb{E}_t\left[U^i_{t+2}\left(b_{i,t}, b^{CP}(H_t); H_t\right)\right] \tag{27}$$

This inequality is satisfied if $Y_{i,b_t} \geq 0$. It is also satisfied if $\bar{R} \geq \frac{(1-\delta)p_i}{1-\delta p_i} \cdot |Y_{i,b_t}|$.

The proof is in Appendix B.2. Equation (27) is the equivalent to the condition for the longest chain rule in (14), now focusing on the subgraph following the checkpoint (rather than the full graph). To understand the result, note that it is never profitable to append a block behind the checkpoint by Proposition 3. Since the checkpoint rule calls miners to work on their preferred longest chain ahead of the checkpoint, the only possibly profitable deviation is to the checkpoint itself. By Lemma 2, if miner i is mining a block with a positive transaction, she prefers to work on her preferred longest chain. However, if the block contains a spend transaction, miner i may face a profitable deviation to the checkpoint to reduce to cost of her spend transaction. Lemma 3 presents a sufficient (but not necessary) condition such that this type of deviation is not profitable. We thus find that the checkpoint rule with a one-block checkpoint lag can achieve consensus for very general transaction data.

Latent checkpoints and the risk of permanent forks. Our model abstracts from network latency. In reality, some forks are non-malicious and arise naturally because miners do not observe all blocks simultaneously. This is why we ruled out terminal blocks as checkpoints. Our checkpoint selection process designates parent blocks of terminal blocks as new checkpoint. One consequence of this selection process is that only longest chains remain ahead of the checkpoint, yielding mild conditions on transaction data to achieve consensus.

However, with more severe network latency, a one-block checkpoint lag induces the risk of permanent forks. Miners reject blocks appended behind the checkpoint, and if latency causes disagreement over the checkpoint itself, they will continue working on separate forks indefinitely. This induces a trade-off between checkpoint lags and the risk of permanent forks, and it may well be optimal in practice to increase the checkpoint lag.

Consider therefore a k-block checkpoint lag, with $k \geq 2$. In perfect analogy to the result of Proposition 1 and the discussion around positive transaction data and miner heterogeneity in Section 3, miners may now find it profitable to deviate from the longest chain rule *ahead* of the checkpoint by working on shorter forks.

To illustrate, suppose $k = 2$. The checkpoint then only updates to the grandparent of terminal blocks (rather than the parent). Reconsider Figure 3 and suppose that the checkpoint is given by b_r. Miner 1 may find it profitable to work on block b_f if her computing power is sufficiently large. Importantly, since all miners continue to reject blocks appended behind the checkpoint, double-spending attacks continue to be unprofitable regardless of the checkpoint lag.

In summary, modifying the longest chain rule to incorporate checkpoints—which we observe in practice as with the Ethereum's proof-of-stake protocol—achieves consensus for all blocks behind the checkpoint. If merchants delay delivering consumption goods until the block containing the relevant transaction has been finalized by the checkpoint, then double-spending cannot occur.

While the literature has identified the threat of double-spending attacks as key obstacle to blockchain technology (Budish, 2025), it turns out to be more difficult to achieve consensus ahead of the checkpoint. First, miners may induce temporary disagreement to save on the cost of spend transactions. Second, in the more general case of larger checkpoint lags, miners might seek to add data contained in shorter forks to the blockchain.

Importantly, a common theme for both deviations from the checkpoint rule is that miner i works on her preferred chain in the hope of other miners switching to this chain the future. In the following section, we modify the checkpoint rule and outline technological requirements necessary for achieving consensus ahead of the checkpoint. Key to these equilibrium strategies is the need to resolve a form of tacit collusion so that miners have no hope of persuading other miners to switch chains.

5 Approval Weight Strategies

We now develop an equilibrium strategy we call the *approval weighted chain rule*. The approval weighted chain rule yields the same outcomes as the longest chain rule (with checkpoints) along the equilibrium path without forks. However, whenever forks arise, it provides better incentives to miners with high degrees of mining power for arbitrary transaction data in any previously solved blocks—at least under a non-trivial restriction on transaction data in the newly added blocks. In other words, we show that the approval weighted chain rule remains an equilibrium even when mining power is concentrated and miners differ in their preference over which chain they would like to become consensus chain.

The idea behind the approval weighted chain rule is to require miners to coordinate their mining effort on the chains that deliver (any) value to the group of miners with the most mining power. We show that off the equilibrium path, this coordination device induces miners to follow the proposed equilibrium strategy even when they have a large degree of mining power.

We define the approval weighted chain rule in steps. First we determine the common part of all chains that include a terminal block in any graph. Next we divide every chain into this common part and an idiosyncratic part. Finally we calculate the approval weight of the idiosyncratic part of each chain as the sum of mining power of miners with positive balances on this idiosyncratic part of the chain. We iterate on this procedure removing terminal blocks with the lowest approval weight until a terminal block remains. Miners are then called to work on this terminal block. If multiple terminal blocks have the same approval weights, then miners are called to work on the longest chain among the chains running to these terminal blocks.

We proceed by developing a set operator that refines any set of terminal blocks to only those with the highest approval weight recursively. By Proposition 3, we limit our attention to the subgraph following the checkpoint after every history, $J(b^{CP}(H_t), G_t)$. To build the operator, we consider first an arbitrary subset of terminal blocks on this subgraph, $\mathcal{S} \subseteq \mathcal{T}\left(J(b^{CP}(H_t), G_t)\right) \subseteq \mathcal{T}(G_t)$. The common blocks corresponding to the set of terminal blocks $\mathcal{S}$ are those which lie on every chain running to a block in the set: $\mathcal{C}^*(\mathcal{S}, G_t) = \bigcap_{b \in \mathcal{S}} \mathcal{C}(b, G_t)$. By construction, the set $\mathcal{C}^*(\mathcal{S}, G_t)$ is non-empty and contains at least the set of blocks on the chain to the checkpoint, $\mathcal{C}(b^{CP}(H_t), G_t)$. The approval weights of each block $b \in \mathcal{S}$ are constructed as follows:

$$P(b, \mathcal{S}, G_t) \ = \ \sum_{i=1}^{N} p_i \cdot \mathbb{1}\left\{ \sum_{b' \in \mathcal{C}(b, G_t) \setminus \mathcal{C}^*(\mathcal{S}, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \right) > 0 \right\}, \qquad (28)$$

Intuitively, the approval weights can be viewed as a score for the idiosyncratic part of each chain leading to a terminal block on the subgraph following the checkpoint. This score adds up the mining power of those miners which have *any* data in the blocks on the chain. It is intuitive that miners with positive transaction data and block rewards have an interest in this chain and benefit if it becomes the consensus chain. The same is true for miners with unvested spend transactions, which enjoy a consumption flow utility of $1/\delta > 1$ as soon as the chain becomes the consensus chain, the checkpoint updates and consumption goods are delivered.

To illustrate the score function, consider Figure 6. Suppose that the checkpoint is given

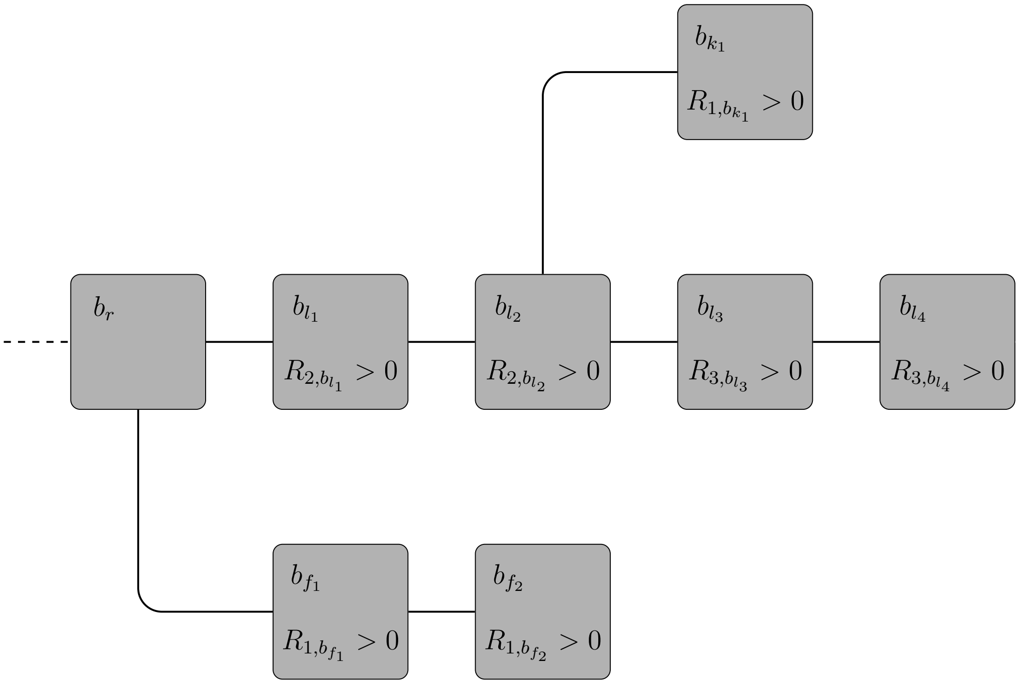

Figure 6: An illustration of approval weights.

by b_r. The blocks are again labeled according to whether they are on the longest chain (b_{l_1}—b_{l_4}) or on one of two forks (b_{f_1}—b_{f_2}) and (b_{k_1}). Suppose for simplicity that the block rewards are the only balances contained in each block. For example, then, only miner 2 has a positive coin balance on block b_{l_1}. Note that only miners 2 and 3 have positive balances on the longest chain, only miners 1 and 2 have positive coin balances on the chain to b_{k_1}, and only miner 1 has positive coin balances on the chain to b_{f_2}. The approval weights associated with the respective chains are then given by

$$\begin{aligned} P(b_{k_1}, \mathcal{T}\left(J(b_r, G_t)\right), G_t) &= p_1 + p_2, \\ P(b_{l_4}, \mathcal{T}\left(J(b_r, G_t)\right), G_t) &= p_2 + p_3, \\ P(b_{f_2}, \mathcal{T}\left(J(b_r, G_t)\right), G_t) &= p_1. \end{aligned} \tag{29}$$

We now define a set operator, $T : \mathcal{S} \to \mathcal{S}$. The operator selects those blocks whose chains have the highest approval weight:

$$T(\mathcal{S}) = \{b \in \mathcal{S} \,|\, P(b, \mathcal{S}, G_t) \geq \max_{b' \in \mathcal{S}} P(b', \mathcal{S}, G_t)\}. \tag{30}$$

To illustrate the effect of the set operator, consider again Figure 6. Suppose now that $p_1 > p_2 > p_3$. Applying the set operator to the set of terminal blocks in this subgraph then yields block b_{k_1}, as it has the highest approval weight.

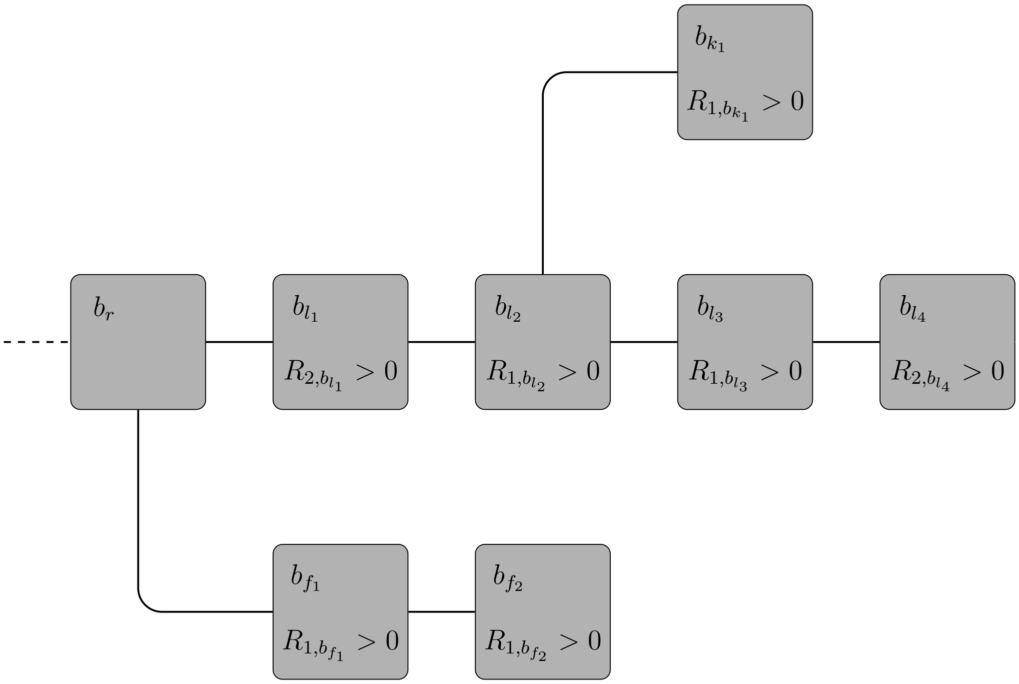

Figure 7: Another illustration of approval weights.

However, Figure 7 presents a different graph which illustrates that applying the set operator once may not suffice to yield a single block. Again suppose that block rewards are the only balances on each block. The approval weights associated with the chains to the terminal blocks are given by

$$\begin{aligned} P(b_{k_1}, \mathcal{T}\left(J(b_r, G_t)\right), G_t) &= p_1 + p_2, \\ P(b_{l_4}, \mathcal{T}\left(J(b_r, G_t)\right), G_t) &= p_1 + p_2, \\ P(b_{f_2}, \mathcal{T}\left(J(b_r, G_t)\right), G_t) &= p_1. \end{aligned} \tag{31}$$

Applying the set operator once yields $\{b_{l_4}, b_{k_1}\}$. Having applied the set operator once, the approval weights of the remaining terminal blocks are given by

$$\begin{aligned} P(b_{l_4}, T\left(\mathcal{T}\left(J(b_r, G_t)\right)\right), G_t) &= p_1 + p_2, \\ P(b_{k_1}, T\left(\mathcal{T}\left(J(b_r, G_t)\right)\right), G_t) &= p_2. \end{aligned} \tag{32}$$

Thus, applying the set operator a second time to the set of remaining terminal blocks yields only b_{l_4}. The reason is that the set of common blocks changes after the first iteration. Initially, the common blocks are the blocks on the chain to the checkpoint b_r. After block b_{f_1} has been removed, the common blocks for chains running to the remaining terminal blocks b_{l_4} and b_{k_1} are the blocks on the chain to b_{l_2}. Hence block b_{k_1} is deleted on the second iteration.

For this reason, we define checkpoint blocks as follows. In any period, we consider the subgraph with the checkpoint block from the previous period as the common root, $J(b^{CP}(H_{t-1}), G_t)$. Next, we find the set of terminal blocks on this subgraph with the highest approval weight by iteratively applying the set operator:

$$\mathcal{A}\big(H_t\big) = \lim_{k\to\infty} T^k\Big(\mathcal{T}\big(J\big(b^{CP}(H_{t-1}), G_t\big)\big)\Big). \tag{33}$$

For simplicity, we focus on histories such that $\mathcal{A}\big(H_t\big)$ is a singleton.[14] We choose the new checkpoint to be the parent block of this terminal block:

$$b^{CP}(H_t) = M\big(\mathcal{A}\big(H_t\big), G_t\big) \tag{34}$$

To illustrate the checkpoint selection with approval weights, consider Figure 8. Suppose that up to b_r there had been no forks in the chain so that the parent of b_r is the initial checkpoint. Suppose further, again purely for simplicity, that block rewards are the only transactions contained in the blocks. Table 2 again describes a possible sequence of added

Period	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$
Block Added	b_{l_1}	b_{f_1}	b_{l_2}	b_{k_1}	b_{l_3}	b_{f_2}	b_{l_4}
Checkpoint	b_r	b_r	b_{l_1}	b_{l_2}	b_{l_2}	b_{l_2}	b_{l_2}

Table 2: A Sequence of Blocks and Checkpoints.

[14] We stress that this is without loss of generality. If the set is not a singleton, we can use the hash of the blocks appended to the checkpoint as tie-breaker, which yields a unique block.

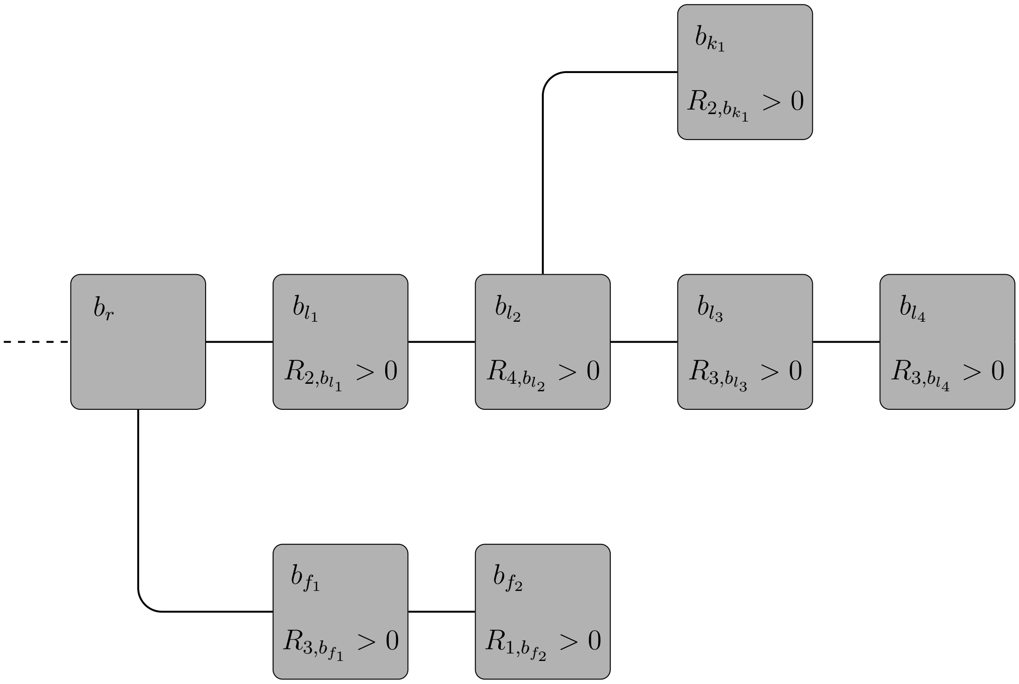

Figure 8: An illustration of checkpoint selection with approval weights.

blocks and the resultant checkpoints, this time assuming that $p_1 > p_2 > p_3 > p_4$. At time $t+1$, block b_{l_1} is added to the single chain and its parent block b_r becomes the new checkpoint. At time $t+2$, the block b_{f_1} creates a fork. The checkpoint only updates to b_{l_1} in the following period when b_{l_2} is appended to b_{l_1} and that chain has the highest approval weights. On the contrary, the checkpoint does not update when b_{l_4} has been appended to b_{l_3}. The reason is that the block b_{k_1} has higher approval weights. This is the key difference relative to the longest chain rule with checkpoints. Importantly, notice that the fork (b_{f_1}—b_{f_2}) branches off behind the checkpoint b_{l_2}. Since we focus on the subgraph following the checkpoint, this fork is ignored at time $t+7$ and in all future periods regardless of its approval weights.

Given checkpoint selection with approval weights, the approval weighted chain rule satisfies

$$\sigma_{i,t}^{AW}(H_t) \;=\; \mathcal{A}\big(H_t\big). \tag{35}$$

In words, the approval weighted chain rule calls miners to append to the block with the highest approval weight ahead of the new checkpoint.[15] This is in contrast to the checkpoint

[15] As for the checkpoint selection, if the set $\mathcal{A}(H_t)$ was not a singleton, the strategy would be adjusted

strategy of the previous section in (23), in which miners were called to work on their preferred longest chain ahead of the checkpoint.

Finally, consider the following restriction on transaction data in the block which miners seek to add in the current time period:

Assumption 2. *For any history H_t, if $|\mathcal{T}\left(J(b^{CP}(H_t), G_t)\right)| \geq 2$, then $Y_{j,b_t} = 0$ for all $j \in \{1, 2, ..., N\}$. Otherwise, transactions are unrestricted.*

Assumption 2 states that, whenever there is a fork ahead of the checkpoint, then the newly added block only contains block rewards but no transaction data. This restriction is important in achieving consensus and permanence under the approval weighted chain rule, and we discuss it in detail at the end of this section.

We now argue that the approval weighted chain rule is an equilibrium strategy for any transaction data on the blockchain and for any distribution of mining power under the restriction of Assumption 2:

Proposition 5. *Under Assumption 2, the approval weighted chain rule is a perfect public equilibrium.*

The proof is in Appendix C. The approval weighted chain rule has two important properties that disincentivize miners from deviating from the equilibrium strategy. First, the mining power of miner i is already included in the approval weight of any chain which miner i might like to select as the consensus chain. Consequently deviating to such a location cannot change the approval weight of the chain. Second, miner i has no incentive to deviate to any chain where her mining power is not already included in the approval weight. As a result of these two features, there is no self-interested deviation where miner i can mine a new block which induces a change in the equilibrium behavior of all other miners in the following period. This ensures that the approval weighted chain rule is an indeed equilibrium.

To illustrate why the approval weighted chain rule is an equilibrium, reconsider the fork which contains a spend transaction for the successful miner and induces short-lived disagreement, as described in the previous section. Miners may find it profitable to create such a fork if they have a lot of mining power and initially other mines will not work on the newly created fork. However, under the approval weight rule, one of two things is true: either all miners ignore this newly created fork in perpetuity since it has lower approval

as follows. Miners are called to work—among the chains with the highest approval weights—on the chain containing the block with the lowest hash among the blocks appended to the checkpoint. We focus on the set of histories such that $\mathcal{A}(H_t)$ is a singleton purely for expositional simplicity.

weights; or all miners work on the new fork if it has higher approval weights. Miner i then either has no hope of turning the fork into the consensus chain, or the new fork immediately becomes the consensus chain and miner i fully bears the disutility of the spend transaction. In either case, it is then (weakly) profitable to follow the equilibrium strategies.

Notice that in equilibrium, since miners have no incentives to deviate from the proposed strategy, there would be no forks (other than by accident). As a result, the approval weighted chain resembles the longest chain since all miners mine a single long chain. Any differences between the approval weighted chain rule and the longest chain rule appear only off the equilibrium path and these differences are important in sustaining equilibrium behavior.

Empty blocks and the trade-off between consensus and liveness. We now argue why the restriction in Assumption 2 is necessary for the approval weighted chain rule to be an equilibrium. Recall that we have taken transactions as given and assumed that all miners mine the same block, modulo block rewards. In reality, miners can choose which transactions to include in their block and can even submit transactions themselves.

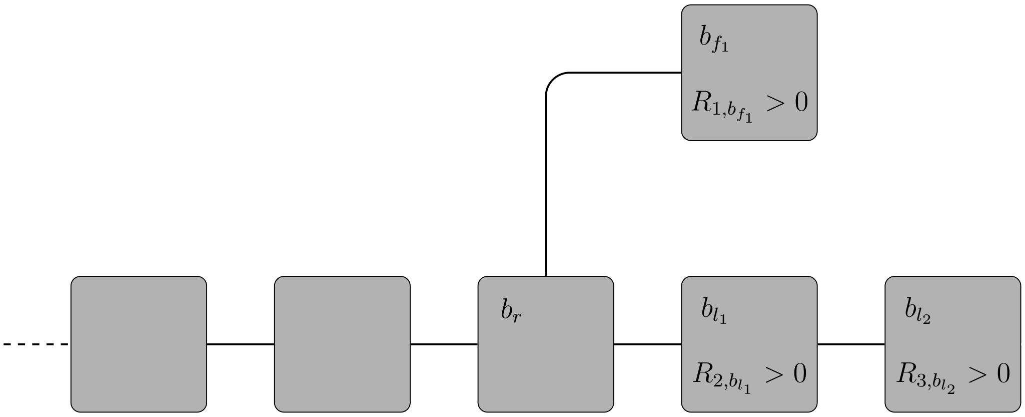

Figure 9: An illustration of the need for empty blocks.

With this in mind, consider Figure 9 and suppose $p_1 > p_2 + p_3$. Miner 1 has solved block b_{f_1}, whereas miners 2 and 3 have solved blocks b_{l_1} and b_{l_2}, respectively. The chain to b_{f_1} has the highest approval weights. Suppose miner 3 is mining a block b_t with a transaction that satisfies $Y_{1,b_t} \neq 0$. That is, she is attempting to add a block which contains a transaction for miner 1, either positive or negative. The approval weighted chain rule calls for her to append her block to block b_{f_1}. However, by appending her block b_{l_2}, she increases the approval weights of that chain by p_1. She thus induces a switch in the behavior of all miners following the approval weighted chain rule in the subsequent period, including miner 1. In a sense, miner 3 can "bribe" miner 1 and thus all other miners to join her preferred chain by either including some existing transaction from the pool of transactions, or by submitting and including a small receive transaction for miner 1 herself. This in turn allows her to capture the utility associated with her blocks rewards in b_{l_2}.

For this reason, one "empty block" in case of a fork ahead of the checkpoint helps re-establish consensus. Interestingly, for the decentralized system described in this paper to maintain and update the transaction ledger in a manner that achieves consensus, it must not include any new transactions for a brief period of time whenever there is disagreement. That is, for the decentralized system to function properly, it needs to *not* function for a short period of time.

Importantly, the concept of an empty block can be interpreted as a deviation from the principle of "liveness," a property that requires the ledger to update continuously. Liveness is often considered a constraint in the design of consensus protocols (see, e.g., Leshno et al. (2024)). Our findings suggest the existence of a trade-off between consensus for the system as a whole on the one hand, and liveness for a short period of time on the other hand.

Checkpoint lags and the number of empty blocks. To illustrate why the approval weight rule remains an equilibrium even for longer checkpoint lags, consider a scenario where the checkpoint updates to the grandparent of terminal blocks (rather than the parent). In the example of Figure 3 with the checkpoint given by b_r, miner i may find it profitable to work on the one-block short fork b_f if her computing power is sufficiently large. Under the checkpoint rule, the motivation for this deviation is to increase the length of the fork, inducing other miners to switch to the fork and turning it into the consensus chain.

However, under the approval weight rule and the restriction of Assumption 2, one of two things is true. Either the fork already has the highest approval weight, in which case all miners are called to work on it anyway, or it has lower approval weights, leaving miner i with no way to increase it. The reason is that all new blocks are empty whenever there is a fork ahead of the checkpoint. In either case, miner i has no incentive to deviate from the proposed equilibrium strategy.

Importantly, since miners solve one block per period, a longer checkpoint lag extends the time required to eliminate all forks ahead of the checkpoint, even when miners adhere to the approval weight rule. This, in turn, raises the number of empty blocks required to achieve consensus. If a fork ahead of the checkpoint exists and miners have the chance to add one non-empty block, then they may have incentives to deviate from the approval weight rule and "bribe" other miners to switch to their preferred chain by increasing its approval weights. The need for empty blocks therefore introduces a second trade-off between checkpoint lags and the duration of violating liveness.

6 Conclusion

In this paper, we demonstrate that incorporating a simple history-dependence in the form of checkpoints into a blockchain protocol can effectively prevent double-spending attacks. This is because miners collectively agree to disregard forks that attempt to reverse previously confirmed transactions. As a result, once a transaction is part of consensus, the corresponding balances cannot be spent a second time. Effectively, we have shown that the norm with Bitcoin of waiting at least six blocks (about one hour) before delivering non-blockchain goods ought to be linked explicitly to the consensus protocol, as is done on the Ethereum network.

Remarkably, achieving consensus 'ahead of the checkpoint' presents a greater challenge. To address this, we propose strategies that guide miners to prioritize forks containing balances of the group with the highest mining power over forks of greater length.

Implementing a checkpoint equilibrium raises two interesting issues. The first is how to publicly track the checkpoint. Part of the attractiveness of the longest chain rule in Bitcoin, see equation (7), is that it depends only on the current graph G_t and nothing from the history. This simplifies implementation since code need only download the current blockchain and calculate the longest chain. Our checkpoint strategy in equation (23) would require monitoring the blockchain for several periods. However, given a blockchain can record arbitrary data, it is interesting to consider how the current blockchain graph could also contain the checkpoint.

The second implementation consideration of our checkpoint equilibrium is network latency. Since the entire network of miners does not see new blocks at the same time it is possible, in fact likely, that forks will occur. In Bitcoin, for example, it takes about 11 seconds for all nodes to hear of a new block. Average new-block arrival time on Bitcoin is designed to be 600 seconds. Solving a block is Poisson and so a second block will arrive before all nodes are informed that a new block has already been solved about 1.8% of the time ($11\ seconds/600\ seconds \approx 1.8\%$). With a longest-chain rule, these forks are relatively innocuous as one of the forks will (randomly with subsequent blocks) emerge as longest. In our checkpoint equilibrium, the same will happen as long as the checkpoint information is not latent. Effectively, this means the checkpoint must be far enough back along the chain from new blocks. If the checkpoint block is too close it is possible miners would disagree about the checkpoint block causing the fork from latency to become permanent. Such disagreement would undermine the usefulness of the blockchain. Optimizing the checkpoint block—choosing the settlement lag—would require comparing the cost of a settlement lag with the likelihood of a permanent fork.

References

Amoussou-Guenou, Y., B. Biais, M. Potop-Butucaru, and S. Tucci-Piergiovanni (2024): "Committee-based blockchains as games between opportunistic players and adversaries," *The Review of Financial Studies*, 37, 409–443.

Auer, R., C. Monnet, and H. S. Shin (2021): "Permissioned distributed ledgers and the governance of money," *Available at SSRN 3770075*.

Bakos, Y. and H. Halaburda (2021): "Permissioned vs Permissionless Blockchain Platforms: Tradeoffs in Trust and Performance," *NYU Stern School of Business working paper*.

Benhaim, A., B. H. Falk, and G. Tsoukalas (2023): "Scaling blockchains: Can committee-based consensus help?" *Management Science*, 69, 6525–6539.

Biais, B., C. Bisiere, M. Bouvard, and C. Casamatta (2019): "The Blockchain Folk Theorem," *The Review of Financial Studies*, 32, 1662–1715.

Bondy, J. A. and U. S. R. Murty (1976): *Graph Theory with Applications*, vol. 290, Macmillan London.

Budish, E. (2025): "Trust at Scale: The Economic Limits of Cryptocurrencies and Blockchains," *The Quarterly Journal of Economics (forthcoming)*.

Buterin, V. and V. Griffith (2017): "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*.

Carlsten, M., H. Kalodner, S. M. Weinberg, and A. Narayanan (2016): "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 154–167.

Chiu, J. and T. V. Koeppl (2022): “The economics of cryptocurrency: Bitcoin and beyond,” *Canadian Journal of Economics/Revue canadienne d’économique*, 55, 1762–1798.

Cong, L. W., Z. He, and J. Li (2021): “Decentralized mining in centralized pools,” *The Review of Financial Studies*, 34, 1191–1235.

Eyal, I. and E. G. Sirer (2018): “Majority is not enough: Bitcoin mining is vulnerable,” *Communications of the ACM*, 61, 95–102.

Fischer, M. J., N. A. Lynch, and M. S. Paterson (1985): “Impossibility of distributed consensus with one faulty process,” *Journal of the ACM (JACM)*, 32, 374–382.

Gans, J. S. and H. Halaburda (2023): “"Zero Cost" Majority Attacks on Permissionless Blockchains,” *NBER Working Paper No. 31473.*

Garratt, R. J. and M. R. van Oordt (2023): “Why fixed costs matter for proof-of-work–based cryptocurrencies,” *Management Science*, 69, 6482–6507.

Halaburda, H., Z. He, and J. Li (2022): “An economic model of consensus on distributed ledgers,” *NBER Working Paper No. 29515.*

John, K., T. J. Rivera, and F. Saleh (2020): “Economic implications of scaling blockchains: Why the consensus protocol matters,” *Available at SSRN 3750467.*

Kang, K.-Y. (2023): “Cryptocurrency and double spending history: Transactions with zero confirmation,” *Economic Theory*, 75, 453–491.

Karakostas, D. and A. Kiayias (2021): “Securing proof-of-work ledgers via checkpointing,” in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 1–5.

Lagos, R. and R. Wright (2005): “A Unified Framework for Monetary Theory and Policy Analysis,” *Journal of Political Economy*, 113, 463–484.

Leshno, J. D., E. Shi, and R. Pass (2024): “On the viability of open-source financial rails: Economic security of permissionless consensus,” *arXiv preprint arXiv:2409.08951.*

Li, J. (2023): “On the security of optimistic blockchain mechanisms,” *Available at SSRN 4499357.*

Lynch, N. A. (1996): *Distributed algorithms*, Elsevier.

Makarov, I. and A. Schoar (2021): "Blockchain Analysis of the Bitcoin Market," *Available at SSRN 3942181.*

Moroz, D. J., D. J. Aronoff, N. Narula, and D. C. Parkes (2020): "Double-spend counterattacks: Threat of retaliation in proof-of-work systems," *arXiv preprint arXiv:2002.10736.*

Nakamoto, S. (2008): "Bitcoin: A peer-to-peer electronic cash system," *Bitcoin White paper.*

Neu, J., E. N. Tas, and D. Tse (2021): "Ebb-and-flow protocols: A resolution of the availability-finality dilemma," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE, 446–465.

Pagnotta, E. S. (2022): "Decentralizing money: Bitcoin prices and blockchain security," *The Review of Financial Studies*, 35, 866–907.

Rocheteau, G. and R. Wright (2005): "Money in Search Equilibrium, in Competitive Equilibrium, and in Competitive Search Equilibrium," *Econometrica*, 73, 175–202.

Saleh, F. (2021): "Blockchain without waste: Proof-of-stake," *The Review of Financial studies*, 34, 1156–1190.

Sankagiri, S., X. Wang, S. Kannan, and P. Viswanath (2021): "Blockchain cap theorem allows user-dependent adaptivity and finality," in *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part II 25*, Springer, 84–103.

A Proofs: Longest Chain Rule

A.1 Proof of Lemma 1

Suppose that the strategy profile σ is such that all miners follow the longest chain rule as in equation (9) from time t onwards. That is, $\sigma_{i,\tau}(H_\tau) = b^*_{i,\tau}$ for all i, after every history H_τ, and for all $\tau \geq t$. Suppose that $\mathcal{B}^{LC}(G_t)$ is not a singleton at time t. Since all miners follow the longest chain rule and only one miner appends a block at time t, there will be a single longest chain at time $t+1$. All miners then append all their new blocks to this consensus chain going forward. Note that blocks on consensus chains have a value of 1 for every miner.

Miner i's expected payoff from following the longest chain rule as in equation (9) is given by:

$$\begin{aligned}
V_{i,t}\left(\sigma; H_t\right) = & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta)q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta}\cdot\lambda_t(b,H_t)\right) \\
& + \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j\cdot\delta\cdot\sum_{\tau=0}^{\infty}\delta^{\tau}(1-\delta)\left(Y_{i,b_t} + \sum_{b'\in\mathcal{C}(b,G_t)}(Y_{i,b'}+R_{i,b'})\right) \\
& + \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j\cdot\delta\cdot\left(\delta\cdot\frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)}\frac{Y^{-}_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right) \\
& + p_i\cdot\delta\cdot\sum_{\tau=0}^{\infty}\delta^{\tau}(1-\delta)\left(Y_{i,b_t}+\bar{R}+\sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)}(Y_{i,b'}+R_{i,b'})\right) \\
& + p_i\cdot\delta\cdot\left(\delta\cdot\frac{Y^{-}_{i,b_t}}{\delta}+\sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)}\left(\frac{Y^{-}_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right)\right) \\
& + \delta^2\cdot\mathbb{E}_t\sum_{\tau=0}^{\infty}\delta^{\tau}(1-\delta)\left(\sum_{\upsilon=0}^{\tau}\sum_{b\in\mathcal{B}(G_{t+\upsilon+1})/\mathcal{B}(G_{t+\upsilon})} Y_{i,b}+R_{i,b}+\delta\cdot\frac{Y^{-}_{i,b}}{\delta}\right)
\end{aligned} \tag{36}$$

The first term is the current flow utility derived from the value of miner i's cumulative balances, plus the consumption flow utility due to spend transactions vesting at time t. The second and fourth terms taken together are the expected discounted sum of future flow utilities derived from current balances and newly added balances at the end of time t on the consensus chain. Miner i selects the longest chain running to $b^*_{i,t}$ as the consensus chain at time t and earns transactions and block rewards from time $t+1$ onwards with probability p_i. The probability that the longest chain running to some $b\in\mathcal{B}(G_t)$ is chosen by some other miner $j\neq i$ given by $\sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j$ for each such block. The third and fifth terms capture the expected consumption flow utility which miner i derives if the blocks containing the corresponding spend transactions become part of consensus. The last term captures the expected discounted sum of flow utilities due to transactions and block rewards in future blocks, which are all added from time $t+2$ onwards to the consensus chain.

The expression for the expected payoff simplifies to

$$
\begin{aligned}
V_{i,t}\left(\sigma;H_t\right) \;=\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y_{i,b}^{-}}{\delta}\cdot\lambda_t(b,H_t) \right) \qquad (37) \\
& + \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i:\, b_{j,t}^{*}=b\}} p_j\cdot\delta\cdot\left(Y_{i,b_t} + \delta\cdot\frac{Y_{i,b_t}^{-}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y_{i,b'}^{-}}{\delta}\cdot\Lambda_t(b',H_t) \right) \right) \\
& + p_i\cdot\delta\cdot\left(Y_{i,b_t} + \bar{R} + \delta\cdot\frac{Y_{i,b_t}^{-}}{\delta} + \sum_{b'\in\mathcal{C}(b_{i,t}^{*},G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y_{i,b'}^{-}}{\delta}\cdot\Lambda_t(b',H_t) \right) \right) \\
& + \delta^2\cdot\mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left[Y_{i,b_{t+\tau+1}} + R_{i,b_{t+\tau+1}} + \delta\cdot\frac{Y_{i,b_{t+\tau+1}}^{-}}{\delta} \right]
\end{aligned}
$$

Now consider an arbitrary deviation to $b\in\mathcal{B}^{LC}(G_t)$, $b\neq b_{i,t}^{*}$, by miner i, for this time period only. Denote this new profile by σ'. As one chain becomes the single longest chain under this profile σ', it remains true that consensus is achieved after the next block has been appended. Recall that future transactions and block rewards are taken as given. Furthermore, the current value of balances in existing blocks is a function of other miners' strategies but not of miner i's own strategy. Hence, the only difference in payoffs from this deviation is in the second term as miner i selects a different longest chain to become the consensus chain if she is successful in appending the next block:

$$\begin{aligned}
V_{i,t}\left(\sigma'; H_t\right) \;=\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y^-_{i,b}}{\delta}\cdot \lambda_t(b,H_t) \right) \\
& + \sum_{b\in\mathcal{B}^{LC}(G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j\cdot\delta\cdot\left(Y_{i,b_t} + \delta\cdot\frac{Y^-_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t) \right) \right) \\
& + \; p_i\cdot\delta\cdot\left(Y_{i,b_t} + \bar{R} + \delta\cdot\frac{Y^-_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t) \right) \right) \\
& + \; \delta^2\cdot\mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left[Y_{i,b_{t+\tau+1}} + R_{i,b_{t+\tau+1}} + \delta\cdot\frac{Y^-_{i,b_{t+\tau+1}}}{\delta} \right]
\end{aligned} \tag{38}$$

The difference in following the suggested tie-breaker and deviation payoff is

$$p_i\cdot\delta\cdot\left(\sum_{b'\in C(b^*_{i,t},G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t) \right) - \sum_{b'\in C(b,G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t) \right) \right) \tag{39}$$

Note that by definition of $b^*_{i,t}$, the above difference is (weakly) positive. Therefore, there is no strictly profitable one-shot deviation.

Conversely, should a candidate equilibrium strategy specify a different tie-breaking rule, then a one-shot deviation applying the rule specified in Equation (9) immediately yields a strictly profitable deviation. □

A.2 Proof of Proposition 1

Rewrite miner i's expected payoff from following the longest chain rule as:

$$\begin{aligned}
V_{i,t}\left(\sigma; H_t\right) \;=\; & \sum_{b \in \mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b} + R_{i,b}) + \frac{Y_{i,b}^{-}}{\delta} \cdot \lambda_t(b, H_t) \right) \qquad (40) \\
+ & \sum_{b \in \mathcal{B}^{LC}(G_t)} \; \sum_{\{j \neq i :\, b_{j,t}^{*} = b\}} p_j \cdot \delta \cdot \left(Y_{i,b_t} + \delta \cdot \frac{Y_{i,b_t}^{-}}{\delta} + \sum_{b' \in \mathcal{C}(b, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y_{i,b'}^{-}}{\delta} \cdot \Lambda_t(b', H_t) \right) \right) \\
+ & \; p_i \cdot \delta \cdot \left(Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y_{i,b_t}^{-}}{\delta} + \sum_{b' \in \mathcal{C}(b_{i,t}^{*}, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y_{i,b'}^{-}}{\delta} \cdot \Lambda_t(b', H_t) \right) \right) \\
+ & \; \delta^2 \cdot \mathbb{E}_t \left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y_{i,b_{t+1}}^{-}}{\delta} \right) \\
+ & \; \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta \cdot \frac{Y_{i,b_{t+\tau+2}}^{-}}{\delta} \right)
\end{aligned}$$

Now consider a one-shot deviation by miner i to some $\hat{b} \in \mathcal{B}^{-1}(G_t)$ before reverting to the longest chain rule. Denote this new profile by σ'. If miner i is successful in appending the next block, then the number of longest chains at time $t+1$ increases by one. Consensus is then stalled until period $t+2$, at which point one of the longest chains including the new one becomes the consensus chain. If miner i is not successful in appending the next block, then consensus is achieved at time $t+1$ as under the longest chain rule.

Miner i's expected payoff from the one-shot deviation is given by

$$\begin{aligned}
V_{i,t}\left(\sigma';H_t\right) \;=\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta)q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta}\cdot\lambda_t(b,H_t)\right) \qquad (41) \\
& + \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i\colon b^*_{j,t}=b\}} p_j\cdot\delta\cdot\left(Y_{i,b_t}+\delta\cdot\frac{Y^{-}_{i,b_t}}{\delta}+\sum_{b'\in\mathcal{C}(b,G_t)}\left(Y_{i,b'}+R_{i,b'}+\frac{Y^{-}_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right)\right) \\
& + \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i\colon b^*_{j,t}=b\}} p_j\cdot\delta^2\cdot\mathbb{E}_t\left(Y_{i,b_{t+1}}+R_{i,b_{t+1}}+\delta\cdot\frac{Y^{-}_{i,b_{t+1}}}{\delta}\right) \\
& + \; p_i\cdot\delta\cdot(1-\delta)\cdot\sum_{\{j\neq i\colon b^*_{j,t+1}=b_{i,t}\}}\frac{p_j}{1-p_i}\cdot\left(Y_{i,b_t}+\bar{R}+\sum_{b'\in\mathcal{C}(\hat{b},G_t)}(Y_{i,b'}+R_{i,b'})\right) \\
& + \; p_i\cdot\delta\cdot(1-\delta)\cdot\sum_{b\in\mathcal{B}^{LC}(G_t)}\sum_{\{j\neq i\colon b^*_{j,t+1}=b\}}\frac{p_j}{1-p_i}\sum_{b'\in\mathcal{C}(b,G_t)}(Y_{i,b'}+R_{i,b'}) \\
& + \; p_i^2\cdot\delta^2\cdot\Bigg(\mathbb{E}_t\left[Y_{i,b_{t+1}}+\bar{R}+\delta\cdot\frac{Y^{-}_{i,b_{t+1}}}{\delta}\right]+Y_{i,b_t}+\bar{R}+\frac{Y^{-}_{i,b_t}}{\delta} \\
& \qquad\qquad + \sum_{b'\in\mathcal{C}(\hat{b},G_t)}\left(Y_{i,b'}+R_{i,b'}+\frac{Y^{-}_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right)\Bigg) \\
& + \; p_i\cdot\sum_{\{j\neq i\colon b^*_{j,t+1}=b_{i,t}\}} p_j\cdot\delta^2\cdot\Bigg(\mathbb{E}_t\left[Y_{i,b_{t+1}}+\delta\cdot\frac{Y^{-}_{i,b_{t+1}}}{\delta}\right]+Y_{i,b_t}+\bar{R}+\frac{Y^{-}_{i,b'}}{\delta} \\
& \qquad\qquad + \sum_{b'\in\mathcal{C}(\hat{b},G_t)}\left(Y_{i,b'}+R_{i,b'}+\frac{Y^{-}_{i,b_t}}{\delta}\cdot\Lambda_t(b',H_t)\right)\Bigg) \\
& + \; p_i\cdot\sum_{b\in\mathcal{B}^{LC}(G_t)}\sum_{\{j\neq i\colon b^*_{j,t+1}=b\}} p_j\cdot\delta^2\cdot\left(\mathbb{E}_t\left[Y_{i,b_{t+1}}+\delta\cdot\frac{Y^{-}_{i,b_{t+1}}}{\delta}\right]+\sum_{b'\in\mathcal{C}(b,G_t)}\left(Y_{i,b'}+R_{i,b'}+\frac{Y^{-}_{i,b'}}{\delta}\cdot\Lambda_t(b',H_t)\right)\right) \\
& + \; \delta^3\cdot\mathbb{E}_t\sum_{\tau=0}^{\infty}\delta^{\tau}\left(Y_{i,b_{t+\tau+2}}+R_{i,b_{t+\tau+2}}+\delta\cdot\frac{Y^{-}_{i,b_{t+\tau+2}}}{\delta}\right)
\end{aligned}$$

The first term is the unchanged current flow utility derived from the value of miner i's cumulative balances. The second and third terms taken together are the expected discounted sum of future flow utilities derived from current balances and newly added balances at time $t+1$ on the time-t longest chains—should another miner be successful at time t and one of the longest chains becomes the consensus chain at time $t+1$—multiplied by their corresponding probabilities with which they will become the consensus chain. The expressions also contain the consumption flow utility derived from unvested spend transactions.

Miner i is successful in appending block $b_{i,t}$ to block $\hat{b}$ with probability p_i. The fourth term is the expected discounted flow utility derived from the balances on this chain at time $t+1$, given the decision of miners $j \neq i$ to also mine on this chain at time $t+1$ and their respective computing powers. The fifth term is the corresponding discounted flow utility on the other longest chains.

The sixth term is the discounted life-time utility derived from the balances (including consumption utility derived from unvested spend transactions) on the fork from $t+2$ onwards if it becomes the consensus chain and miner i was successful at appending blocks in both time periods t and $t+1$. This event occurs probability p_i^2. The seventh term is the corresponding discounted life-time utility if miner i was successful at appending blocks at time t and another miner who was working on the fork at time $t+1$ was successful at appending a block, which has probability $p_i \cdot \sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} p_j$. The eight term is the corresponding discounted life-time utility if another miner who was working on a previous longest chain rather than the fork at time $t+1$ was successful at appending a block, which has probability $p_i \cdot \sum_{\{j \neq i :\, b^*_{j,t+1} = b\}} p_j$ for each block $b \in \mathcal{B}^{LC}(G_t)$.

The final term captures the expected discounted sum of flow utilities due to transactions and block rewards in future blocks, which are all appended to the consensus chain.

Recall that miners take transactions $Y_{i,b}$ as given and note that $\mathbb{E}_t R_{i,b_{t+\tau+1}} = p_i \bar{R}$ for all $\tau \geq 0$. Further note that

$$\sum_{b \in \mathcal{B}^{LC}(G_t)} \sum_{\{j \neq i :\, b^*_{j,t} = b\}} p_j = 1 - p_i.$$

Then taking differences in expected payoffs and simplifying shows that $V_{i,t}(\sigma; H_t) \geq V_{i,t}(\sigma'; H_t)$ if and only if the expression in (14) is satisfied. □

A.3 Proof of Proposition 2

Since $Y_{j,b} = 0$ for all miners j and all blocks b (with the notable exception of $Y_{i,b_{l_1}} < 0$) and $R_{i,b_{l_1}} = R_{i,b_{l_2}} = \bar{R}$, the expression for the expected payoff in (40) simplifies to

$$\begin{aligned} V_{i,t-1}\left(\sigma; H_{t-1}\right) \; &= (1-\delta)(Y_{i,b_{l_1}} + 2\bar{R}) \qquad (42) \\ &+ \; (1-p_i) \cdot \delta \cdot (Y_{i,b_{l_1}} + 2\bar{R}) \\ &+ \; p_i \cdot \delta \cdot (Y_{i,b_{l_1}} + 2\bar{R} + \bar{R}) \\ &+ \; \delta^2 \cdot \mathbb{E}_t \left(R_{i,b_{t+1}}\right) \\ &+ \; \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left(R_{i,b_{t+\tau+2}}\right) \end{aligned}$$

Note that $\mathbb{E}_t\left(R_{i,b_{t+1+s}}\right) = p_i\bar{R}$ for all $s \geq 0$. To ease notation, define $D \equiv Y_{i,b_{l_1}} + 2\bar{R}$. Plugging in and simplifying, we have

$$V_{i,t-1}\left(\sigma; H_{t-1}\right) \; = \; D \; + \; \delta \cdot \frac{p_i\bar{R}}{1-\delta} \; \equiv \; V \qquad (43)$$

The payoff from following the longest chain rule is therefore given by miner i's current balances D plus the expected present value of future block rewards. The deviation payoff is given by

$$\begin{aligned} V_{i,t-1}\left(\sigma'; H_{t-1}\right) \; &= (1-\delta)D \qquad (44) \\ &+ \; (1-p_i) \cdot \delta \cdot V \; + \; p_i \cdot \delta \cdot (1-\delta)D \\ &+ \; p_i \cdot (1-p_i) \cdot \delta^2 \cdot V \; + \; p_i^2 \cdot \delta^2 \cdot (1-\delta)D \\ &+ \; p_i^2 \cdot (1-p_i) \cdot \delta^3 \cdot V \; + \; p_i^3 \cdot \delta^3 \cdot \left[3\bar{R} + (V-D)\right] \end{aligned}$$

The first term on the RHS is the flow utility from balances D at time $t-1$. With probability $(1-p_i)$, some other miner $j \neq i$ appends the new block and extends the longest chain. Miner i then reverts to following the longest chain rule and achieves a life-time payoff of V from time t onwards. This is captured by the second term on the RHS. With probability p_i, miner i is successful and creates the fork at time $t+1$. Since all other miners continue working on the longest chain, miner i's discounted flow utility from balances D at time t is given by $\delta(1-\delta)D$. This is the third term on the RHS.

The fourth term on the RHS captures that, conditional on having created the fork at time $t-1$, with probability $(1-p_i)$ some other miner $j \neq i$ appends the new block and extends the longest chain. Miner i then reverts to following the longest chain rule. The block rewards in b_f are lost and miner i achieves a life-time payoff of V from time $t+1$ onwards. The fifth term is the analogous discounted flow utility at time $t+1$ if miner i has successfully extended the fork to be of equal length (recall that all miners continue working on the previous longest chain).

The sixth term is analogous to the second and fourth term above. The final term is the expected, discounted life-time payoff conditional on succeeding with the double-spending attack, which occurs with probability p_i^3, i.e., the probability that miner i creates and extends the fork at times $t-1$ and t as well as turns it into the only longest chain at time $t+1$. The life-time payoff is then given by the three block rewards earned on the fork as well as the expected present value of future block rewards. Importantly, the payoff does not contain the balances D on the previous longest chain, which are no longer part of consensus.

Algebraic manipulation reveals that

$$V_{i,t-1}\left(\sigma; H_{t-1}\right) - V_{i,t-1}\left(\sigma'; H_{t-1}\right) \;=\; p_i\delta \cdot \left[p_i^2\delta^2 \cdot Y_{i,b_{l_1}} + (1+p_i\delta)\cdot \bar{R}\right] \tag{45}$$

The expression is negative if the inequality in (18) is satisfied. The strategy profile σ' therefore constitutes a profitable deviation, and the claim follows. □

A.4 Proof of Lemma 2

If $Y_{i,b_t} \geq 0$, the condition in (14) becomes

$$\begin{aligned} & Y_{i,b_t} + \bar{R} + \sum_{b' \in \mathcal{C}(b^*_{i,t},G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \\ & \quad \geq \; (1-\delta) \cdot \sum_{\{j\neq i:\, b^*_{j,t+1}=b_{i,t}\}} \frac{p_j}{1-p_i} \cdot \left(Y_{i,b_t} + \bar{R} + \sum_{b'\in\mathcal{C}(\hat{b},G_t)} (Y_{i,b'}+R_{i,b'}) \right) \\ & \qquad + \; (1-\delta) \cdot \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i:\, b^*_{j,t+1}=b\}} \frac{p_j}{1-p_i} \cdot \sum_{b'\in\mathcal{C}(b,G_t)} (Y_{i,b'}+R_{i,b'}) \\ & \qquad + \; \delta \cdot \left\{ p_i + \sum_{\{j\neq i:\, b^*_{j,t+1}=b_{i,t}\}} p_j \right\} \cdot \left(Y_{i,b_t} + \bar{R} + \sum_{b'\in\mathcal{C}(\hat{b},G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \right) \\ & \qquad + \; \delta \cdot \sum_{b\in\mathcal{B}^{LC}(G_t)} \sum_{\{j\neq i:\, b^*_{j,t+1}=b\}} p_j \cdot \sum_{b'\in\mathcal{C}(b,G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \end{aligned} \tag{46}$$

Since there are only longest chains as $\#C(b,G_t) = \#C(b',G_t)$ for every $b, b' \in \mathcal{T}(G_t)$, and by the definition of $b^*_{i,t}$, we have

$$\begin{aligned} \sum_{b'\in\mathcal{C}(\hat{b},G_t)} (Y_{i,b'}+R_{i,b'}) \; & \leq \; Y_{i,b_t} + \bar{R} + \sum_{b'\in\mathcal{C}(\hat{b},G_t)} (Y_{i,b'}+R_{i,b'}) \\ & \leq \; Y_{i,b_t} + \bar{R} + \sum_{b'\in\mathcal{C}(\hat{b},G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \\ & \leq \; Y_{i,b_t} + \bar{R} + \sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \end{aligned} \tag{47}$$

for every $\hat{b} \in \mathcal{B}^{-1}(G_t)$. Plugging in, we find that the LHS of (46) is an upper bound to the RHS of (46). The claim follows. □

A.5 Proof of Lemma 3

If $Y_{i,b_t} < 0$, the condition in (14) becomes

$$\begin{aligned}
&\bar{R} + \sum_{b' \in \mathcal{C}(b^*_{i,t}, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \qquad (48)\\
&\quad \geq\ (1-\delta) \cdot \sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} \frac{p_j}{1-p_i} \cdot \left(Y_{i,b_t} + \bar{R} + \sum_{b' \in \mathcal{C}(\hat{b}, G_t)} (Y_{i,b'} + R_{i,b'}) \right)\\
&\qquad +\ (1-\delta) \cdot \sum_{b \in \mathcal{B}^{LC}(G_t)} \sum_{\{j \neq i :\, b^*_{j,t+1} = b\}} \frac{p_j}{1-p_i} \cdot \sum_{b' \in \mathcal{C}(b, G_t)} (Y_{i,b'} + R_{i,b'})\\
&\qquad +\ \delta \cdot \left\{ p_i + \sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} p_j \right\} \cdot \left(Y_{i,b_t} + \bar{R} + \frac{Y^-_{i,b_t}}{\delta} + \sum_{b' \in \mathcal{C}(\hat{b}, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right) \right)\\
&\qquad +\ \delta \cdot \sum_{b \in \mathcal{B}^{LC}(G_t)} \sum_{\{j \neq i :\, b^*_{j,t+1} = b\}} p_j \cdot \sum_{b' \in \mathcal{C}(b, G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^-_{i,b'}}{\delta} \cdot \Lambda_t(b', H_t) \right)
\end{aligned}$$

We now use the bounds derived in the proof of Lemma 2. Since there are only longest chains as $\#C(b, G_t) = \#C(b', G_t)$ for every $b, b' \in \mathcal{T}(G_t)$, and by the definition of $b^*_{i,t}$, we obtain the following sufficient condition for (48) to hold:

$$\begin{aligned}
\bar{R}\ &\geq (1-\delta) \cdot \sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} \frac{p_j}{1-p_i} \cdot \big(Y_{i,b_t} + \bar{R} \big) \qquad (49)\\
&\quad +\ \delta \cdot \left\{ p_i + \sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} p_j \right\} \cdot \left(Y_{i,b_t} + \bar{R} + \frac{Y^-_{i,b_t}}{\delta} \right)
\end{aligned}$$

The condition is satisfied if $\sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} p_j = 1 - p_i$. Since the RHS of (49) is linear in all terms, it is either maximized or minimized at $\sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} p_j = 0$. If it is minimized, then the condition is always satisfied. Suppose the RHS is maximized at $\sum_{\{j \neq i :\, b^*_{j,t+1} = b_{i,t}\}} p_j = 0$. Plugging in and rearranging, the sufficient condition for (48) to hold becomes (19) as stated in the lemma. The claim follows. □

B Proofs: Checkpoint Rule

B.1 Proof of Proposition 3

The payoff of following the candidate equilibrium strategy is given by

$$\begin{aligned}
V_{i,t}\left(\sigma; H_t\right) \; = \; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b} + R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta} \cdot \lambda_t(b, H_t) \right) \\
& + \sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j \cdot \delta \cdot \left(Y_{i,b_t} + \delta \cdot \frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b}}{\delta} \right) \\
& + \; p_i \cdot \delta \cdot \left(Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta} \right) \\
& + \; \delta^2 \cdot \mathbb{E}_t \left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y^{-}_{i,b_{t+1}}}{\delta} \right) \\
& + \; \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta \cdot \frac{Y^{-}_{i,b_{t+\tau+2}}}{\delta} \right)
\end{aligned} \tag{50}$$

Equation (50) is equivalent to equation (40), now incorporating checkpoints. The first term on the RHS is the flow utility due to balances at time t, plus the consumption utility due to vesting spend transactions. The second and third terms taken together are the expected discounted sum of future flow utilities derived from current balances and newly added balances at time $t+1$ on the consensus chain, plus the consumption utility from vesting spend transactions. Miner i selects the longest chain running to $b^*_{i,t}$ as the consensus chain at time t with probability p_i. The probability that the longest chain running to some $b \in \mathcal{B}^{CP}\big(b^{CP}(H_t), G_t\big)$ is chosen by some other miner $j \neq i$ given by $\sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j$ for each such block. The fourth and fifth term capture the expected discounted sum of flow utilities due to transactions and block rewards in future blocks as well as the corresponding consumption flow utilities, which are all added from time $t+1$ onwards to the consensus chain.

We now consider a deviation to some block $b \notin \mathcal{J}(b^{CP}(H_t), G_t)$. Note that this implies that b_t does not lie on any chain to any block on the subgraph following the checkpoint if miner i appends it. Since all other miners play checkpoint strategies in all time periods, and since miner i herself reverts to playing checkpoint strategies from the next time period, no miner is working on the chain containing block b_t at time $t+1$ and any other future time period. Hence, we have $q_{i,b_t,\tau} = 0$ and $\lambda_\tau(b_t, G_\tau) = 0$ for all $\tau \geq t+1$.

Denoting the strategy profile for this deviation by σ', we have

$$
\begin{aligned}
V_{i,t}\left(\sigma'; H_t\right) \;=\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b} + R_{i,b}) + \frac{Y^-_{i,b}}{\delta} \cdot \lambda_t(b, H_t) \right) \qquad (51) \\
+ \; & \sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j \cdot \delta \cdot \left(Y_{i,b_t} + \delta \cdot \frac{Y^-_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^-_{i,b}}{\delta} \right) \\
+ \; & (1-p_i) \cdot \delta^2 \cdot \mathbb{E}_t \left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y^-_{i,b_{t+1}}}{\delta} \right) \\
+ \; & p_i \cdot \delta \cdot (1-\delta) \cdot \sum_{b\in\mathcal{B}^{CP}(H_t,G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} \frac{p_j}{1-p_i} \sum_{b'\in\mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) \\
+ \; & p_i \cdot \delta \cdot \sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j \cdot \delta \cdot \left(\mathbb{E}_t \left(Y_{i,b_{t+1}} + \delta \cdot \frac{Y^-_{i,b_{t+1}}}{\delta} \right) + \sum_{b'\in\mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^-_{i,b}}{\delta} \right) \\
+ \; & p_i^2 \cdot \delta^2 \cdot \left(\mathbb{E}_t \left[Y_{i,b_{t+1}} + \bar{R} + \delta \cdot \frac{Y^-_{i,b_{t+1}}}{\delta} \right] + \sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^-_{i,b^*_{i,t}}}{\delta} \right) \\
+ \; & \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^\tau \left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta \cdot \frac{Y^-_{i,b_{t+\tau+2}}}{\delta} \right)
\end{aligned}
$$

The first, second and final term on the RHS of (51) are unchanged relative to the strategy profile σ. The third term captures the following. Some miner $j \neq i$ is selected with probability $(1-p_i)$. Consensus is then achieved, the checkpoint updates. Since all miners revert to checkpoint strategies, consensus remains and miner i derives flow utility from positive transactions and block rewards in block b_{t+1} from time $t+2$ onwards.

With probability p_i, no block is added to the one of the longest chains on the subgraph. Hence, the checkpoint does not update. All miners, including miner i, then work on their preferred longest chain on the subgraph. The fourth term captures the time $t+1$ flow utility due to balances in blocks on the longest chains, given the other miners' location choices. The fifth and sixth term are the expected discounted life-time flow utilities associated with the longest chains on the subgraph, multiplied by their respective probabilities of becoming the consensus chains. In addition, they contain the life-time flow utility due to positive transactions and block rewards in block b_{t+1}.

Note that for every $b \in \mathcal{B}^{CP}\big(b^{CP}(H_t), G_t\big)$ we have

$$\begin{aligned} \sum_{b' \in \mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) \;&\leq\; \sum_{b' \in \mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b}}{\delta} \\ &\leq\; \sum_{b' \in \mathcal{C}(b^*_{i,t},G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta} \end{aligned} \tag{52}$$

where the first inequality follows since we only add (weakly) positive objects, and the second inequality follows by definition. This in turn implies that

$$\begin{aligned} &\sum_{b \in \mathcal{B}^{CP}(b^{CP}(H_t),G_t)} \; \sum_{\{j \neq i :\, b^*_{j,t} = b\}} p_j \sum_{b' \in \mathcal{C}(b,G_t)} (Y_{i,b'} + R_{i,b'}) \\ &\qquad\qquad \leq \; \sum_{b \in \mathcal{B}^{CP}(H_t,G_t)} \; \sum_{\{j \neq i :\, b^*_{j,t} = b\}} p_j \left(\sum_{b' \in \mathcal{C}(b^*_{i,t},G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta} \right) \\ &\qquad\qquad \leq \; (1 - p_i) \left(\sum_{b' \in \mathcal{C}(b^*_{i,t},G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta} \right) \end{aligned} \tag{53}$$

Using these inequalities, and collecting the expected transactions and block rewards for block b_{t+1}, it follows that

$$
\begin{aligned}
V_{i,t}\left(\sigma';H_t\right) \;\leq\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta)q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta}\cdot\lambda_t(b,H_t)\right) \\
&+ \sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j\cdot\delta\cdot\left(Y_{i,b_t}+\delta\cdot\frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^{-}_{i,b}}{\delta}\right) \\
&+ \; p_i\cdot\delta\cdot(1-\delta)\cdot\left(\sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta}\right) \\
&+ \; p_i\cdot\delta\cdot(1-p_i)\cdot\delta\cdot\left(\sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta}\right) \\
&+ \; p_i^2\cdot\delta^2\cdot\left(\sum_{b'\in\mathcal{C}(b^*_{i,t},G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^{-}_{i,b^*_{i,t}}}{\delta}\right) \\
&+ \; \delta^2\cdot\mathbb{E}_t\left(Y_{i,b_{t+1}}+R_{i,b_{t+1}}+\delta\cdot\frac{Y^{-}_{i,b_{t+1}}}{\delta}\right) \\
&+ \; \delta^3\cdot\mathbb{E}_t\sum_{\tau=0}^{\infty}\delta^{\tau}\left(Y_{i,b_{t+\tau+2}}+R_{i,b_{t+\tau+2}}+\delta\cdot\frac{Y^{-}_{i,b_{t+\tau+2}}}{\delta}\right)
\end{aligned}
\tag{54}
$$

Then, taking differences, we have

$$
V_{i,t}\left(\sigma;H_t\right) - V_{i,t}\left(\sigma';H_t\right) \;\geq\; p_i\cdot\delta\cdot\left(Y_{i,b_t}+\bar{R}+\delta\cdot\frac{Y^{-}_{i,b_t}}{\delta}\right) \;>\; 0 \tag{55}
$$

From here it follows that any one-shot deviation from the checkpoint strategy to a block behind the checkpoint is strictly not profitable. □

B.2 Proof of Proposition 4

The payoff from following the candidate equilibrium strategy is given by (50). By Proposition (3), and since only chains with one block ahead of the checkpoint remain, we only need to consider deviations from the candidate equilibrium strategy to the checkpoint block itself. Denoting the strategy profile for this deviation to the checkpoint by σ', the corresponding payoff reads

$$
\begin{aligned}
V_{i,t}\left(\sigma'; H_t\right) \;=\; & \sum_{b \in \mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b} + R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta} \cdot \lambda_t(b, H_t) \right) \qquad (56) \\
&+ \sum_{b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_t)} \; \sum_{\{j \neq i :\, b^*_{j,t} = b\}} p_j \cdot \delta \cdot \left(Y_{i,b_t} + \delta \cdot \frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b' \in \mathcal{C}(b, G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b}}{\delta} \right) \\
&+ (1 - p_i) \cdot \delta^2 \cdot \mathbb{E}_t \left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y^{-}_{i,b_{t+1}}}{\delta} \right) \\
&+ p_i \cdot \delta \cdot (1 - \delta) \cdot \sum_{b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_{t+1})} \; \sum_{\{j \neq i :\, b^*_{j,t+1} = b\}} \frac{p_j}{1 - p_i} \sum_{b' \in \mathcal{C}(b, G_{t+1})} (Y_{i,b'} + R_{i,b'}) \\
&+ p_i \cdot \delta^2 \cdot \sum_{b \in \mathcal{B}^{CP}(b^{CP}(H_t), G_{t+1})} \; \sum_{\{j :\, b^*_{j,t+1} = b\}} p_j \cdot \left(\mathbb{E}_t \left[Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y^{-}_{i,b_t}}{\delta} \right] + \sum_{b' \in \mathcal{C}(b, G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b}}{\delta} \right) \\
&+ \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta \cdot \frac{Y^{-}_{i,b_{t+\tau+2}}}{\delta} \right)
\end{aligned}
$$

where $G_{t+1} = G_t \bigcup \left(b_{i,t}, \left(b_{i,t}, b^{CP}(H_t) \right) \right)$ is the graph at time $t{+}1$ should miner i be successful in appending block $b_{i,t}$ to the checkpoint block $b^{CP}(H_t)$ at time t. The first three terms as well as the final term on the RHS are unchanged relative to the payoff under the candidate equilibrium strategy. The fourth term captures the discounted flow utility at time $t+1$ given other miners' location choices if miner i is successful in appending block $b_{i,t}$ to the checkpoint, which occurs with probability p_i. The fifth term captures the expected utility due to the balances on one of the longest chains—i.e., the longest chains at time $t+1$ plus the one added by miner i by creating a fork—becoming the consensus chain from time $t+2$ onwards.

The expression simplifies to

$$\begin{aligned}
V_{i,t}\left(\sigma'; H_t\right) \;=\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta)q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y^-_{i,b}}{\delta}\cdot\lambda_t(b,H_t)\right) \qquad (57)\\
& + \sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_t)} \; \sum_{\{j\neq i:\, b^*_{j,t}=b\}} p_j\cdot\delta\cdot\left(Y_{i,b_t}+\delta\cdot\frac{Y^-_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^-_{i,b}}{\delta}\right)\\
& + \delta^2\cdot\mathbb{E}_t\left(Y_{i,b_{t+1}}+R_{i,b_{t+1}}+\delta\cdot\frac{Y^-_{i,b_{t+1}}}{\delta}\right)\\
& + p_i\cdot\delta\cdot(1-\delta)\cdot\sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_{t+1})} \; \sum_{\{j\neq i:\, b^*_{j,t+1}=b\}} \frac{p_j}{1-p_i}\cdot(Y_{i,b}+R_{i,b})\\
& + p_i\cdot\delta^2\cdot\sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_{t+1})} \; \sum_{\{j:\, b^*_{j,t+1}=b\}} p_j\cdot\left(\sum_{b'\in\mathcal{C}(b,G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^-_{i,b}}{\delta}\right)\\
& + \delta^3\cdot\mathbb{E}_t\sum_{\tau=0}^{\infty}\delta^\tau\left(Y_{i,b_{t+\tau+2}}+R_{i,b_{t+\tau+2}}+\delta\cdot\frac{Y^-_{i,b_{t+\tau+2}}}{\delta}\right)
\end{aligned}$$

Comparing the expressions in (50) and (57), we have $V_{i,t}\left(\sigma; H_t\right) \geq V_{i,t}\left(\sigma'; H_t\right)$ if and only if

$$\begin{aligned}
& Y_{i,b_t}+\bar{R}+\delta\cdot\frac{Y^-_{i,b_t}}{\delta} + \left(Y_{i,b^*_{i,t}}+R_{i,b^*_{i,t}}+\frac{Y^-_{i,b^*_{i,t}}}{\delta}\right) \qquad (58)\\
& \quad \geq\; (1-\delta)\cdot\sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_{t+1})} \; \sum_{\{j\neq i:\, b^*_{j,t+1}=b\}} \frac{p_j}{1-p_i}\cdot(Y_{i,b}+R_{i,b})\\
& \qquad + \delta\cdot\sum_{b\in\mathcal{B}^{CP}(b^{CP}(H_t),G_{t+1})} \; \sum_{\{j:\, b^*_{j,t+1}=b\}} p_j\cdot\left(Y_{i,b}+R_{i,b}+\frac{Y^-_{i,b}}{\delta}\right)
\end{aligned}$$

Rearranging terms, and noting that $b^*_{i,t+1}=b_{i,t}$, the condition in (27) follows. The claim then follows by the proof of Lemmas 2 and 3. □

C Proof of Proposition 5: Approval Weights

First, define b_t^* to be the only element in the set $\mathcal{A}(H_t)$. The payoff of following the candidate equilibrium strategy is then given by

$$
\begin{aligned}
V_{i,t}\left(\sigma; H_t\right) \;=\; & \sum_{b \in \mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t} (Y_{i,b} + R_{i,b}) + \frac{Y_{i,b}^-}{\delta} \cdot \lambda_t(b, H_t) \right) \qquad (59) \\
& +\; (1-p_i) \cdot \delta \cdot \left(Y_{i,b_t} + \delta \cdot \frac{Y_{i,b_t}^-}{\delta} + \sum_{b' \in \mathcal{C}(b_t^*, G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y_{i,b_t^*}^-}{\delta} \right) \\
& +\; p_i \cdot \delta \cdot \left(Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y_{i,b_t}^-}{\delta} + \sum_{b' \in \mathcal{C}(b_t^*, G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y_{i,b_t^*}^-}{\delta} \right) \\
& +\; \delta^2 \cdot \mathbb{E}_t \left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y_{i,b_{t+1}}^-}{\delta} \right) \\
& +\; \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^\tau \left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta \cdot \frac{Y_{i,b_{t+\tau+2}}^-}{\delta} \right)
\end{aligned}
$$

The expression above captures that consensus is achieved and the checkpoint updates in the subsequent period with probability one on the equilibrium path. By definition, the approval weights are weakly increasing when adding a block that only contains block rewards. Further note that the length of the chain to which a block is added is, trivially, strictly increasing. Hence, the checkpoint will update to b_t^* at time $t+1$ on the equilibrium path.

We now consider three possible one-shot deviations from the equilibrium strategy. First, consider a one-shot deviation to some $b \notin \mathcal{J}\big(b^{CP}(H_t), G_t\big)$. Such a deviation is strictly unprofitable by the proof of Proposition 3, replacing the set $\mathcal{B}^{CP}\big(b^{CP}(H_t), G_t\big)$ by $\mathcal{A}(H_t)$.

Second, consider a deviation to some $b \notin \mathcal{A}(H_t)$, $b \in \mathcal{J}\big(b^{CP}(H_t), G_t\big)$. Let the history in the subsequent period, should miner i be successful in appending block $b_{i,t}$ to b, be denoted by H'_{t+1}. Suppose that $b_{i,t} \notin \mathcal{A}(H'_{t+1})$. Since all miners including miner i play equilibrium strategies at time $t+1$, the chain leading to $b_{i,t}$ is abandoned at time $t+1$—just like the chain to any block appended to any $b \notin J\big(b^{CP}(H_t), G_t\big)$. Thus, we can again call to the proof of Proposition 3 to show that such a deviation is strictly unprofitable.

Third, consider a deviation to some $b \notin \mathcal{A}(H_t)$, $b \in \mathcal{J}\big(b^{CP}(H_t), G_t\big)$ but now suppose that $b_{i,t} \in \mathcal{A}(H'_{t+1})$. Since $b_{i,t} \in \mathcal{A}(H'_{t+1})$, and since $\mathcal{A}(H'_{t+1})$ is a singleton, all miners including miner i work on block $b_{i,t}$ at time $t+1$ and consensus is achieved. Note that there are two scenarios in which $b \notin \mathcal{A}(H_t)$ and $b_{i,t} \in \mathcal{A}(H'_{t+1})$, which we discuss in turn.

In the first scenario, there is no fork ahead of the checkpoint $b^{CP}(H_t)$ and the checkpoint itself is the only block $b \notin \mathcal{A}(H_t)$, $b \in \mathcal{J}\big(b^{CP}(H_t), G_t\big)$. Consider therefore a deviation to the checkpoint block $b^{CP}(H_t)$. Denote the strategy profile for such a deviation by σ'. The deviation payoff reads

$$
\begin{aligned}
V_{i,t}\left(\sigma'; H_t\right) \;=\; & \sum_{b \in \mathcal{B}(G_t)} \left((1-\delta) q_{i,b,t}(Y_{i,b} + R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta} \cdot \lambda_t(b, H_t) \right) \\
& + \; (1-p_i) \cdot \delta \cdot \left(Y_{i,b_t} + \delta \cdot \frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b' \in \mathcal{C}(b_t^*, G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b_t^*}}{\delta} \right) \\
& + \; p_i \cdot \delta \cdot \left(Y_{i,b_t} + \bar{R} + \delta \cdot \frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b' \in \mathcal{C}(b^{CP}(H_t), G_t)} (Y_{i,b'} + R_{i,b'}) + \frac{Y^{-}_{i,b^{CP}(H_t)}}{\delta} \right) \\
& + \; \delta^2 \cdot \mathbb{E}_t \left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta \cdot \frac{Y^{-}_{i,b_{t+1}}}{\delta} \right) \\
& + \; \delta^3 \cdot \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^{\tau} \left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta \cdot \frac{Y^{-}_{i,b_{t+\tau+2}}}{\delta} \right)
\end{aligned}
\tag{60}
$$

The only difference between (59) and (60) is that under the deviation profile σ' the current terminal block b_t^* is abandoned should miner i be successful in appending block $b_{i,t}$ to the checkpoint $b^{CP}(H_t)$. It is straightforward to see that $V_{i,t}\left(\sigma; H_t\right) \geq V_{i,t}\left(\sigma'; H_t\right)$, with the inequality strict unless $Y_{i,b_t^*} = R_{i,b_t^*} = 0$.

In the second scenario, there is a fork ahead of the checkpoint $b^{CP}(H_t)$ at time t and hence by Assumption 2 we have $Y_{j,b_t} = 0$ for all miners $j = \{1, 2, ..., N\}$ but $R_{i,b_{i,t}} = \bar{R}$ for miner i. It then must be true that miner i has no data on the chain leading to b ahead of the checkpoint: $\sum_{b' \in \mathcal{C}(b,G_t) \setminus \mathcal{C}(b^{CP}(H_t),G_t)} \left(Y_{i,b'} + R_{i,b'} + \frac{Y^{-}_{i,b'}}{\delta} \right) = 0$. Otherwise, their probability p_i would have already been included in the approval weight of block b by (28), and hence both blocks $b_{i,t}$ and b would have the same approval weights, a contradiction to $b \notin \mathcal{A}(H_t)$ and $b_{i,t} \in \mathcal{A}(H'_{t+1})$.

Denote the strategy profile for such a deviation by σ''. The deviation payoff then reads

$$
\begin{aligned}
V_{i,t}\left(\sigma'';H_t\right) \;=\; & \sum_{b\in\mathcal{B}(G_t)} \left((1-\delta)q_{i,b,t}(Y_{i,b}+R_{i,b}) + \frac{Y^{-}_{i,b}}{\delta}\cdot\lambda_t(b,H_t)\right) \qquad (61)\\
& + \; (1-p_i)\cdot\delta\cdot\left(Y_{i,b_t} + \delta\cdot\frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b^*_t,G_t)}(Y_{i,b'}+R_{i,b'}) + \frac{Y^{-}_{i,b^*_t}}{\delta}\right)\\
& + \; p_i\cdot\delta\cdot\left(Y_{i,b_t} + \bar{R} + \delta\cdot\frac{Y^{-}_{i,b_t}}{\delta} + \sum_{b'\in\mathcal{C}(b,G_t)}\left(Y_{i,b'}+R_{i,b'}+\frac{Y^{-}_{i,b'}}{\delta}\right)\right)\\
& + \; \delta^2\cdot\mathbb{E}_t\left(Y_{i,b_{t+1}} + R_{i,b_{t+1}} + \delta\cdot\frac{Y^{-}_{i,b_{t+1}}}{\delta}\right)\\
& + \; \delta^3\cdot\mathbb{E}_t\sum_{\tau=0}^{\infty}\delta^{\tau}\left(Y_{i,b_{t+\tau+2}} + R_{i,b_{t+\tau+2}} + \delta\cdot\frac{Y^{-}_{i,b_{t+\tau+2}}}{\delta}\right)
\end{aligned}
$$

Since $\sum_{b'\in\mathcal{C}(b,G_t)\setminus\mathcal{C}(b^{CP}(H_t),G_t)}\left(Y_{i,b'}+R_{i,b'}+\frac{Y^{-}_{i,b'}}{\delta}\right) = 0$, we have $V_{i,t}\left(\sigma;H_t\right) \geq V_{i,t}\left(\sigma'';H_t\right)$, with the inequality strict unless $Y_{i,b^*_t} = R_{i,b^*_t} = 0$.

It then follows that there exists no strictly profitable one-shot deviation from the equilibrium strategy, and the claim follows. □