# Getting Blockchain Incentives Right*

Zahra Ebrahimi† Bryan Routledge‡ Ariel Zetlin-Jones§

December 2019

**Abstract**

Blockchain represents a distributed ledger or database technology that allows a group of self-interested users to maintain a ledger without trusted party such as a bank. In this paper, we develop a new, game-theoretic formulation of any blockchain where each user decides how to update the distributed ledger. Blockchains are useful only in so far as the updating strategies of users attain consensus—users agree on which version of the ledger is "correct"—and permanence—users do not have incentives to omit or modify past data. We show currently-implemented strategies—longest chain rules—do not achieve consensus or permanence when users are sufficiently heterogeneous. We go on to prove existence of new equilibrium strategies that attain both consensus and permanence for any degree of heterogeneity. In practice, these equilibrium strategies are robust to so-called 51% attacks. Our results shed light on the important role economic incentives play in determining the resilience of blockchain ledgers.

*We are grateful for the financial support of the PNC Center for Financial Services Innovation at Carnegie Mellon University and The Ripple Foundation.

†Tepper School of Business, Carnegie Mellon University; zebrahim@andrew.cmu.edu.

‡Tepper School of Business, Carnegie Mellon University; routledge@cmu.edu.

§Tepper School of Business, Carnegie Mellon University; azj@andrew.cmu.edu.

# 1 Introduction

In this paper, we propose a new model to analyze blockchain outcomes. By blockchain, we mean a distributed ledger—a record of data, possibly transactions as with bank ledgers—that is 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 party responsible for maintenance and security of the ledger. The paper's main contribution is to propose a new consensus protocol for blockchain-based distributed ledgers that is more robust—it generates consensus and permanence of the underlying data for a significantly wider range of situations than existing protocols.

One novel aspect of a blockchain ledger is that each "write" operation performed on the ledger is "chained" to a previous piece of information. In this sense, one may think of a blockchain as directed graph—a tree—where each node represents a set of information and each path from the origin node to each terminal (leaf) node contains a possible ledger. Since all self-interested users have both read and write access to the blockchain, each user may essentially propose their own version of the ledger. In other words, there may be many possible terminal leafs and, since each path from the origin to a terminal leaf represents a possible ledger, there may be many possible ledgers. For a blockchain ledger to function, then, a necessary condition is that users agree on which is "the" correct ledger—that is, users require consensus.

The fact that all users have write access also implies that blockchain ledgers must attain permanence. In other words, users must know that old data represented on the ledger cannot be changed. Users may change old data by proposing new paths through the blockchain that omit or modify information that resides in other possible ledgers that already exist in the blockchain. If one user can induce the consensus to switch from one path to another, then that user can effectively change the history of data on the consensus chain. For users to be able to trust the data on the ledger, they must know that users are not able to induce such switches in the consensus chain. In this sense, a second necessary condition for a blockchain ledger to function is permanence.

In Bitcoin, the most well understood 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. This ad hoc method of consensus has functioned remarkably well since Bitcoin's creation and yet under some conditions may not attain permanence. For example, Budish (2018) and Biais, Bisiere, Bouvard, and Casamatta

(2018) have both shown that if users' ability to write data to the blockchain is not evenly distributed or the value of modifying the data on the blockchain is sufficiently large, 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 develops a theoretical model to formally analyze blockchain consensus and permanence. In our model, in each period, rational—self-interested—users we call "miners" decide where to write a block of data to the blockchain. Given a locational choice, the likelihood a miner's bock is added to the blockchain depends on her (exogenous) mining power (a probability). If a miner's block of data is added, the block includes a mining reward for that miner—the reward is an increment to the miner's balance of a unit of account on the blockchain.

We first show that longest chain consensus features the same flaws in our model as found in earlier work: longest chain consensus fails to be an equilibrium when mining power is concentrated or when an individual miner has sufficient balances of unit of account on some path that is not the longest chain. These results resemble those found in Budish (2018) and Biais, Bisiere, Bouvard, and Casamatta (2018). We then show that the model admits other equilibria which are more robust—they remain equilibria even when mining power is concentrated or individual miner's have large balances of the unit of account on the blockchain.[1]

The equilibrium strategies we construct have two novel features relative to longest-chain consensus. First, we show that when miners have no incentive to remove data from the blockchain—essentially by mining blocks that omit data previously recorded on the blockchain—then there exists an equilibrium which relies on a form of mining-weighted approval voting. We term this strategy the approval weighted chain strategy. The approval weighted chain equilibrium strategy calls on miners to add blocks to the chain where the total mining power (probability) of miners who have previously mined blocks on the chain is largest. On the equilibrium path, the approval weighted chain consensus generates a graph with a single path of data and thus perfect consensus while disincentivizing deviations by miners with large mining power. In practice, such a strategy is a mild modification of the longest chain strategy.

Second, we show that when miners may have incentives to propose new paths through the blockchain—which admits the possibility of "double-spend" attacks where miners spend

[1]Other related papers the look at blockchain equlibrium include Saleh (2018), Chiu and Koeppl (2017).

their balance of the unit of account multiple times—then a modified version of heaviest-chain consensus which features "checkpoints" is an equilibrium. The 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. And, if no mining rewards vest before they are behind a checkpoint, then miners have no incentives to omit data ahead of the checkpoint. In this way, checkpointed-approval-weighted-chain consensus is a robust equilibrium in the model. Based on our checkpoint results, we show that our model has interesting implications for the types of transactions blockchain-based ledgers are likely to facilitate well.

## 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.** Time is discrete, $t = 0, 1, 2, \ldots$. There are $M$ miners each with a rate of time preference $\delta$. In each period, each miner $i$ proposes a location to add a *block*, $b_i$, of data. We index the block by the name of the miner since each miner may propose to encode different data onto the blockchain. A block consists of three components: hash data, transaction data, and mining rewards.[2] The hash data is determined technologically and is not relevant for our model beyond the fact that it implies a chained data structure. We treat the transaction data as exogenous but consider mining incentives for any possible transaction data (of fixed size). We represent the transaction data in any block $b_i$ as a vector, $\vec{Y}_{b_i} = (Y_{j,b_i})_{j=1,\ldots,M}$ which represents a vector of values for each miner $j$ in the block proposed by miner $i$, $b_i$. In addition, we let the vector $\vec{y}_{b_i}$ denote the mining rewards in block $b$ for miner $i$. We assume that mining rewards have the property that $y_{j,b_i} = \bar{y}$ if $i = j$ and $y_{j,b_i} = 0$ for $j \neq i$ implying that only miner $i$ earns a reward if block $b_i$ is added to the blockchain.

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

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

from $b_0$ to $b$. Let $\mathcal{B}_t(G_t)$ denote the set of all blocks in the graph $G_t$. And let $\mathcal{E}_t(G_t)$ denote 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 $\mathcal{G} = \bigcup_{t=0}^{\infty} \mathcal{G}_t$.

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 $a_{i,t} : G_t \to \mathcal{B}_t(G_t)$. 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.[3] Let $p_i$ denote the probability that miner $i$ successfully adds a block to the existing graph with $\sum_i p_i \leq 1$. 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 each miner $\vec{a}$, the graph in the subsequent period is $G_{t+1} = G_t \bigotimes (a_i, b_i)$ with probability $p_i$ for each $i$ where our notation represents $G_t \bigotimes (a_i, b_i) = G_t \bigcup (b_i, [a_i \to b_i])$. In words, the graph $G_{t+1}$ is the same as the graph $G_t$ but includes a new node, $b_i$ and a new edge from $a_i$ to $b_i$.

Let $H_t^G \in \mathcal{H}_t^= \bigcup_{\tau=0}^{t} \mathcal{G}_t$ denote the public history of the graph in each period. The private history for a miner is the public history combined with the miner's own history of locations where she tried to add blocks in the past. We denote this private history $H_t^i \in \mathcal{H}_t^i = \bigcup_{\tau=0}^{t} \mathcal{G}_t \times \mathcal{B}(G_t)$ and $\mathcal{H}^i = \bigcup_{t=0}^{\infty} \mathcal{H}_t^i$. A strategy for miner $i$ is mapping from the set of all possible miner $i$ histories into a set of pure actions,

$$\sigma_i : \mathcal{H}^i \to \mathcal{B}(G_t). \tag{1}$$

In any period $t$, given any graph $G_t$ and any set of strategies of miners $\vec{a} = (a_1, \ldots, a_M)$, we let $u^i(\vec{a}; G_t)$ denote the period payoff of miner $i$ and $U_t^i(\sigma; H_t^i)$ denote the discounted continuation payoff miner $i$ obtains from period $t$ onwards (where $\sigma = \{\sigma_i\}_{i=1}^m$) for a generic history $H_t^i$.

**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* and note that each chain may represent a different database than any other chain. For any graph, $G_t$ and block $\hat{b} \in \mathcal{B}(G_t)$, define the chain, $C(\hat{b}, G_t)$ as the unique path from $\hat{b}$ back to the

[3] We explore the implications of blockchain latency—that is, the possibility that multiple blocks may be added in a given period although miners may only observe one block addition per period—later.

Genesis block $b_0$ (and note that , $C(\hat{b}, G_t)$ is a subset of $\mathcal{B}(G_t)$ where we suppress the specific links that form the chain from $\hat{b}$ to $b_0$):

$$C(\hat{b}, G_t) = \left\{ \{\hat{b}, b_n, \ldots, b_1, b_0\} \in \mathcal{B}(G_t) \,\middle|\, (\hat{b}, b_n), (b_n, b_{n-1}), ..., (b_1, b_0) \in \mathcal{E}(G_t) \right\} \quad (2)$$

Recall the blockchain protocol imposes that every block has a unique predecessor. Hence, the path backwards from any block, $b$, to the Genesis block is unique. Define $\#C(\hat{b}, G_t)$ as the number of blocks in the chain or the *length* of the chain. And we say that block $b_n$ is on the blockchain $C(\hat{b}, G_t)$ if $b_n \in C(\hat{b}, G_t)$.

**Consensus and Payoffs.** To better understand payoffs, we now propose a specific functional form for the period payoff $u^i$ that represents key features of existing blockchain networks. We base our functional form on a blockchain whose sole purpose is to serve as money (as in Bitcoin). In a monetary setting, it is simplest to think of the data $Y$ and $y$ as representing units of account—*coins*—held on the graph $G_t$. These coin data could be positive or negative with the interpretation that positive data represent transfers received while negative data represent transfers sent. Alternatively, if the underlying data for miner $i$ in block $b$ is denoted $\hat{Y}_{i,b}$, we may assume $Y_{i,b} = v(\hat{Y}_{i,b})$ where $v(\cdot)$ represents each miner's preferences over underlying data. If the miner's preferences over data and units of account ($y$) are quasi-linear, then $Y_{i,b} + y_{i,b}$ represents miner $i$'s value in block $b$.

For any block $\hat{b} \in \mathcal{B}(G_t)$, a miner's balance of coins on any chain may then be represented as

$$\sum_{b \in C(\hat{b}, G_t)} (Y_{i,b} + y_{i,b}). \quad (3)$$

Whether these coins or balances are valuable—that is, whether miners have the option to sell them to others for real-valued goods—depends on whether the coins lie on a ledger in the graph (a path from the origin to a terminal node in $G_t$) that other users use. We link validity to miners' location actions. That is, if a miner choose a location in $\mathcal{B}(G_t)$, then that we say that miner uses the data along the path or *chain* from the origin to that existing block. If more miners use the same chain, then the coins on that chain are more valuable.

A particularly simply way to model the dependence of the value of coins on miners' actions is to assume that only coins that all miners agree are on the chain are valuable. In this case, we may define the set of *consensus blocks* as

$$\mathcal{C}(\vec{a}, G_t) = \bigcap_{i=1}^{m} C(a_i, G_t). \quad (4)$$

Then, a miner's balance aggregated across consensus blocks is simply

$$\sum_{b \in \mathcal{C}(\vec{a}, G_t)} (Y_{i,b} + y_{i,b}). \tag{5}$$

The set of consensus blocks represents the set of blocks that all miners agree (based on their location choices in the current period) are on the blockchain. Note that $\mathcal{C}(a, G_t)$ is not empty by construction since it includes $b_0$, the *genesis block*.

Given a structure to value coins in any graph, we may then define preferences. In particular, we will assume that

$$u^i(\vec{a}; G_t) = (1 - \delta) \sum_{b \in \mathcal{C}(\vec{a}, G_t)} (Y_{i,b} + y_{i,b}) \,. \tag{6}$$

As an example, consider one miner, "Satoshi." Suppose Satoshi has 1 unit of account on the genesis block $y_{\text{Satoshi},b_0} = 1$ and no units of account in any other block on the graph at any future date. Since, by construction, the genesis block is on the consensus chain for any graph and any period, Satoshi's balance on the set of consensus blocks is 1. Then, for every period, for any actions of miners and any graph, Satoshi's utility is simply $(1 - \delta)$. Aggregating Satoshi's utility over time along any infinite history then delivers $\sum_{t=0}^{\infty} \delta^t u^i(\vec{a}_t; G_t) = 1$.

Restricting value to only those coins that lie on consensus blocks is a strong notion of consensus. A relaxed—and smoother—construct that simplifies our subsequent analysis is to value blocks according to the computational mining power allocated to those blocks. We refer to such a valuation as *computational power weighted* preferences. We define preferences as

$$u^i(\vec{a}; G_t) = (1 - \delta) \sum_{b \in \mathcal{B}(G_t)} \frac{\sum_{\{j \neq i : b \in C(a_j, G_t)\}} p_j \, (Y_{i,b} + y_{i,b})}{\sum_{\{j \neq i\}} p_j}. \tag{7}$$

Under computational power weighted preferences represented by (7), miner $i$ receives value for any units of account held on blocks that are on the blockchain associated with some (other) miner's location choice $a_j$. Again, since Satoshi's unit of account is on every miner's blockchain, his or her utility is unchanged. Now, however, to the extent there is disagreement, miners still obtain value from their units of account as long as some other miners apply their mining power to these blocks. Of course, when there is full consensus in the sense that all miners choose the same location, then there exists a single blockchain and in this case only blocks on the blockchain receive their full value.

# 3 Equilibrium Consensus Protocols

In this section, we analyze equilibrium strategies in the blockchain model with no latency. We first develop conditions under which the longest chain strategy fails to be an equilibrium in the blockchain. We then show how the longest chain strategy may be modified to deliver both consensus and permanence of the blockchain for a wider range of parameters improving its security and resilience.

## 3.1 Longest Chain Equilibria

Consider first 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 her potential block. This is a simple coordination mechanism in that it depends only on the current graph $G_t$.

To ease notation, let

$$B^{LC}(H_t^G) = \underset{b \in G_t}{\operatorname{argmax}} \#C(b, G_t) \tag{8}$$

denote the set of blocks in the graph in (public) history $H_t^G$ such that the chain to $b$ has the largest number of blocks. In our model, we may represent the longest chain rule as the strategy satisfying

$$\sigma_i^{LC}(H_t^i) = B^{LC}(H_t^G) \tag{9}$$

along with a tie-breaking rule in case the graph features multiple longest chains so that $B^{LC}(H_t^G)$ is not a singleton. For simplicity, suppose the tie-breaking rule is that all miners choose each block in $B^{LC}(H_t^G)$ with equal probability.

We now argue that under a restriction that all transactions are positive, the longest chain rule is a Nash equilibrium. To prove this result, we show that no one-shot deviations are profitable along the equilibrium path where all other miners play the longest chain rule. Notice that along the equilibrium path associated with the longest chain rule, in each period, the graph takes the form of a single chain.

To see why along such a path no one-shot deviations are profitable, first note that it is sufficient to restrict attention to one-shot deviations that mine the next-to-last block on

the chain. For any other block, if the miner successfully adds her block and then reverts to the equilibrium strategy, then she immediately abandons her block mined away from the longest chain and forgoes the opportunity to have earned the rewards and transactions associated with mining that block to the longest chain.

Consider then the period $t$ tradeoff for miner $i$ of appending her block to the current longest chain—denote the block at the end of the existing chain $b^*$—as opposed to appending her block to the block that precedes $b^*$. In Appendix A, we show that the net benefit of choosing the longest chain over the preceding block satisfies

$$\frac{p_i\delta}{2}(Y_{i,b^*} + y_{i,b^*}) + \frac{p_i\delta}{2}(Y_{i,b_{i,t}} + \bar{y}). \tag{10}$$

The net benefit (10) of following the equilibrium over a one-shot deviation follows from thinking through the outcome should miner $i$ successfully appends her block away from the last block in the current chain. Such a successful deviation the resulting graph features a fork where in the subsequent period, there are two terminal blocks each with the same chain length. The block which was the unique longest chain block in period $t$ is $b^*$, and the additional longest chain block in period $t+1$ includes the block added by miner $i$, $b_{i,t}$. Following this graph, all miners mix between the two forks. Should all miners randomly select miner $i$'s fork, then miner $i$ forgoes the value she had on the previous longest block $(Y_{i,b^*} + y_{i,b^*})$. Moreover, should all miner's randomly select the original longest chain, then miner $i$ forgoes her transactions in her fork $(Y_{i,b_{i,t}} + \bar{y})$. The probability of these outcomes are $p_i/2$.

It follows immediately from (10) that as long as $Y_{i,b} \geq 0$ for all blocks $b$, then there exist no profitable one-shot deviations from the equilibrium path. We summarize this result in the following proposition.

*Proposition* 1 (Longest Chain Rule is a Nash Equilibrium)*: Suppose* $Y_{i,b} \geq 0$ *for all* $b$*. Then the longest chain rule is a Nash equilibrium.*

We now emphasize two limitations associated with the longest chain rule strategy. First, the proposed equilibrium strategy may fail to attain consensus following "small" deviations from the equilibrium path. Specifically, we develop conditions for longest chain to be a public perfect equilibrium and show that these conditions are easily violated. Second, limiting a blockchain to only feature mining rewards or positive transaction data is a severe restriction. In practice, miners (and other users) may wish to spend their mining rewards and or receive units of account for providing off-line goods and services. Proposition 1 also

immediately reveals that this equilibrium may fail to even be a Nash equilibrium when transactions may be negative. With negative transaction data, miners may have incentives to deviate from the Nash equilibrium to omit negative transactions from the blockchain and in this sense the longest chain strategy may fail to attain permanence. Considering both of these limitations in turn provides insights into what equilibrium strategies are likely to attain both consensus and permanence.

### 3.2 Approval Weighting and Perfect Consensus

We first proceed by obtaining conditions such that the longest chain rule is a Perfect Public equilibrium. These conditions shed light on whether the longest chain rule is likely to yield consensus in practice as the data generating process for blockchains are likely to induce forks. We therefore examine conditions under which no profitable one-shot deviations exist from any possible graph of data.

As with studying Nash equilibrium, the only additional relevant (one-shot) deviations are those from a graph which already features at least two equal length longest chains or those which feature at least at least one fork that is at most one block shorter than the longest chain. We argue that the former—incentives to abide the proposed equilibrium with two (or more) equal length chains—requires a tie-breaking rule which calls on miners to choose their most preferred fork. We then show that the latter—incentives to mine the longest chain instead of the end of a shorter fork—imposes a set of restrictions of transactions and mining power.

The necessary tie-breaking rule is intuitive. One miner's location decision does not influence her static payoffs from the graph. However, starting from a graph with multiple longest chains, if the miner has strictly higher transaction data on one of the longest chains, than by mining in that location she strictly increases the likelihood that her preferred chain becomes the consensus chain the in next period. Hence, the only tie-breaking rule that is immune to one-shot deviations is the rule that prescribes miners choose the block on their most preferred longest chain. We define the longest chain rule accounting for such a tie-breaking rule as

$$\sigma_i^{LC}(H_t^i) = \underset{b \in B^{LC}(H_t^G)}{\operatorname{argmax}} \sum_{b' \in C(b, G_t)} (Y_{i,b} + y_{i,b}). \tag{11}$$

Under the tie-breaking rule implied by (11) and the assumption that transaction data is positive ($Y_{i,b} \geq 0$), the only one-shot deviations that could be profitable for miners are those

that occur when the graph features a fork whose length is exactly 1 less than the length of the longest blockchain. For such graphs, there is a unique longest chain where $B^{LC}(H_t^G)$ is a singleton.

In any period $t$ for any such graph, it is then useful to define the weight of miners who would like to see a given block $b_t$ added to a block at the end of a fork shorter than the longest chain, $b$. These are miners for whom if there were in period $t+1$ a tie between the longest chain in period $t$ and this other chain, they would prefer the new chain. We denote this weight $W(b_t, b, G_t)$ and it satisfies

$$W(b_t, b; G_t) = \sum_j p_j \mathbb{1}\left\{ \sum_{b' \in C(b, G_t)} (Y_{j,b'} + y_{j,b'}) + Y_{j,b_t} + y_{j,b_t} > \sum_{b' \in C(B^{LC}(H_t^G), G_t)} (Y_{j,b'} + y_{j,b'}) \right\}. \tag{12}$$

*Proposition* 2 (Longest Chain Rule is a Perfect Public Equilibrium)*: Suppose for all blocks* $Y_{i,b} \geq 0$*. The longest chain rule is a perfect public equilibrium if for every graph* $G_t$ *and for every block* $b \in \{b' : \#C(b', G_t) = \#C(B^{LC}(H_t^G), G_t) - 1\}$*,* $\forall i$*:*

$$\begin{aligned} & Y_{i,b_{i,t}} + \bar{y} \\ & \geq \left( (1-\delta) \frac{W(b_{i,t}, b; G_t) - p_i}{1 - p_i} + \delta W(b_{i,t}, b; G_t) \right) \left( (Y_{i,b_{i,t}} + \bar{y}) + \sum_{b' \in C(b, G_t)} (Y_{i,b'} + y_{i,b'}) - \sum_{b' \in C(B^{LC}(H_t^G), G_t)} (Y_{i,b'} + y_{i,b'}) \right). \end{aligned} \tag{13}$$

The set of conditions represented by (13) ensures that miners prefer to attempt to add their block to the longest chain for any graph. One interpretation of these conditions is to consider a thought experiment where miner $i$ may add one block $b_{i,t}$ to the graph for sure in period $t$. If she adds her block to the current longest chain, that chain remains the consensus chain the subsequent period and miner $i$ earns $Y_{i,b_{i,t}} + \bar{y}$ (in perpetuity) with probability 1. If instead she adds her block to a fork of shorter length, then she earns rewards only if her fork becomes the longest chain. The first term represents the chance her fork becomes the longest chain (after inducing a tie in period $t+1$). When this event happens, she earns her the transaction value associated with her block, $b_{i,t}$, the value of transactions in any block on the location she chose in period $t$ ($b$), and she forgoes any value on the longest chain from period $t$ (transactions on the chain $C(B^{LC}(H_t^G), G_t)$.

To visualize the constraints that arise from (13), consider the graph displayed in Figure 1. The figure shows an example of the graph in period $t$. 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 $\arg\max \#C(b, G_t) = b_{l_2}$, 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. Consider the net benefit to miner 1 of deviating from longest chain and choosing block $b_{f_1}$. Suppose for this example there are no transaction data so $Y_{i,b} = 0$ and the chain only features mining rewards, $y_{i,b}$.

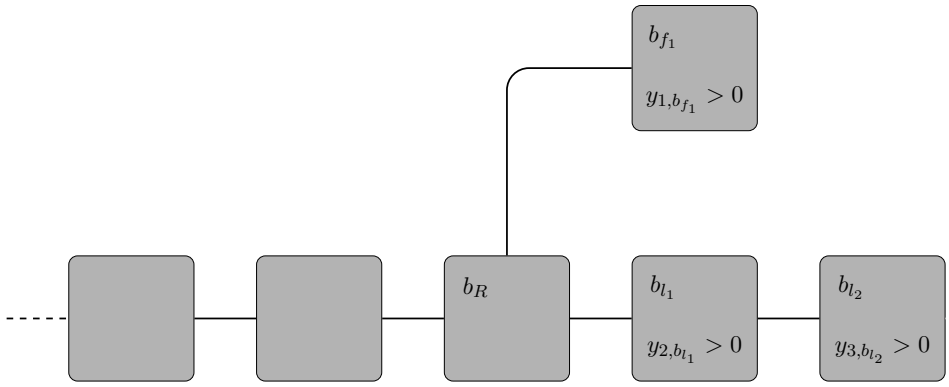


Figure 1: An illustration of incentives to deviate from the longest chain strategy.

Given the mining rewards illustrated in Figure 1, the weight of miners who would like to see fork $b_{f_1}$ extended is simply $p_1$. Hence, condition (13) requires $\bar{y} \geq \delta p_1 2\bar{y}$ or $p_1 \leq 1/(2\delta)$. In other words, should forks appear, if miner 1 has too much weight (say if $\delta \approx 1$ and $p_1 > 0.5$) than she can 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 2 likely imposes stringent limits on the distribution of mining power and these limits are likely to be violated (or provide miners with incentive to acquire mining power such that they are violated). To see these potential constraints, consider (13) as $\delta \to 1$. In this case, (13) simplifies to

$$Y_{i,b_{i,t}} + \bar{y} \geq \frac{W(b_{i,t}, b; G_t)}{1 - W(b_{i,t}, b; G_t)} \left( \sum_{b' \in C(b, G_t)} (Y_{i,b'} + y_{i,b'}) - \sum_{b' \in C(B^{LC}(H_t^G), G_t)} (Y_{i,b'} + y_{i,b'}) \right). \tag{14}$$

If, for example, $W(b_{i,t}, b; G_t) = p_i$ so that miner $i$ is the only miner who would like to see block $b_{i,t}$ added to the fork ending at block $b$, then (14) imposes an upper bound on $p_i$.

Of course, this upper bound may not suffice should other miners have value on the chain ending at block $b$ suggesting that (13) is likely to bind in general.

Proposition 2 reveals that even when a blockchain only features mining rewards, the longest chain rule may not robust as an equilibrium (in the perfect public sense) to general distributions of mining power. In practice, use of Bitcoin reveals that mining power is concentrated (see `https://www.blockchain.com/en/pools`). As a result, it is necessary to develop equilibrium strategies that are robust to concentrated mining.

We now develop an equilibrium strategy we call the *approval weighted chain* rule that yields the same outcomes as the longest chain rule along the equilibrium path, but provides better incentives to miners with high degrees of mining power. In other words, we show that the approval weighted chain strategy remains a perfect public equilibrium even when mining power is concentrated.

The idea behind the approval weighted chain rule is 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 strategy 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 single chain remains.

We proceed by developing a set operator that refines the set of terminal to only those with the highest approval weight recursively. Let $\mathcal{T}(G_t) \subset \mathcal{B}(G_t)$ denote the set of terminal blocks of the graph $G_t$—these are blocks with no edges leading away from the genesis block.

To build the operator, we consider first an arbitrary set of terminal blocks, $S$. The consensus blocks in the set $S$ are those which lie on every chain associated with every block in $S$: $\mathcal{C}(S, G_t) = \bigcap_{b \in S} C(b, G_t)$. We define the common *root* of each of these chains $b_R(S, G_t) = \{b \in \mathcal{B}(G_t) | C(b, G_t) = \mathcal{C}(S, G_t)\}$. Note that by construction, for any set of blocks $S$, the root $b_R(S, G_t)$ is both nonempty and a singleton block. It is of course straight-forward to

compute each miner's balance along the common root chain. Let

$$Y_{i,R}(S, G_t) = \sum_{b \in C(b_R(S,G_t),G_t)} (Y_{i,b} + y_{i,b}) \,. \tag{15}$$

Next, we construct the approval weight of each path leading away from the common root block (net of the balances miners hold on the root chain)—the idiosyncratic part of each chain—as, for each $b \in S$,

$$\mathcal{P}(b, S, G_t) = \sum_{i=1}^{M} p_i \mathbb{1} \left\{ \sum_{b' \in C(b,G_t)} \left(Y_{i,b'} + y_{i,b'}\right) > Y_{i,R}(S, G_t) \right\}. \tag{16}$$

One may view the approval weight as a score for each unique branch of the blockchain leading to a terminal node. This score adds up the mining power of those miners who attain a positive balance on the idiosyncratic component of each branch.

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

$$T(S) = \{b \in S | P(b, S, G_t) \geq \max_{b' \in S} P(b', S, G_t)\}. \tag{17}$$

Given the set operator, $T$, the approval weighted chain strategy satisfies

$$\sigma_i^{AW}(\mathcal{H}_t^i) = \lim_{k \to \infty} T^k(\mathcal{T}(G_t)). \tag{18}$$

We now argue that the approval weighted chain strategy is an equilibrium for any distribution of mining power when the blocks only contain mining rewards, or $Y_{i,b} = 0$ for all blocks $b$. The approval weighted chain strategy has two important properties that disincentivize miners from deviating from the candidate 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 induce a change in the equilibrium behavior of all other miners which ensures the approval weighted chain strategy is an equilibrium. We summarize this discussion in the following proposition.

*Proposition* 3 (Approval Weighted Chain Equilibrium)*: Suppose in every block* $Y_{i,b} = 0$ *and* $y_{i,b}$ *are positive. Then, the approval weighted chain strategy is an equilibrium.*

To illustrate why the approval weighted chain strategy is an equilibrium, consider Figure 2. The blocks are labeled according to whether they are on the longest chain ($b_{l_1}$—$b_{l_4}$), the initial fork ($b_{f_1}$—$b_{f_2}$) or a secondary fork ($b_{k_1}$). Suppose the transactions listed are the only total transactions on each block (they include any relevant mining rewards and only those listed are positive). For example, then, only miner 2 has a positive coin balance on block $b_{l_1}$. The approval weights associated with the chains to the terminal blocks are given by

$$P(b_{k_1}, \mathcal{T}(G_t), G_t) = p_1 + p_2, \quad P(b_{l_4}, \mathcal{T}(G_t), G_t) = p_2 + p_3, \quad P(b_{f_2}, \mathcal{T}(G_t), G_t) = p_1. \tag{19}$$

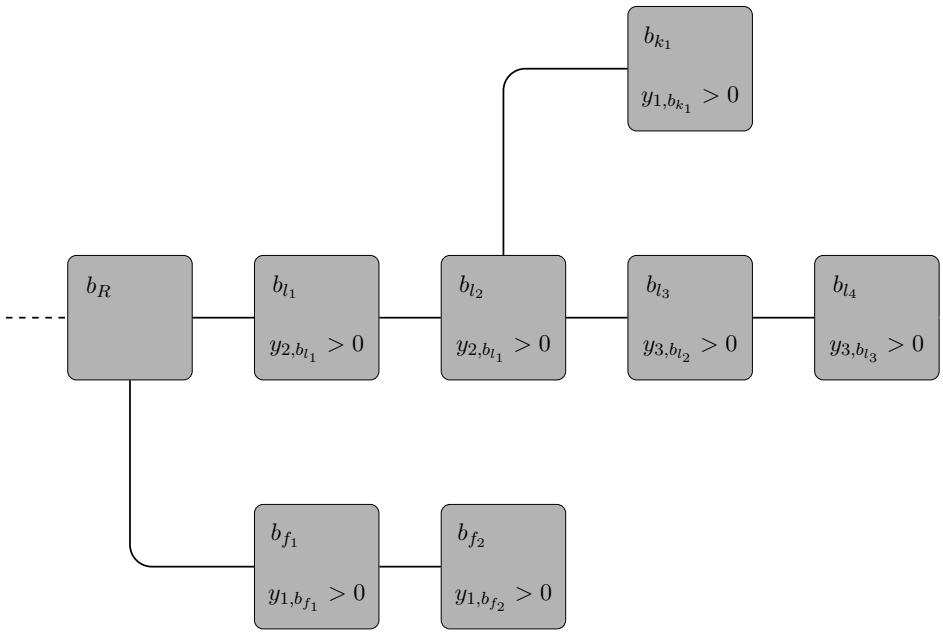


Figure 2: An illustration of approval weighted chain strategies.

Suppose now that $p_1 > p_2 > p_3$. Then, the approval weighted chain strategy calls for all miners to mine $b_{k_1}$—and note this is *not* the longest chain. Consider next the interesting case in which $Y_{1,b_{f_1}} + Y_{1,b_{f_2}} > Y_{1,b_{k_1}}$ so that miner 1 has larger balances on the chain to $b_{f_2}$ than on the chain to $b_{k_1}$. Here, miner 1 might have an incentive to choose location $b_{f_2}$ in the hopes of enlarging her coin balances on the consensus chain. But doing so would not cause the approval weight of any chain through $b_{f_2}$ to change and therefore miner 1 has no ability to induce a switch in behavior by miners 2 and 3.

Notice that in equilibrium, since miners have no incentives to deviate from the proposed strategy, there would no forks. 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 strategy and the longest chain strategy appear only off the equilibrium path and these differences are important in sustaining equilibrium behavior.

We argue that including these transactions by allowing $Y_{i,b} > 0$ is problematic for both longest chain and approval weighted strategies. To see why even restricting attention to $Y_{i,b} > 0$ may be problematic, consider again the example in Figure 2. Suppose miner 3 has a block with transactions $Y_b$ that satisfy $Y_{1,b} > 0$. The approval weighted strategy calls for her to append her block to block $b_{k_1}$. However, by appending her block to block $b_{l_4}$, she may be able to induce a switch in the behavior of miner 1. The reason is that if she successfully adds her block to $b_{l_4}$, then the resulting approval weight of the chain through block $b_{l_4}$ would be $p_1 + p_2 + p_3$. In a sense, miner 3 is "bribing" miner 1 to join her block which is valuable for miner 3 because it allows her to capture the utility associated with her mining rewards in blocks $b_{l_3}$ and $b_{l_4}$. For this reason, we pursue a new type of equilibrium strategy which is robust to more general transaction data.

### 3.3 Checkpoint Strategies

We now consider equilibrium strategies that are robust to positive and negative transaction data. We show that a modification of the approval weighted chain strategy that introduces a form of history dependence is an equilibrium when the blockchain features transaction data. We also modify our preferences to capture the idea that miners receive flow utility associated with spend transactions but that those who deliver the real goods and services associated with this flow only deliver this value once they are confident the spend transaction cannot be removed. The resulting settlement lag, which is a feature of existing blockchains plays a key role in disciplining double spend behavior.

**Double Spending.** To motivate this analysis, first note that one may interpret our previous restriction that the blockchain only represents positive transaction data is that the blockchain data feature only "earnings" of units of account but no "spending." For many blockchains, of course, it is natural to think that miners accumulate balances of the unit of account and then spend them—elements of the transaction data satisfy $Y_{i,b} < 0$—for real-valued goods and services. Spending, the transfer of the unit of account between miners and/or others, can alter incentives. A miner who has a "spend" transaction in block $b$, might have a preference for that block *not* to be part of the future consensus chain which, in principle, would allow the miner to spend that unit of account once again.

The incentive to change or un-do past spending is called "double spending" (a term coined in Brands (1993)) and is a central concern of the Bitcoin protocol of Nakamoto (2008). Double spending, in old-school banking, is very similar to bouncing a check. Satoshi buys groceries and pays with a check. Usually, the grocery store "cashes" the check via a deposit and the bank transfers money from Satoshi's account to the grocery store. However, Satoshi can bounce the check by withdrawing his funds after buying the groceries but before the check is cashed. He then has groceries and the cash he can use to spend again.

On a blockchain, the analogous fraud can happen. See Figure 3. Think of a miner, Satoshi ($m$), who owns Bitcoins in two accounts with public keys $k_m$ and $k'_m$. Satoshi transfers his Bitcoin from account $k_m$ to a grocery store in exchange for groceries (and in particular, value not represented on the blockchain). The transaction is broadcast and eventually ends up in a block. For example, suppose the transaction is part of block $b_{t+1}$. Notice block $b_{t+1}$ is chained to $b_t$ and has subsequent block $b_{t+2}$. Satoshi need not be the creator of any of these blocks. Now, if Satoshi's transaction is large, she has an incentive to focus her mining effort on attaching to block $b_t$ (rather than the longer chain defined by $b_{t+2}$). If Satoshi mines a new version of $b_{t+1}$, call it $b'_{t+1}$ and other blocks follow (by the same miner or by others), then block $b_{t+1}$ is no longer on the consensus chain. In that case, transactions listed in block $b_{t+1}$ "didn't happen."

Under Bitcoin's current protocol, transactions in abandoned blocks simply re-enter the pool of transactions and eventually are included in a future block. All this would be innocuous if the original transaction simply ends up in a later block. However, in $b'_{t+1}$ miner $m$ includes a transaction moving all Bitcoin from $k_m$ to $k'_m$. The transaction involving the grocery store and $k_m$ is no longer valid—the account has "Non-Sufficient Funds." Satoshi has the groceries (real goods) and the Bitcoin in $k'_m$. The anonymous structure of Bitcoin can mask the fact that Satoshi also owns $k_m$ and $k'_m$. The reward for this "double spend" attack is higher the higher is the real value of Bitcoin (more real goods received).[4] Budish (2018) uses this observation to conclude that there is an upper bound on the value of Bitcoin and concludes, in part that Bitcoin is useful only for small-value transactions.

Our model admits double spends if we consider negative transactions on the blockchain. Consider, for example, Figure 3. The top chain, blocks $b_R, b_{l_1}, b_{l_2}$, is the status quo and features a negative transaction for miner $i$ in block $b_{l_1}$. Given a new block of transactions $b_{f_1}$, the longest chain strategy calls for miner $i$ to choose location $b_{l_2}$—the longest (or only) chain. However, miner $i$ now has an incentive to choose block $b_R$, a block before the negative

[4]The incentive to double spend in this example also depends on the other transactions and mining rewards involving $m$ on blocks $b_{t+1}$ and $b_{t+2}$.

transaction appears in the blockchain. Should she mine successfully at this location, she creates a fork. If she is able to mine enough consecutive blocks before the other miners add to $b_{l_2}$ then upon completion of block $b_{f_3}$, she has created a new longest chain. At this point, longest chain calls for all miners to mine block $b_{f_3}$ in which case it becomes feasible for miner $i$ to enter a negative transaction once again $Y_{i,b_{f_4}} < 0$. This new negative transaction is her double spend.

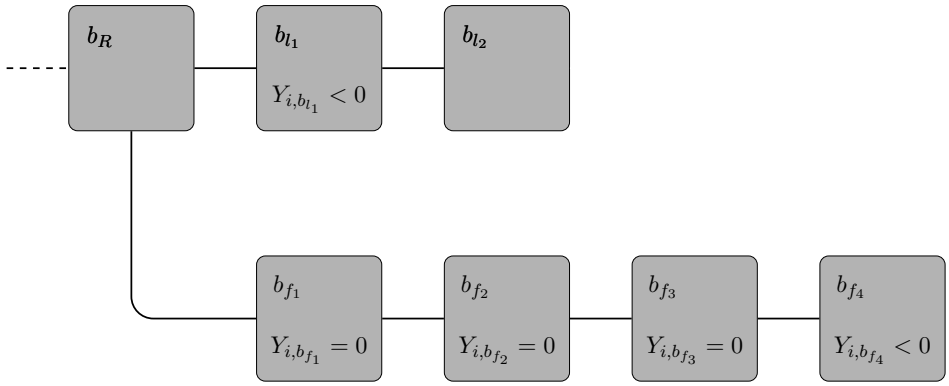


Figure 3: An illustration of a double spend under longest chain strategies.

There are two practical concerns with double spending that limit the usefulness of incentives provided by the longest chain strategy. First, the value of a double spend, $Y$ can be large in absolute value. There is no practical bound on the value of a Bitcoin transaction. And it is possible $Y$ could represent the benefit of many transactions across several blocks. More importantly, and as is central to Budish (2018), the economic value of transactions is endogenous. As the value of Bitcoin rises, the incentive to attempt a double spend increases. Second, mining power is not equal and can be large, $p_m$, so we cannot rely on $p_i$—and, therefore, the likelihood that miner $i$ is able to mine consecutive blocks before others—being small. The commonly described "51% attack" occurs when a miner with $p_m > 0.5$ can generate blocks faster (i.e., will generate a longer chain eventually) then all the other miners.

When we allow for double spend opportunities—when transaction data may be negative—the approval weighted chain strategy mail fail to be an equilibrium in a manner distinct from when the transaction data is positive. Consider the same example from Figure 2. Suppose we introduce a fourth miner, $i = 4$. If miner 4 has no transaction balances on any chain as in the graph in Figure 2, then miner 4 is indifferent between following the recommended location from the approval weighted chain strategy and any other location implying that the approval weighted chain strategy is an equilibrium. Suppose instead that

miner 4 has a negative transaction balance in block $b_{l_1}$. This new version of the graph is displayed in Figure 4.

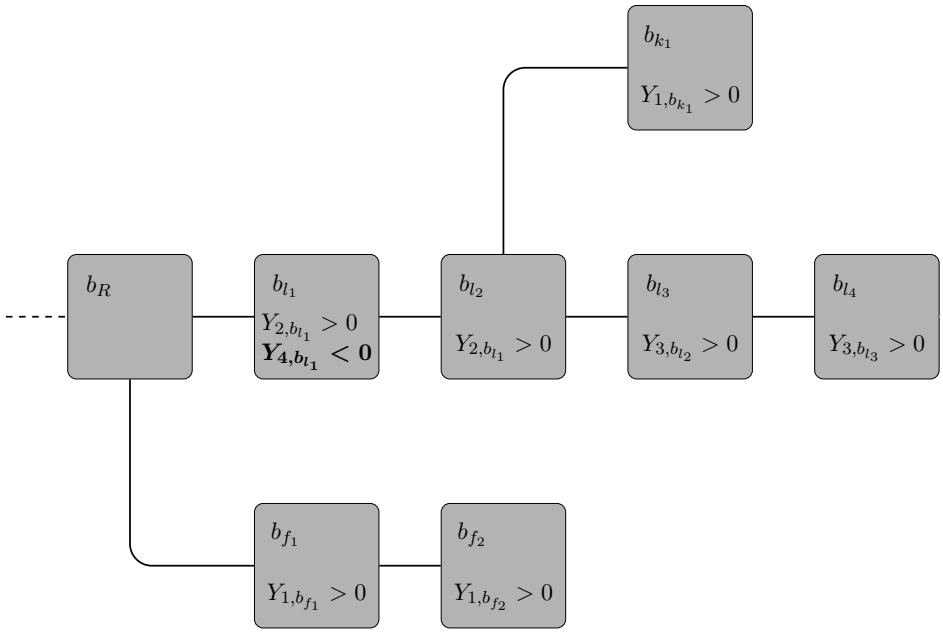


Figure 4: An illustration of a double spend under approval weighted chain strategies.

Suppose that $p_1 > p_4 > p_2 > p_3$. Given our definition of approval weights, no approval weights change with the introduction of a negative transaction for miner 4 in block $b_{l_1}$. As before, the approval weighted chain strategy calls for all miners to choose location $b_{k_1}$.

However, if miner 4 should add any block to the graph, because she will earn the mining reward $y_{4,b}$, her weight would be then added in the subsequent period. It is possible that miner 4's decision is pivotal in the sense that she is able to change the rankings of the approval weights of each possible chain. In this case, miner 4 has an incentive to choose location $b_{f_2}$. If she successfully adds her block, call it $b_{f_3}$ at this location then the blockchain leading from $b_R$ to $b_{f_3}$ would then have the highest approval weight, equal to $p_1 + p_4$. Miner 4 benefits from such a strategy because she induces a switch in consensus from a chain with a negative transaction $Y_{4,b_{l_1}}$ to a chain with a positive transaction $y_{4,b_{f_3}}$.

**Checkpoints.** Our proposed resolution to miner's incentives to omit data from the chain is to introduce *checkpoints*, a form of history dependence, into the equilibrium strategies. The basic idea is that for every graph, agents determine a reference block, or checkpoint

and restrict approval weight to locations on the subgraph that follows the checkpoint. To see why such checkpoints may be useful in disincentivizing double-spend behavior, consider again the graph in Figure 4. Suppose for this graph, the checkpoint is block $b_{l_2}$. Using this checkpoint, the only terminal nodes with positive approval weight are $b_{k_1}$ and $b_{l_4}$. While miner 4 would still like to induce all other miners to switch to a fork following $b_{f_2}$, because this fork is not on the subgraph following the checkpoint, she recognizes that her behavior cannot induce other miners to switch to any chain following $b_{f_1}$ or $b_{f_2}$. As a result, she cannot benefit from deviating and choosing location $b_{f_2}$ as under standard approval weighting. In this manner, checkpoints can eliminate incentives of miners to choose their locations as a way of proposing new blockchain paths that omit certain transactional data which they do not like.

The above description of double spends in our setting omits a critical aspect of double spends in existing blockchains, namely, the exchange of real valued goods or services—offline trade—associated with spend transactions. We introduce this idea now by modifying preferences in our model. To do so, we assume that traders receive a flow payoff associated with a spend transaction of 1 unit of account on the blockchain equal to the present discounted value of forgoing 1 unit of account on the blockchain in perpetuity. The idea is that a spend transaction (in a blockchain with consensus) is permanently on the record and therefore impacts the miner's utility in perpetuity. However, the benefit of a spend transaction is a (nearly) immediate gain in resources. If priced fairly, the gain from spending should roughly equal the cost. Critically, we also assume that miners receive the flow utility associated with spend transactions only when the spend transaction is immutable. In other words, the receivers of the spend transaction would only remit real goods and services once they are (sufficiently) confident that the transactions cannot be removed. If the checkpoint strategy we construct is an equilibrium, then as soon as a block with a spend transaction becomes a checkpoint, all users or miners will know that the spend transaction will not be omitted from future blockchains.

**Checkpoint Preferences and Strategies.** We now define a strategies and preferences with checkpoints. We proceed in two steps. First, we define checkpoints: for any public history $\mathcal{H}_t^G$, the checkpoint, $B^{CP}(H_t^G)$ selects a specific block on the current graph, $G_t$. Strategies are unchanged with the exception that checkpoints are specified as part of the equilibrium. 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 because it 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 strategies that admit the possibility of forks.

*Assumption* 1*: For any checkpoint rule, for all public histories,* $H_t^G$*,* $B^{CP}(H_t^G) \notin \mathcal{T}(G_t)$*.*

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.

Next, we re-define preferences to accommodate surplus flow utility associated with spend transactions.

Let $\lambda(b, Y_{i,b}, H_t^G)$ denote an indicator function which takes the value of 1 if $Y_{i,b} < 0$, $b \in C(B^{CP}(H_t^G), G_t)$, $b \notin C(B^{CP}(H_{t-1}^G), G_{t-1}$. In words, the indicator $\lambda(b, Y_{i,b}, H_t^G)$ is 1 when a block features a negative transaction and period $t$ is the first period that block $b$ is on the chain to the checkpoint block. Recall that the period when $\lambda(b, Y_{i,b}, H_t^G) = 1$ is the first period when miners know for sure that spend transactions will not be omitted from blockchain in any future period. Given $\lambda$, itself a function of the checkpoint strategy $B^{CP}(H_t^G)$, preferences satisfy

$$u^i(\vec{a}; H_t^G) = \sum_{b \in \mathcal{B}(G_t)} \frac{\sum_{\{j \neq i : b \in C(a_j, G_t)\}} p_j \left[ (1-\delta)\left(Y_{i,b} + y_{i,b}\right) - \frac{1}{\delta} Y_{i,b} \lambda(b, Y_{i,b}, H_t^G) \right]}{\sum_{\{j \neq i\}} p_j}. \quad (20)$$

Note that in this formulation, preferences are function of the current actions of miners and the public history of the graph (as opposed to just the current state of the graph). To illustrate the changes we have made to preferences, consider again the case where Satoshi has one unit of account on the genesis block and in every period there is a single chain in the graph. If no other block contains a transaction for Satoshi, then her lifetime utility will continue to be 1.

Consider instead how Satoshi's payoffs change if there is a transaction equal to -1 in the second block. And, suppose for any such graph with a single chain, the checkpoint rule satisfies $B^{CP}(H_t^G) = \mathcal{T}(G_{t-1})$. That is, the checkpoint is the block before the terminal block. In period 2, Satoshi's utility over the graph is 0 because her balance aggregated over blocks is zero. Her utility over the graph in all future periods will continue to be zero.

However, in period 3, the second block in the chain becomes the checkpoint block under the rule above. As a result, her spend transaction in the second block vests and Satoshi earns the flow equal to $1/\delta$. Aggregating and discounting these payoffs from the perspective of period 0, her lifetime utility is 1. Of course, once Satoshi receives the flow utility in period 3, she would benefit from the construction of an alternative path through the blockchain where her total balance on the the chain is 1 instead of 0.

**An Equilibrium Checkpoint Strategy.** To specify an equilibrium checkpoint strategy, it is helpful to introduce two pieces of notation. First, let $J : \mathcal{B}(\mathcal{G}) \times \mathcal{G} \to \mathcal{G}$ represent the subgraph associated with some root block $b' \in \mathcal{B}(G_t)$. Then, given block, $b'$,

$$J(b', G_t) = \{b \in \mathcal{B}(G_t) | \#C(b, G_t) \geq \#C(b', G_t) \text{ and } b' \in C(b, G_t)\}. \tag{21}$$

Second, let $M : \mathcal{B}(G_t) \times G_t \to \mathcal{B}(G_t)$ denote the parent of a block. That is,

$$M(b, G_t) = \{b' : b' \in C(b, G_t) \text{ and } \#C(b', G_t) = \#C(b, G_t) - 1\}. \tag{22}$$

Next, we adjust the scoring function (16) to account for off-path spending as follows. For any $b \in S$,

$$\mathcal{P}(b, S, G_t) = \sum_{i=1}^{M} p_i \mathbb{1}\left\{ \sum_{b' \in C(b, G_t)} \left(Y_{i,b'} + y_{i,b'}\right) - \frac{Y_{i,b'}}{\delta} \mathbb{1}\left\{Y_{i,b'} < 0\right\} \left(1 - \Pi_{\tau=0}^{t} \lambda(b', Y_{i,b'}, H_\tau^G)\right) > Y_{i,R}(S, G_t) \right\}. \tag{23}$$

The additional term in the scoring function adds a miner's weight if they have a negative spend transaction that has not yet paid out its surplus flow. We now define the checkpoint blocks according to

$$B^{CP}(H_t^G) = M\left[\lim_{k \to \infty} T^k \left(\mathcal{T}\left[J(B^{CP}(H_{t-1}^G), G_t)\right]\right), G_t\right] \tag{24}$$

In any period, to find the checkpoint block, we begin by looking for the terminal blocks on the subgraph that has the checkpoint block from last period as the common root $(\mathcal{T}\left[J(B^{CP}(H_{t-1}^G), G_t)\right])$. We choose the new checkpoint to be the parent block of the terminal block among these with the highest approval weight.

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 (again, assuming that $p_1 > p_4 > p_2 > p_3$.

| 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 | $b_R$ | $b_R$ | $b_R$ | $b_{l_2}$ | $b_{l_2}$ | $b_{l_2}$ | $b_{l_2}$ |

Table 1: A Sequence of Blocks and Checkpoints

Given the checkpoint rule, the checkpoint strategy satisfies

$$\sigma_i^{CP}(H_t^i) = \lim_{k\to\infty} T^k(\mathcal{T}(J(B^{CP}(H_t^G), G_t)). \tag{25}$$

Since the checkpoint strategy has the same features as the approval weight strategy (but limited to the subgraph following the checkpoint), it necessarily provides the same incentives for consensus as the more general approval weight strategy. The key feature of this new equilibrium is that no miner has an incentive to deviate when they have spend transactions that are "ahead of" the checkpoint. Blocks "ahead of" the checkpoint are blocks whose parent (or whose parent's parent, etc) is the checkpoint block.

Notice, in such cases, if the spend is on the chain with the highest approval weight, miners expect to receive a net zero flow associated with their spend and their surplus utility if the spend transaction ends up on or behind the checkpoint in the next period. Should they deviate in a manner that would omit their spend transaction from the chain with the highest approval weight, then they earn zero utility (in expectation) associated with their spend transaction and their surplus utility. Consequently, their only concern is the mining reward which they expect to earn on any node that has or will have the highest approval weight. Of course, once the spend transaction vests, miners would like to deviate and mine blocks behind the checkpoint which omit their spend transaction, but they recognize that all such blocks will be ignored by all other miners.

There remains one potential problem with the checkpoint strategy. When a graph in a given period features a fork on the checkpoint block, given arbitrary, positive transaction data, miners may have incentives to deviate from the checkpoint strategy if they are able to "bribe" large miners to follow them. A natural conjecture is that recognizing this limitation, miners may choose to not submit or accept transaction data when checkpoints feature forks as they understand that this can incentivize deviations by other miners. If the transactions following a fork only feature mining rewards, then such incentives to deviate are not present and will induce consensus among miners. This consensus will cause an update to the graph and an update to the checkpoint. When the checkpoint updates, the new checkpoint will

not feature a fork and will tolerate positive and negative transaction data without providing incentives to deviate. We therefore argue that the checkpoint strategy is an equilibrium on a subset of all possible graphs that have the property that in any period in which a checkpoint has multiple child-blocks, the transaction data are all zero.

We restrict transactions in the following manner.

*Assumption* 2*: For any public history $H_t^G$, if the cardinality of the terminal blocks in the subgraph with the root block equal to the checkpoint is (weakly) larger than 2, then each miner receives transactions $\vec{Y}_{b_i} = 0$. Otherwise, transactions are unrestricted.*

*Proposition* 4 (Checkpoint Equilibrium)*: Under assumption 2, the checkpoint strategy is an equilibrium.*

# 4 Conclusion

Allowing everyone write-access to a database—particularly a database of financial transactions—sounds unworkable. A novel aspect of blockchain technology is to facilitate this decentralized database system by bundling transactions in blocks and then linking the blocks to create a tree. The technological requirement that transactions in a new block can only reference transactions in blocks along the chain from the new block back to the genesis block creates a coherent database of an ordered list of transactions. In this paper, we have focused on two incentive components of this system: consensus and permanence. Consensus is the equilibrium property that all miners choose the same block (hence chain) as a predecessor for their new block of transactions. Permanence is the equilibrium property that miners do not choose "old" blocks as predecessors that would create an alternate chain to eliminate blocks previously on the consensus chain. Both these properties are necessary if a blockchain technology is to be viable.

The perfect public equilibrium we propose is similar to the longest chain mechanism currently implemented in many blockchain instances, like Bitcoin. First, we adapted the coordination consensus mechanism to allow for non-equal mining power. Second, we add a "checkpoint" that, in equilibrium, limits the set of potential predecessor blocks to recent added blocks. In a financial setting such as Bitcoin, the checkpoint dovetails with a settlement lag. Anyone accepting Bitcoin will sensibly wait until the transaction is in a block and behind the checkpoint before delivering the off-blockchain physical good. Effectively, the norm (not a formal rule) with Bitcoin is that a seller receiving Bitcoin wait at least six

blocks (about one hour) before delivering the non-blockchain goods. We have shown how this norm ought to be linked explicitly to the consensus protocol.

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 equations (8) and (9), 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 (25) would require monitoring the blockchain for several periods before knowing the consensus block to choose as a predecessor. 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 ($11\ seconds/600\ seconds \approx 1.8\%$, see Decker and Wattenhofer (2013)). 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. For example, the six block norm currently used on Bitcoin is well outside uncertainty created by network latency. If however, the checkpoint block is too close to the current block 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


Biais, B., C. Bisiere, M. Bouvard, and C. Casamatta (2018): "The blockchain folk theorem," Toulouse University Working Paper.

Brands, S. (1993): "Untraceable off-line cash in wallet with observers," in *Annual international cryptology conference*, pp. 302–318. Springer, https://bitcoin.stackexchange.com/questions/87567/who-first-defined-coined-the-double-spending-problem.

Budish, E. (2018): "The Economic Limits of Bitcoin and the Blockchain," National Bureau of Economic Research Working Paper.

Chiu, J., and T. V. Koeppl (2017): "The economics of cryptocurrencies–bitcoin and beyond," Queens University Working Paper.

Decker, C., and R. Wattenhofer (2013): "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*, pp. 1–10. IEEE.

Nakamoto, S. (2008): "Bitcoin: A peer-to-peer electronic cash system," Discussion paper, Bitcoin Project, White paper `http://bitcoin.org/bitcoin.pdf`.

Saleh, F. (2018): "Blockchain without waste: Proof-of-stake," McGill Working Paper.

# A Proofs of Propositions

## A.1 Proof of Proposition 1: Longest Chain Rule is a Nash Equilibrium

Consider the longest chain strategy,

$$\sigma_i^{LC}(H_t^i) = \arg\max_{b \in G_t} \#C(b, G_t)$$

with the augmented rule that if $L = \arg\max_{b \in G_t} \#C(b, G_t)$ is not a singleton, $\sigma_i^{LC}(H_t^i) = 1/\#L$.

Consider any history $H_t^i$ induced by the equilibrium profile $\sigma_i^{LC}$. The graph in period $t$ is a single chain and all graphs following this history are a single chain. As a result, discounted utility from period $t$ on (along the equilibrium path) satisfies

$$\mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^\tau u^i(\sigma^{LC}, G_{t+\tau}) \tag{26}$$

$$= \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^\tau (1-\delta) \left[ \sum_{b \in \mathcal{B}(G_{t+\tau})} (Y_{i,b} + y_{i,b}) \right] \tag{27}$$

$$= \mathbb{E}_t \sum_{\tau=0}^{\infty} \delta^\tau (1-\delta) \left[ \sum_{b \in \mathcal{B}(G_t)} (Y_{i,b} + y_{i,b}) + \sum_{\upsilon=1}^{\tau} \sum_{b \in \mathcal{B}(G_{t+\upsilon})/\mathcal{B}(G_{t+\upsilon-1})} (Y_{i,b} + y_{i,b}) \right] \tag{28}$$

$$= \sum_{b \in \mathcal{B}(G_t)} (Y_{i,b} + y_{i,b}) + \mathbb{E}_t \sum_{\tau=1}^{\infty} \delta^\tau \sum_{b \in \mathcal{B}(G_{t+\tau})/\mathcal{B}(G_{t+\tau-1})} (Y_{i,b} + y_{i,b}) \tag{29}$$

$$= \sum_{b \in \mathcal{B}(G_t)} (Y_{i,b} + y_{i,b}) + \mathbb{E}_t \sum_{\tau=1}^{\infty} \delta^\tau \left[ p_i (Y_{i,b_{i,t+\tau-1}} + \bar{y}) + \sum_{j \neq i} p_j Y_{i,b_{j,t+\tau-1}} \right] \tag{30}$$

Consider any other strategy, $\sigma_i$. Suppose $\sigma_i$ is a one shot-deviation from longest chain rule in history $H_t^G$. If $\#C(\sigma_i, G_t) < \#C(\sigma_i^{LC}, G_t) - 1$, then $\sigma_i$ cannot induce a change in the equilibrium strategies of other miners. If miner $i$ successfully adds her block, she earns no utility associated with it (it has zero consensus of other miners). On net, she loses $\delta^{t+1} p_i (Y_{i,b_{i,t}} + \bar{y})$.

If instead $\#C(\sigma_i, G_t) = \#C(\sigma_i^{LC}, G_t) - 1$, then with probability $p_i$, when she adds her block, there are now two terminal blocks and so all miners mine each block with probability $1/2$.

On path, her continuation utility is

$$\sum_{b\in\mathcal{B}(G_t)}(Y_{i,b}+y_{i,b})+\mathbb{E}_t\sum_{\tau=1}^{\infty}\delta^{\tau}\left[p_i(Y_{i,b_{i,t+\tau-1}}+\bar{y})+\sum_{j\neq i}p_jY_{i,b_{j,t+\tau-1}}\right] \tag{31}$$

Off-path, her utility from this deviation is

$$\sum_{b\in\mathcal{B}(G_{t-1})}(Y_{i,b}+y_{i,b})+(1-\delta)(Y_{i,b^*}+y_{i,b^*}) \tag{32}$$

$$+p_i\delta\left[\frac{(1-\delta)}{2}(Y_{i,b^*}+y_{i,b^*})+\frac{(1-\delta)}{2}(Y_{i,b_{i,t}}+\bar{y})\right] \tag{33}$$

$$+p_i\delta\left[\frac{1}{2}\sum_{\tau=1}^{\infty}\delta^{\tau}\mathbb{E}_t\left[(1-\delta)(Y_{i,b^*}+y_{i,b^*})+p_i(Y_{i,b_{i,t+\tau}}+\bar{y})+\sum_{j\neq i}p_jY_{b_{j,t+\tau}}\right]\right] \tag{34}$$

$$+p_i\delta\left[\frac{1}{2}\sum_{\tau=1}^{\infty}\delta^{\tau}\mathbb{E}_t\left[(1-\delta)(Y_{i,b_{i,t}}+\bar{y})+p_i(Y_{i,b_{i,t+\tau}}+\bar{y})+\sum_{j\neq i}p_jY_{b_{j,t+\tau}}\right]\right] \tag{35}$$

$$+\delta\sum_{j\neq i}p_j\left[(Y_{i,b^*}+y_{i,b^*})+Y_{j,b_{j,t}}+\sum_{\tau=1}^{\infty}\delta^{\tau}\mathbb{E}_t\left[Y_{i,b^*}+y_{i,b^*}+p_i(Y_{i,b_{i,t+\tau}}+\bar{y})+\sum_{j\neq i}p_jY_{b_{j,t+\tau}}\right]\right] \tag{36}$$

Line (32) represents the lifetime utility the miner receives from blocks in the graph before the terminal block (these appear on any graph following the one-shot deviation) plus her stage-payoff from block $b^*$ (where all the other miners mine in period $t$. Line (33) represents her stage payoff in period $t+1$ if she is successful—there are two chains of equal length and miner's split their mining power evenly. Line (34) is her continuation payoffs when she successfully mines in period $t$ and the mixing in period $t+1$ ends up on the original chain. Line (35) is her continuation payoffs when she successfully mines in period $t$ and the mixing in period $t+1$ ends up on the new chain. Line (36) is her continuation payoffs when she is not successful in period $t$.

Simplifying her deviation payoffs, we have

$$\sum_{b\in\mathcal{B}(G_{t-1})}(Y_{i,b}+y_{i,b})+(1-\delta)(Y_{i,b^*}+y_{i,b^*})+\delta\sum_{\tau=1}^{\infty}\delta^{\tau}\mathbb{E}_t\left[p_i(Y_{i,b_{i,t+\tau}}+\bar{y})+\sum_{j\neq i}p_jY_{b_{j,t+\tau}}\right]$$
$$+p_i\delta\left[\frac{1}{2}(Y_{i,b^*}+y_{i,b^*})+\frac{1}{2}(Y_{i,b_{i,t}}+\bar{y})\right]+\delta(1-p_i)(Y_{i,b^*}+y_{i,b^*})+\delta\sum_{j\neq i}p_jY_{j,b_{j,t}} \tag{37}$$

Differencing (37) from (31), we have

$$\begin{aligned}
&\sum_{b\in\mathcal{B}(G_t)}(Y_{i,b}+y_{i,b})+\mathbb{E}_t\sum_{\tau=1}^{\infty}\delta^{\tau}\left[p_i(Y_{i,b_{i,t+\tau-1}}+\bar{y})+\sum_{j\neq i}p_jY_{i,b_{j,t+\tau-1}}\right]\\
&-\sum_{b\in\mathcal{B}(G_{t-1})}(Y_{i,b}+y_{i,b})-(1-\delta)(Y_{i,b^*}+y_{i,b^*})-\delta\sum_{\tau=1}^{\infty}\delta^{\tau}\mathbb{E}_t\left[p_i(Y_{i,b_{i,t+\tau}}+\bar{y})+\sum_{j\neq i}p_jY_{b_{j,t+\tau}}\right]\\
&-p_i\delta\left[\frac{1}{2}(Y_{i,b^*}+y_{i,b^*})+\frac{1}{2}(Y_{i,b_{i,t}}+\bar{y})\right]-\delta(1-p_i)(Y_{i,b^*}+y_{i,b^*})-\delta\sum_{j\neq i}p_jY_{j,b_{j,t}}\\
=&(Y_{i,b^*}+y_{i,b^*})+\delta\left[p_i(Y_{i,b_{i,t}}+\bar{y})\right]-(1-\delta)(Y_{i,b^*}+y_{i,b^*})\\
&-p_i\delta\left[\frac{1}{2}(Y_{i,b^*}+y_{i,b^*})+\frac{1}{2}(Y_{i,b_{i,t}}+\bar{y})\right]-\delta(1-p_i)(Y_{i,b^*}+y_{i,b^*})\\
=&\frac{p_i\delta}{2}(Y_{i,b^*}+y_{i,b^*})+\frac{p_i\delta}{2}(Y_{i,b_{i,t}}+\bar{y})
\end{aligned}$$

And this difference is positive as long as $Y, y \geq 0$.

By the one-shot deviation principle (which applies since $\delta < 1$), since there is no profitable one-shot deviation from the proposed equilibrium path, the longest chain must be a nash equilibrium.

## A.2 Proof of Proposition 2: Longest Chain Rule is a Perfect Public Equilibrium.

To prove this result, we first prove an intermediate lemma which shows that if longest chain is subgame perfect, then the tie-breaking rule in case a graph exhibits a fork with at least two equal length longest chains must call for miners to mine their most preferred block.

*Lemma 5: If longest chain is a perfect public equilibrium, then for any graph $G_t$ such that $B^{LC}(H_t^G) = \text{argmax}_{b\in G_t} \#C(b, G_t)$ is not a singleton, the longest chain rule satisfies*

$$\sigma_i^{LC}(H_t^i) = \arg\max_{b\in B^{LC}(G_t)}\sum_{b'\in C(b,G_t)}(Y_{i,b'}+y_{i,b'}). \tag{38}$$

To prove the lemma, denote $b_{i,t}^* = \arg\max_{b\in B_t^{LC}}\sum_{b'\in C(b,G_t)}(Y_{i,b'}+y_{i,b'})$. Now consider an arbitrary deviation to $b \in B_t^{LC}$ from $b_{i,t}^*$ by $i$. The difference in following the suggested tie-breaker and deviation payoff is

$$\delta\left(E_t[U_t^i(H_t^i,\sigma)]-E_t[U_t^i(H_t^i,\sigma')]\right)=\delta p_i\left(\sum_{b'\in C(b_{i,t}^*,G_t)}(Y_{i,b'}+y_{i,b'})-\sum_{b'\in C(b,G_t)}(Y_{i,b'}+y_{i,b'})\right) \tag{39}$$

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

Conversely, should a public perfect equilibrium specify a different tie-breaking rule, then a one shot deviation applying the rule specified in (38) immediately yields a profitable deviation.

Using the tie breaking rule in Lemma 5, it is straightforward to develop conditions such that longest chain is a perfect public equilibrium.

To ease notation, let $b^* = B^{LC}(H^G_t)$. Along the equilibrium path, the miner earns utility equal to

$$\sum_{b' \in C(b^*, G_t)} (Y_{i,b'} + y_{i,b'}) + \mathbb{E}_t \sum_{\tau=1}^{\infty} \delta^\tau \left[ p_i (Y_{i,b_{i,t+\tau-1}} + \bar{y}) + \sum_{j \neq i} p_j Y_{i,b_{j,t+\tau-1}} \right] \tag{40}$$

As with determining when the longest chain is a Nash equilibrium, the interesting one-shot deviations to consider are those in which a miner chooses a block $b$ such that $\#C(b, G_t) = \#C(b^*, G_t) - 1$. In this case, the miner induces a tie in the subsequent period which may lead other miners to follow the deviating miner's chain as proposed by the tie-breaking rule. Let $\mathcal{I}(b_{i,t}, b; G_t)$ denote the set of miners who would choose to mine at $b_{i,t}$ linked to $b$ when

tied rather than $b^*$. Then, for such deviations, a miner's expected utility satisfies

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= (1-\delta) \sum_{b' \in C(b^*, G_t)} (Y_{i,b'} + y_{i,b'}) + \delta \sum_{j \neq i} p_j \left( \sum_{b' \in C(b^*, G_t)} (Y_{i,b'} + y_{i,b'}) + (1-\delta) Y_{i, b_{j,t}} \right) \\
&+ \delta(1-\delta) p_i \left[ \frac{W(b_{i,t}, b; G_t)}{1-p_i} \left( \sum_{b' \in C(b, G_t)} (Y_{i,b'} + y_{i,b'}) + Y_{i, b_{i,t}} + \bar{y} \right) + \left( 1 - \frac{W(b_{i,t}, b; G_t)}{1-p_i} \right) \sum_{b' \in C(b^*, G_t)} (Y_{i,b'} + y_{i,b'}) \right. \\
&+ \delta^2 p_i^2 \left( \sum_{b' \in C(b, G_t)} (Y_{i,b'} + y_{i,b'}) + Y_{i, b_{i,t}} + \mathbb{E}_t Y_{i, b_{i,t+1}} + 2\bar{y} \right) \\
&+ \delta^2 p_i \sum_{\substack{j \in \mathcal{I}(b_{i,t}, b; G_t) \\ j \neq i}} p_j \left( \sum_{b' \in C(b, G_t)} (Y_{i,b'} + y_{i,b'}) + Y_{i, b_{i,t}} + \bar{y} + \mathbb{E}_t Y_{i, b_{j,t+1}} \right) \\
&+ \delta^2 p_i \sum_{j \notin \mathcal{I}(b_{i,t}, b; G_t)} p_j \left( \sum_{b' \in C(b^*, G_t)} (Y_{i,b'} + y_{i,b'}) + \mathbb{E}_t Y_{i, b_{j,t+1}} \right) + \delta^2 \sum_{j \neq i} p_j \left( Y_{i, b_{j,t}} + \mathbb{E}_t [Y_{i, b_{t+1}} + y_{i, b_{t+1}}] \right) \\
&+ \left. \sum_{\tau=t+2}^{\infty} \delta^{\tau-t} (1-\delta) \left\{ \mathbb{E}_t \sum_{s=t+3}^{\tau} \sum_{b' \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b'} + y_{i,b'}) \right\} . \right. \qquad (41)
\end{aligned}
$$

The difference in following the longest chain versus the deviation satisfies

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma^{LC}, G_\tau) - \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \left( 1 - (1-\delta) \frac{W(b_{i,t}, b; G_t)}{1-p_i} - \delta(p_i + W(b_{i,t}, b; G_t)) \right) (Y_{i, b_{i,t}} + \bar{y}) \\
&\qquad - \left( (1-\delta) \frac{W(b_{i,t}, b; G_t)}{1-p_i} + \delta(p_i + W(b_{i,t}, b; G_t)) \right) \left( \sum_{b' \in C(b, G_t)} (Y_{i,b'} + y_{i,b'}) - \sum_{b' \in C(b^*, G_t)} (Y_{i,b'} + y_{i,b'}) \right) \qquad (42)
\end{aligned}
$$

Under the conditions of the Proposition, this difference is (weakly) positive ensuring that the longest chain strategy is a public perfect equilibrium.

### A.3 Proof of Proposition 3: Approval Weight Equilibrium

For miner $i$ equilibrium payoff is

$$u^i(\vec{a}, G_t) + E_t[U_t^i(H_t^i, \sigma)]$$

Payoff of deviation is

$$u^i(\vec{a}, G_t) + E_t[U_t^i(H_t^i, \sigma')]$$

Where $\sigma'$ represents the expected set of strategies following the deviation. The difference in equilibrium versus deviation payoff, therefore, can be reduced to $E_t[U_t^i(H_t^i, \sigma)] - E_t[U_t^i(H_t^i, \sigma')]$.

Denote $k'(b, G_t) = \max\{k : b \in T^k(\mathcal{T}(G_t))\}$.

There are two cases to consider:

1. To mine

$$b' : b' \in \mathcal{T}(G_t), \text{and} \mathbb{1}\left\{\sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) > Y_{i,R}(T^{k'}(\mathcal{T}(G_t)), G_t)\right\} = 1$$

   In this case, if $i$ succeeds in appending the next block, $b_{i,t}$, to $b'$

$$\mathcal{P}(b', T^{k'}(\mathcal{T}(G_t)), G_t) = \mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(G_{t+1})), G_{t+1})$$

   Expected utility from period $t+1$ on, along the equilibrium path

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma^{AW}, G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \sum_{s=t+1}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + p_i(Y_{i,b_{i,t}} + \bar{y}) + \sum_{j \neq i} p_j Y_{i,b_{j,t}} \right. \\
&\left. + \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right]
\end{aligned}
$$

Off-path:

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \sum_{j \neq i} p_j Y_{i,b_{j,t}} + \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right]
\end{aligned}
$$

This means that in $k'^{th}$ round, the approval weights have not changed, and therefore

$$E_t[U^i_t(H^i_t, \sigma)] - E_t[U^i_t(H^i_t, \sigma')] = \delta p_i(Y_{i,b_{i,t}} + \bar{y})$$

2. To mine

$$b' : b' \in \mathcal{T}(G_t), \text{and} \mathbb{1}\left\{ \sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) > Y_{i,R}(T^{k'}(\mathcal{T}(G_t)), G_t) \right\} = 0$$

In this case, if $i$ succeeds in appending the next block, $b_{i,t}$, to $b'$

$$\mathcal{P}(b', T^{k'}(\mathcal{T}(G_t)), G_t) + p_i = \mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(G_{t+1})), G_{t+1})$$

Now there are two cases to consider

(a)

$$\mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(G_{t+1})), G_{t+1}) > \mathcal{P}(b^*_t, T^{k'}(\mathcal{T}(G_t)), G_t)$$

Where $b^* = \lim_{k\to\infty} T^k(\mathcal{T}(G_t))$. In this case, $b_{i,t} = \lim_{k\to\infty} T^k(\mathcal{T}(G_{t+1}), G_{t+1})$. Therefore expected utility from period $t+1$ on, along the equilibrium path is

$$\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma^{AW}, G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b\in C(b^*_\tau, G_\tau)} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b\in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \sum_{s=t+1}^{\tau} \sum_{b\in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b\in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + p_i(Y_{i,b_{i,t}} + \bar{y}) + \sum_{j\neq i} p_j Y_{i,b_{j,t}} \right. \\
&\left. + \sum_{s=t+2}^{\tau} \sum_{b\in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right]
\end{aligned}$$

Off-path:

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ p_i \left( \sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) + Y_{i,b_{i,t}} + \bar{y} \right) \right. \\
&+ \sum_{j \neq i} p_j \left( \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + Y_{i,b_{j,t}} \right) \\
&\left. + \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right]
\end{aligned}
$$

And

$$
E_t[U_t^i(H_t^i, \sigma)] - E_t[U_t^i(H_t^i, \sigma')] = \delta p_i \left( \sum_{b \in C(b^*)} (Y_{i,b} + y_{i,b}) - \sum_{b \in C(b')} (Y_{i,b} + y_{i,b}) \right)
$$

Due to the assumption $\mathbb{1}\left\{ \sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) > Y_{i,R}(T^{k'}(\mathcal{T}(G_t)), G_t) \right\} = 0$, the above difference is positive.

(b)

$$
\mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(G_{t+1})), G_{t+1}) < \mathcal{P}(b^*_t, T^{k'}(\mathcal{T}(G_t)), G_t)
$$

In this case, since approval weights in the $k'^{th}$ round do not change, we have the off-path expected payoff:

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (Y_{i,b} + y_{i,b}) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t}(1-\delta) \left[ \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \sum_{j \neq i} p_j Y_{i,b_{j,t}} + \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) \right]
\end{aligned}
$$

And

$$E_t[U_t^i(H_t^i,\sigma)] - E_t[U_t^i(H_t^i,\sigma')] = \delta p_i \left(Y_{i,b_{i,t}} + \bar{y}\right)$$

## A.4 Proof of Proposition 4: Checkpoint Equilibrium

As before, the difference between equilibrium and deviation payoff for all miners is

$$E_t[U_t^i(H_t^i,\sigma)] - E_t[U_t^i(H_t^i,\sigma')]$$

Where $\sigma'$ represents the expected set of strategies following the deviation.

Next thing to note is that following the equilibrium strategies at time $t$, i.e. mining $b_t^*$, results in $B^{CP}(H_{t+1}^G) = b^*$. While any deviation, i.e. mining $b'$, can be weakly (or possibly strictly) profitable if and only if it results in $B^{CP}(H_{t+1}^G) = b'$.

Denote $k'(b, B_t^{CP}, G_t) = \max\{k : b \in T^k(\mathcal{T}(J(B^{CP}(H_t^G), G_t)))\}$.

There are two cases of deviation to consider:

- To mine
$$b' : b' \in \mathcal{T}(J(B^{CP}(H_t^G), G_t)),$$
such that
$$\sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) - \frac{Y_{i,b}}{\delta} \mathbb{1}\{Y_{i,b} < 0\} (1 - \Pi_{\tau=0}^{t} \lambda(b, Y_{i,b}, H_\tau^G)) > Y_{i,R}(T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), G_t)$$
In this case, if $i$ succeeds in appending the next block, $b_{i,t}$, to $b'$, the score for $b_{i,t}$ will be positive if and only if $B_{t+1}^{CP} = b'$, as stated previously. However since
$$\begin{aligned}&\mathcal{P}(b', T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), J(B^{CP}(H_t^G), G_t)) \\ &= \mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_t^G), G_{t+1}))\end{aligned}$$
we can conclude that
$$\begin{aligned}&\mathcal{P}(b_t^*, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_t^G), G_{t+1})) \\ &> \mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_{t+1}^G), G_{t+1}))\end{aligned}$$

This means that approval weight of $C(b_{i,t}, G_{t+1})$ will be equal to zero, due to $B^{CP}(H^G_{t+1}) \notin C(b_{i,t}, G_{t+1})$. Therefore expected utility from period $t+1$ on, along the equilibrium path

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma^{CP}, G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (1-\delta)(Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b} \lambda(b, Y_{i,b}, H^G_\tau) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \Bigg[ (1-\delta) \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) \\
&+ (1-\delta) \sum_{s=t+1}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-2}} \lambda(b_{\tau-2}, Y_{i,b_{\tau-2}}, H^G_\tau) \Bigg] \\
&= \delta \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \delta \mathbb{E}_t (Y_{i,b_t} + y_{i,b_t}) - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H^G_{t+1}) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H^G_\tau) \right] \\
&= \delta \left[ \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + p_i (Y_{i,b_{i,t}} + \bar{y}) + \sum_{j \neq i} p_j Y_{i,b_{j,t}} \right] - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H^G_{t+1}) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \delta^{\tau-t} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H^G_\tau) \right]
\end{aligned}
$$

Off-path:

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (1-\delta)(Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b} \lambda(b, Y_{i,b}, H^G_\tau) \right] \\
&= \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \Bigg[ (1-\delta) \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) \\
&+ \mathbb{E}_t \left[ (1-\delta) \sum_{s=t+1}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-2}, Y_{i,b_{\tau-2}}, H^G_\tau) \right] \Bigg] \\
&= \delta \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \delta \sum_{j \neq i} p_j Y_{i,b_{j,t}} - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H^G_{t+1}) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \delta^{\tau-t} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H^G_\tau) \right]
\end{aligned}
$$

And

$$E_t[U^i_t(H^i_t, \sigma)] - E_t[U^i_t(H^i_t, \sigma')] = p_i \delta (Y_{i,b_{i,t}} + \bar{y})$$

- To mine

$$b' : b' \in \mathcal{T}(J(B^{CP}(H^G_t), G_t)),$$

such that

$$\sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) - \frac{Y_{i,b}}{\delta} \mathbb{1}\{Y_{i,b} < 0\} (1 - \Pi^t_{\tau=0} \lambda(b, Y_{i,b}, H^G_\tau)) \leq Y_{i,R}(T^{k'}(\mathcal{T}(J(B^{CP}(H^G_t), G_t))), G_t)$$

In this case, if $i$ succeeds in appending the next block, $b_{i,t}$, to $b'$, again, the score for $b_{i,t}$ will be positive if and only if $B^{CP}_{t+1} = b'$.

Therefore, there are two cases to consider

1.

$$B^{CP}_{t+1} = b'$$

This corresponds to

$$\begin{aligned}
&\mathcal{P}(b', T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), J(B^{CP}(H_t^G), G_t)) + p_i \\
&= \mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_t^G), G_{t+1})) \\
&> \mathcal{P}(b_t^*, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_t^G), G_{t+1}))
\end{aligned}$$

In this case, $b_{i,t} = \lim_{k\to\infty} T^k(\mathcal{T}(J(B^{CP}(H_{t+1}^G), G_{t+1})))$. Therefore expected utility from period $t+1$ on, along the equilibrium path

$$\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma^{CP}, G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \left[ \sum_{b\in C(b_\tau^*, G_\tau)} (1-\delta)(Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b} \lambda(b, Y_{i,b}, H_\tau^G) \right] \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \Bigg[ (1-\delta) \sum_{b\in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) \\
&+ (1-\delta) \sum_{s=t+1}^{\tau} \sum_{b\in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-2}} \lambda(b_{\tau-2}, Y_{i,b_{\tau-2}}, H_\tau^G) \Bigg] \\
&= \delta \sum_{b\in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \delta \mathbb{E}_t (Y_{i,b_t} + y_{i,b_t}) - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H_{t+1}^G) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b\in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H_\tau^G) \right] \\
&= \delta \left[ \sum_{b\in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + p_i (Y_{i,b_{i,t}} + \bar{y}) + \sum_{j\neq i} p_j Y_{i,b_{j,t}} \right] - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H_{t+1}^G) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \delta^{\tau-t} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b\in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H_\tau^G) \right]
\end{aligned}$$

Off-path:

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \left[ \sum_{b \in C(b^*_\tau, G_\tau)} (1-\delta)(Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b} \lambda(b, Y_{i,b}, H^G_\tau) \right] \\
&= \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \left[ (1-\delta) \left( p_i \sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) + \sum_{j \neq i} p_j \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) \right) \right. \\
&\left. + \mathbb{E}_t \left[ (1-\delta) \sum_{s=t+1}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-2}, Y_{i,b_{\tau-2}}, H^G_\tau) \right] \right] \\
&= p_i \left( \delta \sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) + \delta(Y_{i,b_{i,t}} + \bar{y}) - Y_{i,b'} \lambda(b', Y_{i,b'}, H^G_{t+1}) \right) \\
&+ \sum_{j \neq i} p_j \left( \delta \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \delta Y_{i,b_{j,t}} - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H^G_{t+1}) \right) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \delta^{\tau-t} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H^G_\tau) \right] \right]
\end{aligned}
$$

And

$$
\begin{aligned}
&E_t[U^i_t(H^i_t, \sigma)] - E_t[U^i_t(H^i_t, \sigma')] = p_i \delta \times \\
&\left( \sum_{b \in C(b^*)} (Y_{i,b} + y_{i,b}) - \mathbb{1}\left\{ Y_{i,b^*_t} < 0 \right\} \frac{Y_{i,b^*_t}}{\delta} - \sum_{b \in C(b')} (Y_{i,b} + y_{i,b}) + \mathbb{1}\left\{ Y_{i,b'} < 0 \right\} \frac{Y_{i,b'}}{\delta} \right)
\end{aligned}
$$

We have assumed

$$
\begin{aligned}
&\sum_{b \in C(b', G_t)} (Y_{i,b} + y_{i,b}) - \frac{Y_{i,b}}{\delta} \mathbb{1}\left\{ Y_{i,b} < 0 \right\} \left( 1 - \Pi^t_{\tau=0} \lambda(b, Y_{i,b}, H^G_\tau) \right) \leq \\
&Y_{i,R}(T^{k'}(\mathcal{T}(J(B^{CP}(H^G_t), G_t))), G_t)
\end{aligned}
$$

Also note that

$$\sum_{b\in C(b^*)} (Y_{i,b} + y_{i,b})$$
$$= Y_{i,R}(T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), G_t) + Y_{i,b^*} + y_{i,b^*}$$

Therefore, if $Y_{i,b^*} \geq 0$, we have

$$\sum_{b\in C(b')} (Y_{i,b} + y_{i,b}) - \mathbb{1}\left\{Y_{i,b'} < 0\right\} \frac{Y_{i,b'}}{\delta}$$
$$\leq \sum_{b\in C(b',G_t)} (Y_{i,b} + y_{i,b}) - \frac{Y_{i,b}}{\delta} \mathbb{1}\left\{Y_{i,b} < 0\right\} (1 - \Pi_{\tau=0}^{t} \lambda(b, Y_{i,b}, H_\tau^G)) \leq$$
$$Y_{i,R}(T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), G_t)$$
$$\leq \sum_{b\in C(b^*)} (Y_{i,b} + y_{i,b})$$

Which means the difference in payoffs is positive.

Next if $Y_{i,b^*} < 0$, we have

$$\sum_{b\in C(b')} (Y_{i,b} + y_{i,b}) - \mathbb{1}\left\{Y_{i,b'} < 0\right\} \frac{Y_{i,b'}}{\delta}$$
$$\leq \sum_{b\in C(b',G_t)} (Y_{i,b} + y_{i,b}) - \frac{Y_{i,b}}{\delta} \mathbb{1}\left\{Y_{i,b} < 0\right\} (1 - \Pi_{\tau=0}^{t} \lambda(b, Y_{i,b}, H_\tau^G)) \leq$$
$$Y_{i,R}(T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), G_t)$$
$$= \sum_{b\in C(b^*)} (Y_{i,b} + y_{i,b}) - (Y_{i,b^*} + y_{i,b^*})$$
$$\leq \sum_{b\in C(b^*)} (Y_{i,b} + y_{i,b}) - \mathbb{1}\left\{Y_{i,b_t^*} < 0\right\} \frac{Y_{i,b_t^*}}{\delta}$$

and finally we are again left with a positive difference.

2.
$$B_{t+1}^{CP} \neq b'$$

This corresponds to

$$\mathcal{P}(b', T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_t))), J(B^{CP}(H_t^G), G_t)) + p_{i_1}$$
$$= \mathcal{P}(b_{i,t}, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_t^G), G_{t+1}))$$
$$< \mathcal{P}(b_t^*, T^{k'}(\mathcal{T}(J(B^{CP}(H_t^G), G_{t+1}))), J(B^{CP}(H_t^G), G_{t+1}))$$

In this case, since $B_{t+1}^{CP} \notin C(b_{i,t}, G_{t+1})$, we have off-path payoff:

$$
\begin{aligned}
&\mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} u^i(\sigma', G_\tau) \\
&= \mathbb{E}_t \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \left[ \sum_{b \in C(b_\tau^*, G_\tau)} (1-\delta)(Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b} \lambda(b, Y_{i,b}, H_\tau^G) \right] \\
&= \sum_{\tau=t+1}^{\infty} \delta^{\tau-t} \Bigg[ (1-\delta) \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) \\
&+ \mathbb{E}_t \left[ (1-\delta) \sum_{s=t+1}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-2}, Y_{i,b_{\tau-2}}, H_\tau^G) \right] \Bigg] \\
&= \delta \sum_{b \in C(b^*, G_t)} (Y_{i,b} + y_{i,b}) + \delta \sum_{j \neq i} p_j Y_{i,b_{j,t}} - Y_{i,b^*} \lambda(b^*, Y_{i,b^*}, H_{t+1}^G) \\
&+ \mathbb{E}_t \sum_{\tau=t+2}^{\infty} \delta^{\tau-t} \left[ (1-\delta) \sum_{s=t+2}^{\tau} \sum_{b \in \mathcal{B}(G_s)/\mathcal{B}(G_{s-1})} (Y_{i,b} + y_{i,b}) - \frac{1}{\delta} Y_{i,b_{\tau-1}} \lambda(b_{\tau-1}, Y_{i,b_{\tau-1}}, H_\tau^G) \right]
\end{aligned}
$$

And

$$
E_t[U_t^i(H_t^i, \sigma)] - E_t[U_t^i(H_t^i, \sigma')] = p_i \delta \left( Y_{i,b_{i,t}} + \bar{y} \right)
$$