Predictability, innovation, and competition in Bitcoin's mining market

Robert Parham* Einar Kjenstad†

October 2, 2018

Abstract

We construct and estimate a dynamic oligopoly model of the Bitcoin mining market. Mining equipment manufacturers produce differentiated durable capital goods and endogenously choose optimal investments in R&D. Miners make dynamic purchase decisions based partly on beliefs regarding manufacturers' future choices. We show that policy-relevant values, such as aggregate R&D investment by manufacturers and network energy consumption, are predictable given only a Bitcoin price-path. We further show the industry is uniquely suited to test the impact of product market competition on innovation, a much-debated subject in the economics of R&D literature.

JEL classifications: G23; E42; L11; L63; O31
Keywords: Bitcoin; Crypto-currency; R&D; Innovation; Dynamic Oligopoly; Predictability

[0]The authors wish to thank Bill Wilhelm, Ron Kaniel, Neng Wang, Yakov Amihud, Marc Bevand and seminar participants at the IDC-Herzliya summer conference for useful comments. A previous version of this paper circulated under the name "The predictable cost of Bitcoin".

*University of Virginia (robertp@virginia.edu)
†Aarhus University (einar.kjenstad@econ.au.dk)

1. Introduction

In 2011 Mt. Gox pivoted from a website dedicated to exchanging "Magic the Gathering" playing cards into the first Bitcoin exchange. By doing so, it created a liquid market for trading these Internet novelties, thus establishing a well-defined price for them. That year, Bitcoin miners, who secure the Bitcoin ledger and receive compensation in return, generated an aggregate income of 20 million USD.[1] They did so by solving cryptographic "puzzles" on their personal computers, as part of an intricate mechanism which gives rise to a trustless, cryptographically-secure, distributed ledger of transactions: the Bitcoin blockchain. Six years later, in 2017, Bitcoin miners generated an aggregate income of 3 billion USD, representing a compound annual growth rate of ~ 130 percent. During this period, crypto-currency mining and specifically the mining equipment manufacturers industry, took on an oligopolistic market structure with strong competition, high R&D intensity, and unique predictability features.

Our paper has two objectives: to explore these unique features of crypto-currency mining in the context of the most prominent one, Bitcoin, and to use the ensuing industrial organization to examine the relation between product market competition (PMC) and innovation. On the first front, we follow in the footsteps of recent inquiries into the economic underpinnings of crypto-currencies, such as Yermack (2015), Athey, Parashkevov, Sarukkai, and Xia (2016), Biais, Bisière, Bouvard, and Casamatta (2017), and Huberman, Leshno, and Moallemi (2017). The paper closest to ours in this literature is Huberman et al. (2017), who concentrate on Bitcoin's transaction fees and the congestion queuing game they induce. Huberman et al. (2017) assume miners are only awarded with transaction fees, ignoring the role of block rewards. We assume transaction fees are negligible, which is the case in practice, and focus instead on block rewards. In this sense, our analysis is complementary to theirs. Further, Huberman et al. (2017) refrain from discussing the structure of competition in the mining market, stating that it is left for future research. We fill this gap.

[1] Calculated as the value of block rewards on the day they were granted.

On the second front, there is much work, both theoretical and empirical, on the relation between PMC and innovation, going back at least to Schumpeter (1942). The debate on whether PMC decreases the rate of industry innovation, as Schumpeter (1942) claims, increases it, as Arrow (1962) claims, or has an inverted-U[2] relation with it, as Scherer (1967) claims, has not yet been settled. See, e.g., Gilbert (2006) for a review of the literature.

Rather than conducting a cross-industry analysis, as in Blundell, Griffith, and Van Reenen (1999), who find a decreasing relation, and Aghion, Bloom, Blundell, Griffith, and Howitt (2005), who find an inverted-U relation, we follow Goettler and Gordon (2011) in focusing on a single industry. Goettler and Gordon (2011) construct a structural model of dynamic oligopoly, with endogenous innovation and durable goods. They then estimate it using data from the PC microprocessor industry to assess the effects of competition between Intel and AMD on innovation and profits. They argue that the rate of innovation in the microprocessor industry would have been 4.2% higher had Intel been a monopoly, consistent with Schumpeter (1942). Furthermore, they show that this result depends on the level of R&D spillovers between Intel and AMD.

We extend the analysis of Goettler and Gordon (2011) in three ways. First, their empirical results are regarding a duopoly with no entry and exit, while we consider an oligopolistic industry characterized by high dynamism, in which technological leadership "changes hands" often. Second, the product produced by this industry - specialized integrated circuits (ICs) only usable for Bitcoin mining - is essentially a machine for "printing money," with no scrap value. We can therefore directly calculate the expected value of the ICs to consumers (i.e., Bitcoin miners) rather than estimate complicated consumer demand models. Finally, the unique predictability features of Bitcoin's mechanism design imply that there are few model parameters, and most of those are observable, leading to very clean empirical estimation. An important similarity between our analysis and that of Goettler and Gordon (2011) is the availability of a direct measure of innovation - increase in the ratio of hash power to energy

[2]Innovation rises but then falls with the intensity of PMC, such that innovation is lower at monopolistic and perfectly competitive industries, but higher at some intermediate level of oligopolistic competition.

consumption - which alleviates the customary need to rely on indirect measures, such as patent counts.

We begin by analyzing the Bitcoin mechanism design. We show that the BTC-denominated[3] reward to miners, inflation rate, and total float are predictable. We use these facts to construct a dynamic, Markov-Perfect Nash Equilibrium (MPNE) oligopoly model of the two-sided Bitcoin mining market. IC manufacturers produce mining ICs and sell them to Bitcoin miners, while endogenously choosing an optimal investment in R&D to improve the quality of their ICs. Miners make dynamic IC purchase decisions, based on beliefs regarding the ICs' lifetime values, which are affected among other factors by the miners' forward-looking estimates of BTC prices and the number, and vintage, of all ICs in the market during future periods.

BTC price is the only exogenous variable in our analysis. While much economic work regarding Bitcoin concentrates on the price of BTC, we sidestep questions as to the fundamental value of Bitcoin, and whether BTC prices are "rational" or exhibit a "bubble".[4] Instead, we concentrate on the real outcomes of Bitcoin *given* a BTC price-path. As such, our analysis is a thought experiment answering three questions with practical and policy implications. Given a possible future price path: What would be the aggregate energy consumption of the Bitcoin network? What would be the aggregate R&D expenditure of IC manufacturers? What would be a rational price for a mining IC of any given quality?

Much attention has been directed at Bitcoin mining's energy consumption, describing it as an "environmental disaster" [Bloomberg, April 2013], stating that it is comparable with the energy consumption of Ireland [O'Dwyer and Malone (2014)], or that "one bitcoin transaction now uses as much energy as your house does in a week" [Motherboard, November 2017]. To our knowledge, ours is the first model capable of *predicting* energy costs for future price paths, by taking into account the optimizing behavior of all agents.[5]

[3]To reduce the confusion between Bitcoin the mechanism and Bitcoin the unit of account induced by it, we refer to the unit of account as BTC throughout the paper.

[4]E.g., Cheah and Fry (2015), Urquhart (2016), Nadarajah and Chu (2017).

[5]Vries (2018) conducts an empirical analysis of Bitcoin's energy consumption, and publishes the "Bitcoin

The popular media neglects to discuss the second major input of Bitcoin mining: ICs and the R&D they encapsulate. We show that the Bitcoin protocol gives rise to an R&D arms-race between IC manufacturers, by creating a "tragedy of the commons" situation. Notably, this is true even if there is only one monopolist manufacturer, who must then compete with his own past IC sales. This R&D arms-race forces IC manufacturers to invest heavily in R&D. We follow Griliches (1998) and Goettler and Gordon (2011) in assuming R&D spillovers from the technological leader to laggards. We further discuss the existence of cross-industry R&D spillovers, from IC manufacturers to the microelectronics industry at large. Our model can quantify the aggregate R&D investment by IC manufacturers for future price paths, and help assess the positive R&D externality created by Bitcoin mining.

A feature of our model, with significant importance to practitioners in the industry, is its ability to estimate the expected value of an IC. This value depends on expectations regarding optimal production, pricing, and R&D choices of existing IC manufacturers; entry of new manufacturers; future rewards to miners; BTC price dynamics; and miners' choice to deactivate old ICs as they become unprofitable. Our model is the first to capture this complex set of conditions and yield an expected value for an IC, given its quality and current market conditions.

We estimate the model using data on Bitcoin dynamics gathered directly from the blockchain, and hand-collected data on the release dates and technical features of Bitcoin mining ICs. We test the estimated model's predictive ability using the observed historical path of Bitcoin prices. We show that it predicts the Bitcoin mining industry's evolution well, on dimensions such as network hash rate, innovation rate, and IC prices. We then ask: what if Bitcoin were to become a store-of-value (akin to gold), within the next five years, as some commentators claim? We show that in this state of the world, the Bitcoin mining industry

Energy Consumption Index" on-line, based on his results. He does not purport to provide predictive ability, and his model is severely flawed by the assumption that IC manufacturers produce no profits ("We know that, in equilibrium, not even [...] the largest manufacturer of new Bitcoin mining machines [...] should be able to generate a profit"). Our model highlights the fact that IC manufacturers, who own the only scarce resource required for mining - IC chips of high quality - extract *all* the available rents. This fact materially changes the calculations conducted by Vries (2018).

would consume xxx USD/year in energy, and its aggregate R&D expenditure would be xxx.

The current draft of the paper fully specifies the oligopoly model, but provides empirical results from a simpler model in which only a monopolist manufacturer exists. Estimation of the oligopoly model is technically complex, and we have yet to gain sufficient confidence that we squashed all bugs in the oligopoly estimation code. But as conference submission deadlines are upon us, we defer the discussion of results regarding PMC's impact on innovation to a later draft. The next section describes and formalizes the process of Bitcoin mining, establishing the predictability of miner rewards. Section 3 describes the simplified monopoly model and the full oligopoly model, and discusses theoretical implications of the models. Section 4 presents the empirical estimation and tests the estimated model's ability to predict the data in-sample. Section 5 explores the relation between innovation and competition in our model. Section 6 concludes.

2. Bitcoin's mining market

We begin this section by describing the mining of new BTC in the process of block creation, the concept of mining difficulty, and the dynamic adjustment of mining difficulty as a response to changes in the network hash rate. We then discuss the two forms of miner compensation - block rewards and transaction fees. As BTC price is the core exogenous variable in our analysis, we also briefly discusses price formation and the extant price-path of BTC. We conclude with a description of mining technology evolution and the resulting industrial organization of the IC manufacturing industry. Readers who are not already familiar with Bitcoin's organizing primitives (including the ledger, hash functions, and the blockchain) are encouraged to first read Appendix A for a non-technical intro to Bitcoin and crypto-currencies. In what follows we assume familiarity with these primitives.

Before diving into the mechanics of Bitcoin mining, it is worth reviewing Bitcoin's theoretical underpinning. In the words of Yermack (2017): "*In his conception of the Bitcoin*

blockchain as a distributed open source ledger, Nakamoto (2008) implemented an idea very similar to Kocherlakota's (1998) "money is memory" theory, although Nakamoto does not seem to have been aware of this economists work. Kocherlakota reasons that agents treat money as a store of value because they believe that each owner of a coin obtained the money by delivering goods or services to the coins prior owner, who in turn did the same with the predecessor owner. Kocherlakotas formal model shows that any economic allocation achieved through the use of money could be replicated if all agents knew the complete history of everyones exchanges and kept a running account of their net contributions to the economy, using each agents net economic surplus earned as a signal of their claim against other agents with net deficits. Nakamoto wrote that We define an electronic coin as a chain of digital signatures. In other words, each Bitcoin is made valuable by the ability to attach to it the memory of its previous exchanges."

2.1. Mining BTCs

The need to maintain a distributed consensus regarding a consistent ledger is at the heart of Bitcoin's mechanism design. This is achieved by requiring that new blocks are only added to the blockchain if they contain a verifiable costly signature, or *proof-of-work*. Proof-of-work allows any member of the Bitcoin network to independently verify that a large amount of computing power was required to create a block. As shown by Biais et al. (2017), this means that the Markov-Perfect Nash Equilibrium is for all miners to add truthful blocks to the longest existing blockchain, hence guaranteeing a consistent ledger. The requirement for a costly signature amounts to requiring that a block's hash will be lower than some dynamically determined and widely known value - the inverse of the current *difficulty* level - such that a higher mining difficulty requires a lower hash value. To allow modifying a block's hash without replacing the transactions contained therein, a block definition includes a nuisance field known as the *nonce*, which can be set to any numeric value. Changing the nonce changes the hash value of the entire block, with every change in the nonce constituting

a uniform draw from the domain of possible hash values. Re-calculating the block's hash multiple times with different nonce values will eventually yield a hash value appropriate for the current required difficulty, and is the only way to find such a hash.

The proof-of-work mechanism induces repeated distributed trust-less "tournaments", in which miners compete to create the next block on the chain (by finding a nonce yielding a valid hash value). Each miner's probability of winning a tournament depends on the amount of computing power he invests to search for an appropriate nonce, relative to the aggregate computing power invested by all miners. The miner who wins the tournament is awarded a number of *newly minted* BTCs for his effort, as well as possible *transaction fees* - existing BTCs from transactors whose transactions he included in the new block. It is important to note that the mining process is the origin of all existing BTCs - they were all initially awarded to miners for creating blocks.

To formalize the discussion, consider the n^{th} block in the blockchain, for some arbitrary $n \in \mathbb{N}$. Let $\Delta_n \in \mathbb{R}^+$ denote the time between the creation of block $n-1$ and block n, and let $t_0 \in \mathbb{R}^+$ denote the time at which the "genesis block" (block 0) was created.[6] The time t_n at which block n is created can now be written as

$$t_n = t_0 + \sum_{j=1}^{n} \Delta_j \tag{1}$$

If we let $\mathbf{M}_n$ denote the set of all active miners during block n, and let $h_n^m \in \mathbb{R}^+$ denote the average hash rate[7] of miner $m \in \mathbf{M}_n$ during the mining of block n, then we can write the average *network hash rate* during block n as

$$h_n = \sum_{m \in \mathbf{M}_n} h_n^m \tag{2}$$

[6]Time units should be taken to be in a format amenable to simple arithmetic. One such unit system is seconds since some baseline date, e.g. midnight at 1/1/1970, as in UNIX time.

[7]The average number of hash-function computations performed by the miner per unit of time. The customary units for hash rates are GH/s, or 10^9 hash calculations per second.

Next, let $d_n \in \mathbb{R}^+$ denote the widely-known dynamic hashing difficulty for block n, in terms of expected number of hashes required to find a valid nonce. The expected time between the mining of block $n-1$ and block n can now be written as

$$\mathbb{E}\left[\Delta_n\right] = \frac{d_n}{h_n} \tag{3}$$

The dynamic choice of difficulty is designed to ensure that $\mathbb{E}[\Delta_j] = \Delta^T \equiv 10$ minutes, for all j. Put differently, the Bitcoin mechanism adjusts the difficulty to ensure that some miner will win the tournament and create a new block every ten minutes, on average, *regardless of the network hash rate* h_n. Crucially, the addition of more computing power dedicated to mining does not mean more BTC are mined - it just makes mining BTC harder.

Difficulty adjustments happen every 2016 blocks (approximately two weeks). When a block n for which n is a multiple of 2016 is mined, all peers calculate $(t_n - t_{n-2016})$, the total creation time of the previous 2016 blocks. They then set the value of d_{n+1} (and up to d_{n+2016}) such that

$$d_{n+1} = d_n \cdot \frac{2016 \cdot \Delta^T}{t_n - t_{n-2016}} \tag{4}$$

To highlight how the dynamic difficulty adjustment is able to respond to changes in network hash rate, Figure 1 presents the average network hash rate and the dynamic difficulty since Bitcoin inception. Mining difficulty closely follows the aggregate hash rate. This leads us to our first observation

Observation 2.1. The mining time of block n is predictable

Which follows from equation 1, if the difficulty adjustment procedure ensures $\mathbb{E}[\Delta_j] = \Delta^T$. The timing of block n is only *approximately* predictable because of the discrete nature of the difficulty adjustment period and the stochastic nature of block arrival. If the network hash rate changes sufficiently quickly during an adjustment period, then $(t_n - t_{n-2016})$ will somewhat deviate from its expected value, $(2016 \cdot \Delta^T)$. Across Bitcoin's life, the network hash rate has grown by almost 1% *per day*, though in recent years, the pace has stabilized

at around 0.42% per day. At these paces, the theoretical durations of adjustment periods are 94% and 97% of the expected value, respectively. In practice, the average time per block across all $\sim 530,000$ blocks since Bitcoin's inception is 9.41 minutes.

This striking increase in network hash rate is illustrated in Figure 1 by the horizontal level lines, depicting the hashing abilities of several computing systems. Note that the axes in Figure 1 are in log-scale, and that in early 2010, the hash rate of the entire network could be attained by one modern GPU. By mid-2011, the network hash rate rose to a level attainable using *all* of the world's 500 top supercomputers, combined.[8] This phenomenal climb in computing power was enabled by repeated quality improvements in mining technology - improvements which are at the heart of our investigation - and we will consider them further after discussing miner rewards.

2.2. Block rewards

As compensation for his effort, the miner who creates block n is awarded $a_n \in \mathbb{R}^+$ newly minted BTC, known as the *block reward*. The block reward was set at $a_0 = 50$ BTC for the first block created, and the Bitcoin mechanism dictates it halves every $\Delta^A \equiv 210,000$ blocks, such that $a_{210,000} = 50$, but $a_{210,001} = 25$. Note that this prescription leads to a hard limit on the number of BTC to ever exist, at $21,000,000$, because

$$210,000 \cdot 50 \cdot \Sigma_{i=0}^{\infty}(0.5)^i = 21,000,000$$

We can write a_n, the block reward for block a, as

$$a_n = a_0 \cdot 2^{-\lfloor \frac{n}{\Delta^A} \rfloor} \tag{5}$$

[8]This increase was made possible by the introduction of computing systems specifically optimized for Bitcoin mining, as discussed in Section 2.4.

and the total BTC float (number of BTC in circulation) at block n, f_n, as

$$f_n = \sum_{k=1}^{n} a_k \tag{6}$$

Note that because block rewards are newly minted BTC, there is an inherent BTC-denominated inflation rate in the Bitcoin network. Put differently, block rewards are financed by inflation. The inflation rate at block n, i_n, can be expressed as

$$i_n \equiv \frac{f_n - f_{n-1}}{f_{n-1}} = \frac{r_n}{f_{n-1}} \tag{7}$$

These definitions allow us to make our second observation by combining Observation 2.1 with Equations 5, 6, and 7

Observation 2.2. The block reward a_n, the BTC float f_n, and the inflation rate i_n are predictable

Figure 2 follows this observation by presenting the number of BTC in circulation and the number of awarded BTC per block up to the year 2035. The vertical line represents June 30^{th} 2018. Data to the left of the line are based on actual observations while data to the right are based on Equations 5 and 6.[9] Note that $\sim 80\%$ of BTC to ever exist have already been mined.

Figure 3 presents the BTC-denominated yearly inflation rate for the same period. As in Figure 2, data to the left and right of the vertical line are actual observations and imputations from Equation 7, respectively. We can see that Bitcoin began with a period of very high inflation, but current inflation is below 5% in yearly terms. The predictability of Bitcoin's inflation, and the inability of central banks or other entities to determine or influence it, is one of the core design principles of Bitcoin, as outlined by its creator(s) in Nakamoto (2008).

A second form of compensation to the miner who creates block n is optional transaction

[9] We elect 2035 as the ending period because by mid-2035, 99% of BTC to ever exist will have already been mined.

fees. These fees are paid by transactors whose transaction the miner includes in the block, and are a transfer of BTC from transactors to miners, rather than newly created BTC. Miners have discretion as to which transactions they include in a block they attempt to sign, and if there are more transactions waiting to be committed to the blockchain than can fit in a single block, rational miners will choose to include those transactions which promise them the highest transaction fees. Transactors, in turn, will attach transaction fees to their transactions if there is congestion in the network, to make sure miners give higher priority to their transactions. Thus, an important feature of transaction fees is that they are *congestion dependent*, as discussed by Huberman et al. (2017). In this sense, congestion in the Bitcoin mechanism is "not a bug but a feature". This is further highlighted by Nakamoto (2008), who write(s) "In a few decades when the reward gets too small, the transaction fee will become the main compensation for nodes. I'm sure that in 20 years there will either be very large transaction volume or no volume."

Figure 4 depicts the dependence of transaction fees on congestion by presenting the number of daily transactions and the ratio of transaction fees to total miner reward (i.e., transaction fees plus block rewards) for the duration of Bitcoin's existence. As transaction load increased above $\sim 200,000$ transactions per day - the maximum network load - transaction fees increased significantly. Transaction fees later returned to being an insignificant part of miner compensation, following the decrease in transaction load. The maximum network load is a technical limit of the Bitcoin network which is *unrelated* to the computing power used to authenticate transactions.[10] Using more computing power for mining therefore has no impact on the maximum network load.

Because transaction fees depend on transaction load, which in turn depends on the rate of Bitcoin adoption and other exogenous factors, our analysis does not assume any transaction fee predictability. Hence, the model presented in Section 3 makes the conservative assump-

[10]Recall that transactions need to be packed into blocks, and blocks are created at a pre-determined rate. The number of transactions that can fit in a block is limited by the Bitcoin protocol, due to a limit on block size. Nicolas (2014) discusses the problem of limited block size from an economic perspective.

tion that transaction fees are always zero, but can easily be modified to assume transaction fees constitute any percentage of total miner reward.

2.3. The price of BTC

We now turn our attention to the main exogenous variable of our analysis - BTC price - equivalently defined as the USD/BTC exchange rate. As mentioned in Section 1, we take no stand regarding the process of BTC price formation, or whether prices are "rational". We are merely interested in understanding real-world outcomes, *given* a price level. Nevertheless, stylized facts about BTC's historical price path are crucial for our analysis, and so Panel (a) of Figure 5 presents the BTC price since 2010 and up to July 1^{st} 2018.

Several facts stand out in Panel (a) of Figure 5. First, the exceptional price appreciation. BTC price exhibits cumulative annual growth rate of ~ 420 percent, from 0.066 USD/BTC in August 2010 to $\sim 6,300$ USD/BTC in July 2018. Second, prices exhibit periods of significant increase followed by decline (or "bubbles"). Note that the late-2017 period was neither exceptional, nor the most pronounced. This is noteworthy, as the popular media usually presents BTC price graphs in linear (rather than log) scale, as depicted in Panel (b). The difference in conclusions the unaware reader may derive from Panels (a) and (b) of Figure 5 is striking. In our view, it is misguided to observe a phenomenon which is exponential in nature on a linear scale, as it obscures important patterns of said phenomenon.

Let $b_t \in \mathbb{R}^+$ denote BTC price in terms of nominal USD at an arbitrary time t. Recall that Equation 1 defines the creation time of block n as t_n. We can write $z_n \in \mathbb{R}^+$, the market capitalization[11] of Bitcoin at time t_n as

$$z_n = b_{t_n} \cdot f_n \tag{8}$$

We can also write the block reward to the miner who creates block n in USD terms, sans

[11] Ignoring the phenomena of lost BTC, discussed in Appendix A.

transaction fees, as

$$g_n = b_{t_n} \cdot a_n \tag{9}$$

Combining Observation 2.2 with Equations 8 and 9, we have

Observation 2.3. The Bitcoin market cap z_n and the miner reward in USD terms, g_n, are functions of only BTC price at time t_n.

Put differently, if we can form an expectation regarding BTC price at some future time, we can calculate the miner reward at that time.

To conclude the discussion of BTC price, Figure 6 displays BTC's actual market cap since 2010 and up to July 1^{st} 2018. Figure 6 also includes level lines presenting the July 2018 values of the M2 monetary US aggregate, and the total value of the world's gold reserves, for comparison.[12] While the popular media often speculates regarding Bitcoin becoming a "reserve currency" or a "store of value akin to gold", Figure 6 puts these concepts in context.

2.4. Mining history and IC manufacturers

We now return our attention to the phenomenal increase in network hash rate exhibited in Figure 1. To rationalize this increase, we provide a brief overview of the historical evolution of mining technology which led to it. This will help highlight the rapid innovation in Bitcoin mining, and the ensuing industry structure. See Taylor (2017) for more information on the evolution of Bitcoin hardware.

The first open source Bitcoin software client was released on January 9^{th} 2009. Few people were aware of Bitcoin at the time or bothered to mine it, so the difficulty level d_n was initially very low. The low difficulty meant little computing power was required to win the "tournament" and receive BTC awards, such that using general purpose CPUs[13]

[12] M2 estimate is from FRED; Gold reserves are calculated based on the estimate that 165,000 metric tons of gold were mined in human history, from http://numbersleuth.org/worlds-gold/.

[13] Central Processing Units - the integrated circuits that conduct computations in everyday computers.

was sufficient to mine Bitcoin. It is widely accepted in the Bitcoin community that the mysterious Satoshi Nakamoto mined over a million BTC during the "CPU era".

Bitcoin's popularity steadily grew among both technology enthusiasts and consumers using it for anonymous on-line transactions - mostly illegal drug purchases on Silk Road. The growing popularity of Bitcoin led to increased mining, which in turn led to higher difficulty via the adjustment mechanism described by Equation 4. By the end of 2010 mining using CPUs became unprofitable, as the expected value of BTCs mined using the CPU was lower than the cost of energy consumed by the CPU. It was the first time of many in which difficulty grew to a level that made the probability of winning the "tournament" using current mining equipment (e.g., CPUs) vanishingly small. Miners looked for better ways to mine Bitcoin, and concentrated their attention on GPUs[14].

GPUs are designed for repetitive "number crunching", whereas CPUs are designed for "executive decision making", so GPUs were better at quickly and efficiently calculating hashes. The first open-source program to enable GPU mining was released in October 2010. The arms race continued - GPUs were widely available because many existing computers already included powerful GPU units - and as they were deployed to Bitcoin mining, the difficulty again rose quickly. By June 2011, FPGAs[15] became all the rage among Bitcoin miners.

FPGAs represented the first step towards specialization in Bitcoin mining. Unlike GPUs, they were not already widely deployed, and had to be specifically purchased and fitted for the task of Bitcoin mining. Furthermore, they had few alternative uses[16] so they represented a sunk capital cost with low scrap value from an economic perspective. FPGAs provided a five-fold increase in power efficiency, but their initial capital cost and low scrap value relative to GPUs meant GPUs remained economically viable. Both FPGAs and GPUs

[14]Graphical Processing Units - computer chips originally designed for high-quality graphic processing.

[15]Field-Programmable Gate Arrays - hardware components which can be programmed for a specific calculation.

[16]FPGAs can be reprogrammed for other tasks (hence, "field-programmable"), but have no wide consumer use like CPUs and GPUs.

became obsolete for Bitcoin mining soon after the introduction of the first ASICs (Application Specific Integrated Circuits) on Januray 2013.

Application specific integrated circuits (henceforth ICs) are computer chips designed solely for a single purpose - like computing Bitcoin hashes. As such, they are significantly faster and more power efficient at this task than general-purpose hardware, such as CPUs, GPUs and FPGAs. The first publicly available IC - Avalon 1 - has such high efficiency it mined over $500 worth of BTC during its first day of operation. The introduction of ICs did not stop the mining arms race. On the contrary - it now became even more fierce. Several mining IC manufacturers emerged, competing for market dominance and introducing improved ICs in short development cycles. This led to swift obsolescence of previous ICs that were the state of the art only months prior. By January 2018, merely 5 years after the introduction of Avalon 1, ASICMiner - one of the manufacturers in this new oligopoly - introduced the AM0815 IC, which is almost a 1000-fold faster than the Avalon 1, and 100-fold more energy efficient.

ICs changed the Bitcoin mining market from an egalitarian market with low barriers-to-entry to a market dominated by a few IC manufacturers, who control production of the scarce resource required for mining. In this way, ICs gave rise to an oligopolistic industry structure, in which miners are but economic "peons". IC manufacturers now extract most of the value created by Bitcoin mining, and make the economically significant decisions. The next section provides a formal model of this new industry structure.

3. Model

We model Bitcoin mining as an infinite-horizon dynamic oligopoly industry. IC manufacturers produce differentiated durable capital goods (mining ICs) and endogenously choose optimal investment in R&D to improve the quality of future ICs. Miners make dynamic purchase decisions, based on beliefs regarding the IC's lifetime value. The income generated

by an IC per-period depends on: (1) BTC price, (2) BTCs awarded per period, and (3) the IC's computing power relative to total computing power dedicated to mining Bitcoins during the period. Demand is hence driven by miners' forward-looking estimates of BTC prices and the number, and vintage, of all ICs in the market during future periods.

Section 3.1 begins the discussion by presenting a simpler monopoly model, in which only one IC manufacturer exists. This is done both for ease of exposition, and because it is the limiting case of an oligopolistic industry in which the manufacturers form a perfect cartel. Building on intuition from the monopoly model, we introduce a full oligopoly model in Section 3.2. The timing of within-period actions by manufacturers and miners is provided in Figure 7. Model notation is summarized in Table 1.

3.1. A monopolistic IC industry

3.1.1. The manufacturer

A monopolist IC manufacturer faces a dynamic production-investment decision in discrete time, in which she needs to decide how many ICs to produce and how much to invest in R&D each period. At the beginning of every period $t \in \mathbb{N}$, she observes her current quality level $q_t \in \mathbb{N}$; BTC price $b_t \in \mathbb{R}^+$; and BTC float $f_t \in \mathbb{R}^+$. The price of BTC is exogenous to the model, and we assume it follows an $AR(1)$ process in logs, such that

$$\log(b_{t+1}) = \rho_b \log(b_t) + \sigma_b \epsilon_t \tag{10}$$

with $\epsilon_t \sim \mathbf{N}(0,1)$, and $\rho_b > 0, \sigma_b > 0$ parameters. The law of motion for BTC float is a deterministic function defined by the Bitcoin protocol, as described in Section 2, such that

$$f_{t+1} = \Psi(f_t) \tag{11}$$

for some deterministic function Ψ. The manufacturer's quality level q_t is endogenous to the model, and progresses in time based on her investment in R&D, in a manner described shortly.

At the beginning of period t, the manufacturer also observes the previous period's *capital vintage structure* - the total number of ICs of each quality level that existed in the market during period $(t-1)$. Let Q_{t-1} be a column vector whose q^{th} element $Q_{t-1}^{(q)} \in \mathbb{N}$ denotes the number of ICs with quality q that existed in the market at the end of period $(t-1)$.[17] The state of the economy at the beginning of period t can now be summarized by the tuple

$$\mathbf{S}_t^- = \langle Q_{t-1}, q_t, b_t, f_t \rangle \tag{12}$$

The manufacturer must choose an amount of ICs to produce and sell this period, c_t, and an amount to spend on R&D, r_t. All ICs produced at time t are of her current quality level, q_t. The manufacturer's decision at time t is summarized by the tuple

$$\mathbf{X}_t = \langle c_t, r_t \rangle \tag{13}$$

The manufacturer's lifetime value given a beginning-of-period state of the economy $\mathbf{S}_t^-$ is recursively defined by the Bellman equation

$$U\left(\mathbf{S}_t^-\right) = \max_{\mathbf{X}_t} \left\{ \Omega\left(\mathbf{S}_t^-, \mathbf{X}_t\right) + \beta^{IC} \mathbb{E}\left[U\left(\mathbf{S}_{t+1}^-\right) \mid \mathbf{S}_t^-, \mathbf{X}_t \right] \right\} \tag{14}$$

in which $\Omega\left(\mathbf{S}_t^-, \mathbf{X}_t\right)$ is her expected per-period payoff. The manufacturer's inter-temporal discount rate is denoted $\beta^{IC} > 0$.

[17] Assume for now only $\bar{Q}$ qualities are ever achievable, to constraint the cardinality of Q_{t-1}. We will later follow Goettler and Gordon (2011) in referring only to relative qualities to remove this assumption but still constraint the cardinality of Q_{t-1}.

Her expected per-period payoff is given by

$$\Omega\left(\mathbf{S}_t^-, \mathbf{X}_t\right) = c_t \cdot \left(\mathbb{E}\left[p_t \middle| \mathbf{S}_t^-, \mathbf{X}_t\right] - mc\right) - r_t \tag{15}$$

which is the straightforward production-investment per-period payoff function, with an expected market-clearing price p_t and a constant marginal cost of production $mc > 0$.[18]

Note that the next-period pre-entry state $\mathbf{S}_{t+1}^-$ in Equation 14 includes the manufacturer's next period quality q_{t+1}. The relevant quality measure in Bitcoin mining is the effectiveness of an IC in transforming electricity expenses to hashes. We follow Pakes and McGuire (1994) in using a "quality ladder", with a successful R&D investment resulting in a quality improvement of exactly one step up the ladder. This is a consequence of quality representing an index, which is later calibrated such that the underlying efficiency gain matches observed data. The manufacturer's next period quality follows

$$q_{t+1} = \begin{cases} q_t + 1, & \text{with probability } \Theta\left(r_t\right) \\ q_t, & \text{otherwise} \end{cases} \tag{16}$$

with the quality improvement function Θ defined as

$$\Theta\left(r_t\right) = \frac{\left(r_t/\alpha_1\right)^{\alpha_2}}{1 + \left(r_t/\alpha_1\right)^{\alpha_2}} \tag{17}$$

following Goettler and Gordon (2011). The R&D productivity parameters $\alpha_1 > 0$ and $\alpha_2 > 0$ will play a crucial role in determining industry structure. The functional form in Equation 17 implies that if the manufacturer invests α_1 in R&D, she has a 50% chance of advancing to the next quality level. The rate by which the success probability increases with increased investment is controlled by α_2.

[18] We assume a constant marginal cost of production for simplicity, but also because this is the observed pattern in the Bitcoin mining market over its existence, as well as in the microelectronics industry at large.

The law of motion for the capital vintage structure is then

$$Q_t = Q_{t-1} + c_t \cdot \mathbf{1}_{q_t} \tag{18}$$

in which $\mathbf{1}_{q_t}$ is a column vector of the same cardinality as Q_{t-1}, with zeros in all elements besides element q_t, that is equal to 1. Before we proceed to describe the miner's problem, it will be useful to define the state of the economy in period t after the manufacturer sold her vintage of c_t ICs as

$$\mathbf{S}_t^+ = \langle Q_t, q_t, b_t, f_t \rangle \tag{19}$$

3.1.2. The miners

There is free entry into the "prospective miner" market, so many new potential miners are born every period. A potential miner can choose whether to buy a single IC for the market clearing price p_t. If he chooses to buy, he enters the mining market and mines BTC in every future period to the extent that it is profitable to do so. A new miner's entry decision is forward-looking in that the maximum price he is willing to pay for an IC is set with rational expectations about the manufacturer's current and future IC sales and R&D investments. The miner is willing to pay less for an IC today if he expects high future IC sales or ICs of higher quality. This is because the competitive nature of mining will mean that the extra/better ICs will decrease the future profitability of the mining IC he intends to buy. A miner will enter the market if and only if

$$\mathbb{E}\left[V\left(\mathbf{S}_t^+, \mathbf{X}_t, q_t\right) \mid \mathbf{S}_t^-\right] - p_t \geq 0 \tag{20}$$

with $V\left(\mathbf{S}_t^+, \mathbf{X}_t, q_t\right)$ denoting the lifetime value of a miner owning an IC of quality q_t when the state of the economy post-entry is $\mathbf{S}_t^+$ and the manufacturer's decisions this period are $\mathbf{X}_t$. The information set available to form the miner's expectations as to lifetime IC value, the market structure post-entry, $\mathbf{S}_t^+$, and the manufacturer's decisions $\mathbf{X}_t$, is the state of

the economy pre-entry, $\mathbf{S}_t^-$. We assume that the pre-entry market vintage structure Q_{t-1} is observable to all prospective miners (it is a component of $\mathbf{S}_t^-$) as a way to impose rational expectations of the manufacturer's future choices. Note that due to the assumed free entry into the "prospective miner" market, Equation 20 holds with equality every period.

A miner's lifetime value is recursively defined by the Bellman equation

$$V\left(\mathbf{S}_t^+, \mathbf{X}_t, q\right) = \Pi\left(\mathbf{S}_t^+, q\right) + \beta^M \mathbb{E}\left[V\left(\mathbf{S}_{t+1}^+, \mathbf{X}_{t+1}, q\right) \mid \mathbf{S}_t^+, \mathbf{X}_t\right] \tag{21}$$

in which $\Pi\left(\mathbf{S}_t^+, q\right)$ is the expected per-period payoff of a miner owning an IC of quality q when the capital vintage structure is $\mathbf{S}_t^+$. The miner's inter-temporal discount rate is $\beta^M > 0$. Note that the miner's IC quality, q, remains the same and does not update with time. This is because the basic unit the model tracks is the IC, and an "upgrade" decision by a miner is modeled as the entry of a new miner purchasing a new IC.

The per-period payoff function $\Pi\left(\mathbf{S}_t^+, q\right) \geq 0$ captures the miner's decision whether to operate his mining IC in a given period. Once an entry decision was made, the cost of an IC is sunk, and the economically relevant choice is whether to activate the IC the entire period, part of the period, or remain inactive, considering the expected revenue and operation cost (i.e., energy consumption). We assume for simplicity there is no cost to "storing" the IC when it is inactive. The payoff function also captures the competitive nature of mining, in which a single miner's share of the aggregate mining revenue depends on his computing power relative to the total *active* computing power of the network during the period.[19]

We write the per-period payoff as

$$\Pi\left(\mathbf{S}_t^+, q\right) = \begin{cases} b_t\left(f_{t+1} - f_t\right)\Phi\left(\mathbf{S}_t^+, q, q_t^*\right) - e_M, & \text{if } q \geq \lceil q_t^* \rceil \\ 0, & \text{if } q = \lfloor q_t^* \rfloor \\ 0, & \text{if } q < \lfloor q_t^* \rfloor \end{cases} \tag{22}$$

[19] As described in Section 2, while a single miner faces a stochastic "lottery", the introduction of mining pools smooths his earnings to a level which allows us to ignore the stochastic nature of mining rewards.

The three cases correspond to: (1) a fully active miner earning a positive payoff, (2) a miner active only part of the period earning zero payoff, and (3) a miner inactive during the entire period, also earning zero payoff.[20] The miner's payoff in case (1) equals the aggregate revenue of all miners, times that miner's share of the aggregate revenue, minus his energy cost, $e_M > 0$.[21] The aggregate revenue is the price of BTC multiplied by the number of new BTCs awarded to miners during the period, $(f_{t+1} - f_t)$. The miner's share of this aggregate revenue is captured by $\Phi\left(\mathbf{S}_t^+, q, q_t^*\right)$, which is a function of the current capital vintage structure, the miner's quality level, and a per-period activation threshold, $q_t^* \in \mathbb{R}^+$.

The activation threshold q_t^* is the value solving the equation

$$b_t \left(f_{t+1} - f_t\right) \Phi\left(\mathbf{S}_t^+, \lfloor q_t^* \rfloor, q_t^*\right) - e_M = 0 \tag{23}$$

which states that the payoff of a miner with IC of quality $\lfloor q_t^* \rfloor$ is zero. Before discussing the miner's threshold policy, we introduce its core component, the miner's earning share function

$$\Phi\left(\mathbf{S}_t^+, q, q_t^*\right) = \frac{\exp\left(q\delta\right)}{\sum_{q' \geq \lceil q_t^* \rceil} \left[Q_t^{(q')} \exp\left(q'\delta\right)\right] + \left(\lceil q_t^* \rceil - q_t^*\right) Q_t^{(\lfloor q_t^* \rfloor)} \exp\left(\lfloor q_t^* \rfloor \delta\right)} \tag{24}$$

The numerator of Equation 24 is simply the computing power of a single IC with quality q. As in Goettler and Gordon (2011), IC computing power is defined on an exponential grid with step size $\delta > 0$, i.e., an IC with quality $q+1$ performs a factor δ more calculations per second than an IC with quality q. The denominator of Equation 24 has two parts. The first part captures the aggregate computing power of fully active miners (case (1) of Equation 22), and the second part captures the aggregate computing power of miners that are active only part of the period (case (2) of Equation 22). The share of time type (2) miners are active is captured by the term $(\lceil q_t^* \rceil - q_t^*)$.[22]

[20] For this equation to hold with exact equality, one needs to assume that the scale of each IC is small relative to the aggregate computing power of the entire Bitcoin network.

[21] Energy cost is *normalized* - rather than assumed - to be quality-independent, as it determines the relative scale of mining ICs.

[22] An example might be useful here. Suppose $q_t^* = 2.7$; this implies miners of type 3 are fully active

Consider now the per period payoff for the marginal miner with arbitrary quality q, and assume all other miners are following the threshold policy defined in Equation 22. If $q \geq \lceil q_t^* \rceil$, the miner can earn positive payoff by being active, and hence will be active the entire period. If $q < \lfloor q_t^* \rfloor$, the miner's energy cost will be higher than his expected revenue even if he chooses to be activate for only a small part of period t, and hence he will remain inactive the entire period. If $q = \lfloor q_t^* \rfloor$, a sub-game "tragedy of the commons" situation [Hardin (1968)] arises.

Note that if all miners of type $q = \lfloor q_t^* \rfloor$ are active for a time share of exactly $(\lceil q_t^* \rceil - q_t^*)$ out of period t, then their per-period payoff is exactly zero by Equation 23 which defines q_t^*. In this situation, if any of them decides to be active for longer, then all their payoffs, including his, will be negative. If they all agree to only be active $(\lceil q_t^* \rceil - q_t^*)/2$ out of period t, they will all earn some positive payoff. If, however, all other miners of type $q = \lfloor q_t^* \rfloor$ follow the agreement to only be active for $(\lceil q_t^* \rceil - q_t^*)/2$, the marginal miner can increase his profit by being active for more than $(\lceil q_t^* \rceil - q_t^*)/2$. Hence, the equilibrium of the sub-game *without cooperation* is that all miners of type $q = \lfloor q_t^* \rfloor$ are active for $(\lceil q_t^* \rceil - q_t^*)$ and earn zero per-period payoff (see, e.g., Clark (1990), ch. 5.4).

3.2. An oligopolistic IC industry

We now generalize the monopoly model presented in Section 3.1 into a model of dynamic oligopoly, in the spirit of Pakes and McGuire (1994), Ericson and Pakes (1995), Pakes and McGuire (2001), and Goettler and Gordon (2011). The discussion in this section concentrates on the differences in modeling the monopoly and oligopoly cases. Because the only salient feature of an IC is its quality, all manufacturers with a specific quality level q are essentially selling an undifferentiated good (relative to each other). A symmetry argument then implies that their policies are identical, and we can consider only the optimal policies

(and earn positive per-period payoff), and miners of type 2 are active only 30% of the time (and earn zero per-period payoff). Hence, miners of type 2 only contribute 30% of their *maximum* computing power to the denominator.

of a representative member of each quality level. Put differently, our equilibrium concept is now a symmetric MPNE. Henceforth, when we refer to a manufacturer by her quality level q, we implicitly assume that all other manufacturers of quality q follow the same policy.

3.2.1. The manufacturers

The economy now consists of $M \in \mathbb{N}$ independent IC manufacturers, each faced with a similar dynamic production-investment decision as the monopolist in Section 3.1. At the beginning of period t, an arbitrary manufacturer m observes the current BTC price b_t, the BTC float f_t, her own quality q_t^m, and the previous period's capital vintage structure Q_{t-1}. The laws of motion for b_t and f_t, as defined by Equations 10 and 11 respectively, still hold.

The manufacturer m also observes the *industry structure* - the number of IC manufacturers of each quality level. Let W_t be a column vector whose q^{th} element $W_t^{(q)} \in \mathbb{N}$ denotes the number of IC manufacturers with quality q that exist at the market in period t. Note that manufacturer m is also included in the industry structure, i.e., $W_t^{(q_t^m)}$ counts her as well.[23] We re-define the state of the economy at the beginning of period t to be the tuple

$$\mathbf{S}_t^- = \langle W_t, Q_{t-1}, b_t, f_t \rangle \tag{25}$$

Next, we generalize the quantity and R&D investment decisions for the oligopoly case. Let C_t be a column vector whose q^{th} element $C_t^{(q)} \in \mathbb{N}$ denotes the amount of ICs produced and sold by a *single* manufacturer of quality q. Similarly, $R_t^{(q)} \in \mathbb{R}^+$ is the amount a single manufacturer of quality q chooses to spend on R&D. Manufacturers' decisions can be summarized by the tuple

$$\mathbf{X}_t = \langle C_t, R_t \rangle \tag{26}$$

[23] As before, assume for now that only $\bar{Q}$ qualities are ever achievable, to constraint the cardinality of W_t.

We re-write the law of motion for Q_t, the capital vintage structure, as

$$Q_t = Q_{t-1} + C_t \circ W_t \tag{27}$$

in which $\circ$ is the Hadamard (entrywise) product. The state of the economy after all manufacturers sold their ICs is now

$$\mathbf{S}_t^+ = \langle W_t, Q_t, , b_t, f_t \rangle \tag{28}$$

A manufacturer of quality q has a lifetime value which is recursively defined by the Bellman equation

$$U\left(\mathbf{S}_t^-, q\right) = \max_{\mathbf{x}_t^{(q)}} \left\{ \Omega\left(\mathbf{S}_t^-, \mathbf{X}_t^{(q)}, q\right) + \beta^{IC} \mathbb{E}\left[U\left(\mathbf{S}_{t+1}^-, q'\right) \mid \mathbf{S}_t^-, \mathbf{X}_t^{(q)} \right] \right\} \tag{29}$$

Note that the value function U now takes q as a parameter. This means Equation 29 defines a set of value functions, one for each quality level. A manufacturer of quality q only determines her own choices, but forms rational expectations regarding the decisions of manufacturers of all other qualities. This includes rationally expecting all competitors of the same quality as her to reach the same decisions as she.

The manufacturer's per-period payoff function is

$$\Omega\left(\mathbf{S}_t^-, \mathbf{X}_t^{(q)}, q\right) = C_t^{(q)} \cdot \left(\mathbb{E}\left[P_t^{(q)} \mid \mathbf{S}_t^-, \mathbf{X}_t^{(q)} \right] - mc \right) - R_t^{(q)} \tag{30}$$

in which P_t is a price vector, organized similar to the C_t and R_t vectors. Note that the market is not segmented by IC quality, such that $P_t^{(q)}$ - the market clearing price for a specific quality q - depends on the manufacturing decisions of *all* manufacturers, regardless of quality.

Note that the expected utility of a manufacturer in time t, given by Equation 29, depends on her expectations as to her next period quality, q', but also on her expectations of all other manufacturers' qualities, as captured by the progression of the industry structure W_t.

We follow the literature on R&D spillovers, reviewed by Griliches (1998), in assuming it is harder to innovate at the technological frontier than when the manufacturer is a technological laggard. To capture this logic, we re-define the quality law of motion to be

$$q' = \begin{cases} q+1, & \text{with probability } \Theta\left(R_t^{(q)}, \bar{q}_t - q\right) \\ q, & \text{otherwise} \end{cases} \tag{31}$$

in which $\bar{q}_t$ is the quality level of the technological leader at time t (i.e., the highest quality level in the industry). The quality improvement function Θ is now defined as

$$\Theta\left(R_t^{(q)}, \bar{q}_t - q\right) = \frac{\left(R_t^{(q)}/\alpha\right)^{\alpha_2}}{1 + \left(R_t^{(q)}/\alpha\right)^{\alpha_2}} \tag{32}$$

with α defined to be

$$\alpha = \frac{\alpha_1}{(\bar{q}_t - q + 1)^{\alpha_3}} \tag{33}$$

in which $\alpha_1 > 0$ and $\alpha_2 > 0$ are R&D productivity parameters similar to Equation 17, and $\alpha_3 > 0$ is a spillover parameter, capturing the differential difficulty of improving quality for laggards vs. leaders.

Together, Equations 31, 32, and 33 describe the stochastic quality progression of all existing manufacturers. The next section discuss manufacturer entry and exit dynamics, allowing us to characterize the law of motion for W_t, the industry structure.

3.2.2. Manufacturer entry and exit

To begin the discussion of entry and exit dynamics, it is useful to define $\mathring{q}_t \in \mathbb{N}$, the *obsolete quality* at time t, as the minimal value for which the inequality

$$\mathbb{E}\left[V\left(\mathbf{S}_t^+, \mathbf{X}_t, \mathring{q}_t + 1\right) \mid \mathbf{S}_t^-\right] \geq mc \tag{34}$$

still holds. A manufacturer of quality $\mathring{q}_t + 1$ can still garner a price higher than the marginal cost of production, but a manufacturer of quality $\mathring{q}_t$ can not, as the market clearing price for her ICs is lower than *mc*. We assume there is no benefit to the laggard from producing ICs at a loss (e.g., due to *learning by doing*), though the model does not preclude a case in which the leader finds it profitable to sell ICs at a loss, if that is a strategically beneficial behavior to her. Hence, any manufacturer with quality $q \leq \mathring{q}_t$ will not produce any ICs at time t, and we define them to have exited the industry with no scrap value.

New prospective manufacturers are born every period. A prospective manufacturer can pay an endogenous entry sum r_t^e, defined such that $\Theta\left(r_t^e, \bar{q}_t - \mathring{q}_t\right) = 0.5$, and with probability 0.5 become a manufacturer of type $\mathring{q}_t + 1$ at the next period. As in Ericson and Pakes (1995), we assume new entrants make and observe their entry decisions *sequentially* to insure the number of new entrants at period t is well-determined. We further assume new entrants make their entry decisions after observing the time t innovation outcomes of manufacturers. This entry mechanism is similar to assuming free entry into the ranks of manufacturers with the obsolete quality $\mathring{q}_t$, and requiring obsolete manufacturers to make a sufficiently large investment in R&D to return into the industry. Furthermore, this mechanism insures that $U\left(\mathbf{S}_{t+1}^{-}, \mathring{q}_t + 1\right)$ never deviates significantly from $2 \cdot r_t^e$.

We can now characterize the law of motion for W_t, the industry structure, as

$$W_{t+1}^{(q)} = \begin{cases} 0, & \text{if } q \leq \mathring{q}_t \\ l^- \cdot W_t^{(q)} + l^0, & \text{if } q = \mathring{q}_t + 1 \\ l^- \cdot W_t^{(q)} + l^+ \cdot W_t^{(q-1)}, & \text{if } q > \mathring{q}_t + 1 \end{cases} \tag{35}$$

in which l^0 captures the number of new entrants, l^- the share of quality q manufacturers who failed to innovate during period t, and l^+ the share of quality $(q-1)$ manufacturers who successfully innovated during period t to become quality q manufacturers in period $(t+1)$. Completing the formal characterization of l^0, l^-, l^+ is left for a later draft.

3.2.3. The miners

The miners' problem changes little between the monopoly and oligopoly industry structures. Hence, most changes described in this section are notational rather than substantial.

As in the monopoly case, we assume free entry into the "prospective miner" market. A new miner can choose whether to buy a single IC of quality q for the market clearing price $P_t^{(q)}$. The new miner's entry decision is again forward-looking, but now depends on his rational expectations as to *all* manufacturers' current and future productions and R&D investments. This includes manufacturers that have yet to enter the manufacturing market. A miner will purchase an IC with quality q if and only if

$$\mathbb{E}\left[V\left(\mathbf{S}_t^+, \mathbf{X}_t, q\right) \mid \mathbf{S}_t^-\right] - P_t^{(q)} \geq 0 \tag{36}$$

Again, due to the assumed free entry into the "prospective miner" market, Equation 20 holds with equality every period, for *every* offered quality q.

The miner's lifetime value function $V\left(\mathbf{S}_t^+, \mathbf{X}_t, q\right)$, the miner's per-period payoff $\Pi\left(\mathbf{S}_t^+, q\right)$, the equation defining the activation threshold q_t^*, and the miner's earning share function $\Phi\left(\mathbf{S}_t^+, q, q_t^*\right)$ do not change and are defined as in Equations 21, 22, 23, and 24, respectively. Note, however, that these are all functions of the current state of the economy, $\mathbf{S}_t^+$, which was redefined in Equation 28 to include W_t, the industry structure.

3.3. Equilibrium

Formal definition of the pure-strategy symmetric markov-perfect nash equilibrium is left for a later draft.

3.4. Model discussion

We begin the discussion by considering the position of miners. Miners are "economic peons" in the Bitcoin market, as highlighted by Equations 20, 36, and especially 21. Free entry

to the prospective miner market implies miners extract no consumer surplus when buying an IC. The IC price in equilibrium is at a level sufficient only to compensate them for their risk-adjusted cost of capital, as captured by β^M. Notably, while miners are "peons", they are not myopic. A rational miner needs to consider the dynamic problem facing manufacturers, as well as BTC price trends, and how these will affect the profitability of an IC he intends to purchase.

This result is a natural economic outcome, as the only scarce resource required for Bitcoin mining is ICs. It can also be seen in Equation 21, which contains no miner choice, except the trivial decision to activate or deactivate the IC in any period, as noted in Equation 22. The fact that miners capture no surplus led to the creation of so-called "egalitarian" crypto-currencies, with mining algorithms immune to being encapsulated in an IC. None of these crypto-currencies have been able to challenge the hegemony of Bitcoin.

The observation that miner are "peons" stipulates that all economic power resides with manufacturers. It is therefore not surprising that Bitmain - one of the leading IC manufacturers - which was started in 2012 as a kick-starter project, had profit from operations of over a Billion USD in 2017. One may not extrapolate this result, though, as the evident profitability of IC manufacturing for crypto-currency mining has attracted a considerable number of new entrants, as the model predicts.

A monopolistic miner's problem is simpler than that of an oligopolistic one. She needs to balance extracting value today, miners' rational expectations as to her value extraction tomorrow, and R&D expenditures that will allow her to depreciate all existing ICs and begin value extraction anew. If the monopolist manufacturer was able to commit to never manufacture ICs again, she could manufacture a single IC, and sell it for a price which captures all of the expected future value of the Bitcoin mining market. Limited commitment on the side of the monopolist manufacturer is hence critical for any dynamics to arise.

In the absence of commitment, the monopolist manufacturer must optimally choose how many ICs to sell every period as to balance her time-preference with her ability to extract

value in the next periods and her R&D schedule. When the monopolist is facing a satiated market, investment in R&D is more valuable as a successful innovation will effectively depreciate existing ICs. This is not physical depreciation, as ICs are long-lived, but technological depreciation - the old ICs are unable to compete with new ICs due to their inferior computational efficiency. Following a technological advancement, the monopolist manufacturer will reduce R&D expenditure and harvest the economic value of the recent superior technology.

An oligopolistic manufacturer has to account for the actions of other manufacturers, which changes the industry dynamics considerably. A technological leader can no longer slowly skim the industry, and must sell ICs quickly, before a laggard manufacturer innovates. Miners rationally predict this behavior, and are willing to pay less for an IC of high quality than in the monopoly case. Laggards are also driven to sell ICs more quickly, before the leader depreciates the market with her superior sales. Laggards are also forced to innovate quickly, lest they fall sufficiently behind the leader to become obsolete.

Because ICs "print money", the Bitcoin mining industry's dynamics are purely driven by R&D. Brand value, marketing efforts, consumer awareness - these traditional drivers of demand have little importance in this market. Notably, IC manufacturers to-date are also "fabless", meaning they do not manufacture their own ICs. They only conduct the R&D process of designing a lithography blueprint of the IC and outsource production to major manufacturers (e.g., TSMC). Operational efficiency is therefore not a major factor of industry dynamics. This clean connection between R&D and market structure is partly why we find the Bitcoin mining industry to be useful in uncovering the relation between innovation, including R&D spillovers, and competition.

Further discussion of the oligopoly model dynamics is left for a later draft.

4. Estimation

We now present an empirical estimation of the model, using data on Bitcoin dynamics gathered directly from the blockchain, and hand-collected data on the release dates and technical features of Bitcoin mining ICs. As a first step, we make two adjustments to the presented model, to facilitate estimation. We then describe the calibration of parameters that are observable or can be credibly calibrated with sufficient accuracy. The remaining, unobservable parameters, are estimated using the method of simulated moments - an indirect-inference, minimum-distance estimation method that matches moments of distributions in simulations of the model to observed moments in the data.

Prior to estimation, we must choose a time interval for model analysis, i.e., the time between the beginning of period t and period $t+1$. While BTC price data are available at high-frequency, our IC-related data are available only at monthly frequency. Conversely, choosing a longer time period - such as quarterly or yearly analysis - is problematic due to the short history of Bitcoin. We therefore conduct all analysis at a monthly frequency.

4.1. Model adjustments

To estimate the model numerically, we need to make two adjustments to it. The first adjustment is an immaterial translation from absolute to relative qualities, as in Goettler and Gordon (2011). This adjustment constrains the cardinality of the quality indexed vectors $(Q_t, W_t, C_t, R_t, P_t)$. Note that the entry process described in Section 3.2.2 implies there is no need to track the quantity of obsolete ICs (those with quality $q \leq \mathring{q}_t$). Hence, the only ICs we need to track are those with $q \in [\mathring{q}_t + 1, \bar{q}]$, where $\bar{q}$ is the frontier quality. Equations 24 and 33 are the only ones which directly use quality levels, and are both robust to shifting all qualities by the same factor. We therefore shift all qualities such that $\mathring{q}_t = 0$ for all t.

The second adjustment is material but necessary. Recall that the number of BTC awarded per block halves every $210,000$ blocks. This discrete drop in miner rewards causes disconti-

nuities in the miner's value function defined in Equation 21 (i.e., it becomes a step function in the variable f_t) due to horizon effects. These discontinuities also cause the manufacturer's value function defined in Equation 29 to be discontinuous, and stymie the numerical stability of the recursive value-function-iteration process required to solve the model.

We solve this numerical problem by modifying the model to include a smooth decrease in the number of BTC awarded per block, rather than a discrete drop, while maintaining the total number of BTC awarded. In practice, this means we assume the block reward for the genesis block was 70 BTC, and it declines monthly by 1.43 percent. While this adjustment somewhat modifies the incentives faced by miners, its economic significance is limited due to the quick obsolescence of ICs observed in practice. As described below, the median "lifetime" of an IC, from being at the technology frontier ($q = \bar{q}$) to being obsolete ($q = \mathring{q}$) is less than 1.5 years. This means that miners have very little time-discounting effects and we can use the adjusted model, with minor changes, to achieve good predictive accuracy even when nearing a "halvening" event.

Assuming a smooth decrease in awarded BTC has a second numerical benefit - we are now required to track one less state variable. Equations 22 and 23 are the only ones that directly use the BTC float f_t, and they only use it in conjunction with the BTC price b_t to calculate total miner reward per period. It is therefore sufficient to track the dynamics of miner reward per period, rather than the separate dynamics of BTC price and BTC float.

4.2. Calibrated parameters

The first parameters we calibrate are the ones pertaining to the per-period miner reward dynamics. These parameters embed the BTC price dynamics and the smooth decrease in BTC awarded per-block, as discussed in Section 4.1. Recall that Equation 10 defined the law of motion for (log) BTC price as

$$\log(b_{t+1}) = \rho_b \log(b_t) + \sigma_b \epsilon_t$$

with $\epsilon_t \sim \mathbf{N}(0,1)$, and parameters $\rho_b > 0, \sigma_b > 0$. We estimate this equation using monthly BTC price data, obtained from http://blockchain.info. We then apply the smooth BTC-award decrease to it. The uncorrected auto-regressive coefficient for $log(b_t)$ is $\rho_b = 1.01$, indicating an average 1% price appreciation per *month*, across Bitcoin's lifetime. The corrected coefficient is $\rho_b = 1.004$, indicating that mining rewards appreciate by an average 0.4 percent per month. The average monthly price-volatility is 17.6 log-percentage points. While appreciating mining rewards fit the observed data, we also test the model with lower ρ_b values, as this exponential growth trend seems difficult to maintain.

The assumption that all ICs have equal electricity cost ec is a normalization which sets the relative scale of an IC. It is also a realistic assumption, because ICs require standardized power supply modules. These power modules are generally available as standard 500 Watt units. We set ec to a value of 30 USD, which is the cost to operate a 500 Watt unit continuously for a month, including cooling overhead. This amount assumes electricity cost of 0.07 USD/KwH, as per Taylor (2017).

The cost of producing a single 500 Watt "mining rig", which is in practice composed of dozens of ICs, is taken from industry estimates. It includes the fabrication cost of the IC (photo-lithography of the IC onto a silicone wafer), the packaging of ICs into an integrated-circuit board, and the cost of a power supply unit. We verify the industry estimates by manually checking what are the minimum prices for which manufacturers sold ICs when these ICs were close to obsolescence, as our model predicts this value should be close to mc. Following these investigations, we set mc to a value of 400 USD. Note that we assume mc does not change for new generations of ICs, for three reasons. First, this is the observed pattern in industry estimates of mining IC marginal costs. Second, this is the pattern in near-obsolescence manufacturer prices. Third, this has been the general microelectronics industry trend throughout its existence.

Our baseline calibration of the quality improvement factor δ is informed by hand-collected data on mining ICs used in practice for Bitcoin mining. The Bitcoin community is excep-

tionally tech-savvy, and has diligently documented the technical details of each IC, as those pertain to its profit-generating ability. Table 2 presents those technical details for the major ICs used in the Bitcoin mining market. The details we collect are: the manufacturer; release date; energy efficiency (Joules/GH); capacity (GH/sec); and energy consumption (Watts). The general pattern is of a doubling of energy efficiency, our core measure of quality, between succeeding quality generations. We hence set $\delta = log(2)$, because our quality scale is logarithmic.

Finally, the time discount parameters, β^{IC} and β^{M} are both set at 0.98. Recall that the analysis is monthly, so this represents a risk-adjusted discount rate of roughly 2 percent per month. Different reasonable parametrizations of the time-discount parameters change little in our empirical analysis, because the rate of technological depreciation is so high. Put differently, in a standard $r + \delta$ user-cost analysis, the discount rate r is eclipsed by the depreciation parameter δ. Panel (a) of Table 3 summarizes the values of our calibrated parameters.

4.3. Estimated parameters

The key parameters characterizing the manufacturer's R&D policy are unobservable. As discussed above, these parameters in essence determine the industry structure for the Bitcoin mining market, because of Bitcoin's unique features. We obtain estimates of these parameters using indirect inference (Gourieroux, Monfort, and Renault, 1993; Gourieroux and Monfort, 1996). In particular, we use the method of simulated moments to extract key moments from data generated by model simulations. We compare these moments to observed moments in the data. We repeat this process with different model parameters $\Gamma \equiv< \alpha_1, \alpha_2, \alpha_3 >$ such as to minimize the distance between the simulated and observed moments, for some distance metric. We use the inverse of the variance-covariance matrix of data moments as our distance metric, as it is the efficient GMM weighting matrix.

For any candidate set of parameters Γ in our estimation routine, we solve the model

numerically via value-function-iterations on the appropriate Bellman equations. We obtain value and policy functions for the monopolist manufacturer as well as miners of all tracked qualities. We use the manufacturer policy function and the miner entry and activation rules to simulate industry progression over a panel of width S and length T. We start each panel s with observed data in January 2013 as the initial condition, where $T = 66$ is equal to the number of periods in our sample. We then compare the moments of the generated simulated data with moments of observed data.

Identification relies on the chosen moments being sensitive to the parameters of the model. Though all parameters are jointly identified using all moments, we outline several prominent moments and their economic intuition in establishing parameter estimates. The observed frequency of innovation by the technological leader is key in identifying the R&D efficiency parameter α_1. The observed frequency of innovation by laggards is able to identify the R&D spillover parameter α_3. We use Bitcoin price changes as exogenous variation in manufacturer income to identify the elasticity of R&D success probability w.r.t. R&D expenditure, α_2.

It is important to note that the current draft only includes estimation results from the simplified monopoly model. This is in conflict with the observed data generating process (i.e., reality), in which the industry is a competitive oligopoly. Hence, the estimated parameters, as presented in Panel (b) of Table 3, are inherently biased. The nature of the bias and the difference between the monopoly and oligopoly estimates is of importance to our discussion of competition and innovation in Section 5.

A BETTER DISCUSSION OF PARAMETER ESTIMATION AND MATCHED MOMENTS IS LEFT FOR A LATER DRAFT.

4.4. Tests of model estimation

We present two tests of the predictive ability of the model. Both tests require the model to replicate the observed network hash rate and innovation rate during the 2013-2018 time period. We begin the test on January 2013 because the first IC - Avalon 1 - was introduced on

February 2013. In the first test, the model is provided with both the BTC price path and the observed timing of major innovations during the period. It is also provided with data on the network hash-rate on January 2013. We then simulate the progression of the market vintage structure, which evolves based on the manufacturer's policies, and the ensuing network hash rate. Figure 8 presents the observed and simulated time paths from this analysis. We can see the model tracks the observed data well, across more than 5 years and more than 5 orders of magnitude increase in hash rate.

In our second test, we simulate the model *without* providing it the timing of innovations - i.e., we allow for endogenous innovation. We simulate the model 100 times from the 2013 initial market condition and using only the observed price path. The mean predicted network hash rate path across 100 simulations, as well as its 95% confidence bound and the observed network hash rate are presented in Figure 9. It is striking that the model can predict an equilibrium outcome, 5 years ahead, with excellent accuracy. We surmise that the Bitcoin mechanism design lends itself to significant predictability.

5. Competition and innovation

Discussing the impact of product-market competition on industry innovation is left for a later draft.

6. Conclusion

In this paper, we first analyze predictability properties of Bitcoin's mechanism design. We then use them to construct a dynamic model of the Bitcoin mining market, featuring a durable-goods IC manufacturing oligopoly with endogenous innovation and R&D spillovers. We estimate the model using data directly from the Bitcoin blockchain, as well as hand-collected data on the technical specifications and release dates of mining ICs.

We show the model is able to predict policy-relevant aggregate values such as energy

consumption of the Bitcoin network and R&D expenditure by IC manufacturers, for a given price-path. Notably, we show the model is able to predict salient feature of the Bitcoin mining market - network hash rate - several years ahead, given only a price-path, with significant accuracy. The model is also able to quantify the value of an IC given market-conditions - a much sought-after capability among practitioners. While we only analyze the Bitcoin mining market, the predictability component of our paper is applicable to other crypto-currencies using a proof-of-work mechanism, which include six of the top ten crypto-currencies by market capitalization.

We describe why the setting of Bitcoin's mining market is uniquely suited to bear on the question of how product-market competition affects industry innovation. The fact that mining ICs are essentially "money printing" simplifies the analysis. More importantly, it implies commonplace non-R&D drivers of product-market competition are significantly less important in this market - facilitating a clean analysis of R&D's impact on outcome variables of interest.

One of our goals is to highlights unique economic features of crypt-currency, such as the "tragedy of the commons" inherent in proof-of-work mining, and the negative network effects, in which the selling of new ICs *reduces* the value of existing ICs. Such features can be used as natural experiments to examine economic questions that are not directly related to the nature of crypto-currencies. This is especially true in the face of a surge in crypto-currency varieties, and the range of economic mechanisms they employ, even if some of those crypto-currencies are economically ill-conceived.

References

Aghion, P., Bloom, N., Blundell, R., Griffith, R., Howitt, P., 2005. Competition and Innovation: An Inverted-U Relationship. Quarterly Journal of Economics 120, 701–728.

Arrow, K., 1962. Economic Welfare and the Allocation of Resources for Invention. In: *The Rate and Direction of Inventive Activity: Economic and Social Factors*, UMI, vol. I, pp. 609–626.

Athey, S., Parashkevov, I., Sarukkai, V., Xia, J., 2016. Bitcoin Pricing, Adoption, and Usage: Theory and Evidence. mimeo .

Biais, B., Bisière, C., Bouvard, M., Casamatta, C., 2017. The blockchain folk theorem. mimeo .

Blundell, R., Griffith, R., Van Reenen, J., 1999. Market share, market value and innovation in a panel of British manufacturing firms. Review of Economic Studies 66, 529–554.

Cheah, E.-T., Fry, J., 2015. Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. Economics Letters 130, 32–36.

Clark, C. W., 1990. Mathematical Bioeconomics: The Optimal Management of Renewable Resources. Wiley, New-York, NY, second ed.

Dwork, C., Naor, M., 1993. Pricing via Processing or Combatting Junk Mail. Advances in Cryptology CRYPTO' 92 pp. 139–147.

Ericson, R., Pakes, A., 1995. Markov-perfect industry dynamics: A framework for empirical work. Review of Economic Studies 62, 53–82.

Ferguson, N., Schneier, B., 2003. Practical Cryptography. Wiley, New-York, NY.

Gilbert, R. J., 2006. Competition and innovation. Journal of Industrial Organization Education 1, 1–23.

Goettler, R. L., Gordon, B. R., 2011. Does AMD Spur Intel to Innovate More? Journal of Political Economy 119, 1141–1200.

Gourieroux, C., Monfort, A., Renault, E., 1993. Indirect inference. Journal of Applied Econometrics 8, 85–118.

Gourieroux, C. A., Monfort, A., 1996. Simulation Based Econometric Methods. Oxford University Press, Oxford, UK.

Griliches, Z., 1998. The Search for R&D Spillovers. In: *R&D and Productivity: The Econometric Evidence*, University of Chicago Press, pp. 251–268.

Hardin, G., 1968. The Tragedy of the Commons. Science 162, 1243–1248.

Harvey, C. R., 2014. Cryptofinance. mimeo .

Huberman, G., Leshno, J. D., Moallemi, C., 2017. Monopoly without a monopolist: An Economic analysis of the bitcoin payment system. mimeo .

Kocherlakota, N. R., 1998. Money Is Memory. Journal of Economic Theory 81, 232–251.

Liu, X., Lan, J., Shenoy, P., Ramaritham, K., 2006. Consistency maintenance in dynamic peer-to-peer overlay networks. Computer Networks 50, 859–876.

Nadarajah, S., Chu, J., 2017. On the inefficiency of Bitcoin. Economics Letters 150, 6–9.

Nakamoto, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. Www.Bitcoin.Org .

Nicolas, H., 2014. The Economics of Bitcoin Transaction Fees. mimeo .

O'Dwyer, K. J., Malone, D., 2014. Bitcoin Mining and its Energy Footprint. Proceedings of the 25th IET Irish Signals & Systems Conference pp. 280–285.

Pakes, A., McGuire, P., 1994. Computing Markov-perfect Nash equilibria: numerical implications of a dynamic differentiated product model. RAND Journal of Economics 25, 555–589.

Pakes, A., McGuire, P., 2001. Stochastic algorithms, symmetric Markov perfect equilibrium, and the 'curse' of dimensionality. Econometrica 69, 1261–1281.

Scherer, F. M., 1967. Market Structure and the Employment of Scientists and Engineers. American Economic Review 57, 524–531.

Schumpeter, J. A., 1942. Capitalism, Socialism and Democracy. Routledge, London.

Taylor, M. B., 2017. The Evolution of Bitcoin Hardware. Computer 9, 58–66.

Urquhart, A., 2016. The inefficiency of Bitcoin. Economics Letters 148, 80–82.

Vries, A. D., 2018. Bitcoin's Growing Energy Problem. Joule 2, 801–809.

Yermack, D., 2015. Is Bitcoin a Real Currency? an Economic Appraisal. In: Chuen, D. L. K. (ed.), *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*, Academic Press, pp. 31–44.

Yermack, D., 2017. Corporate governance and blockchains. Review of Finance 21, 7–31.

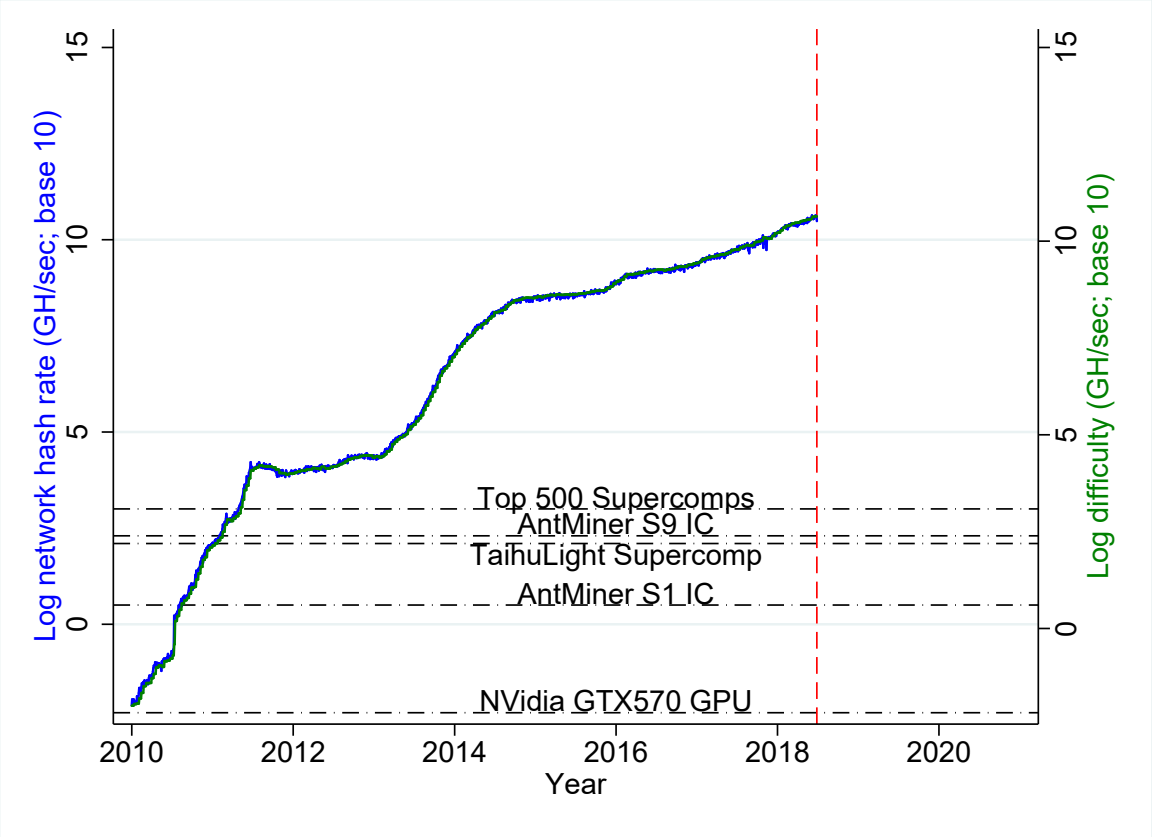


Fig. 1. Hash rate and difficulty. The left axis of this figure presents the base-10 log of Bitcoin's network hash rate over time. The right axis presents the base-10 log of mining difficulty. Both values are in terms of 10^9 hashes per second (GH/sec). The vertical line marks July 1^{st} 2018. The horizontal lines represent the current hashing power of relevant computing platforms. Data are from the Bitcoin blockchain.

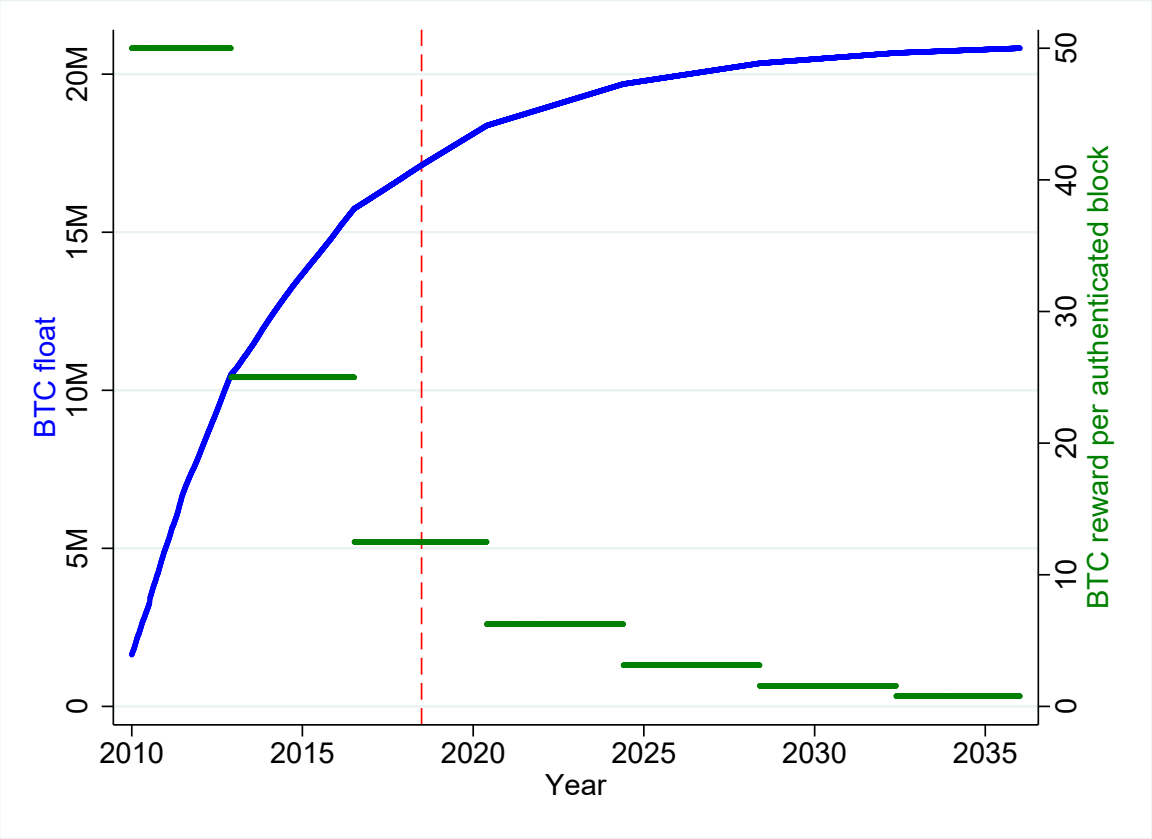


Fig. 2. BTC float and block reward. The left axis of this figure presents the actual and predicted BTC float (number of BTC in circulation) over time. The right axis presents the actual and predicted BTC reward per authenticated block. The vertical line marks July 1^{st} 2018. Data on actual values are from the Bitcoin blockchain. Future predictions are based on Equations 5 and 6.

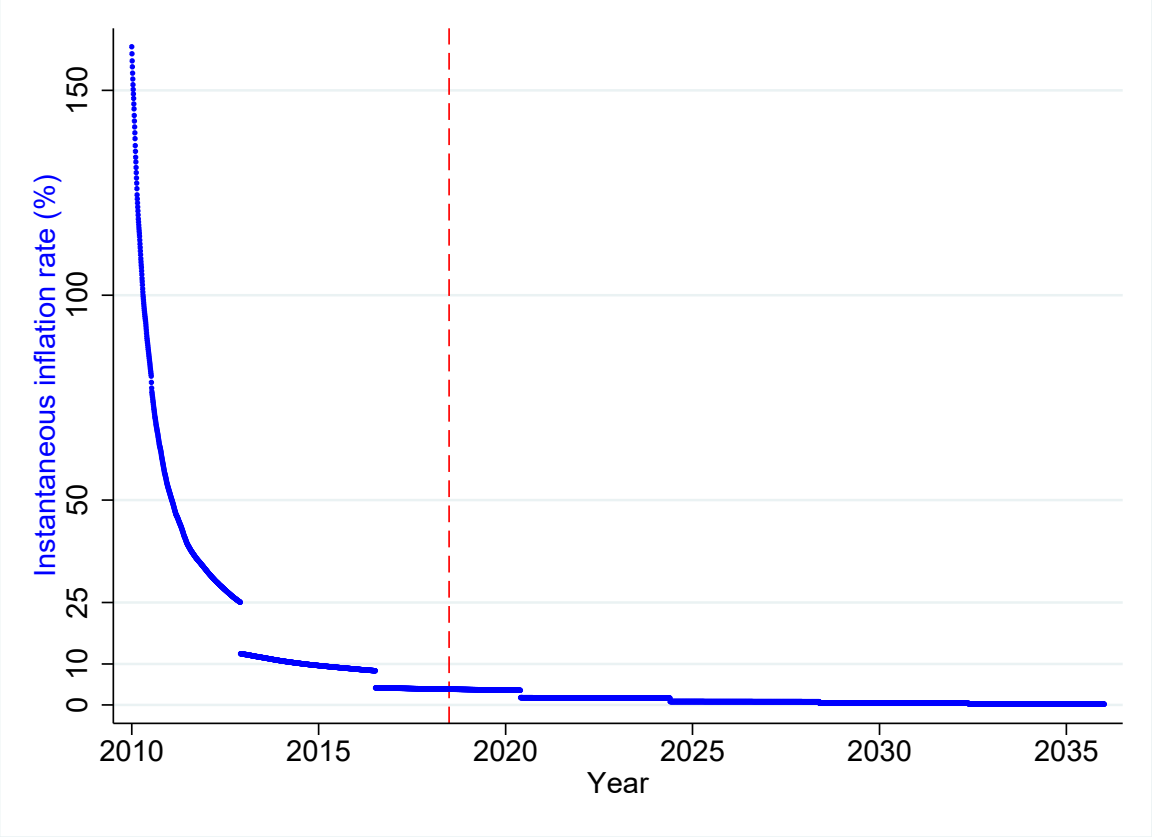


Fig. 3. Bitcoin inflation rate. This figure presents the instantaneous inflation rate, in yearly percentage terms, of the Bitcoin network. The vertical line marks July 1^{st} 2018. Data on actual values are from the Bitcoin blockchain. Future predictions are based on Equation 7.

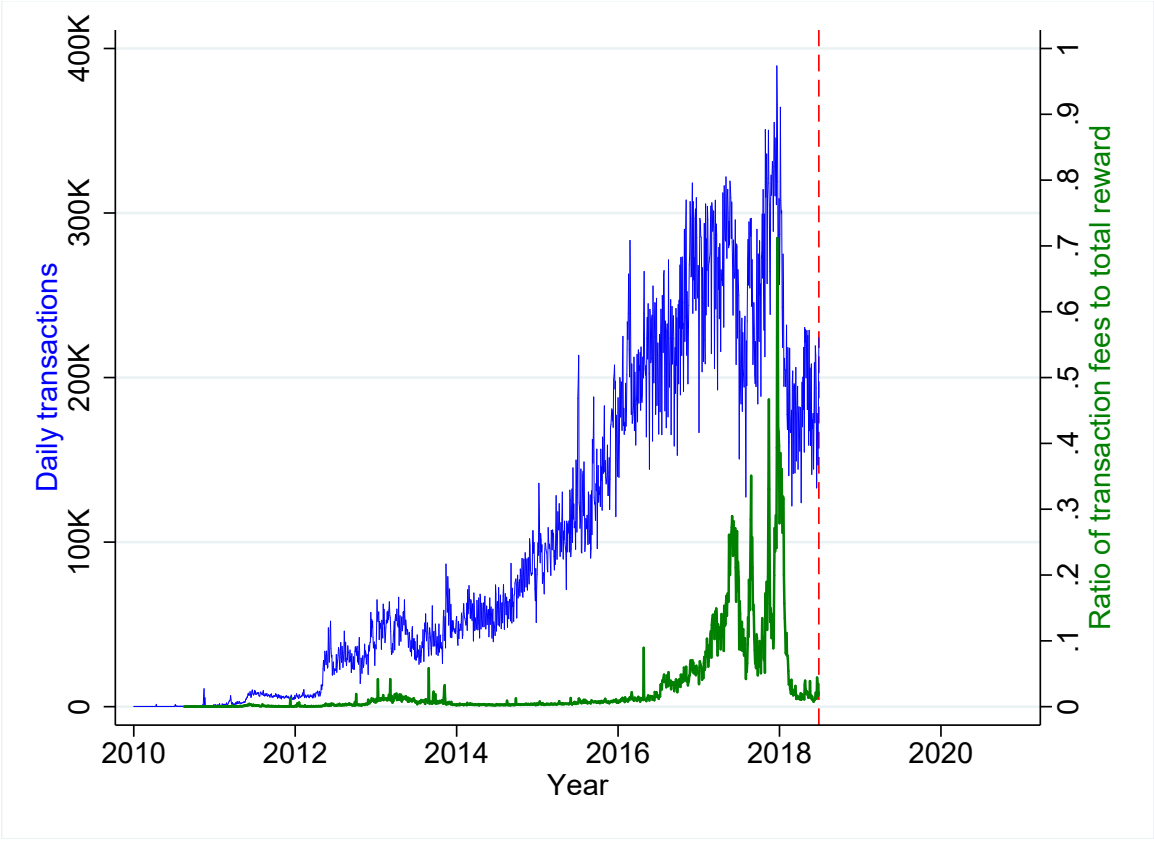
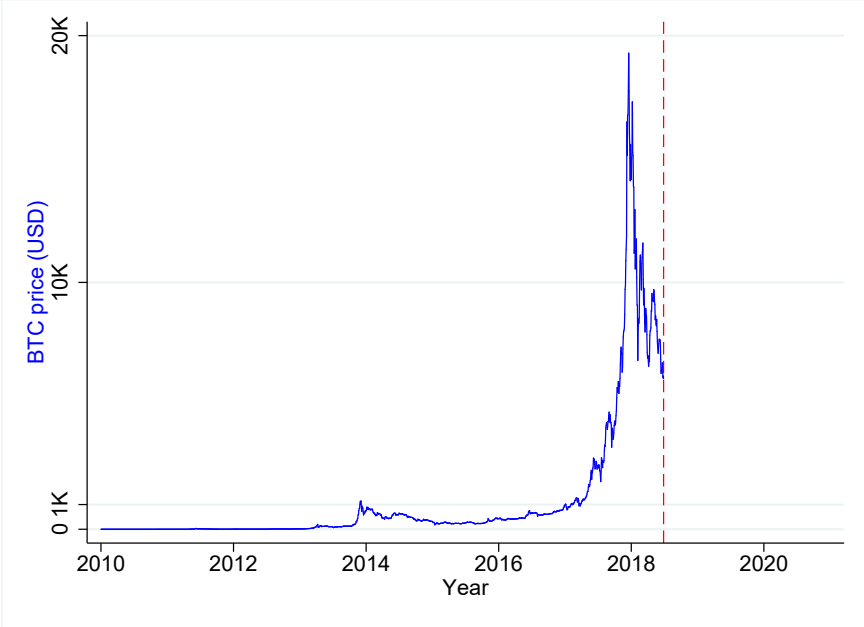


Fig. 4. Transaction rate and fee ratio. The left axis of this figure presents the number of daily transactions on the Bitcoin network. The right axis presents the ratio of transaction fees paid to miners out of total miner compensation (i.e., transaction fees + block rewards). The vertical line marks July 1^{st} 2018. Data are from the Bitcoin blockchain.

Panel (a)



Panel(b)

Fig. 5. BTC price. Panel (a) of this figure presents the base-10 log of BTC market price, in USD terms. Panel (b) presents the raw BTC market price, also in USD terms. The vertical lines mark July 1^{st} 2018. Price data are from http://blockchain.info.

Fig. 6. BTC market capitalization. This figure presents the market capitalization of the outstanding BTC float, in USD terms. The vertical line marks July 1^{st} 2018. Price data are from http://blockchain.info, and BTC float data are from the Bitcoin blockchain. The horizontal lines are the July 2018 values of the US M2 monetary aggregate (top) and the world's gold reserves (bottom).

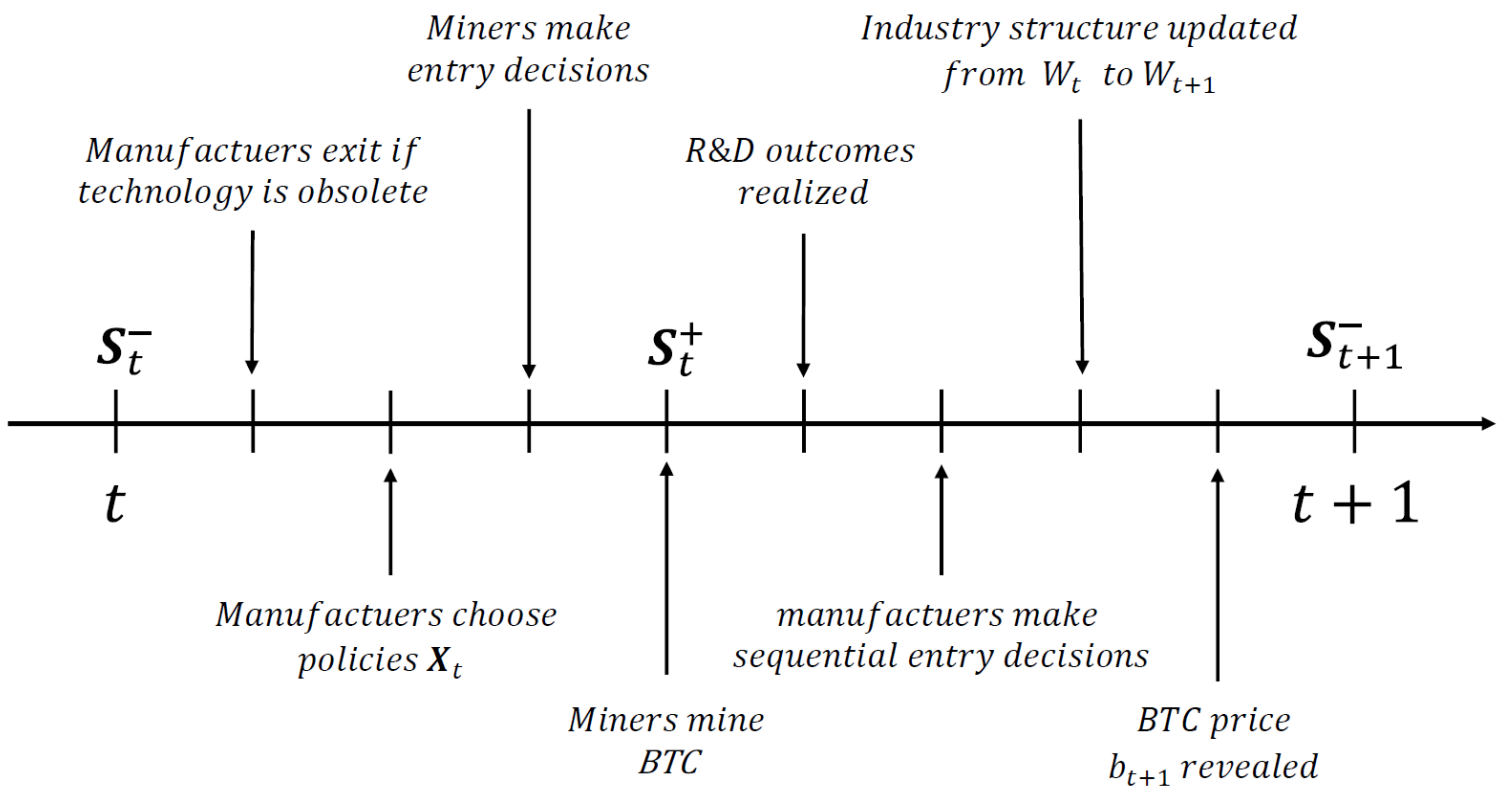


Fig. 7. Within-period model timing. This figure presents the within-period timing of actions by miners and manufacturers in the model. It further depicts the information available to agents at every decision point.

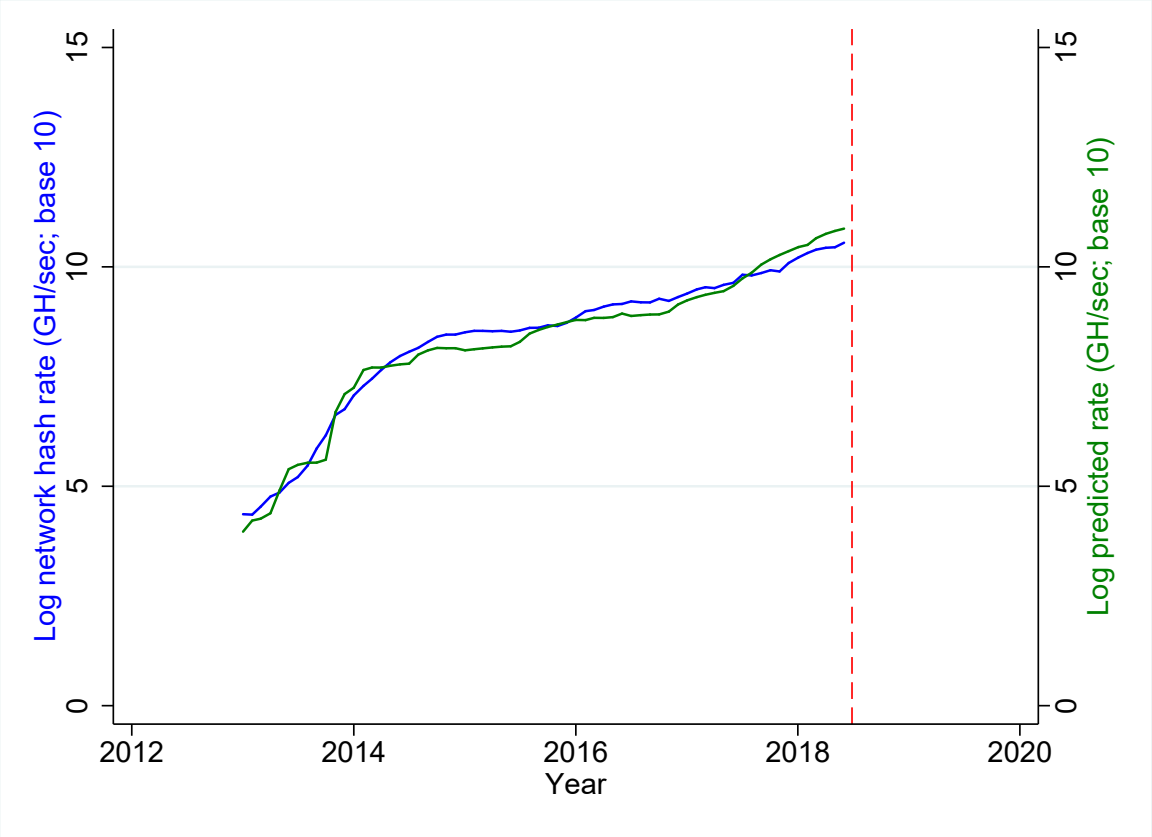


Fig. 8. Predictability with exogenously given innovation. The left axis of this figure presents the base-10 log of the Bitcoin network hash rate, since introduction of the first mining IC - Avalon 1 - on February 2013. The right axis presents base-10 log simulated (predicted) hash rate from the estimated model. The simulation was provided actual data on BTC price path and the timing of IC innovations, but not on the quality or amount of IC sold by manufacturers. The vertical line marks July 1^{st} 2018. Data on network hash rate are from the Bitcoin blockchain; data on BTC price path are from http://blockchain.info; data on the timing of IC innovations were hand-collected by the authors.

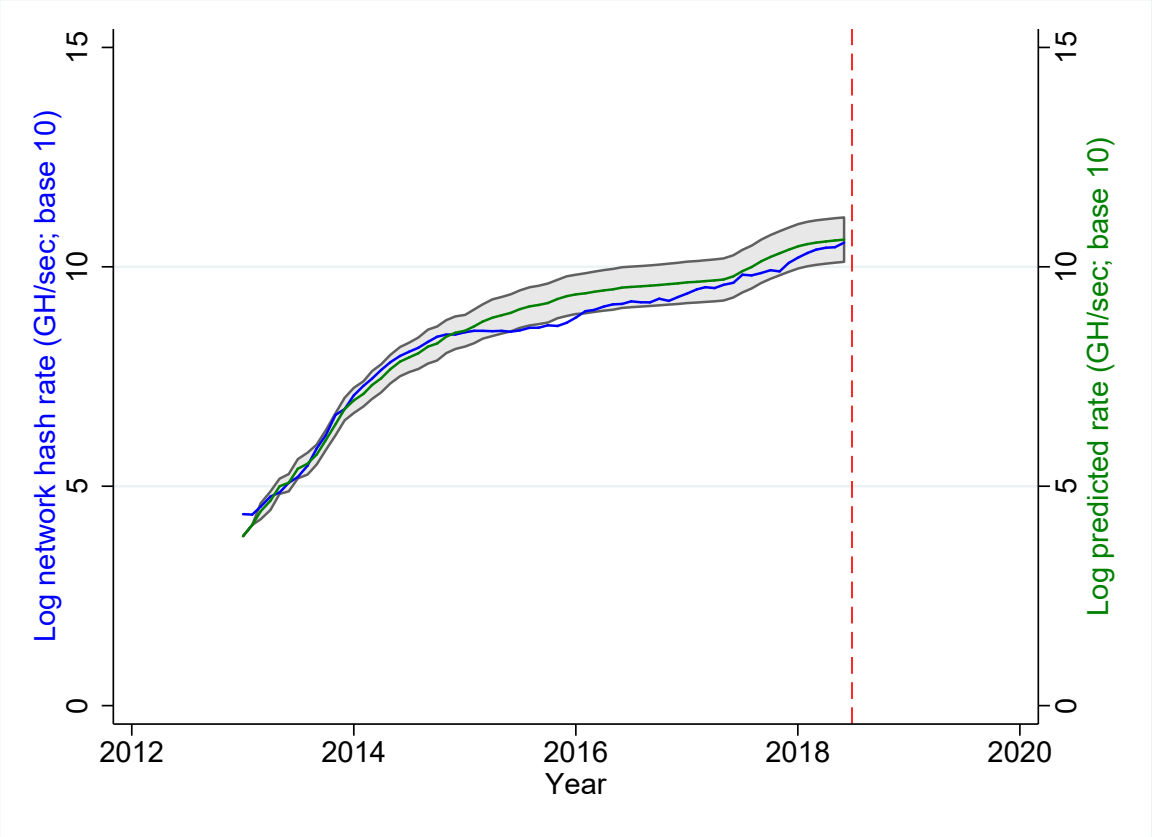


Fig. 9. Predictability with endogenous innovation. The left axis of this figure presents the base-10 log of the Bitcoin network hash rate, since introduction of the first mining IC - Avalon 1 - on February 2013. The right axis presents the mean, across 100 repetitions, of the base-10 log simulated (predicted) hash rate from the estimated model, along with its 95% confidence interval. The simulation was provided actual data on BTC price path, but *not* on the timing of IC innovations or the quality and amount of IC sold by manufacturers. The vertical line marks July 1^{st} 2018. Data on network hash rate are from the Bitcoin blockchain; data on BTC price path are from http://blockchain.info.

Table 1
Summary of notation used

This table defines the notation used in the paper, with the exception of model parameters, which are defined in Table 3.

State variables	
Q_t	Market structure vector whose q^{th} element denotes the number of ICs with quality q that exist in the market during period t
W_t	Industry structure vector whose q^{th} element denotes the number of IC manufacturers of quality q that exist during period t
P_t	Prcie vector whose q^{th} element denotes the price charged for a single IC with quality q during period t
f_t	BTC float during period t
b_t	BTC price, in USD terms, in period t
$\mathbf{S}_t^-$	Tuple $< Q_{t-1}, W_t, f_t, b_t >$ with time t state variables measured before miner entry
$\mathbf{S}_t^+$	Tuple $< Q_t, W_t, f_t, b_t >$ with time t state variables measured after miner entry
Actions	
C_t	Production vector whose q^{th} element denotes the amount of ICs produced by one IC manufacturer of quality q
R_t	R&D vector whose q^{th} element denotes the R&D spending by one IC manufacturer of quality q
$\mathbf{X}_t$	Tuple $\langle C_t, R_t \rangle$ with manufacturer production and R&D decision vectors
Functions	
$V\left(\mathbf{S}_t^+, \mathbf{X}_t, q\right)$	Miner lifetime value for an IC of quality q
$\Pi\left(\mathbf{S}_t^+, q\right)$	Miner per-period profit function for an IC of quality q
$\Phi\left(\mathbf{S}_t^+, q, q_t^*\right)$	Share of aggregate income accruing to an IC of quality q when the cutoff quality level is q_t^*
$U\left(\mathbf{S}_t^-, q\right)$	Manufacturer lifetime value for a manufacturer with quality q
$\Omega\left(\mathbf{S}_t^-, \mathbf{X}_t^{(q)}, q\right)$	Per-period profit of a manufacturer with quality q
$\Theta\left(R_t^{(q)}, \bar{q}_t - q\right)$	Probability of a successful innovation for a manufacturer with quality q and R&D spending $R_t^{(q)}$ when the technological frontier is $\bar{q}$

Table 2
Technical specification of Bitcoin mining ICs

This table presents hand-collected release dates and technical specifications for key Bitcoin mining ICs during our sample period.

Manuf.	IC Name	Release Date	J/GH	GH/sec	Watts
Avalon					
	A3256 (Avalon 1)	February 2013	6.6	0.295	1.95
	A3255 (Avalon 2)	August 2013	2	1.5	3
	A3233 (Avalon 3)	April 2014	0.86	7	6
	A3222 (Avalon 4)	September 2014	0.5	25	12.5
	A3218 (Avalon 6)	September 2015	0.285	35	10
	A3212 (Avalon 721)	September 2016	0.12	83	10
	A3210 (Avalon 841)	January 2018	0.095	130	12.3
BitFury Group					
	BF756C55	June 2013	0.8	2.7	2.16
	BF864C55	April 2014	0.5	3.8	1.9
	BF 28nm	April 2015	0.2	10	2
	BF8162C16	December 2015	0.11	100	11
Bitmain Technologies Ltd.					
	BM1380 (S1)	Novemeber 2013	1	2	2
	BM1382 (S3)	April 2014	0.6	16.5	10
	BM1384 (S5)	September 2014	0.35	15.1	5.36
	BM1385 (S7)	August 2015	0.22	32.5	7.12
	BM1387 (S9)	June 2016	0.12	71.4	8.5
ASICMiner					
	BE100	April 2013	6	0.34	2
	BE200	July 2014	0.66	9.6	6.34
	AM0815	February 2018	0.055	200	11

Table 3
Model parameters

This table defines the model parameters, and presents their baseline estimated/calibrated values.

Parameter	Baseline	Description
Panel (a) - Calibrated parameters		
ρ_b	1.004	$AR(1)$ coefficient of miner reward
σ_b	0.176	Standard deviation of innovation to BTC price
mc	400	Marginal cost of manufacturing an IC, in USD terms
ec	30	Per-period electricity cost of an active IC, in USD terms
δ	2	Factor by which computational efficiency increases when innovation occurs
β^{IC}	0.98	Manufacturer risk-adjusted time-discounting factor
β^{M}	0.98	Miner risk-adjusted time-discounting factor
Panel (b) - Estimated parameters		
α_1	5.2E+7	R&D baseline productivity
α_2	0.5	R&D productivity elasticity
α_3	na	R&D spillover elasticity

Appendix A. Bitcoin mechanism design

This appendix introduces the mechanics of the Bitcoin protocol and mechanism design. It builds on concepts from computer network design and cryptography, but does not assume prior knowledge in these fields.

A.1. The ledger

At its core, the Bitcoin network is simply a distributed ledger - a type of public database that is shared, replicated, and synchronized among the members of a peer-to-peer computer network[1]. The distributed ledger records transactions among network participants, thus keeping track of asset ownership. Every participant in the network has a unique identifier, known as an *address*. One person or entity may create and use many addresses. An example address may be $1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa$[2]. The ledger, hence, is just a list of valid transactions between addresses, essentially stating "on date *ddd* at time *ttt* address *xxx* transferred *bbb* BTC to address *yyy*".

The validity of the transaction is verified before it is posted, by making sure that on date *ddd* at time *ttt*, address *xxx* did indeed have *bbb* BTC, and that the (anonymous) owner of address *xxx* is really the one who issued the transfer order. Authenticating the transaction issuer is achieved using a cryptographic mechanism known as *public-key cryptography*[3]. When a Bitcoin address is created, the owner of the address also creates two *keys*, a public-key, which is disseminated widely, and a private-key, which is known only to the owner. When the owner issues a transaction request, they "sign" it by attaching an encrypted version of the transaction request, using their private-key for the encryption process. An important feature of public-key cryptography is that anyone holding the public-key can decipher this signature and verify it matches the transaction it is signing. Because only the account owner has the

[1] A peer-to-peer network is composed of equally privileged, equipotent participants. This is in contrast to a *client-server* network, in which clients request services and resources from centralized servers.

[2] Also known as "The Genesis Address", this is the address created by Satoshi Nakamoto, to which the first 50 BTC ever created were deposited.

[3] See Ferguson and Schneier (2003) for further details.

private-key, this signature verifies they are indeed the ones who issued the transaction.

A few features of Bitcoin implied by this use of public-key cryptography are worth noting. First, an address's keys are mathematically related to the address and cannot be changed. Second, anyone holding the private-key for an address is considered the address owner, so if a private-key is revealed/stolen, the holder of the private-key can transfer all BTC from that address to another (i.e., BTC is a bearer asset, not registered asset).[4] Third, if an address's private-key is permanently lost (say, due to a computer hard-drive crash), all the BTC in that address are "lost" and can never be used again. By some estimates[5], 10-25 percent of the final BTC supply of 21 million have already been lost, mostly in Bitcoin's early years.

A.2. Ledger consistency

The peer-to-peer distributed architecture is one of the core design principles of Bitcoin. There is no central entity that all participants need to implicitly or explicitly trust. This architectural choice is not without price, though. With a peer-to-peer distributed system, even if all peers are trustworthy, problems of consensus and consistency arise[6]. This is even more so in an untrusted peer-to-peer network in which one can not necessarily trust information from any other peer. An example will be useful.

Suppose at time t_0, the network is consistent, i.e., all peers agree on the current allocation (ownership) of all BTC. Suppose that at the t_0 allocation, address *xxx* owns one BTC. At $t_1 > t_0$, two transactions are issued on the network: one in which address *xxx* transfers one BTC to address *yyy*, and one in which address *xxx* transfers one BTC to address *zzz*. Importantly, the first transaction is issued to a Bitcoin peer in the UK, while the second one is issued to a Bitcoin peer in Australia[7]. Both peers verify that address *xxx* does indeed own 1 BTC, and that the transfer order is properly signed by the owner of the *xxx* address. They

[4] A famous example is Bloomberg TV's Matt Miller who flashed the private-key of his Bitcoin address to the camera. A viewer then extracted the funds to his own address, posted the details on reddit.com, and offered to return the BTC to a new address Miller will provide, which he did.

[5] See https://letstalkbitcoin.com/blog/post/rise-of-the-zombie-bitcoins for discussion.

[6] See, e.g., Liu, Lan, Shenoy, and Ramaritham (2006).

[7] Specifically, Melbourne, the best city in the world.

then propagate information regarding this transactions to their closest peers, so those peers can update their ledgers as well. This process repeats at the peers receiving the transactions, which update their ledgers and propagate the message forward. At some point a Bitcoin peer, which received the UK transaction first and already updated its ledger, will receive the Australian transaction and reject it, because the address *xxx* does not have sufficient funds. Conversely, the UK transaction will be rejected by all the Bitcoin peers which received the Australian transaction first. Following this process, the Bitcoin network is no longer consistent, because the peers of the network do not agree on the current allocation of BTC.

Solving this consistency problem and the accompanying so-called *double-spending* problem is the key innovation described by Nakamoto (2008). To resolve it, they introduce the concept of a chain of authenticated blocks, or *blockchain*, protected by a *proof-of-work* mechanism. Following the exposition by Harvey (2014), it will be useful to introduce the concept of a *hash-function* before discussing these two mechanisms.

A.3. Hash functions

Hash-functions are a class of mathematical functions with many uses in the field of cryptography. The key properties of a good cryptographic hash-function are:

1. Size - it maps data d of arbitrary size to data h of fixed size
2. Determinism - the same input data d always hash to the same hash value h
3. One-way - given a hash value h it is difficult to find data d s.t. $h = hash(d)$
4. No-collision - it is difficult to find two different data d_1 and d_2 s.t. $hash(d_1) = hash(d_2)$

Informally, these properties mean that a malicious adversary cannot replace or modify the input data d without changing its hash. Thus, if two data strings d_1 and d_2 have the same hash, one can be very confident that d_1 is identical to d_2. Additionally, these properties mean that hash functions are highly sensitive to changes in their input, i.e. a single bit change in the input data from d_1 to d_2 should cause the hash h_2 to be nearly independent of h_1.

The specific cryptographic hash function used by Bitcoin is called SHA-256. SHA stands for "Secure Hash Algorithm", and 256 indicates that the resulting hash is 256 bit long (or 64 hexadecimal letters, as they are usually presented). Table A.1, adapted from Harvey (2014), presents the SHA-256 hash of three different strings: "Hello, world!0", "Hello, world!1", and "Hello, world!4250"[8]. Note the four leading zeros at the beginning of the third hash - they will play an important role below.

A.4. The blockchain

So how do hash-functions, blockchains, and proofs-of-work come together to solve the double-spending problem? The key idea is packing many individual BTC transactions into a *block*. A block is just a collection of transactions which are valid at the time of block creation, and are internally consistent[9]. A useful metaphor is that the entire Bitcoin ledger is a book, and a block is just a single page of that book. The block further contains a "header", with useful data on the block, such as the block number and other useful fields, and crucially, the hash of the previous block in the ledger (a hash of the previous "page").

By requiring each block to contain a hash of the previous block, and due to the special properties of cryptographic hash-functions, a unique chain of blocks (the *blockchain*) is created. Block N includes the hash of block $N-1$, and block $N-1$ includes the hash of block $N-2$, and so on until block number 1 (known in Bitcoin parlance as "The genesis block"). Any change to the contents of any of the blocks between 1 and N (such as tampering with or altering any transactions contained in these blocks) will cause the hash of block N to change, which will be observable by all peers. By further introducing a way of ensuring there is a single chain of blocks that all peers agree on, the Bitcoin protocol solves the problem of consistency and double-spending. In effect, it makes the blockchain *immutable*.

To ensure the existence of a single chain that all peers agree on, Nakamoto (2008) suggests

[8]You are encouraged to verify these are indeed the hash values by conducting the hash yourself at http://xorbin.com/tools/sha256-hash-calculator.

[9]Because all the transactions that form a block are available to the block creator at the time of block creation, they can verify that there are no double spending instances within that block.

using a cryptographic mechanism known as proof-of-work, initially proposed by Dwork and Naor (1993) to combat junk mail. The key idea is requiring some form of *verifiable costly signature* of each block before it is accepted to the chain, while rewarding the signing of such blocks with newly minted BTC. By making signing a block costly, and simultaneously making it rewarding, Bitcoin creates a competitive market for block signing services (or "mining", due to the newly granted BTC). An outcome of this market design is the emergence of a single chain everyone agrees on. Signing blocks is costly because a block's hash is required to be lower than some widely known value, which amounts to requiring the hash have a pre-determined number of leading zeros. The only way to find a hash that abides by this requirement is by investing much computing power.

An important outcome of the blockchain mechanism design described above is the *longest chain rule*. This rule implies all peers in the network work towards extending (adding blocks to) the longest[10] chain of blocks known to them. Note the longest chain rule is an unenforceable guideline, but Biais et al. (2017) discuss why abiding by the longest chain rule is the optimal choice for all peers, in the sense of being a Markov perfect equilibrium of the induced dynamic stochastic game.

An example of the longest chain rule is presented in Figure A.1. Assume a peer knows of a blockchain of length 80, and receives a message regarding an authenticated new block (denoted 81a) from one of its neighbors, quickly followed by messages from other peers regarding two other new blocks, 81b and 81c, all of which extend block number 80. Figure 1(a) presents the state of the peer at this point. The peer goes ahead and verifies each of the new blocks, by making sure all transactions within it are valid and the hash at the top of the block is identical to the hash of block 80. Finally, the peer verifies each block has the appropriate proof-of-work, by hashing it with the provided nonce and verifying the number of leading zeros in the hash is appropriate. If all tests pass, the peer accepts the blocks as valid candidates, and starts working towards extending the chain past block 81a, which is

[10] A fine point is that "length" is calculated not by the number of blocks on the chain but by the total verifiable work invested in signing the entire chain.

the first block (of the three) it received. Figure 1(b) presents the state of the peer trying to extend the chain past block 81a.

After a few minutes, the peer receives a message regarding block 82b. Block 82b is a valid block that extends block 81b, which is a valid block the peer is aware of but is not trying to extend. Because 82b is a valid extension of 81b, and 81b is a valid extension of 80, the longest chain the peer is now aware of is 82 blocks long, whereas the current chain he is working towards extending (the one ending with block 81a) is only 81 blocks long. The longest chain rule means the peer ceases their attempt to extend 81a and starts trying to extend 82b. Figure 1(c) presents the state of the peer after that change.

A.5. Majority rules and a 51% attack

You may have noticed in Figure A.1 that from our peer's perspective, the longest chain changed from being $1, ..., 80, 81a$ to being $1, ..., 80, 81b, 82b$. What of the transactions which were encapsulated in block $81a$? Most of them are likely to also be contained in blocks $81b$ and $82b$, but such a requirement is not enforceable. Hence, some transactions which were once accepted as part of the longest chain may cease to be part of the chain, effectively disappearing. We conclude the introductory discussion of Bitcoin mechanics by considering how this issue is resolved.

To understand the issue, it is useful to consider an example of what a rogue peer could do if it could create a longer chain of blocks than the rest of the network[11]. Figure A.2 presents how such an ability would allow them to unwind a transaction by:

1. Creating two payments with the same bitcoins: one to an online retailer, the other to themselves (another address they control).
2. Only broadcast the payment to the retailer.
3. When the payment gets added in an honest block, the retailer sends them goods.

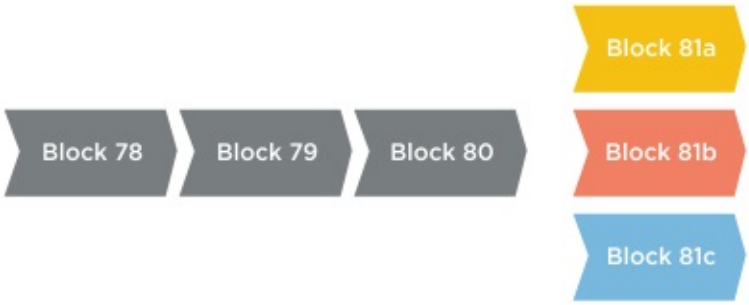
[11]This part follows closely the exposition in https://bitsonblocks.net/2015/09/21/a-gentle-introduction-to-bitcoin-mining/

4. Secretly create a longer chain of blocks which swaps out the payment to the retailer, and swaps in the payment to themselves.
5. Publish the longer chain. If the other nodes are playing by the longest chain rule, they will ignore the honest block with the retailer payment, and continue to build on the longer (dishonest) chain. The honest block is said to be orphaned and does not exist to all intents and purposes.
6. The original payment to the retailer will be deemed invalid by the honest nodes because those BTC have already been spent (in the longer dishonest chain).

Importantly, this attack vector is only valid if the rogue peer could create a longer chain of blocks than the rest of the network could. Because creating blocks is expensive, the rogue peer will only be able to sustain faster block creation if it controls more than 51% of the aggregate mining computing power, giving this attack its name - "the 51% attack"[12]. If honest nodes control more than 51% of computing power, only the last few block of a chain are in any risk of changing. Hence, one could simply wait until sufficiently many blocks are added to the chain past the block containing one's transaction. This will make sure the block one's transaction appears in is "baked-in" to the blockchain and will not be replaced. That is indeed the path advocated by Nakamoto (2008). Current rules-of-thumb recommend waiting at least 6 blocks (or an hour on average, because blocks are authenticated every 10 minutes) before assuming a transaction is permanently part of the chain[13].

[12]This follows from Bitcoin's underlying philosophy, which considers the "truth" to be whatever 51% of computing power says it is.

[13]There are 10 documented cases of 2 block unwinding, one of 3 block unwinding, and none of longer unwindings.



(a) Step 1

(b) Step 2

(c) Step 3

Fig. A.1. Longest chain rule. This figure presents an example of using the longest chain rule. Panel (a) is the state of the Bitcoin peer after receiving notice of three new blocks extending block 80. Panel (b) describes the peer mining on the longest chain known to it. Panel (c) describes orphaning of block 81a because of a new longest chain, containing 81b and 82b. Source: http://bitsonblocks.net/

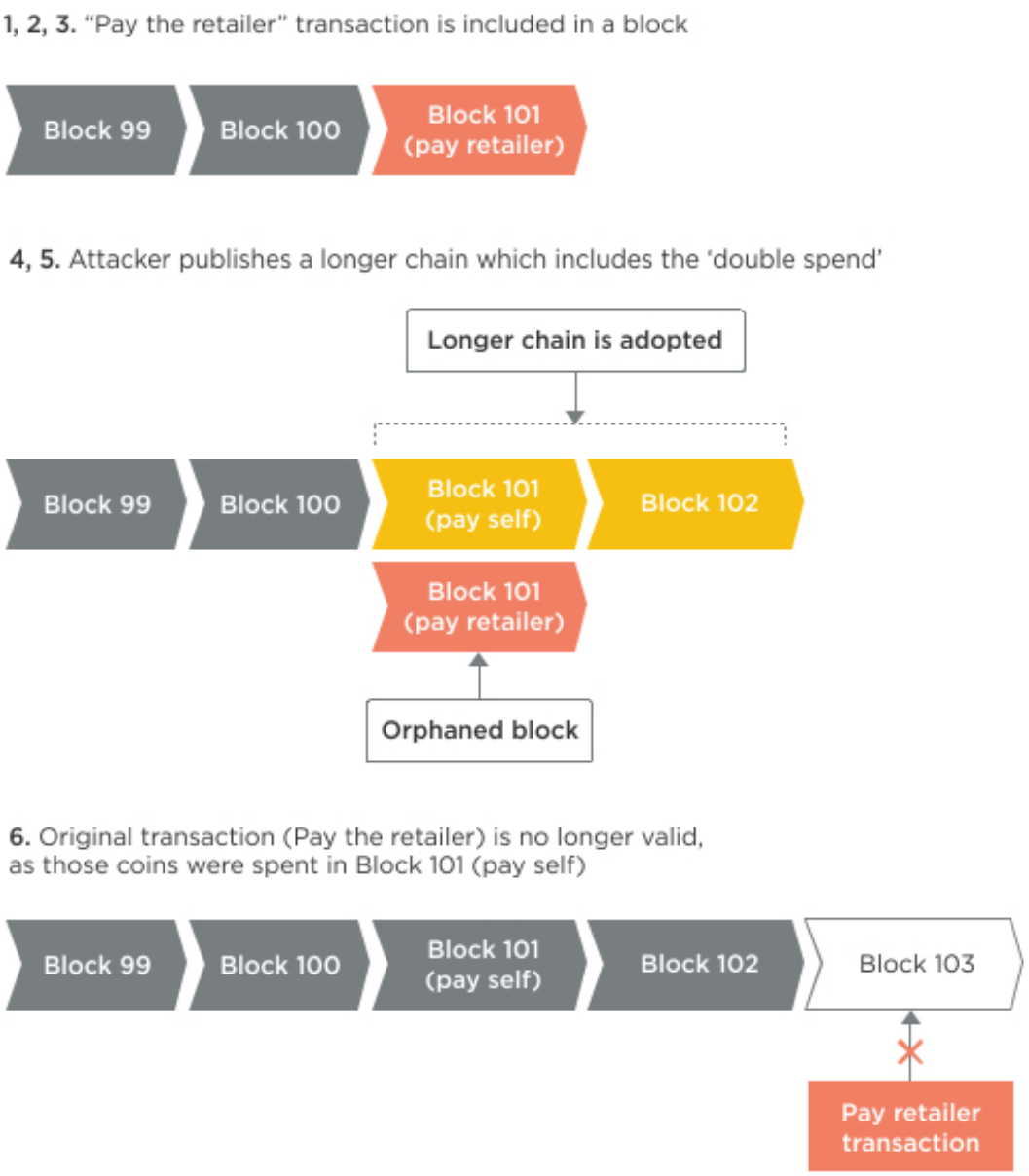


Fig. A.2. A 51% attack. This figure presents an example of a 51% attack. The attacker pays an online retailer, but then orphans the payment block and introduces a newer longest-chain in which the payment is to themselves. Source: http://bitsonblocks.net/

Table A.1
Hash function examples

This table presents the results of calculating the SHA-256 cryptographic hash value for the string "Hello, world!", concatenated with three different numerical values - 0, 1, and 4250. Source: Harvey (2014).

String	SHA-256 hash	Leading zeros
Hello world!0	`1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64`	0
Hello world!1	`e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8`	0
Hello world!4250	`0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9`	4