Miner Collusion and the Bitcoin Protocol

Alfred Lehar *
Haskayne School of Business
University of Calgary

Christine A. Parlour†
Haas School of Business
UC Berkeley

October 6, 2022

Abstract

Bitcoin users can offer fees to the miners who record transactions on the blockchain. We document the blockchain rarely runs at capacity, even though there appears to be excess demand and higher fee orders are not always prioritized. We show this is inconsistent with competitive mining, but is consistent with miners exercising market power. If users believe that only high fee transactions will be executed expeditiously then we show how strategic capacity management can be used to increase fee revenue. Using a novel data set, we present evidence consistent with strategic capacity management. We show that mining pools facilitate collusion, and estimate that they have extracted least 300 million USD a year in excess fees by making processing capacity artificially scarce.

Keywords: Decentralized Finance, Transaction Costs, Implicit Collusion

*email:alfred.lehar@haskayne.ucalgary.ca.

†email:parlour@berkeley.edu. We thank Christina Atanasova, Bruno Biais, Lin William Cong, Jaisun Li, Ye Li, Andreas Park, Fahad Saleh, Asani Sarkar, Shu Yan, Peter Zimmerman, and seminar participants at Carnegie Mellon, CBER, Cowles Foundation Economics of Cryptocurrencies 2021, SF BlockChain Week, IWF-SAS, Philadelphia Fed FinTech Conference, Toulouse School of Economics, WFA 2020, EFA 2020, the China FinTech Research Conference, 2021, INFORMS 2020, the University of Graz, P2P Financial Systems 2020, Toronto FinTech Conference 2020, 3rd UWA Blockchain, Cryptocurrency and FinTech conference 2020, 4th SAFE Market Microstructure Conference 2020. Lehar is grateful to the Canadian Securities Institute Research Foundation, CFI/JELF, and SSHRC under grant 435-2021-0605 for financial support.

1 Introduction

Bitcoin is the first successful proof of concept of a decentralized financial product. The original white paper by Nakamoto presented a system that obviates the need for trusted intermediaries because "competitive miners" record and settle all Bitcoin transactions. The code assigns Bitcoin as an incentive for miners to do this. The first observed departure from zero direct cost settlement was that, spontaneously over time, consumers added fees to their transactions to encourage speedy settlement. In this paper, we shed light on the determinants of Bitcoin transaction fees and provide evidence that decentralized miners act in a way consistent with fee maximization, as a monopolist intermediary might.

Our analysis is based on three data sources. Our core data comprises 685,406,634 processed transactions and the fees associated with them, gleaned from the blockchain. These data span the earliest days of Bitcoin to November 9, 2021. In addition, we have two data sources on transaction demand including aggregated mempool data and a shorter but more detailed mempool data set. We believe that to date, this is the most comprehensive Bitcoin transaction data set that has been analyzed.

Following the Industrial Organization literature, we proceed in three steps. First, we compare mined blocks to outcomes that are predicted under competitive mining. Second, we compare mined blocks to outcomes consistent with the exercise of market power through strategic capacity management. Finally, we provide a simple quantification of the loss of consumer surplus from this strategic behavior. We find it is close to 1.6 billion USD or approximately 300 million USD a year since the advent of mining pools. These costs have been borne by the Bitcoin users.

We establish that a competitive miner maximizing revenue in each block will optimally fill each block to capacity (if there is demand) and will optimally choose those transactions with the highest fees attached. We compare these predictions to the actual history of Bitcoin transactions. Contrary to predictions from competitive mining, we document that the Bitcoin blockchain rarely operates at full capacity. Indeed, there is no single day in which the blockchain has run at full capacity, even though there appears to be excess demand for transaction processing in the mempool. We further show that low fee transactions are frequently processed into a block even though higher fee transactions are waiting in the mempool. The foregone revenue from mining blocks with excess capacity ("money left on the table") appears to be large.

Market power under the Bitcoin protocol can only be exercised in a novel way. Unlike standard goods markets, Bitcoin miners cannot set higher fees, as these are solely determined by users of the system. They also cannot increase their share of mined blocks, because the rate at which blocks are produced is essentially determined by a random number generator which keeps the arrival rate of blocks constant. (We delve into the mechanics of the process in more detail in the body of the paper.) The only variable over which miners have direct control at a high frequency is the choice of transactions that they incorporate into a block, and how they use block capacity.

To show how strategic capacity management can lead to higher miner revenue, we present a simple example. The intuition is straightforward: suppose that all users believe that transactions with high fees are processed immediately, but transactions with low fees will always be delayed for multiple blocks. Thus, arriving agents effectively face a menu that relates fees to waiting times.

Given this menu, high valuation users with a demand for immediacy will always submit high fees. For a random arrival of users, credibly implementing such a menu means that sometimes blocks will be processed with spare capacity (to enforce waiting times) and some transactions will be processed even if transactions with higher fees attached to them remain in waiting. Both of these predictions are consistent with our data.

In our sample of over 685 million observations, we document that users who are more likely to have higher valuations for consummated transactions are more likely to pay a higher fee, which is consistent with rent extraction. Specifically, transactions that are more likely to originate from institutional sources (proxied through day of the week), transactions that are more likely to be arbitrage trades (proxied by the Kimchi premium), transactions that involve gambling sites and exchanges and transactions associated with rapid redeployment pay higher fees.

The mechanics of the Bitcoin system allows us to provide more evidence. First, blocks are deemed mined when a random number with specific characteristics is found. As this is a computational exercise, the length of time between blocks is random. Using this feature, we demonstrate that the probability of mining an empty block is higher if the previous blocks arrived at a higher frequency and were fuller. We note that the probability of mining an empty block is not affected by the size of waiting transactions. Second, valid transactions are broadcast to the network and stored in mempools awaiting processing. Using our detailed mempool data set, we document price priority violations in which higher fee transactions are left waiting in the mempool while lower fee transactions are processed. Thus miners do not process transactions based on a simple priority rule, but a combination of price and waiting time consistent with a menu.

To understand how strategic capacity management can be maintained, we characterize a miner's intertemporal profit function. We show that a miner's relative hash rate (effectively the probability that they are randomly chosen to mine the next block) affects the weight they put on the profitability of future blocks and the extent to which their actions affect current users of the system. Intuitively, miners that are more likely to mine a block, care more about the profitability of future blocks and also have a bigger effect on how the overall system currently performs. In short, higher relative hash rates capture miners' incentive and ability to affect profitability. Our empirical proxy for large relative hash rates is the Hirschmann-Herfindahl index. In all our regressions, we find that interaction terms are significant. Under competitive mining this variable should not affect observables.

We document that the supply of mining and in particular mining concentration (in the form of mining pools) affects transaction fees. To further investigate the effect of mining pools, we find that fee dispersion is higher when mining capacity is more concentrated and more mining is done by pools. Consistent with strategic capacity management, blocks tend to be fuller after periods of low mining output and more empty after periods of high mining output. This effect is more pronounced the higher mining concentration.

A simple way to implicitly enforce collusion is to punish deviators. The consensus protocol means that there can be forks that are subsequently abandoned. These are called "orphan blocks." The miner who produced the orphan block loses the valuable coinbase (the fixed reward for mining a block). In a hand collected sample of orphan blocks, we document that these are more likely to have been mined by pools that fill blocks to capacity. This is consistent with a punishment meted out to miners deviating from collusive strategy.[1]

Finally, we have access to two natural experiments. First, the Xinjiang coal mine disaster, as it lead to a shut-down of miners, lead to an increase in the relative hash rates of the unaffected miners, but had no plausible effect on demand from users. We find that after this disaster, high value users paid higher fees consistent with more effective strategic capacity management. We also consider the Silk Road closure. This corresponds to a demand shift and a decrease in high value users. We find that after this there is no significant change in capacity usage, but we observe lower fee dispersion consistent with lower revenue extraction.

There is a recent literature that uses blockchain data to test theories of bitcoin value determination. For example, Pagnotta (2021) considers the equilibrium tradeoff in the Bitcoin system between prices and security, while Biais, Bisiére, Bouvard, Casamatta, and Menkveld (2020) provides a rational model and estimation of value.

The literature on transaction fees in blockchain systems is small: Easley, O'Hara, and Basu (2019) explain the observed shift from no-fee to fee paying transactions and model the interactions of fee payments and waiting times. While the focus of their empirical analysis is on the time series of average transaction fees our paper documents a huge variation of Bitcoin transaction within blocks analyses the cross section of transaction fees. Huberman, Leshno, and Moallemi (2017) compare Bitcoin to a traditional payment system and derive closed form solutions for equilibrium fees. In their framework, the chain runs at full capacity.

This research on fees fits into a larger body of literature that focuses on the economics and incentives in blockchain ecosystems (among others Abadi and Brunnermeier (2018), Cong and He (2019), Budish (2018)) and the impact on financial markets (e.g. Malinova and Park (2017) or Brauneis, Mestel, Riordan, and Theissen (2018)). Cong, He, and Li (2019) analyze the incentives for miners to form pools to tradeoff risky mining against the amount pools charge to miners. Other research focuses on the pricing of crypto-currencies in the market including frictions causing pricing differences (e.g. Hu, Parlour, and Rajan (2018), Makarov and Schoar (2018), Choi, Lehar, and Stauffer (2018)).

A literature in Industrial Organization investigates non-competitive behavior in markets. While collusive behavior is difficult to detect empirically, deviations from a competitive benchmark are easier to observe. This is a large literature, but examples include Kawai and Nakabayashi (2022) who consider rebidding in Japanese procurement auctions which is inconsistent with competitive behavior, while Bajari and Ye (2003) consider construction firms in the midwest. Deviations from a competitive behnchmark in the financial markets were also observed by Christie and Schultz (1994).

Strategic manipulation of capacity to extract rents has also been analyzed in the industrial organization literature. For example, Gilbert and Klemperer (2000) show that in markets for which consumers have to make a fixed investment to enter a market, rationing may induce more entry and thus be profitable, while Denicolò and Garella (1999) show that rationing may allow a durable good monopolist to maintain high prices. Similarly, in the operations research literature, rationing has been shown to be optimal to induce consumers to accelerate purchases (Liu and

[1]We thank Bruno Biais for suggesting this test.

van Ryzin (2008)) and to convince consumers to ascribe a higher value to the good (see for example Debo, Parlour, and Rajan (2011) and Debo, Rajan, and Veeraraghavan (2020)).

An interesting contemporaneous theory paper, Malik, Aseri, Singh, and Srinivasan (2019) considers consumers who decide between processing payments with Bitcoin or going to the banking system. They provide conditions under which increasing Bitcoin capacity leads large miners to collude tacitly to undo such increases by only partially filling blocks; this crowds out low value payments who prefer to use the banking system. They further show that providing incentives for miners to operate at full capacity increases system security risk which could reduce value. We too consider how miners strategically manipulate capacity but our focus is on whether the blockchain system is competitive. Our predictions and findings on the within block dispersion of fees are inconsistent with their framework.

It is important to our framework that servicers can choose which transactions to include. The lack of commitment in the protocol is the starting point for Basu, Easley, O'Hara, and Sirer (2019). The authors observe that over time the prices for the same service have fluctuated, and argue that there is no dominant strategy equilibrium. They propose a mechanism that is manipulation proof as the number of users and miners increases.

Finally, although we are the first paper to document strategic capacity management in the Bitcoin system, there is a computer science literature on miner extractable value, which is another way in which miners exert market power. Briefly, this literature documents that miners on the Ethereum blockchain systematically front run arbitrage trades. This further supports the idea that miners are profit maximizing and seek revenue from all aspects of the mining process. The seminal paper in this literature is Daian, Goldfeder, Kell, Li, Zhao, Bentov, Breidenbach, and Juels (2019), while recent finance contributions include Lehar and Parlour (2022) and Capponi, Jia, and Wang (2021).

2 The Bitcoin Protocol and Data

The Bitcoin system comprises a network of transaction processors (typically called miners). A transaction is initiated when a user announces or broadcasts it to the network. Then, each processor determines if the transaction is valid (i.e., the Bitcoin have not been spent elsewhere) and stores it. Each processor keeps an independent inventory of valid potential transactions called the "mempool."

A miner creates a block and appends it to previous ones by choosing valid transactions from their own mempool and being the first to solve a computational puzzle on that data. This process is referred to as "proof of work." A feature of the proof of work protocol is that solving the puzzle and creating a block only depends on a miner's relative computing power (called a hash rate). Once a block is mined, it is broadcast to the network and all processors update their own mempools. While each mempool is specific to each processor, empirical results from computer science indicate that mempools across processors are effectively the same.[2]

[2]See for example, Dae-Yong, Meryam, and Hongtaek (2020).

Miners who produce a valid block are compensated in two ways. First, they automatically receive a per block reward called the coinbase. This was initially set at ₿50 and is programatically cut in half every 210,000 blocks (roughly every four years). Second, and more germane to our analysis, users who want their transaction to be included in a block can offer fees to miners. A Bitcoin transaction comprises inputs and outputs. Fees are offered implicitly as the difference between these inputs and outputs. For example, a submitted transaction might call ₿2.2 as an input but only assign ₿2.18 as output. Miners retain the difference (₿0.02) and pay it to themselves if they successfully mine the block.

Our sample comprises all blocks from the Genesis block (January 3, 2009) to block number 708,957 (November 9, 2021) and includes 1,749,089,936 inputs and 1,873,784,894 outputs. In total we have 685,406,634 transactions. For each block we observe the coinbase, the inputs and outputs (and hence fees paid to the miners), and the "size" of each transaction.

While bytes is the usual metric to describe data size, some nuances of the Bitcoin protocol mean that it is more useful to describe the "weight" of transactions. Unless otherwise stated, in the rest of the paper, when we refer to size or block capacity, we work with weights which reflect true capacity utilization. (In Appendix C we provide further institutional details on the measurement of block capacity and data size.) It is important to note that there is a technological limit on the total data size of transactions that can be processed into one block. Thus, block capacity should be a binding constraint in the presence of high demand.

From the blockchain we observe precisely how much capacity was used in each block. However, because of the distributed structure of mempools, we observe information about demand for this capacity imprecisely. We have obtained two sets of mempool data to measure transaction demand. First, we have partially aggregated mempool data that provides information on the number and size of transactions grouped into 45 fee buckets. These data are snapshots taken every minute from Dec 16, 2016 to the end of our sample. Second, we set up two nodes and collected a shorter sample from block 620,591 to block 708,957 or from March 7, 2020 to the end of our sample. In the latter data set, we observe precisely when each transaction was broadcast to the network and entered our mempool, its weight, the fee, any dependencies on other unmined transactions, and if and when it was eventually mined. We also collect information on the weight, time, and transaction count of the block in which each transaction was mined. Details of both data sets appear in Appendix D.

Some of the control variables that we use in our analyses are blockchain specific. We describe them in detail here and present summary statistics in Table 1. We show how they affect fees in Appendix E.

Recipient wallet owners that spend their funds very quickly have demonstrated an immediate need for funds. We define rest time as the minimum time (in blocks) until the first output of a transaction is spent again. Similarly, we use a dummy variable for transactions for which an output is spent within one block.

Other transactions are used to insert data into the blockchain; such transactions have no obvious need for speed. We use a dummy variable for these data insertion transactions. These so-called "Op-Ret" transactions are used to store data from 2nd layer applications on the Bitcoin

	Observations	Mean	Std.Dev.	Median	Min	Max
Fee (thsd. Satoshi)	685,406,634	37	2,031	11	0	29,124,090
Fee (USD)	685,406,634	3.4	38	.33	0	135,576
Inputs (BTC)	685,406,634	9.8	332	.083	0	550,000
Inputs (USD)	685,406,634	80,569	6,149,485	290	0	6,051,963,392
Blocksize (weight)	685,406,634	3,501,619	1,011,423	3,992,992	704	4,000,000
Tx-Size (weight)	685,406,634	1,897	8,289	904	248	3,998,628
Data insertion	685,406,634	.072	.26	0	0	1
Resttime (blocks)	679,884,420	694	6,387	5	0	650,022
Spent next block	685,406,634	.13	.33	0	0	1
₿Price (USD)	685,406,634	10,506	14,742	6,433	0	68,387

Table 1. Summary statistics of raw data *Fee (Satoshi)* is the transaction fee in thousand Satoshi. One bitcoin is 100 million Satoshi. *Fee (USD)* is the transaction fee in USD, *Inputs (thsd. Sat.)* is the sum of input values for the transaction measured in thousands of Satoshi excluding coinbase transactions, *Inputs (USD)* is the sum of input values for the transaction measured in USD excluding coinbase transactions, *Blocksize* is the size of the block measured in weight units, *Tx-Size* is the size of the transaction measured in weight units, *Data insertion* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Resttime* is the average time (measured in blocks) until transaction outputs are re-spent, *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, and *₿ Price (USD)* is the Bitcoin price in USD.

blockchain.

We also use variables that are related to both the capacity of the block and the size of the transactions. Transaction size is the physical size of all inputs and outputs in bytes or weight units. The transaction size represents an opportunity cost for miners as block space is limited. Blocksize is the absolute size of the block in weight units.

The average value of Bitcoin transaction was for ₿9.81 (one Bitcoin equals 100 million Satoshi) while the largest transaction was for ₿550,000.00 on Nov 16, 2011 at a zero fee.[3] The largest transaction in dollar terms occurred on Feb 19^{th}, 2021 when ₿108,010.99 valued at over USD 6.05 billion changed wallet for a fee of 67,600 Satoshi or USD 37.88.[4] In our sample there are 366,288 transactions with a value of more than USD 10 million. Of those, 127,338 were processed with a fee of less than USD 5, while the average fee for those transactions was USD 39.31.

Most transactions have a small data size, as the mean is 1,897.40 weight units while the median is 904.00 weight units. However, the largest transaction in our sample consumes the entire block 364,292 and has a size of 3,998,628 weight units.

We identified 181 transactions with fees over USD 10,000. These transactions have an average input of 1,056,696 USD, and an average fee of ₿6.07 or (USD 16,080 at the time). For our regressions, we winsorize the fee data, the transaction size, the inputs, and the restime at the 99.9% level. (The coinbase of the Genesis block has never been spent, implying a resttime equal to the sample length.) Our results are robust to different levels of winsorizing.

[3] see transaction 29a3efd3ef04f9153d47a990bd7b048a4b2d213daaa5fb8ed670fb85f13bdbcf

[4] see transaction 2fa413fed0aac4c8c85ad8aa96636bda5c30a8283dcad988f045fb395b4f01e9. In the smallest transaction somebody sent zero bitcoin with a fee of zero in transaction 3a5e0977cc64e601490a761d83a4ea5be3cd03b0ffb73f5fe8be6507539be76c.

Bitcoin fees also vary tremendously over time. To control for this time variation we include day-fixed effects in our regression analysis. To control for variation of fees across miners we cluster standard errors per block.

3 Competitive Mining

Bitcoin mining is done for profit, but the mining industry differs from a standard goods market in two important ways. First, miners do not directly set prices for processing transactions. As mentioned above, the fixed per block fee or coinbase is determined by the protocol and fees attached to individual transactions are chosen by users. Second, market share (or blocks mined) only depends on computing capacity. Thus, miners do not compete on price or quantity as in standard goods markets. In light of these differences, we present a simple framework to develop hypotheses on the outcomes that characterize competitive mining.

Consider the profit maximization problem of miner i who is one of N miners. Miners choose transactions from a mempool and process them into blocks. Transactions waiting in the mempool are submitted at discrete fees that are ordered from $0 < f_1 < \ldots f_m$. Let $g_j(t)$ denote the weight of transactions in the mempool prior to block t in fee bucket f_j. Then, the mempool available at time t can be succinctly described as $G_t = \{g_j(t)\}_{j=0}^{m}$.

Let $\alpha_j^i(t)$ denote the fraction of mempool orders in fee bucket j that miner i plans to incorporate into the block t, where $0 \leq \alpha_j^i(t) \leq 1$. Then, miner i's realized profit from successfully mining the block at time t is

$$\pi_t^i \quad = \quad C_t + \sum_{j=0}^{m} f_j \alpha_j^i(t) g_j(t). \tag{1}$$

Here, C_t is the coinbase which is not under the miner's control. Going forward, we assume this to be zero.

Miner i maximizes Equation (1) by choosing fractions $\{\alpha_j^i(t)\}_{j=0}^{m}$ for each block subject to the physical capacity constraint:

$$\sum_{j=0}^{m} \alpha_j^i(t) g_j(t) \leq \kappa, \tag{2}$$

where κ is the fixed block capacity.

Given Equation 1, it is clear that the capacity constraint, Equation 2, must bind. If there is a positive fee transaction waiting in the pool, a miner can increase profits by including that in the block. It is also clear that profit maximizing miners in choosing between two transactions, will choose the higher fee one.

Proposition 1 *Suppose that all miners take the composition of the mempool at any time t as given, then in the solution to each miner's profit maximization problem:*

i. If demand in the mempool is greater than block capacity, so $\sum_{j=1}^{m} g_j(t) \geq \kappa$, then the capacity constraint (Equation 2) will bind every period. That is, there will not be unused capacity in mined blocks.

ii. If demand in the mempool is greater than block capacity, so $\sum_{j=1}^{m} g_j(t) > \kappa$, then a miner will not optimally choose a lower fee transaction if a higher fee transaction is available. Or if $\alpha_j^i(t) > 0$, then $\alpha_{j'}^i(t) = 1$, for all $j' > j$.

To see how competitive mining translates into the data, we present a simple example based on three user types that arrive every after every block has been mined. The timeline is presented in Figure 1. New users that arrive are added to the mempool, while users that are processed into a block are removed from the mempool. For simplicity, we assume that each block has space for three transactions.

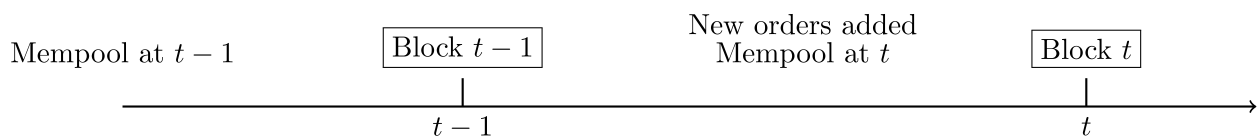

Figure 1. Sequence of Events from Block to Block

Suppose that there are three types of Bitcoin users, denoted h, m, ℓ. Each differs in how much they value a transaction and how long they are willing to wait (in blocks). The h value users values transactions at 4, but only if their order is executed in the next block. The m user values transactions at 2 and is willing to wait up to 1 block, while the l value user obtains 1 from a transaction, and is willing to wait up to 2 blocks. For simplicity if a users' order does not transact with the required number of blocks (labelled "patience"), their payoff is zero. The value of a user's consummated transaction and the maximum number of blocks they are willing to wait are illustrated in Table 2.

Type	Value	Patience
h	4	0
m	2	1
ℓ	1	2

Table 2. Types that arrive before each block is mined. Each type has a value for processed transaction, and a maximum number of blocks that they are willing to wait ("patience").

Example 1 (Competitive Mining) *Suppose that there is a minimum fee of $\epsilon > 0$. If each user submits at least the minimum fee, then a profit maximizing competitive miner will always include a fee paying transaction if there is space in the block. Given these assumptions, Figure*

2 illustrates the orders flowing into the mempool and their time stamp and the consequent blocks formed from the mempool.

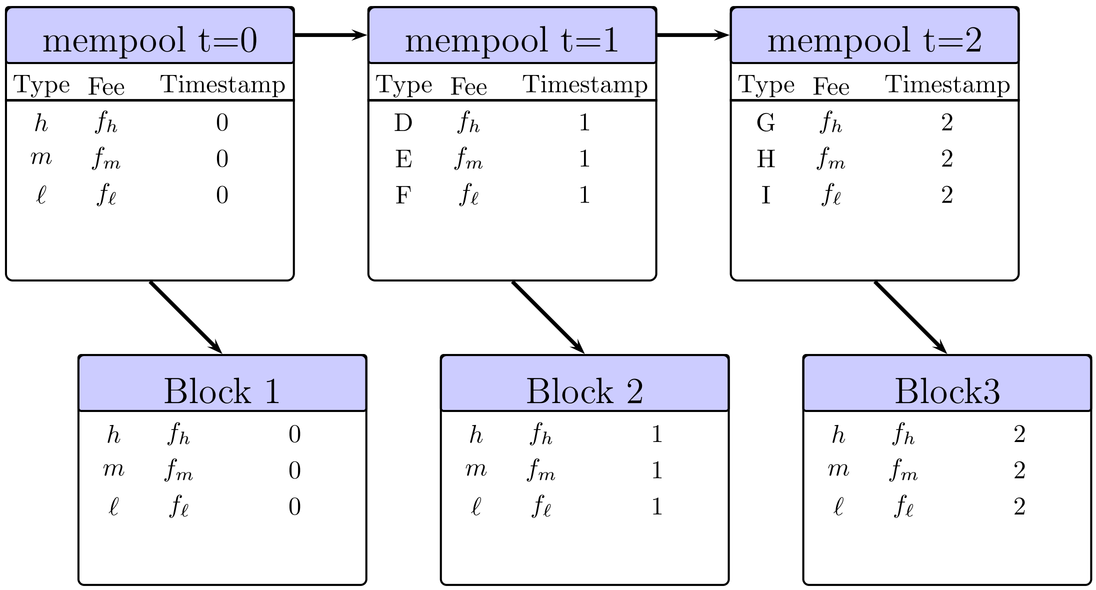

Figure 2. How mempool orders (top) translate into mined blocks(bottom) under competitive mining. Each mined block contains the timestamp of when the user entered the mempool. If the block is filled every period with incoming orders, then $f_h = f_m = f_\ell = \epsilon$, as there is no incentive to submit higher fees. Blocks are full.

The fee revenue from each block would be $f_h + f_m + f_\ell$. Consider the optimal fee: If miners fill up each block, there is no incentive for any of the users to submit a fee above the minimum – in this case ϵ. Therefore, each user will submit the minimum fee so $f_h = f_m = f_\ell = 1$ and the winning miner will gain fee revenue of 3ϵ per block.

We reformulate Proposition 1 into two hypotheses that we can examine in the data to determine if miners are price takers – competitive – and profit maximizers:

Hypothesis 1 *Suppose that all miners are competitive profit maximizers, then if there is excess demand in the mempool, we will not observe unused capacity in mined blocks.*

Hypothesis 2 *Suppose that all miners are competitive profit maximizers, then if there is excess demand in the mempool, we will not observe mined blocks with low fee transactions if higher fee transactions are waiting.*

3.1 Competitive mining in the data

Following Hypothesis 1 we should not observe unused capacity on the blockchain when orders are waiting and from Hypothesis 2 we should not observe low fee transactions being processed when high fee transactions are waiting in the mempool. We investigate each in turn.

3.1.1 Unused Capacity (Hypothesis 1)

Our definition of unused capacity is conservative: we define a block as full if at most an additional 2,000 weight units could have been processed in that block. (In our complete sample the median transaction size has a weight of 904. Thus, at least two more median transactions could have been included one of our "full" blocks.)

Definition 1 *A block is full if 2,000 or fewer additional weight units could have feasibly been included.*
A block is empty if it contains no transactions.

Figure 3 plots the fraction of total blocks per day that are mined full, those that are mined completely empty and the average used capacity per block per day.

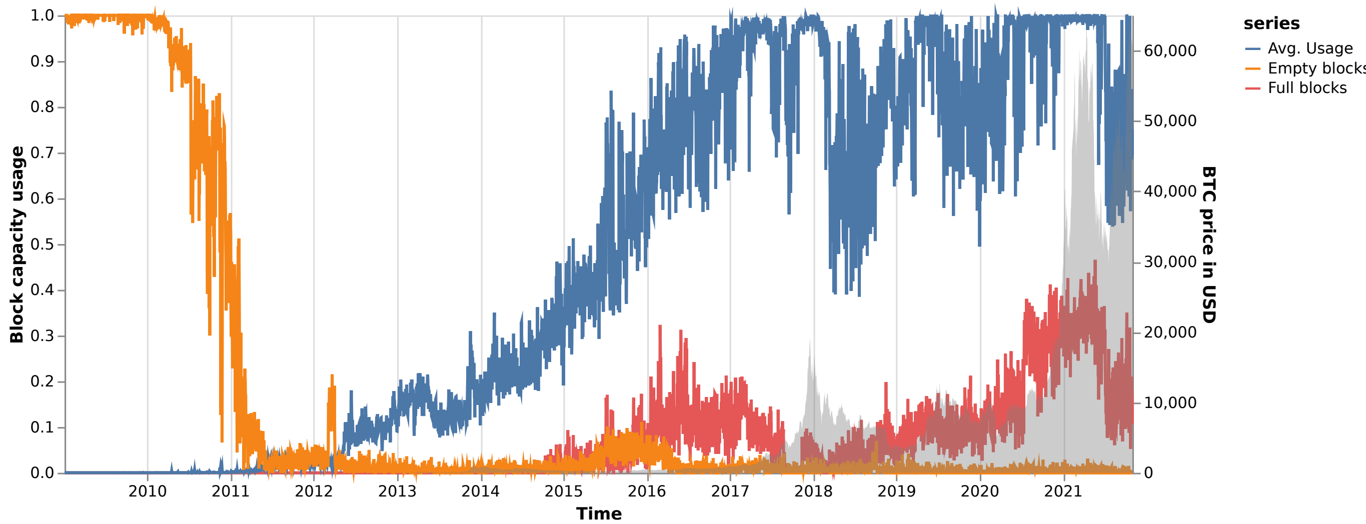

Figure 3. Fraction of full (red) and empty (orange) blocks and average capacity usage (blue) per day measured on the left axis. Days are defined over UTC. Daily USD BTC price depicted in grey (right axis).

From Figure 3, in the early days of Bitcoin, most blocks were mined empty. However, during the 2017 and 2021 runups in Bitcoin prices, the blockchain also did not run at full capacity. For example, on Dec 17, 2017 even though Bitcoin was trading at a then record price over USD 19,000, blocks 499704 and 499763 were mined empty by BTTC pool. This was not a technical problem: On the same day, BTTC pool also mined 5 non-empty blocks.[5] Using part of our mempool data we find that in the one hour interval around the time that block 499704 was mined, more than 130,000 transactions were waiting to be mined. More broadly, we find that for all blocks mined in or after 2016 about 0.87% are empty. In unreported results we find a similar magnitude for all of the top five mining pools in our sample.

[5]Over all of December 2017, 40 empty blocks were mined, by 11 different mining pools. The largest pool, AntPool, also mined the largest number of empty blocks.

In addition to empty blocks, the second source of unused capacity is in blocks that contain some transactions, but are not full. In the blockchain's busiest month so far, April 2021, on average there was space for an extra 3,047 transactions per day.[6] For the broader sample since Jan 1 2014 on average there was space for an additional 184,711 transactions per day. For context, during the same time 228,289 transactions were processed per day. We emphasize that in spite of this unused capacity, unconsummated orders, often with fees attached, were waiting to be added to the blockchain. For 93.13% of the blocks in our sample all the excess capacity could have been completely filled with transactions from the mempool and further 6.87% could be partially filled. Only for 0.0027% of blocks in our sample the mempool was exhausted.[7] While our calculations are based on our mempool data, specialized mining pools have access to better hardware and more peer connections and therefore have better and more up-to-date information on potential transactions. We conclude that we have conservative snapshots of their mempools.

To quantify how much money miners would have made if they had filled all blocks to capacity, we use the minute by minute mempool data, in which partially aggregated transactions are grouped by fee buckets based on sat/byte. Combining the excess capacity per block with information in the mempool about the latent demand, we calculate an upper bound on the cost of leaving transactions unmined in the mempool.

Definition 2 *Money left on the table is the additional fee revenue obtainable from filling empty and partially filled blocks with the highest fee transactions waiting in the mempool.*

For each block, we match the excess capacity in the block with the excess demand for transactions in the mempool at the time the block was mined.[8] Figure 4 plots the total money left on the table by miners per day. Foregone revenue or money left on the table is observed consistently throughout our sample. The "money left on the table" is economically meaningful amounting to USD 121.96 million over the whole sample. We conclude that there is consistently unused capacity in mined blocks which could have been filled with fee-paying, waiting transactions. This is inconsistent with profit maximizing competitive mining.

3.1.2 Priority Violations (Hypothesis 2)

The second hypothesis is that a competitive profit maximizing miner will not process low fee transactions from the mempool when high fee transactions are also waiting. We call these priority violations.

Definition 3 *A priority violation occurs if a block contains a transaction and there are at least five transactions waiting in the mempool that*

[6] We base our calculation on the median transaction size of 904 weight units.

[7] We are able to match mempool snapshots to 264,597 mined blocks. Out of these 242,755 could be completely filled, and 17,897 could be partially filled. 3,938 blocks were already completely full, and for 7 blocks the mempool snapshot was empty. Note that invalid transactions are not broadcast and immediately discarded from the mempool.

[8] Our calculation is an upper bound because it is done with replacement as we cannot distinguish individual transactions with this data set.

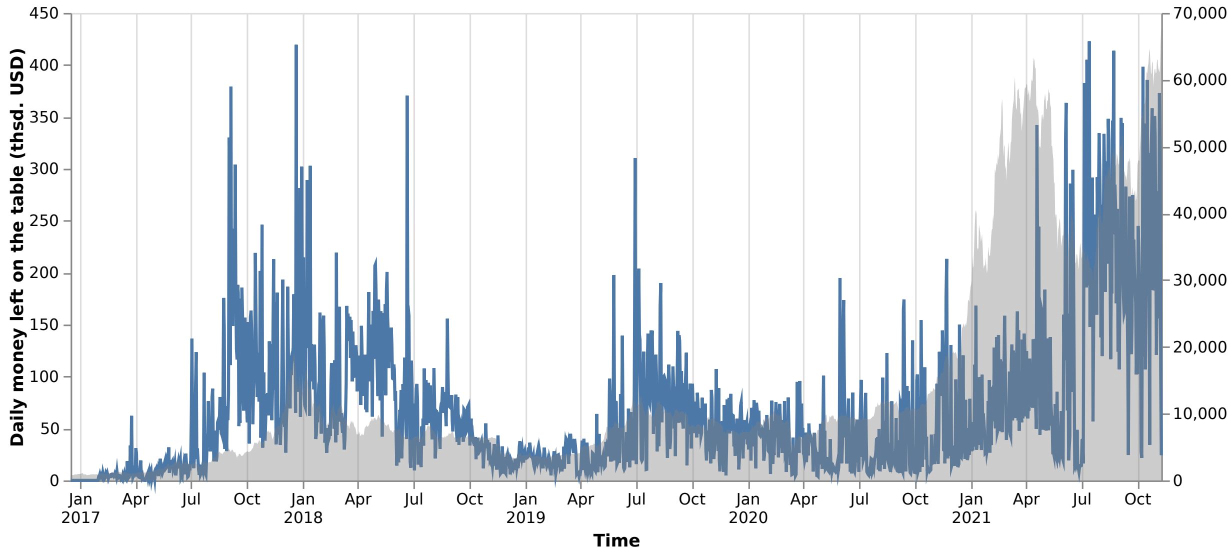

Figure 4. Money left on the table by miners (MLOT) and Bitcoin price. MLOT is computed by filling up empty capacity on the blockchain with unmined mempool transactions offering the highest fee/weight and is aggregated per day (blue, left axis). BTC prices in grey (right axis)

i. *have a higher fee,*

ii. *were waiting up to ten blocks,*

iii. *had a fee greater than 50 cents,*

iv. *were eventually mined.*

We use our detailed transaction level mempool data between blocks 620,591 and 708,957 from March 7, 2020 to November 9,2021. Our definition of priority violations is conservative because we explicitly rule out "stalled transactions." All included transactions were eventually mined. We also eliminate transactions that are initiated by miners and chains of transactions that are dependent on each other. (Such transactions may have artificially low fees.) We end up with over 144 million transactions. Figure 5 plots daily average violations. We note that violations are frequent and high and affect up to 90% of daily processed transactions.

While the existence of priority violations contradict the assumption of competitive mining, Table 3 shows the results of a probit regression explaining the probability that a transaction suffers a priority violation. The blockweight and the number of transactions in the pool do not determine the probability of priority violations. We find no significant difference across mining pools in our sample.

Neither Hypothesis 1 or Hypothesis 2 and the assumption of competitive mining are consistent with the data.

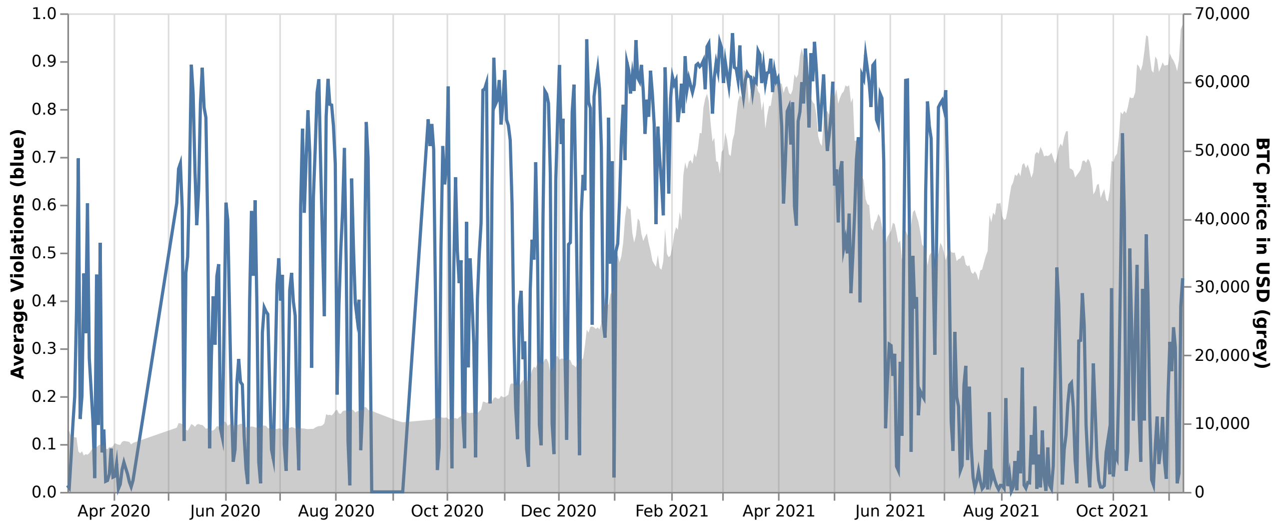

Figure 5. Daily average violations (blue). Days are defined over UTC. Daily USD BTC price depicted in grey (right axis).

Fee per weight (USD)	8.207***	7.361***	7.348***
	(104.15)	(94.22)	(94.04)
₿Price (USD)	0.00192*	0.00218**	0.00222**
	(1.85)	(2.07)	(2.11)
Blocksize (weight)		1.218***	1.219***
		(36.56)	(36.50)
Inputs (USD)		-0.00156***	-0.00156***
		(-41.83)	(-41.83)
Data insertion		-0.215***	-0.215***
		(-48.67)	(-48.64)
Resttime (blocks)		-10.21***	-10.21***
		(-73.44)	(-73.43)
Spent next block		-0.0353***	-0.0351***
		(-11.86)	(-11.81)
Constant	-0.808***	-5.556***	-5.565***
	(-34.49)	(-41.47)	(-41.04)
Observations	124,405,631	124,405,631	124,405,631

Table 3. Probit regression explaining the probability of a priority violation. A priority violation is defined as the inclusion of a lower fee transaction in a block when a higher fee transaction was left waiting in the mempool. *Fee per weight* is fee paid by a transaction over its size measured in weight units converted to USD. *Blocksize* is the weight of the block in million weight units, *Sum Inputs (USD)* is the sum of input values for the transaction measured in million USD, *Data insertion* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, and *Resttime* is the average time (measured in million blocks) until transaction outputs are re-spent. Standard errors are clustered per block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

4 Exercising market power in the Bitcoin protocol

As we have observed, unlike standard product markets, a miner cannot directly affect the fees they obtain, or their market share. Fees are set by users. Further, market share – the probability of mining a block – is determined by miner i's computing power or relative hash rate. Block-by-block, miners only have discretion in the orders that they select from the mempool.

For a miner to exercise market power profitably, the contemporaneous choice of orders must affect the profitability of future orders. To shed light on this, and to formulate our hypotheses on market power, we put more structure on the mempool from which the miner chooses transactions to process every block. This will allow to characterize miner i's dynamic optimization problem.

As described in the previous section, the mempool before block t comprises all previously submitted but currently unprocessed orders. Recall, we described it succinctly as $G_t = \{g_j(t)\}_{j=0}^{m}$, where $g_j(t)$ was the weight of unprocessed transactions at fee level f_j. The mempool changes block by block as some orders are processed and therefore leave, while others are added. Incoming orders and their associated fees are based on random external factors and individuals' optimal choices based on their beliefs about how the Bitcoin protocol works. We denote by ϵ_t, any external shock that affects the mempool at block t.

Users' beliefs about the system are based on the past history of mined blocks and the mempool (both of which are observable). Let h_t denote the history of mined blocks prior to block t. Then, the mempool at time t is $G_t(h_t, \epsilon_t)$. More specifically, the history comprises, for each block, the transactions chosen by the successful miner, and the remaining set of transactions in the mempool. Or, if I_τ is an indicator for the winning miner at τ, who chooses $\{\widetilde{\alpha}_j(\tau)\}_{j=0}^{m}$, the history is:

$$h_t = \left\{ I_\tau \left(\sum_{j=0}^{m} f_j \widetilde{\alpha_j}(\tau) g_j(\tau) \right), \qquad G_\tau(h_\tau, \epsilon_\tau) \right\}_{\tau=0}^{\tau=t-1}. \tag{3}$$

Given this relationship, a miner's dynamic profit function is

$$\Pi_t^i = \underbrace{\sum_{j=0}^{m} f_j \alpha_j^i(t) g_j(t)}_{\text{Current Block}} + \underbrace{\sum_{\tau=1}^{\infty} E\left[\chi^i \sum_{j=0}^{m} f_j \alpha_j^i(t+\tau) g_j(t+\tau) \mid G_{t+\tau}(h_{t+\tau}, \epsilon_\tau) \right]}_{\text{Future Blocks}}. \tag{4}$$

The payoff to the current block solely depends on the existing mempool (summarized by fees and weights or $\{f_j, g_j(t)\}_{j=0}^{m}$) and the choices that the miner makes (α_j^i). By contrast, the payoff to future blocks depends on the cumulative effect of these choices (through the history) and the miner's relative hash rate, χ^i.

The hash rate affects the profitability of future blocks in two specific ways. First, it is the probability that miner i will find the correct random variable before others and mine a block. In this way, it acts as a discount factor that determines how important mining future blocks is to miner i's profit. Second, it is the probability that users of the blockchain will have observed the realization of miner i's processing strategy. Or, it is the probability with which miner i has affected and will affect the realized history and hence the flow of orders into the mempool. In short, χ^i also captures the extent to which miner i affects users' beliefs about the Bitcoin protocol.

Proposition 2 *There is a relative hash rate, $0 < \bar{\chi} < 1$, so that if $\bar{\chi} < \chi^i$, processor i affects the realized history observed by Bitcoin users and optimally exercises market power.*

Intuitively, if χ^i is 1, then the miner succeeds in every block, and is a monopolist, so his processing strategy is the processing strategy of the Bitcoin protocol. Conversely, as $\chi^i \downarrow 0$ choices made by the miner are not expected to have an effect on subsequent order arrivals. This corresponds to competitive mining.

If miners are exercising market power, they can only do so by the transactions they select as they form blocks. We therefore consider the implications of miners strategically delaying transactions: they implement a strategy so that the lower the fee, the longer the wait. Users observe this, and understand the relationship between wait time and fees. Therefore, users with an urgent need to have their transaction incorporated into a block attach the highest fee possible to their order to get it processed quickly.

Example 2 (Capacity management as market power) *Strategic capacity management can increase miner fee revenue. Recall, that there are three types of users who arrive every period. Assume miners adhere to the processing strategy illustrated in Table 4. Specifically, if a user submits a fee of $4 - \epsilon$, the transaction is processed immediately, if they submit a fee of $2 - \epsilon$, a wait time of 1 block is imposed. If they submit a fee of $1 - \epsilon$, a wait time of 2 blocks is imposed.*

Fee	Wait Time(blocks)
$4 - \epsilon$	0
$2 - \epsilon$	1
$1 - \epsilon$	2

Table 4. Fees and wait times imposed under strategic capacity management.

Faced with this menu, users submit fees equal to their valuations. The imposed wait time induces users to bid their valuations. Consider users with value 4. If she submits a fee of $4-\epsilon$, he obtains a benefit of ϵ. If he submits a lower fee ($2 - \epsilon$), he will wait for two blocks, which reduces his value for the transaction to 0. The waiting cost means that he will not submit a lower fee. A similar logic applies to the users with valuation 2 and 1 respectively.

High value users only submit high fees if lower values have longer wait times. Thus, to sustain the fee structure in Table 4, miners have to let transactions with lower fees wait in the mempool even though there is empty capacity in the block.

This strategy induces the block composition illustrated in Figure 6. Every block, one transaction from each of h, m, ℓ enters the mempool. We label each of these with a subscript indicating the timestamp from when they arrive. The high fee transactions with fee 4 are processed immediately. Waiting times are imposed on the other orders. Specifically, the order submitted by m_0 at fee of 2 is not mined immediately, but is kept in the mempool even though there is unused capacity and the transaction is only processed in Block 2. The order l_0 is kept back for two blocks and only processed in Block 3. Notice, that it is processed even though a higher fee order (m_2 with a fee of 2) is waiting in the mempool. These "priority violations" are a consequence of SCM and will occur more for lower fee transactions. We will analyze priority violations in greater detail in Section 3.1.2.

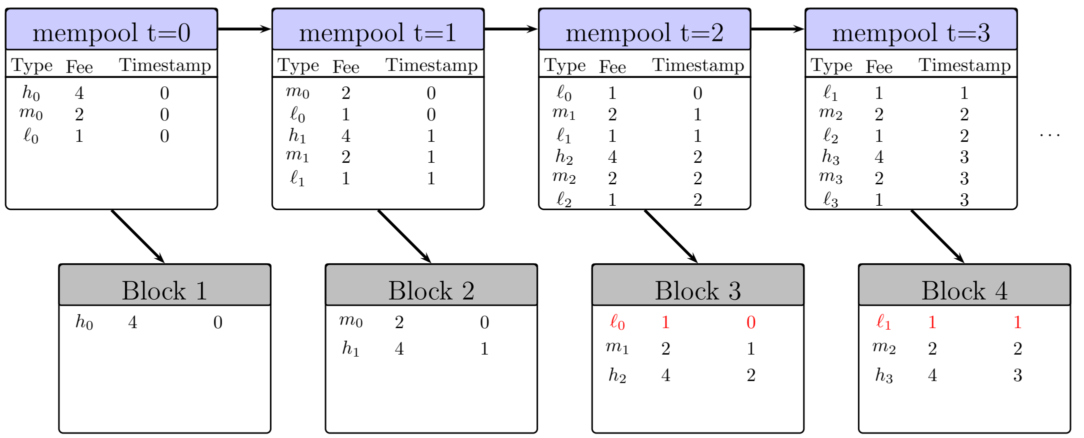

Figure 6. Block composition under strategic capacity management. High fee transactions are processed immediately, whereas lower fee transactions are made to wait. In this rendering, we have let ϵ go to zero.

Two observations:

i. The steady state per block revenue from this strategy is $7 - 3\epsilon$ per block, more than the revenue under competitive mining.

ii. There are priority violations: Credibly imposing wait times on low bidders requires that they have to be kept waiting in the mempool. This means that sometimes m types are kept waiting. If they were not, they would have no incentive to submit higher fees.

Various hypotheses are consistent with revenue maximizing strategic capacity management. First, are implications on how block capacity is used. Second, there are implications on the observed fees, and third there are implications on how relative hash rates affect mined block characteristics.

If miners are exercising market power and using strategic capacity management, then we will observe both partially filled or empty blocks and priority violations. The analysis of the previous section is consistent with Hypothesis 3.

Hypothesis 3 *(Mined Blocks and Mempool) If miners are exercising market power:*

(i) The mempool may contain transactions after a partially empty or fully empty block is mined.

(ii) A mined block may include transactions with a fee lower than existing transactions in the mempool (priority violations).

Strategic capacity management has implications for the observed fees. First, if SCM is successful, then users with higher valuations, or users whose valuations increase will submit higher fees. The second implication is that fee dispersion within a block will be high. The intuition is that if SCM is applied optimally, high value users will be induced to submit the highest possible fees, while those with lower valuations will submit low fees and wait. This is consistent with both the examples – under competition users retain surplus, while under market power, miners extract surplus.

Hypothesis 4 *(Fees) If miners are exercising market power through strategic capacity management:*

(i) Users with higher valuations or whose valuations have increased will submit more extreme bids

(ii) Fee dispersion within a block will be high.

Finally, as we have argued, the exercise of market power is related to the relative hash rate of miners. We emphasize that this variable should have no explanatory power under competitive mining or if congestion drives mined blocks. Our empirical measure of the relative hash rate is the share of daily blocks mined by specific mining pools. To compute this, we collect miners' signatures from each block's coinbase transaction. Unlike any other transactions, the Bitcoin in the coinbase are newly created and therefore do not originate from another wallet. Miners use the space reserved for the input script to insert data into the blockchain. This space is used to send messages to the community on topics as diverse as miners' opinion on Bitcoin improvement proposals, or philosophy,[9] but most important for our purpose is that it usually contains a signature identifying the mining pool. We automatically search for commonly used signatures and then manually examine unidentified blocks for re-occurring signature patterns. We compute the daily Hirschman Herfindahl index (HHI) of mining concentration as the sum of the squared shares of each mining pool computed over the day where the block is mined.

Hypothesis 5 *(Relative hash rate)*

(i) The higher the HHI, the stronger the results in each of the other hypotheses.

(ii) Increases in the relative hash rate increases the incentive to exercise market power.

[9]e.g. 'Welcome to the real world.' in block 328465, or 'smile to life and life will smile back at you' in block 328444, or 'the Lord of the harvest, that he send forth labourers into his harvest' in Block 143822.

For succinctness, we include our concentration measure (HHI) in tests of the other hypotheses. We present more direct evidence on the effect of the relative hash rate and our measure, HHI in Section 4.5 below. However, as we present evidence related to our other hypotheses, we note the amplifying effect of this variable in the regressions.

4.1 Exercising Market Power in the Data

We address each of the hypotheses in the data. First, however we provide anecdotal evidence that is consistent with strategic capacity management. Anecdotal evidence suggests that users do believe that there is relationship between fees and waiting times. Online fee calculators such as the one shown in Figure 7 provide users with a real time estimate on the fee they have to post to be confirmed with a 90% probability within 1,2,3,4,5, and 6 blocks, respectively.[10]

4.2 Hypothesis 3 (Mined Blocks and Mempool Usage)

As we have observed, our analysis of the previous Section 3.1 is consistent with Hypothesis 3. We provide additional evidence.

The Bitcoin protocol calibrates the difficulty, i.e. the number of leading zeros that a block hash has to have to qualify as valid, in such a way that on average a new block is added every ten minutes. Yet the times between blocks as they are recorded on the blockchain vary widely because mining a successful block is purely random and so sometimes blocks are found very quickly and sometimes it takes a long time. Figure 8 illustrates the time between blocks.[11] In the graph we focus on blocks after block 100,000 because dispersion in times between blocks was higher in the early days of Bitcoin. Very few blocks have more than 50 minutes between them and these observations are omitted from the graph.

Strategic capacity management can increase revenue because miners delay low fee transactions. To keep delay consistent miners compensate for the variation in the arrival-time of blocks as documented in Figure 8 by adjusting block usage. Specifically, we would expect that if a few blocks arrive close together, subsequent blocks would be more empty as miners delay patient types. Similarly after a long interval between blocks, subsequent blocks would be fuller to accommodate the patient types.

The length of time between blocks clearly affects the size and composition of the mempool. To

[10]See for example `https://www.buybitcoinworldwide.com/fee-calculator/` or `https://bitcoinfees.github.io/#30m`.

[11]Clock mis-alignments can be an issue in Bitcoin. We find 13,848 cases in the early years of our sample for which a block has an earlier time-stamp than its predecessor. This is technically impossible. Each block contains information from the previous block, which links the blocks together in a blockchain. The only rational explanations for the inconsistency in time-stamps is improper alignment of miners' clocks. To accommodate potential synchronization problems in miners' clocks the Bitcoin protocol allows a block to have time-stamp up to two hours earlier than the previously mined block. We adjust for these mis-measurements heuristically by assuming that a block with an impossible timestamp has been mined half way between the two neighbouring blocks. Despite these problem cases for the vast majority of the sample the time-stamps seem to be properly recorded.

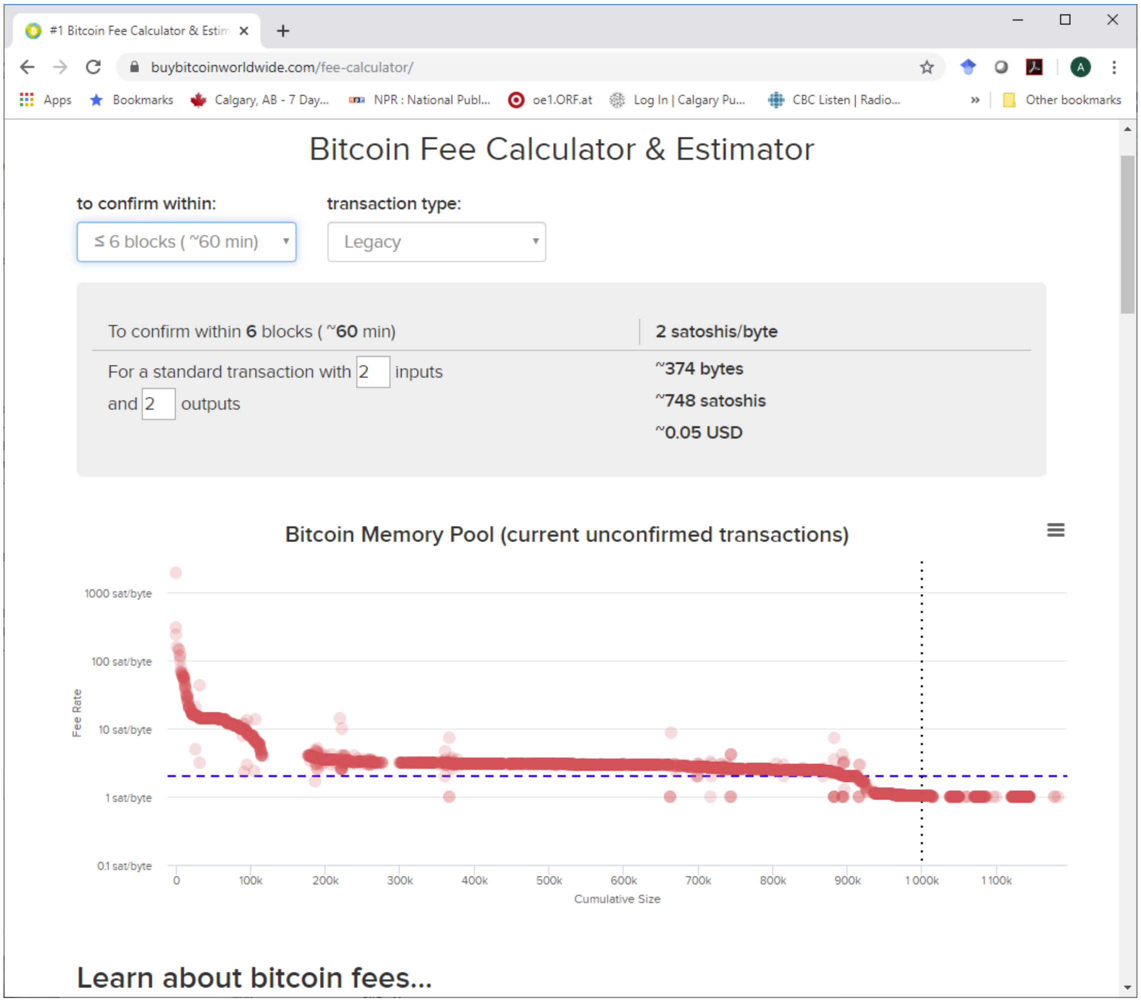

Figure 7. Screenshot from a Bitcoin fee calculator that provides real time estimates of fees that users have to offer to get confirmed with 90% probability within 1,2,3,4,5, and 6 blocks, respectively.

eliminate changes in the mempool as a reason for patterns in capacity usage, we control for the size of the mempool but more importantly we interact arrival times with the HHI. If used block capacity is simply a pass-through from changes in the mempool, the HHI should be irrelevant.

We regress block-weight on dummies that record if the previous block was mined more than 7 minutes outside the expected 10 minute interval. Columns (1) and (2) of Table 5 document that blockweight decreases in the HHI of mining concentration. In column (3) we find that block weight increases when the last block was minted more than 17 minutes ago, but only when the HHI is high and that block weight decreases when the last block was mined less than 3 minutes ago and when the HHI was high. In the last column we document for a smaller sample that our findings are robust with respect to the size of the mempool. Our findings are consistent with the idea that when miners find it easier to collude, capacity management by mining pools is more prevalent.

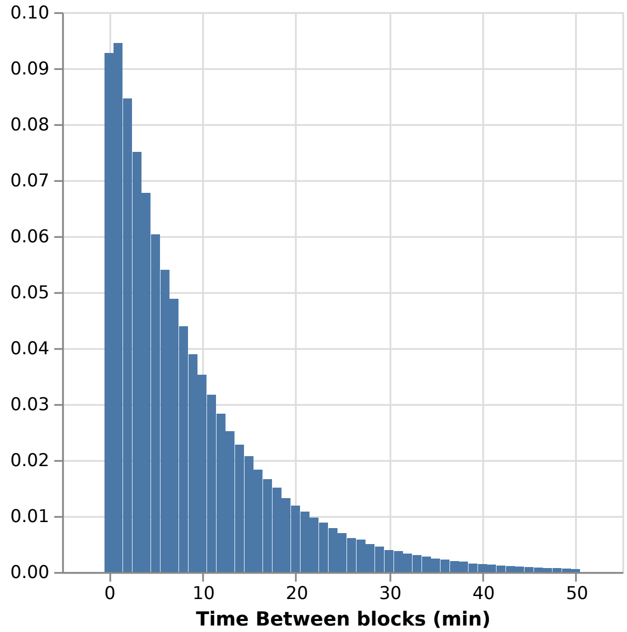

Figure 8. Histogram of time between blocks with blockheight larger than 100,000, capped at 50 minutes.

Another way to manage capacity is by including empty blocks. In Table 6 we document that the probability of an empty block being mined increases in the number of blocks found in the last hour and in the used capacity over the last hour.[12] This evidence is consistent with miners delaying transactions and thus inserting empty blocks when by chance too many blocks were found.

4.3 Hypothesis 4 (Higher Value Users Pay more in Fees)

In Appendix E, we analyze transaction and blockchain specific variables that drive fees and then provide evidence that is consistent with the idea that high value users pay higher fees. We document that users who spend their Bitcoin faster pay higher fees.

Additionally, if miners are successful in exercising market power, then higher value users should be induced to pay more. This has two implications. First, in times when values are higher, holding fixed blockchain characteristics, fees should be higher and second, market power should induce fee dispersion.

To identify high value traders whose values change over time, we consider cross-exchange arbitrageurs. Arbitrageurs' preference for immediacy is higher, the higher the price differential between exchanges as they need to us the blockchain to move Bitcoin between exchanges. We focus on the well-documented "Kimchi premium," which is the relative price difference of Bitcoin

[12] Our findings are similar with two hour windows.

HHI mining activity	-9.013***	-10.28***	-10.21***	-4.665***
	(0.488)	(0.511)	(0.517)	(1.024)
Last block > 17 min old			-0.0354	-0.0571
			(0.0272)	(0.0579)
HHI x Last block > 17 min old			0.680***	0.919
			(0.259)	(0.698)
Last block < 3 min old			0.0957***	0.132***
			(0.0218)	(0.0404)
HHI x Last block < 3 min old			-0.667***	-1.156**
			(0.213)	(0.480)
Sum Inputs (USD)		0.000776***	0.000776***	0.000353***
		(0.0000440)	(0.0000439)	(0.0000312)
Data		-1.203***	-1.204***	-1.736***
		(0.0689)	(0.0689)	(0.0742)
Resttime		0.0983***	0.0981***	0.0947***
		(0.00987)	(0.00986)	(0.0151)
Size mempool				0.00242***
				(0.000510)
R^2	0.0377	0.0700	0.0702	0.0682
Observations	221,709	221,709	221,709	122,765

Table 5. Regression results of blockweight on the frequency of recent blocks. *HHI mining activity* is the is the Herfindahl–Hirschman index of daily mining shares, *Last block > 17 min old* and *Last block < 3 min old* are dummy variables equal to one if the most recent block was mined more than 17 or less than 3 minutes ago, respectively, *Sum Inputs (USD)* is the average input transaction value measured in million USD per block, *Data* is the fraction of data insertion transactions (identified by the OP_RET instruction in the script) in the block, *Resttime* is the per block average of the time (measured in thousand blocks) until transaction outputs are re-spent, and *Size mempool* is the size of transactions waiting in the mempool measured in million weight units. Standard errors are clustered by day. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

Number of blocks (1h)			0.00345	0.00671**
			(0.00218)	(0.00300)
Number of blocks (2h)	0.00269*	0.00609***		
	(0.00162)	(0.00202)		
Size mempool		-0.00100***		-0.00101***
		(0.000182)		(0.000182)
R^2				
Observations	371,927	264,597	371,927	264,597

Table 6. Probit regression explaining the probability of mining a sparse block. A sparse block is defined as a block with at most 30 transactions, *Number blocks (1h)* and *Number blocks (2h)* are the number of blocks mined in the previous hour/two hours, *Size mempool (MB)* is the size of the mempool in million weight units. All regressions include week fixed effects. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

in the US and Korea.[13]

[13] The Kimchi premium is calculated as the absolute value of the Bitcoin price at Korbit in Korea converted to USD minus the Bitcoin price on Coinbase in the US as a percentage of the US price.

We match minute time-stamped prices from Korea and the US with block creation times. We interact a dummy for payments being made to and from wallets that can be identified as exchanges with the absolute value of the Kimchi premium and find that fees on transactions to and from exchanges increase in the Kimchi premium.

Our findings are in Table 7. Exchange is a dummy that identifies 35,163,580 payments to and from known exchange wallets. The interaction term of Exchange and Kimchi premium is statistically and economically significant. For an increase in the Kimchi premium of 10 percentage points, users are willing to pay USD 3.22 more in fees. We note that our analysis is most likely an underestimate of how the demand for immediacy affects fees for two reasons. First, we cannot identify all payments to exchanges. Observed high fees to exchanges might therefore coincide with similar high fee payments to unidentified exchanges making it harder to identify any effect in the data. Second, arbitrage between KRW and USD is only one of many potential trading strategies to exploit price differences, within one country, or between countries. We might therefore also observe high fees for payments to exchanges at times when two different markets have a large price difference which would not be captured in our regression. The results are robust to the introduction of Segwit and other control variables. Our finding cannot be driven by general variation in fees over time as we include day fixed effects. We note that the effect is stronger, when mining is more concentrated. Once again, we defer a complete discussion of HHI, our concentration measure, to Section 4.5 below.

Different trader types plausibly have different values for transactions and different costs of waiting. More sophisticated investors such as institutional investors or hedge funds are more likely to be active during the week and when the Bitcoin futures at CME are trading.[14]

We regress fees on explanatory variables and time dummies to capture institutional trading. The results are presented in Table 8. Average transaction fees on weekends are 37 cents lower than week days (Column (2)) which, while small, is economically meaningful as it is close to the median transaction fee for the whole sample, which is 33 cents. Also, fees gradually increase during the work week so that the highest observed fees are on Fridays, which is also the settlement day for futures (Column (1)). Fees are also higher by about half a dollar, more than the median fee, whenever the futures market is open (Column (3)). The effects are stronger if mining is more concentrated. We defer further discussion of Column (4), and the HHI variable, to Section 4.5 below.

It is possible that CME trading hours are picking up higher transaction demand that is unrelated to futures and hence institutional trading. To ensure that our findings are not driven by specific characteristics of CME futures trading hours we perform a robustness check using a 15 day window around December 18, 2017 when futures were first traded. These results appear in column (5). The dummy *CME trading hours* is one during the regular trading hours of Bitcoin futures at CME, *Post Dec 18th* is a dummy equal to one after December 18, 2017 and *CME Futures trading* is the product of *CME trading hours* and *Post Dec 18th*. We find that fees during CME trading hours are significantly higher once futures trading starts.

[14] BTC futures at CME trade Sunday to Friday from 6pm to 5pm EDT with a daily one hour break between 5pm and 6pm EDT. Futures were first traded on December 18, 2017. Settlement is on the last Friday of the contract month. For this analysis we convert all block timestamps from UTC to Eastern Time with the appropriate adjustment for summer daylight savings time.

Kimchi Premium × exchange	15.56***	16.00***	13.41***	15.38***	12.86***
	(118.96)	(144.70)	(135.86)	(39.25)	(36.98)
Kimchi Premium × exchange × HHI				6.204*	5.589*
				(1.71)	(1.71)
Exchange		2.079***	1.568***	2.080***	1.568***
		(233.95)	(215.76)	(234.25)	(215.34)
Kimchi Premium		-14.51***	-12.48***	-14.54***	-12.50***
		(-56.35)	(-51.43)	(-56.10)	(-51.07)
Blocksize (thsd. weight)			0.0000565***		0.0000565***
			(74.32)		(74.30)
Tx-Size (hundred weight)			0.0141***		0.0141***
			(375.81)		(375.80)
Inputs (USD)			0.00000894***		0.00000894***
			(418.58)		(418.56)
Data insertion			-0.145***		-0.145***
			(-41.41)		(-41.29)
resttime			-0.000290***		-0.000290***
			(-136.00)		(-136.07)
Spent next block			0.472***		0.472***
			(209.33)		(209.34)
R^2	0.514	0.517	0.581	0.517	0.581
Observations	662,729,303	662,729,303	662,729,303	662,729,303	662,729,303

Table 7. Regression results of fees in USD including payments to exchanges and the size of the Kimchi premium. *Kimchi Premium* is the absolute value of the Bitcoin price in Korea converted to USD minus the Bitcoin price in the US as a percentage of the US price, *Exchange* is a dummy equal to one if the transaction involves a wallet identified as belonging to an exchange, *HHI* is the Herfindahl–Hirschman index of daily mining shares, *Post Segwit* is a dummy equal to one after the introduction of Segwit, *Block weight* is the size of the block measured in weight units, *Transaction Weight* is the size of the transaction measured in weight units, *Sum Inputs (USD)* is the sum of input values for the transaction measured in USD, *Data* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, *Resttime* is the average time (measured in blocks) until transaction outputs are re-spent. Regressions include day fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

Bitcoin is a pseudo-anonymous system, while wallet addresses are public and payments can be observed moving from one wallet to another, the identity of the wallet owner is usually unknown. In some cases, e.g. voluntary disclosure, or court proceedings, the owner of some addresses becomes public. In addition, gambling sites often use vanity addresses, such as '1dice....' or '1Lucky....' which are easily identified and more importantly are reused.[15] Once an address is known, other addresses controlled by the same wallet can be inferred in a process commonly known as address clustering see e.g. Reid and Harrigan (2013) or Foley, Karlsen, and Putniņš (2018). If multiple addresses are used as inputs in the same transaction these addresses most

[15] Addresses are encoded in a Base58 alphabet (i.e. there are 58 possible 'letters' consisting of upper case, lower case letters and numbers with some combinations dropped that are often mixed up when printed on paper, e.g. capital i and lower case L) and start with 1. To get an address starting with '1Lucky' one has to try $58^5 \approx 656$ *million* combinations. Vanity address companies offer computing resources to find custom Bitcoin addresses.

likely belong to the same person because the private key has to be used to sign the transaction.[16]

We use data from lists of known addresses and are able to identify the sender for 18,123,498 transactions, out of which 7,250,374 (2.05% of all transactions) were initiated by an exchange and 10,873,124 (3.07% of all transactions) that were initiated by a gambling site.[17] Similarly, we are able to identify 32,771,960 payments to an exchange and 23,099,021 payments to a gambling site. Table 9 presents our findings for fees in USD. Notably, flows to and from exchanges transact at substantially higher than average fees. Since we control for day fixed effects our results cannot be driven by more exchange flows occurring on days when fees are generally higher. Our findings are also not driven by outliers as the data is winsorized. Transactions flowing into exchanges pay on average USD 3.97 more than the average fee paid on the same day. This is economically large, given that the median fee for the whole sample is USD 0.20. Flows from exchanges pay USD 2.20 more than same-day average. Gamblers also pay significantly higher fees. Traders moving funds in and out of exchanges and gamblers put a high value on immediate execution. Consistent with revenue maximization, such transactions pay higher fees.

Our last piece of evidence that there is heterogeneity in users' valuation is from the shutdown of the dark net website, Silk Road. The operator, Ross Ulbricht, was arrested by government agents at the Glen Park Branch Library in San Francisco in the afternoon of October 1, 2013.[18] Payments for drugs, and illegal guns were made in bitcoin and are plausibly time sensitive.

We examine all transaction in a four week window around the closure and present our findings in Table 10. There is a 12 cent drop in fees for transactions that are in the upper quartile of the fee distribution in the days following the closure of Silk Road. This quantity is economically significant as the median transaction fee for this sub-sample is 6.1 cent. The bitcoin blockchain was not capacity constrained with an average blockweight of 787,000 well below the maximum of 4 million. Our findings are robust to miner fixed effects (column(3)) and a longer event window (column(4)).

[16]One notable exception are anonymizing services combine transactions of several users into one large transaction so that it is not that clear who paid whom. See e.g. Möser and Böhme (2017).

[17]The data are primarily from `walletexplorer.com` and are available from the authors upon request.

[18]Because our data is encoded in UTC time we set the event date for our analysis to October 2^{nd} UTC time.

	Whole Sample				Event
	(1)	(2)	(3)	(4)	(5)
Block Size (Weight)	0.0000544***	0.0000576***	0.0000712***	0.0000715***	0.00524***
	(0.00000103)	(0.00000100)	(0.000000928)	(0.000000928)	(0.00122)
Transaction Weight	0.0146***	0.0146***	0.0146***	0.0146***	0.0161***
	(0.0000384)	(0.0000384)	(0.0000384)	(0.0000384)	(0.000230)
Sum Inputs (Sat)	0.0151***	0.0151***	0.0152***	0.0152***	0.0286***
	(0.0000530)	(0.0000529)	(0.0000531)	(0.0000531)	(0.00105)
Data	-0.0435***	-0.0470***	-0.0466***	-0.0477***	1.672***
	(0.00405)	(0.00404)	(0.00404)	(0.00404)	(0.249)
Time until spent	-0.000363***	-0.000363***	-0.000365***	-0.000365***	-0.00116***
	(0.00000239)	(0.00000240)	(0.00000239)	(0.00000239)	(0.0000446)
Spent next block	0.532***	0.532***	0.531***	0.531***	1.626***
	(0.00270)	(0.00270)	(0.00270)	(0.00270)	(0.0939)
Monday	0.386***				
	(0.0100)				
Tuesday	0.495***				
	(0.00985)				
Wednesday	0.531***				
	(0.00984)				
Thursday	0.616***				
	(0.0100)				
Friday	0.618***				
	(0.0107)				
Saturday	0.295***				
	(0.0104)				
Weekend		-0.377***			
		(0.00604)			
CME Futures trading			0.468***	-0.103***	2.405***
			(0.00895)	(0.0281)	(0.396)
HHI				-0.917***	
				(0.161)	
CME Futures trading × HHI				6.521***	
				(0.363)	
CME trading hours					-0.796***
					(0.233)
Post Dec 18th					7.496***
					(0.339)
R^2	0.552	0.552	0.551	0.551	0.240
Observations	685,406,634	685,406,634	685,406,634	685,406,634	11,035,388

Table 8. Regression results of fees in USD - day of the week and opening hours of the futures market. The regression includes day of the week dummies, *Weekend* is a dummy set to one if the day is either Saturday or Sunday, *HHI* is the Herfindahl–Hirschman index of daily mining shares, *CME Futures trading* is a dummy that is set to one during the trading hours of Bitcoin futures at the CME after Dec 18, 2017, the day when Bitcoin futures started trading, *CME trading hours* is a dummy that is set to one during the hours when Bitcoin futures trade at the CME for the whole sample period, i.e. also before Dec 18, 2017, *CME trading hours* is a dummy that is set to one after Dec 18, 2017, *Block weight* is the size of the block measured in weight units, *Transaction Weight* is the size of the transaction measured in weight units, *Sum Inputs (USD)* is the sum of input values for the transaction measured in USD, *Data* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, and *Resttime* is the average time (measured in blocks) until transaction outputs are re-spent. Regressions include week fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

To exchange	1.848***	1.362***		
	(0.00452)	(0.00358)		
To gambling	0.447***	0.328***		
	(0.00143)	(0.00128)		
From exchange			2.636***	1.982***
			(0.00849)	(0.00674)
From gambling			0.140***	0.0376***
			(0.000907)	(0.00116)
Blocksize (thsd. weight)		0.0000568***		0.0000573***
		(0.000000731)		(0.000000733)
Tx-Size (hundred weight)		0.0139***		0.0139***
		(0.0000366)		(0.0000368)
Inputs (USD)		0.00000881***		0.00000909***
		(2.10e-08)		(2.15e-08)
Data insertion		-0.0892***		-0.133***
		(0.00341)		(0.00349)
Resttime		-0.000243***		-0.000284***
		(0.00000206)		(0.00000209)
Spent next block		0.443***		0.452***
		(0.00214)		(0.00217)
R^2	0.523	0.583	0.518	0.581
Observations	685,950,180	685,950,180	685,950,180	685,950,180

Table 9. Regression results of fees in USD and known wallet addresses. *To exchange* and *From exchange* are dummy variables set to one if the transaction makes a payment to or receives a payment from a wallet identified as belonging to an exchange, respectively. *To Gambling* and *From Gambling* are dummy variables set to one if the transaction makes a payment to or receives a payment from a wallet identified as belonging to a gambling site, respectively. *Block weight* is the size of the block measured in weight units, *Transaction Weight* is the size of the transaction measured in weight units, *Sum Inputs (USD)* is the sum of input values for the transaction measured in USD, *Data* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, and *Resttime* is the average time (measured in blocks) until transaction outputs are re-spent. Regressions include day fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

	(1)	(2)	(3)	(4)
Post closure	-0.00464***	-0.0102**	-0.00861*	-0.0263***
	(0.000464)	(0.00370)	(0.00460)	(0.00849)
High Fee	0.247***	0.199***	0.199***	0.175***
	(0.0343)	(0.0149)	(0.0153)	(0.0454)
Post closure * High Fee	-0.104***	-0.102***	-0.103***	-0.0742**
	(0.0275)	(0.0261)	(0.0271)	(0.0352)
Block Size (Weight)		-0.00903	-0.00439	0.0213*
		(0.00533)	(0.00428)	(0.0119)
Transaction Weight		0.0514**	0.0515**	0.0916***
		(0.0223)	(0.0223)	(0.0316)
Sum Inputs (USD)		20.16***	20.13***	30.47***
		(3.880)	(3.865)	(2.906)
OPRET		-1.449**	-1.443**	-2.436**
		(0.633)	(0.627)	(0.992)
Time until spent		-0.145**	-0.146**	-0.242***
		(0.0643)	(0.0649)	(0.0682)
Spent next block		0.00461	0.00438	0.0313
		(0.0144)	(0.0144)	(0.0217)
R^2	0.000182	0.00244	0.00245	0.00233
Observations	1,531,227	1,531,227	1,531,227	3,035,234

Table 10. Regression results of fees in USD around the closure of the Silk Road darknet marketplace. *Post closure* is is a dummy set to one post the closure of the site, *High Fee* is a dummy equal to one for all fees is the top quartile of the fee distribution, *Block weight* is the average daily size of blocks measured in million weight units, *Tx weight* is the daily average weight of transactions measured in thousand weight units, *Sum Inputs (USD)* is the daily average input transaction value measured in million USD, *Data* is the fraction of daily data insertion transactions (identified by the OP_RET instruction in the script), *Resttime* is the daily average of the time (measured in thousand blocks) until transaction outputs are re-spent. Days are defined in UTC. Standard errors are clustered by miner. Columns (3) and (4) include miner fixed effects. Columns (1)-(3) compare two weeks before and after the event, Column (4) examines four weeks before and after the event. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

4.4 Hypothesis 4: High fee dispersion

High fee dispersion is consistent with revenue maximization to induce induce different fees from different types. If agents believe that mining is competitive, it is difficult to envisage a rational agent paying such excessive fees given that transactions with lower fees were recently included in blocks.

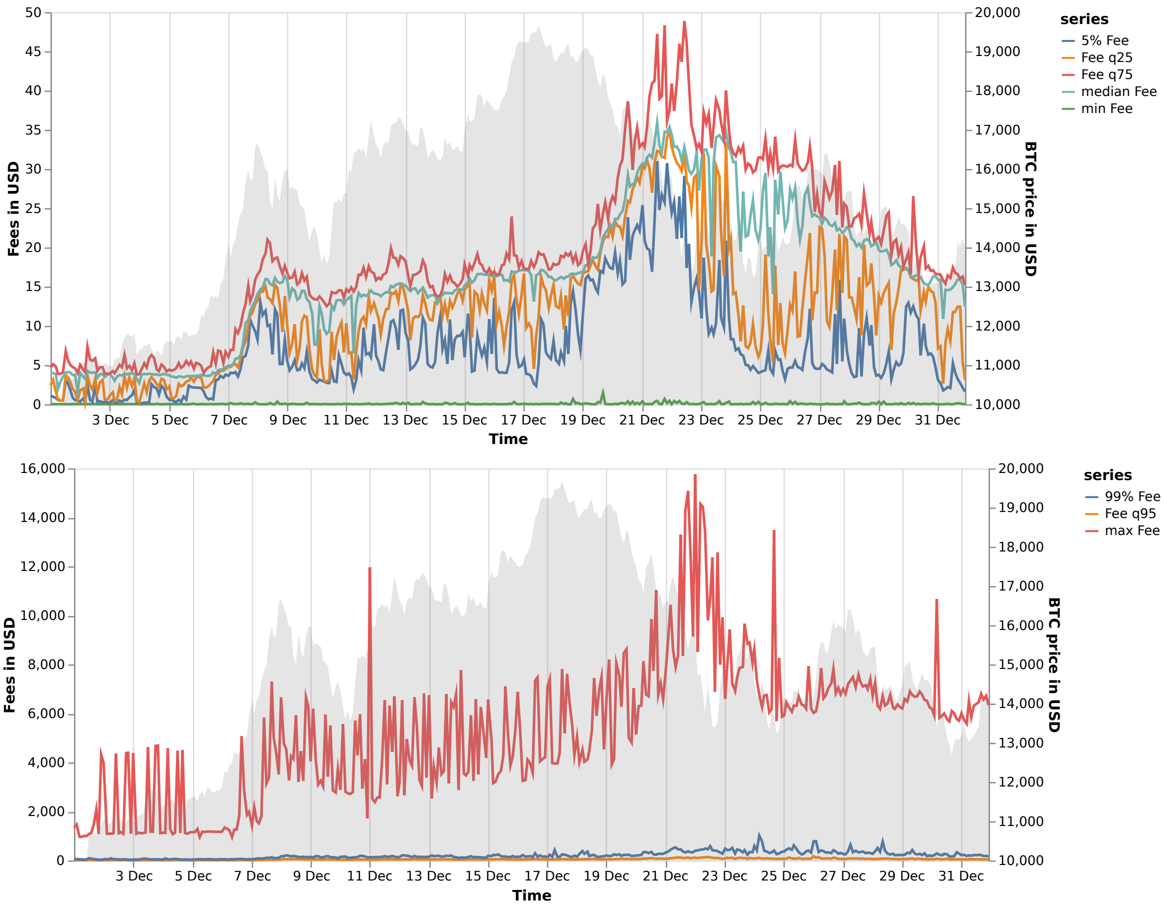

Figure 9. Fees in USD (right axis) and Bitcoin price in USD (left axis) in December 2017. Lower (upper panel) and higher (lower panel) quantiles of the fee distribution for two hour windows in December 2017.

Figure 9 illustrates Bitcoin fee heterogeneity in December 2017, which was the peak of Bitcoin prices in our sample. The left panel illustrates that fees up to the median were less than USD 35. By contrast, from the right panel the maximum fee was USD 14,174.64, and many other transactions paying several thousand dollars in fees. Overall, there are 80 transactions in our sample with fees greater than USD 10,000, of which 51 occur between Dec 20, 2017 and Dec 24, 2017. However over these same five days 1,674,141 transactions were processed out of which 752 had no fee and 16,191 transactions are mined with fees less than USD 5.[19]

[19]In unreported results we document a similar pattern around another peak in Bitcoin in March 2021.

Violations (lag)	17.19***	10.34***	17.03***	-23.48***
	(19.85)	(10.42)	(20.21)	(-3.12)
Violations (lag 2)		9.952***		
		(10.32)		
HHI			95.30***	-53.48***
			(4.86)	(-3.09)
Violations (lag) x HHI				384.5***
				(5.25)
R^2	0.422	0.496	0.443	0.476
Observations	561	560	561	561

Table 11. Regression results of the daily fee spread defined as the difference of the 90% and 10% quantile in USD. *Violations (lag)* is the is the fraction of transactions that had a priority violation on the previous day, *Violations (lag 2)* is the is the fraction of transactions that had a priority violation two days ago, and *HHI* is the Herfindahl–Hirschman index of daily mining shares. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

We examine how priority violations affect future fee spreads consistent with the exercise of market power presented in Hypothesis 4. We regress daily fee spreads, defined as the difference of the 90% and the 10% quantile of the daily fee distribution on the lagged fraction of transactions that face a priority violation as defined in Definition 3. From the results in Table 11 we see that past priority violations lead to higher fee dispersion. This effect is stronger when mining is more concentrated, consistent with Hypothesis 5. Overall, the finding that past priority violations lead to higher fees is consistent with rational users being induced to attach larger fees to their orders to avoid longer wait times.

4.5 Hypothesis 5: Relative Hash rate and HHI

In our formulation of the dynamic optimization problem (Equation 4), we highlighted the importance of the relative hash power. A higher relative hash rate means that a miner is more likely to produce a block and more likely to affect users' beliefs about the relationship between fees and waiting times. In what follows we consider an exogenous change in relative hash rates and how HHI (our empirical measure of hash rates) interacts with other hypotheses.

After a series of coal mining accidents, authorities closed a mine and shut down electricity in Xinjiang province on the weekend of April 17-18, 2021. This event is described in detail in Makarov and Schoar (2021), who point out that worldwide Bitcoin mining capacity dropped by 35% as many miners were without power. However, the relative hash rate of the unaffected miners increased.

For our empirical test we regress fees in USD on a dummy which is set to one after the event and a dummy set to one for all transactions in the highest fee quartile. We are interested in the coefficient of the interaction between these two dummies which measures the change in the fee for the impatient users caused that is attributable to the Xinjiang event. We present the results in Table 12. Consistent with Hypothesis 5, we find that fees increase after the Xinjiang event for impatient types. Users who pay fees in the top quartile of the fee distribution pay between

	(1)	(2)	(3)	(4)
Post closure	2.193***	4.397***	4.452***	1.517***
	(0.121)	(0.279)	(0.286)	(0.231)
High Fee	49.57***	-12.43***	-12.50***	-6.145***
	(0.827)	(1.555)	(1.561)	(1.048)
Post closure * High Fee	-0.762	39.99***	40.03***	34.19***
	(1.303)	(1.402)	(1.414)	(0.990)
Block Size (Weight)		-8.815	7.332	4.853
		(10.94)	(5.099)	(3.673)
Transaction Weight		14.52***	14.52***	12.00***
		(0.442)	(0.442)	(0.258)
Sum Inputs (USD)		11.64***	11.63***	12.77***
		(0.744)	(0.743)	(0.744)
OPRET		-0.229	-0.157	-2.584***
		(1.360)	(1.364)	(0.718)
Time until spent		-1.984***	-1.975***	-1.640***
		(0.153)	(0.152)	(0.0905)
Spent next block		7.922***	7.930***	6.092***
		(0.647)	(0.648)	(0.382)
R^2	0.0191	0.646	0.646	0.605
Observations	4,090,911	4,090,911	4,090,911	8,057,893

Table 12. Regression results of fees in USD around the Xinjiang incident. *Post closure* is is a dummy set to one post the the shutdown of electricity in Xinjiang, *High Fee* is a dummy equal to one for all fees is the top quartile of the fee distribution, *Block weight* is the average daily size of blocks measured in million weight units, *Tx weight* is the daily average weight of transactions measured in thousand weight units, *Sum Inputs (USD)* is the daily average input transaction value measured in million USD, *Data* is the fraction of daily data insertion transactions (identified by the OP_RET instruction in the script), *Resttime* is the daily average of the time (measured in thousand blocks) until transaction outputs are re-spent. Days are defined in UTC. Standard errors are clustered by miner. Columns (3) and (4) include miner fixed effects. Columns (1)-(3) compare one week before and after the event, Column (4) examines two weeks before and after the event. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

USD 34 and 40 more in fees post the Xinjian shutdown. Average fees in the two week period prior to the event were USD 20.16. Our findings are robust to miner fixed effects (column(3)) and a longer event window (column(4)).

Results from the natural Xinjiang experiment are consistent with the regressions in our tests of the other hypotheses. In all the regressions (presented in Tables 5, 7, and 8), effects are stronger if HHI is higher. Thus miners and mining concentration are crucial for the quality of Bitcoin settlement.

To further consider the effect of mining concentration we examine how it affects fee dispersion across our whole sample. We define feespread as the difference between the 90% and the 10% quantile of fees in a given block, standardized by the average fee. We choose this measure of fee dispersion over, say, a standard deviation, to reduce the influence of outliers. We then investigate how this variable is affected by mining concentration. Table 13 presents our findings. The feespread (i.e., dispersion) increases in both the HHI and mining pool's aggregate share of mining activity. This finding is consistent with Hypothesis 5.

HHI mining activity	23.85***	23.83***	12.31***			
	(2.901)	(2.901)	(4.592)			
Fraction mined by pools				9.487***	9.488***	4.990***
				(0.952)	(0.952)	(0.773)
New entrant		-0.175	-0.537***		-0.422***	-0.561***
		(0.151)	(0.190)		(0.151)	(0.188)
Post Segwit	4.829***	4.828***	3.839***	3.764***	3.762***	3.616***
	(0.193)	(0.193)	(0.225)	(0.184)	(0.184)	(0.227)
Block weight	0.677***	0.677***	0.188***	0.637***	0.638***	0.150***
	(0.0334)	(0.0334)	(0.0363)	(0.0284)	(0.0284)	(0.0376)
Average tx weight	0.0629**	0.0629**	0.620***	0.0636**	0.0635**	0.622***
	(0.0308)	(0.0308)	(0.0365)	(0.0310)	(0.0310)	(0.0362)
Sum Inputs (USD)	0.00598***	0.00598***	0.00165***	0.00531***	0.00531***	0.00137***
	(0.000457)	(0.000457)	(0.000264)	(0.000427)	(0.000427)	(0.000248)
Data	-10.10***	-10.10***	-0.999***	-9.911***	-9.911***	-0.957***
	(0.441)	(0.441)	(0.267)	(0.476)	(0.476)	(0.270)
Resttime	0.837***	0.838***	0.0184	1.071***	1.072***	0.227***
	(0.0934)	(0.0935)	(0.0768)	(0.0856)	(0.0856)	(0.0774)
Size mempool			0.0948***			0.0939***
			(0.00335)			(0.00328)
R^2	0.238	0.238	0.531	0.249	0.249	0.533
Observations	366,775	366,775	262,801	366,775	366,775	262,801

Table 13. Regression results of fee spread per block defined as the difference of the 90% and 10% quantile in USD. *HHI mining activity* is the Herfindahl–Hirschman index of daily mining shares, *Fraction mined by pools* is the daily fraction of blocks mined by identifiable pools, *New entrant* is a dummy set to one for all blocks mined within the first 30 days of a mining pool's operations, *Post Segwit* is a dummy equal to one after the introduction of Segwit, *Block weight* is the size of the block measured in million weight units, *Average tx weight* is the average weight of transactions in the block in thousand weight units, *Sum Inputs (USD)* is the average input transaction value measured in million USD per block, *Data* is the fraction of data insertion transactions (identified by the OP_RET instruction in the script) in the block, *Resttime* is the per block average of the time (measured in thousand blocks) until transaction outputs are re-spent, and Size mempool is the size of transactions waiting in the mempool measured in million weight units. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

4.6 Market Power and Implicit Collusion

In standard models of collusion, strategic firms tradeoff profit from cooperation against the payoff from one shot deviation. Typically, cooperation involves splitting the market. A firm can deviate from collusive behavior with a small price reduction and so capture the entire market. Other firms react by inflicting long term "punishment" strategies such as marginal cost pricing. If the payoff from one-shot deviation is small relative to intertemporal profits, each firm's incentive compatibility constraint is satisfied and tacit collusion can be maintained.[20]

[20]There is a large literature on tacit collusion dating back to Stigler (1964) and the more formal analysis in Green and Porter (1984).

Under the Bitcoin protocol, this incentive compatibility constraint is easily satisfied. First, the probability of mining a block is determined solely by the relative hash rate (χ in our notation). Miners therefore cannot easily increase their market share. Second, the only way to increase the profitability of a mined block is to fill it up with lower fee orders – under tacit collusion the higher fee orders would have been allocated to the block. So, the payoff to one-shot deviation is only slightly larger than the cooperative payoff.

Thus, in the Bitcoin proof-of-work protocol, the payoff to deviating payoff is low, and the payoff to tacit collusion is increasing in relative hash rates. Further, the higher the relative hash rate, the more likely is a pool to mine a future block. Thus, if a mining pool has a sufficiently high relative hash rate, strategic capacity management to increase future fees is a dominant strategy.

There are two other features of the Bitcoin protocol that facilitate tacit collusion. First, the system is transparent by design. Orders are transmitted to all nodes, and thus the mempool is effectively known by all. In addition, mining pools frequently sign the blocks that they mine. There is no system reason for doing this. However, other, unsuccessful, mining pools have a credible way of checking whether the block was mined at full capacity or under-capacity.

Second, the system provides a way for collective punishment: orphaned blocks.[21] In Bitcoin, consensus on the correct ledger is reached by blockchain length. A fork is created when two miners simultaneously add two valid blocks to an existing blockchain. Subsequently miners can add blocks to either fork but only the branch that becomes longer is recognized as the true blockchain and the shorter branch becomes "orphaned". Transactions recorded in orphaned blocks are treated as if they never happened. Rather than an inadvertent fork, miners could deliberately cause a fork by ignoring the block of a (deviating) miner and focus their efforts on another branch. With enough computing power (as in the case of a coalition of large mining pools) they can build a longer blockchain and so strategically orphan a block. The miner of the orphaned block would not only lose the fee revenue from that block but also the often more valuable coinbase, the reward for finding the block.

To test this possible disciplining channel, we manually collect data on orphaned blocks from various sites on the internet.[22] Since orphan blocks are rare we end up with a small sample of 57 orphan blocks from January 2016 to August 2019. We then compare the mining behavior of the pool, whose block was orphaned - the victim, to that of other mining pools in a window of 8,000 blocks (approximately 4 days) around the orphaned block. Victims are generally large pools, the median victim ranks third in mining share at the time of the orphaned block. In the first column of Table 14 we examine mining behavior before the orphan block. We regress blockweight of all blocks mined by large mining pools (at least 5% mining share) on a dummy for the victim. Event fixed effects control for inter-temporal variation in blockweights. Our findings are consistent with the idea that miners that deviate from strategic capacity management by mining larger blocks

[21] We thank Bruno Biais for this suggestion.

[22] For example `https://bitcoinchain.com/block_explorer/orphaned` Orphaned blocks are not consistently stored in the local database of a bitcoin node. Orphaned blocks are transmitted and thus stored in the local database as long as there is uncertainty which branch of the blockchain will succeed. Nodes do not transmit blocks that are known to be orphan. Thus longer running nodes have more orphan blocks in their local storage (see `https://bitcoin.stackexchange.com/questions/93455/why-do-two-different-fully-synced-bitcoin-core-nodes-differ-in-the-blockchain-si`).

are more likely to be victim of an orphan attack. Given the median transaction weight of 904 units, victims include about 164 transactions more per block than other large mining pools. The second column also includes data after the orphan attack. While we cannot show that that the victim reduces blocksize after the orphan block we do find that other large mining pools increase blocksize as well, consistent with the idea of a transition away from the collusive equilibrium.

Victim	148573.9***	125017.8**
	(48304.7)	(50702.0)
Post		51361.5**
		(22269.2)
Victim × Post		5322.5
		(69978.6)
R^2	0.696	0.608
Observations	354	723

Table 14. Regression explaining blockweight around orphan block events. Victim is a dummy equal to one for the mining pool whose block was orphaned. Post is a dummy equal to one after the orphan block. Dummies are included for each orphan block event. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

The price of Bitcoin itself might also determine miner behaviour. Chen, Dou, Guo, and Ji (2020) show that in profitable industries tacit collusion is easier to maintain. Firms that are further away from the distress barrier have a longer horizon and thus value the long term benefits of collusion over the short term gains from deviation. In Bitcoin mining block rewards, which are paid in Bitcoin, are a significant component of firm's profits. As Bitcoin prices move, mining capacity will adjust but given that mining farms take time to build we expect recent bitcoin returns to be positively associated with miners' profitability. To test how miners' behaviour is affected by profitability we regress the observed fee spread on the return of Bitcoin measured in USD over the previous month and present our findings in Table 15.

In line with Chen, Dou, Guo, and Ji (2020) we find that fee spreads are positively associated with past Bitcoin returns. And have some explanatory power even when controlling for miner concentration (column (3)). However, past returns become insignificant when controlling for HHI and the size of the mempool (column (4)).

If there are a large number of miners, the probability of "winning the nonce" and processing a block is small, individual miners have an incentive to fill up a block with all positive fee transactions, which would not induce strategic capacity management. It is well known that in repeated games it is more difficult to sustain collusive equilibria as the number of players increases. This suggests that mining pools provide an economic role besides diversifying risk for individual participants. By acting collectively, each mining pool effectively reduces the set of strategic players and so makes it easier to enhance revenue.

4.7 Economic impact of collusion

To approximate the economic impact of tacit collusion, we use the block-by-block observed fee distribution. Specifically, for each block we define excessive fees as the sum of all fees above the

	(1)	(2)	(3)	(4)
Return Bitcoin	5.067***	3.419***	3.433***	-0.196
	(0.671)	(0.538)	(0.535)	(0.396)
HHI mining activity			24.02***	12.40***
			(2.856)	(4.612)
Post Segwit		4.560***	4.960***	3.824***
		(0.171)	(0.191)	(0.224)
Block weight		0.484***	0.577***	0.193***
		(0.0315)	(0.0367)	(0.0369)
Average tx weight		0.0657**	0.0643**	0.620***
		(0.0320)	(0.0317)	(0.0364)
Sum Inputs (USD)		0.00609***	0.00571***	0.00165***
		(0.000477)	(0.000460)	(0.000264)
Data		-10.80***	-9.932***	-0.990***
		(0.469)	(0.428)	(0.271)
Resttime		0.634***	0.756***	0.0232
		(0.0902)	(0.0906)	(0.0757)
Size mempool				0.0950***
				(0.00349)
R^2	0.0222	0.242	0.247	0.531
Observations	366,775	366,775	366,775	262,801

Table 15. Regression results of fee spread on past bitcoin returns. The fee spread is defined per block as the difference of the 90% and 10% quantile of fees in USD. *Return Bitcoin* is the one month trailing return of the USD/BTC rate. *HHI mining activity* is the Herfindahl–Hirschman index of daily mining shares, *Post Segwit* is a dummy equal to one after the introduction of Segwit, *Block weight* is the size of the block measured in million weight units, *Average tx weight* is the average weight of transactions in the block in thousand weight units, *Sum Inputs (USD)* is the average input transaction value measured in million USD per block, *Data* is the fraction of data insertion transactions (identified by the OP_RET instruction in the script) in the block, *Resttime* is the per block average of the time (measured in thousand blocks) until transaction outputs are re-spent, and *Size mempool* is the size of transactions waiting in the mempool measured in million weight units. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

25% quantile. We note that fee distributions have a long right tail, and so our estimates are not sensitive up to the 70% quantile.

Definition 4 *Excessive fees per block are those paid above the 25% quantile.*

In Table 16 we present the results from a regression of the excess fees per block as a fraction of total fees on the HHI of mining concentration and on the fraction of blocks mined by pools. Excessive fees increase in both measures of mining concentration, consistent with capacity management. The results are statistically and economically significant. A ten percentage point increase in the fraction of block mined by pools coincides with a 3 percentage point rise in fees. An increase in the HHI of 0.05, which corresponds to a transition from 5 to 4 equally sized miners, causes excessive fees to increase by approximately 2.15 percentage points.

The sum of fees paid in all transactions is USD 2,313,958,909.48. For the whole sample these

HHI mining activity	1207.8***	5757.8***	5374.4***
	(106.9)	(151.4)	(162.3)
Mined by pool		378.9***	
		(21.07)	
Post Segwit		5743.4***	5329.8***
		(24.53)	(26.91)
Block weight (mill.weight)		733.5***	760.8***
		(6.551)	(7.020)
Average tx weight (thsd.weight)		-7.742***	-8.041***
		(1.147)	(1.136)
Sum Inputs (mill.USD)		2.994***	2.815***
		(0.0201)	(0.0202)
Data		-14362.3***	-13546.1***
		(88.15)	(88.24)
Resttime (thsd.blocks)		0.180***	0.215***
		(0.0108)	(0.0108)
Pool fixed effects	no	no	yes
R^2	0.000180	0.259	0.274
Observations	708,958	563,774	563,774

Table 16. Regression explaining excessive fees in USD. *Excessive fees* are defined as fees per block over the 25th percentile, *HHI mining* is the Herfindahl–Hirschman index of daily mining shares, *Mined by pool* is a dummy set to one for blocks mined by identifiable pools, *Post Segwit* is a dummy equal to one after the introduction of Segwit, *Block weight* is the size of the block measured in million weight units, *Tx weight* is the average weight of transactions in the block in thousand weight units, *Sum Inputs (USD)* is the aggregate input transaction value of the block measured in million USD, *Data* is the fraction of data insertion transactions (identified by the OP_RET instruction in the script), *Resttime* is the block average of the time (measured in thousand blocks) until transaction outputs are re-spent. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

excessive fees sum to USD 1,596,867,940.26. To make the result robust to outliers we re-compute excessive fees and winsorize fees per day at the 99% quantile. After winsorizing, excessive fees for entire whole sample amount to USD 1,112,045,082.65. Overall excessive fees paid due to strategic capacity management are between half and two thirds of total fees paid.

5 Conclusion

We have documented stylized facts about the Bitcoin protocol. In particular, we observe that there appears to be both excess capacity and priority violations. Indeed, a significant portion of blocks are empty or not at capacity. We note that this is consistent with revenue enhancing strategic capacity management. Indeed, the rise of fees coincided with the rise of mining pools. Given that the idea behind the Bitcoin system was to provide a completely decentralized way of transferring value based on competitive mining, the possibility that there could be collusive equilibria in proof of work raises questions about market power in decentralized finance.

Much of the computer science literature has focused on "51% attacks" as a threat to proof of work consensus. Our formulation of the objective function and analysis indicates another threshold of concern. Sufficiently concentrated hash power facilitates collusion and non-competitive behavior even if there is no possibility of a "double spend." In short, concentrated mining power less that 51% permits the exercise of market power and makes tacit collusion optimal.

The evidence we present suggests that one implementation of decentralized finance may operate in a way that is observational equivalent to traditional finance. Indeed, our analysis has highlighted the cost to the consumer of the higher fees they pay because of strategic unused capacity. However, we note that there is a positive side to these rents. Higher profits are one way to ensure that miners will view participating as a valuable exercise which ensures the continuity and stability of the Bitcoin protocol. Similar to financial intermediaries, market power and the ability to extract rents provide an incentive to continue.

References

Abadi, J., and M. Brunnermeier, 2018, "Blockchain economics," working paper, National Bureau of Economic Research.

Bajari, P., and L. Ye, 2003, "Deciding Between Competition and Collusion," *Review of Economics and Statistics*, 85(4), 971–989.

Basu, S., D. Easley, M. O'Hara, and E. Sirer, 2019, "Towards a Functional Fee Market for Cryptocurrencies," *Cornell Working Paper*.

Biais, B., C. Bisiére, M. Bouvard, C. Casamatta, and A. J. Menkveld, 2020, "Equilibrium Bitcoin Pricing," *working paper*.

Brauneis, A., R. Mestel, R. Riordan, and E. Theissen, 2018, "A high-frequency analysis of bitcoin liquidity," .

Budish, E., 2018, "The economic limits of bitcoin and the blockchain," working paper, National Bureau of Economic Research.

Capponi, A., R. Jia, and Y. Wang, 2021, "The Evolution of Blockchain: from Lit to Dark," *Columbia University Working Paper*.

Chen, H., W. Dou, H. Guo, and Y. Ji, 2020, "Feedback and contagion through distressed competition," *The Rodney L. White Center Working Papers Series at the Wharton School, Jacobs Levy Equity Management Center for Quantitative Financial Research Paper*.

Choi, K. J., A. Lehar, and R. Stauffer, 2018, "Bitcoin Microstructure and the Kimchi premium," .

Christie, W. G., and P. H. Schultz, 1994, "Why do NASDAQ Market Makers Avoid Odd-Eighth Quotes?," *The Journal of Finance*, 49(5), 1813–1840.

Cong, L. W., and Z. He, 2019, "Blockchain disruption and smart contracts," *The Review of Financial Studies*, 32(5), 1754–1797.

Cong, L. W., Z. He, and J. Li, 2019, "Decentralized mining in centralized pools," working paper, National Bureau of Economic Research.

Dae-Yong, K., E. Meryam, and J. Hongtaek, 2020, "Examining Bitcoin mempools Resemblance Using Jaccard Similarity Index," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 287–290. IEEE.

Daian, P., S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, 2019, "Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges," *arXiv preprint arXiv:1904.05234*.

Debo, L. G., C. A. Parlour, and U. Rajan, 2011, "Signallng Quality via Queues," *Management Science*, 58(5), 44–55.

Debo, L. G., U. Rajan, and S. Veeraraghavan, 2020, "Signaling Quality via Long Lines and Uninformative Prices," *Manufacturing and Service Operations Management*, 22(3), 513–537.

Denicolò, V., and P. G. Garella, 1999, "Rationing in a Durable Goods Monopoly," *Rand Journal of Economics*, 30(1), 44–55.

Easley, D., M. O'Hara, and S. Basu, 2019, "From mining to markets: The evolution of bitcoin transaction fees," *Journal of Financial Economics.*

Foley, S., J. R. Karlsen, and T. J. Putniņš, 2018, "Sex, drugs, and bitcoin: how much illegal activity is financed through cryptocurrencies?," *Review of Financial Studies, Forthcoming.*

Gilbert, R. J., and P. Klemperer, 2000, "An Equilibrium Theory of Rationing," *Rand Journal of Economics*, 31(1), 1–21.

Green, E. J., and R. H. Porter, 1984, "Noncooperative Collusion under Imperfect Price Information," *Econometrica*, 52(1), 87–100.

Hu, A. S., C. A. Parlour, and U. Rajan, 2018, "Cryptocurrencies: stylized facts on a new investible instrument," working paper.

Huberman, G., J. Leshno, and C. C. Moallemi, 2017, "Monopoly without a monopolist: An economic analysis of the bitcoin payment system," .

Kawai, K., and J. Nakabayashi, 2022, "Detecting Large-Scale Collusion in Procurement Auctions," *The Journal of Political Economy*, 130(5), 1585–1629.

Lehar, A., and C. Parlour, 2022, "Battle of the Bots: Flash loans, Miner Extractable Value and Efficient Settlement," working paper, working paper.

Liu, Q., and G. J. van Ryzin, 2008, "Strategic Capacity Rationing to Induce Early Purchases," *Management Science*, 54(6), 1115–1131.

Makarov, I., and A. Schoar, 2018, "Trading and arbitrage in cryptocurrency markets," working paper.

——— , 2021, "Blockchain analysis of the bitcoin market," working paper, National Bureau of Economic Research.

Malik, N., M. Aseri, P. V. Singh, and K. Srinivasan, 2019, "Why Bitcoin will fail to scale?," *Tepper Working Paper.*

Malinova, K., and A. Park, 2017, "Market design with blockchain technology," *Available at SSRN 2785626.*

Möser, M., and R. Böhme, 2017, "The price of anonymity: empirical evidence from a market for Bitcoin anonymization," *Journal of Cybersecurity*, 3(2), 127–135.

Pagnotta, E., 2021, "Decentralizing Money: Bitcoin Prices and Blockchain Security," *The Review of Financial Studies*, forthcoming.

Reid, F., and M. Harrigan, 2013, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, pp. 197–223.

Stigler, G., 1964, "A Theory of Oligopoly," *The Journal of Political Economy*, pp. 44–61.

Appendix

A Proofs

Proof of Proposition 1

i. Suppose that there is a solution to the per period optimization problem in which the capacity constraint does not bind. Specifically, a competitive miner has chosen $\sum_{j=0}^{m} \hat{\alpha_j^i}(t)$, such that $\sum_{j=0}^{m} \hat{\alpha_j^i}(t) g_j(t) < \kappa$. By assumption, $\sum_{j=1}^{m} g_j(t) \geq \kappa$. Thus, there exists a fee level j, and a weight ϵ so that the miner can increase $\hat{\alpha_j^i}(t)$ by ϵ and increase fee revenue by $f_j \epsilon$. Therefore $\sum_{j=0}^{m} \hat{\alpha_j^i}(t)$ was not a solution to the profit maximization problem.

ii. Suppose that there is a solution to the optimization problem in which $\alpha_j(t) > 0$, and some $\alpha_i(t) < 1$, for $i > j$. The miner could simply reduce $\alpha_j(t)$ by ϵ and increase $\alpha_i(t)$ by ϵ. Revenue would increase by $\epsilon(f_i - f_j)$, so $\alpha_j(t) > 0$, and some $\alpha_i(t) < 1$, for $i > j$ was not a solution to the optimization problem.

■

Proof of Proposition 2

Recall,

$$\Pi_t^i \quad = \quad \sum_{j=0}^{m} f_j \alpha_j^i(t) g_j(t) + \sum_{\tau=1}^{\infty} E\left[\chi^i \sum_{j=0}^{m} f_j \alpha_j^i(t+\tau) g_j(t+\tau) \mid G_{t+\tau}(h_{t+\tau}, \epsilon_\tau)\right]. \qquad (5)$$

For $\chi = 0$, the problem is that of the competitive miner, whereas for $\chi = 1$, the problem is of a monopolist miner. The result follows. ■

B Eliminating Technical or Mechanical reasons for unused capacity

Difficulty: Mining a full and empty blocks are equally difficult. Miners hash over data that include the root of the Merkle tree that contains all transaction information. The size of this root is independent of the number of transactions in the block.

New transactions: Miners can change the set of transactions at any time during the mining process and changes do not affect the probability of finding a valid block. So, if a higher fee transaction arrives before the correct nonce is found, the miner could replace a low fee transaction without affecting the probability of finding a valid nonce. Miners do not have to keep block space set aside should higher fee transactions arrive during the mining process.

Time to verify: Miners have to compile a candidate block from transactions in the mempool on which they want to mine. Transactions that are in the mempool have to be verified before being included in a candidate block. Verification includes, for example, a check of the signature and processing of the script. Verification is somewhat computationally expensive but is completed as transactions enter the mempool and thus before a candidate block is composed. At the time a new candidate block is compiled only relatively trivial consistency checks have to be performed which take usually less than 1 millisecond.[23] Overall the process from block discovery to compiling a candidacy block for mining transpires in less than a second.

Network Latency: When a new block is mined it has to be transmitted to other nodes so that they know the block's hash which they have to include in their own block. Miners also have to know which transactions have been included in the previous block so that they do not include the same transactions in their own block. Anecdotal evidence suggests block validation times for fast hardware to be 45 milliseconds.[24] Because of special high speed connections between miners latency has been dramatically reduced in the bitcoin network. Most miners participate in Fibre (Fast Internet Bitcoin Relay Engine), which is a special network protocol started in 2016 to deliver Bitcoin blocks around the world with delays as close to the physical limits of signal transmission as possible. In addition, compact blocks, as outlined in Bitcoin Improvement Proposal (BIP) 152 have drastically reduced block transmission times. At the time of writing the paper, the median transmission time of blocks is on the order of 5 milliseconds.[25]

There is no direct connection between the probability of a block being empty and the time elapsed since the previous block was mined, which would be the case of latency was a problem. Starting in 2013 we find that the average time after which an empty block was mined to be 9.66 minutes compared to 9.42 minutes for a full block. We find 45,459 non-empty blocks mined within less than minute, and 2,159 non-empty blocks mined within less than 5 seconds. Similarly we find 2,048 empty blocks mined more than 10 minutes

[23]see e.g. `https://bitcoin.stackexchange.com/questions/84045/block-verification-time`.

[24]See e.g., `https://bitcoin.stackexchange.com/questions/50349/how-long-does-block-validation-take`.

[25]See data from `http://bitcoinfibre.org/stats.html`

after their predecessor. In Subsection 4.5 we provide evidence that instead of occurring randomly, empty blocks are correlated with recent capacity usage.

Ethereum (pre-Merge) operated as proof of work with mining pools, a mempool, and transactions that need to be confirmed, yet it is designed to add a new block to the chain every fifteen seconds.[26] Transaction validation is more complicated on Ethereum as transactions can be complex programs that change the state of the virtual machine. Ethereum demonstrates that its is technically possible to add blocks with transactions within a short period of time.

Specific Miners: Our results on empty blocks and excess capacity are consistent over time and across pools and so it is unlikely that they are due to random technical problems or due to specific miners. When confronted over mining empty blocks Jihan Wu from Antpool tweeted in 2016 'sorry, we will continue mining empty blocks. This is the freedom given by the Bitcoin protocol.'

C Block capacity post Segwit

In the original design, Satoshi Nakamoto introduced a 1MB limit to Bitcoin blocks. For technical reasons, however, effective block size was smaller until May 15, 2013.[27] An upgrade, Segregated Witness (Segwit), was implemented on August 24, 2017. Briefly, Segwit is a way to store signatures and scripts associated with compliant transactions in a special area of a block (the witness section).[28] We emphasize that adoption of this technology was voluntary and Bitcoin users slowly converted to the new system.

It is important to note that post Segwit size in bytes is neither an accurate measure of block capacity nor of transaction size. A fully compliant Segwit transaction takes about 25% the space (in bytes) of a traditional transaction because some components such as scripts are outsourced to the witness area and thus not part of the official block. However if coins are spent from an address locked up before the roll-out of Segwit, the full features of Segwit cannot be used and hence such transactions cannot take full advantage of the capacity increase. In short, they take up more space. Segwit did not quadruple capacity: As pre-Segwit transactions are replaced with Segwit compliant transactions, block capacity gradually increased. To deal with this heterogeneity in transaction types, the concept of transaction weight was introduced with Segwit.

[26] We downloaded data on over a million Ethereum blocks ranging from block 10,000,000 to 11,722,614. The average time between blocks is 13.32 seconds, the median is 9 seconds. 25% of blocks are mined within four seconds.

[27] Block size was limited by the number of database locks required to process a block (at most 10,000). This limit translated to around 500-750 thousand bytes, and was forgotten until March 11, 2013, when an upgrade to V0.8.0 with a switch of databases caused an unplanned fork in the blockchain. After resolving the crisis, the community reached a consensus to remove this unknown limit and a hardfork was scheduled and cleanly activated on May 15, 2013. Subsequently, for the first time, 1MB became the effective maximum block size. Details of this system change are available at https://en.bitcoin.it/wiki/Block_size_limit_controversy,https://blog.bitmex.com/bitcoins-consensus-forks/

[28] The first block exceeding 1MB limit was block 481,947 mined on Aug 25, 2017 with a size of 1032 KB.

A big part of the physical space that transactions take up in a block are the locking and unlocking scripts. Segregated Witness (Segwit) compliant transactions outsource these scripts into a separate data structure, the witness. The witness structure is organized as Merkel tree, a data structure where leaves hold data and each node is a hash of the underlying nodes. The root of the tree is linked to the Bitcoin block by including the root-hash in the coinbase transaction.

Segwit transactions are designed to be backward compatible. There are two basic types of Segwit transactions, Pay-to-Witness-Public-Key-Hash (P2WPKH) and Pay-to-Witness-Script-Hash (P2WPSH). In the former the locking script is marked as Segwit by including the Segwit version number (currently 0) followed by a 20 byte hash of the public key. The signature and the full public key required for unlocking the Bitcoin are outsourced to the witness block. For P2WPSH transactions the locking script consists of the version number followed by a 32 byte hash of the unlocking script. These transactions are also often referred to as Bech32 transactions. A non-native and inefficient way of implementing Segwit transactions is to embed them in a classic Pay-to-Script-Hash (P2SH) transaction.

Outsourcing part of the transaction to the witness section reduces the amount of effective space a transaction takes up in the block. The measure of transaction size in bytes includes both the transaction and the witness data to make the measure comparable to pre-segwit transactions. For most purposes, e.g. to measure capacity use, the transaction size in bytes is not a useful measure because segwit-, partial segwit, and non-segwit transactions can be included in a block. To address this problem Bitcoin introduced a measure of transaction "weight." The weight of a transaction that does not take advantage of segwit is 4 times its size in bytes. The weight for a fully segwit compliant transaction is obtained by multiplying components that are part of the block (inputs, outputs, input- and output counts, version, and lock-time) by 4 and multiplying witness components by 1 and then adding up the weighted components. In our sample the weight is between 1.2 and 4 times the size in bytes.

The Segwit update did not have a big impact on variables of economic interest. The blue line in Figure 10 shows that the introduction of SegWit brought no immediate increase in capacity. The average weight per block stays between 3 and 4 MB, the latter being the maximum amount. The reason that Segwit brought no sharp increase unused transaction capacity is because of its slow adoption. The red line shows the fraction of transactions that use some Segwit features. Adoption is slow peaking at 15% after fifty days. With most transactions using the pre-Segwit format not much new capacity on the blockchain is being created. The green line illustrates the total fee revenue per block. While there is a peak around the introduction of Segwit the variation in fee revenue per block seems of similar or smaller magnitude than other variations in total fee revenue. It seems that there is no unusual variation in miners' fee revenue around the Segwit inroduction.

D Mempool data

We collect two sets of mempool data to examine transaction demand for Bitcoin. The partially aggregated dataset is used for the money-left-on-the-table calculation in Section 3, the detailed mempool data is used in Section 3.1.2.

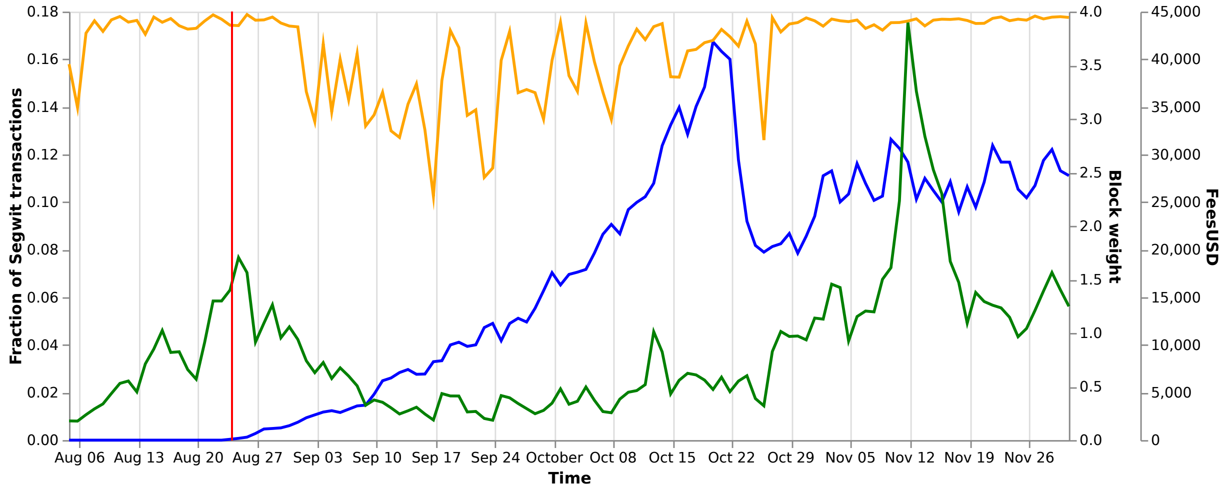

Figure 10. Average Weight per block in million weight units (orange), fraction of Segwit Transactions (blue), and Fee Revenue in USD per block 20 days before to 100 days after the introduction of Segwit (green). Days are defined over UTC.

D.1 Partially aggregated data

We collect minute by minute snapshot data of the mempool from Jochen Hoenicke's website, `https://jochen-hoenicke.de/queue/#0,all`. The data ranges from Dec 16, 2016 to the end of our sample period. For each snapshot, transactions are grouped into 46 fee buckets based on sat/byte and contain for each bucket the number of transactions in that bucket at that time, the sum of fees offered by all transaction in that bucket, and the size of all transactions in the bucket. The sample contains over 2.67 million snapshots with a total of 122.63 million time/bucket observations. There are some gaps in the data, most likely because outages of the server collecting the data. Out of 2.67 million snapshots we observe 7,914 snapshots that are more than 70 seconds apart, with the longest gap being 35 hours.

We match mined blocks to mempool data based on the timestamp that the block was mined and by looking for sharp drops in the size and the number of transactions in the pool. We identify these drops as blocks being mined. We cannot reconcile blocks based on the timestamp alone, as timestamps of blocks are sometimes inaccurate. We therefore have a record of the mempool immediately after a block was mined. For our estimate of money left on the table we start filling any empty blockspace with transactions from the highest fee/byte bucket, until we exhaust this bucket and so forth until the block is full.

Because mempool data is specific to each node, any individual miner may face a different mempool. However, we note that transactions which enter the mempool are shared via peer-to-peer communication. We expect that miners have better hardware, faster connections, and are connected to more peers than our data source. Therefore we provide a conservative estimate of the money miners appear to leave on the table.

D.2 Detailed data (collected from our node)

We set up our own Bitcoin node and collect the precise composition of the mempool on a transaction level for a subsample from block 620,591 to block 708,957 or from March 7, 2020 to November 9,2021. We restrict our sample to 149,506,850 transactions that were eventually mined. This is conservative as some transactions with positive fees, which could be priority violations, were never included in a block and thus purged from the mempool. For each transaction we observe precisely when it entered the mempool, its weight, the fee, any dependencies on other unmined transactions, and if and when it was eventually mined. We also collect information on the weight, time, and transaction count of the mined blocks.

Most miner initiated transactions are not in our sample because these transactions will not show up in the public mempool. Instead, miners directly include those transactions in the blocks they mine. To capture any remaining transactions that could be associated with miners we collect all addresses from miners over the whole sample using coinbase transactions, i.e. those are the addresses that the block rewards were paid to. We eliminate 20,420 transactions that have a miner address as input from our sample. We also eliminate 25,080,799 dependent transactions from our sample as they might also have lower fees and thus bias our results.

E Block Characteristics and fees

In Tables 17 and 18, we regress fees in Satoshis and USD respectively on transaction characteristics. Fees are higher when transaction space is scarce (larger blocksize weight) and when the transaction is larger in weight (Transaction Weight) and value (Sum Inputs). Transaction size has an economically significant impact on fees. Specifically, adding one input (a typical segwit compliant input has a weight of 344) to a transaction (the average transaction has 2.5 inputs in our sample) increases the fee by 595 Satoshi or USD 0.05. While this may sounds low in absolute terms it is relatively high given that the median fee over the whole sample is USD 0.33. The dollar value of the transaction (SumInputs), for example, only has a small and economically insignificant effect on fees, which is consistent with the fact that the miner's opportunity cost for including a transaction in a block of limited size is determined by the transaction size and independent of the value. We find that data-insertion transactions post lower fees in BTC. Fees are also higher when the funds are spent sooner (lower rest time) and especially if the output was spent in the next block. Resttime is measured in blocks, which are mined every 10 minutes on average. People spending their funds in the next block pay on average USD 0.47 (more than the median fee) more, and users spending a week ($\sim$1008 blocks) earlier pay on average 26 cent more in fees. Receivers that are keen to spend their coins sooner put a higher value on the execution. Consistent with discriminatory service those users pay higher fees as miners are able to price discriminate by delaying users that put a low priority on execution.

We stress that our findings are not driven by the peaks in Bitcoin prices in 2017 and 2021. The last column of each table shows the results for the subsample excluding all transactions excluding start of Nov 2017 to end of January 2017, start of February 2021 to end of June 2021, and after October 1st 2021.

	Whole Sample						w/o Peak
Block Size (Weight)	0.000830***					0.000799***	0.000782***
	(0.0000104)					(0.00000998)	(0.0000101)
Transaction Weight		0.175***				0.173***	0.172***
		(0.000428)				(0.000425)	(0.000493)
Sum Inputs (Sat)			0.223***			0.156***	0.136***
			(0.000632)			(0.000548)	(0.000537)
OPRET				-3.889***		-2.268***	-2.664***
				(0.0434)		(0.0421)	(0.0395)
Spent next block					5.997***	5.697***	5.035***
					(0.0205)	(0.0198)	(0.0174)
Resttime					-0.00368***	-0.00378***	-0.00321***
					(0.0000213)	(0.0000200)	(0.0000204)
R^2	0.336	0.434	0.342	0.336	0.339	0.440	0.382
Observations	685,406,634	685,406,634	685,406,634	685,406,634	685,406,634	685,406,634	604,128,793

Table 17: **Regression results of fees in thousand Satoshis (1 BTC is 100 million Satoshis).** *Block weight* is the size of the block measured in hundred thousand weight units, *Transaction Weight* is the size of the transaction measured in thousand weight units, *Sum Inputs (BTC)* is the sum of input values for the transaction measured in BTC, *Data* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, *Resttime* is the average time (measured in blocks) until transaction outputs are re-spent. Regressions include day fixed effects. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

	Whole Sample						w/o Peak
Block Size (Weight)	0.0000616***					0.0000607***	0.0000495***
	(0.000000804)					(0.000000751)	(0.000000700)
Transaction Weight		0.0148***				0.0142***	0.0130***
		(0.0000388)				(0.0000375)	(0.0000407)
Inputs (USD)			0.0000111***			0.00000930***	0.00000742***
			(2.36e-08)			(2.18e-08)	(1.83e-08)
Data				-0.161***		0.0689***	-0.0880***
				(0.00369)		(0.00365)	(0.00257)
Spent next block					0.565***	0.472***	0.302***
					(0.00238)	(0.00217)	(0.00131)
Resttime					-0.000360***	-0.000268***	-0.000118***
					(0.00000230)	(0.00000208)	(0.00000149)
R^2	0.516	0.567	0.532	0.516	0.518	0.580	0.364
Observations	685,406,634	685,406,634	685,406,634	685,406,634	685,406,634	685,406,634	604,128,793

Table 18: **Regression results of fees in USD.** *Block weight* is the size of the block measured in thousand weight units, *Transaction Weight* is the size of the transaction measured in hundred weight units, *Sum Inputs (USD)* is the sum of input values for the transaction measured in USD, *Data* is a dummy set to one of a transaction inserts non-transactional data (identified by the OP_RET instruction in the script), *Spent next block* is a dummy set to one an output of a transaction was re-spent within one block, *Resttime* is the average time (measured in blocks) until transaction outputs are re-spent. Regressions include day fixed effects. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.