NBER WORKING PAPER SERIES

UNRULY BY DESIGN:
FEE VOLATILITY AND STRATEGIC ATTACKS IN BITCOIN MINING

Fabian Schär
Dario Thürkauf
David Yermack

Working Paper 35443
http://www.nber.org/papers/w35443

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
July 2026

Unruly by Design: Fee Volatility and Strategic Attacks in Bitcoin Mining
Fabian Schär, Dario Thürkauf, and David Yermack
NBER Working Paper No. 35443
July 2026
JEL No. G23

ABSTRACT

We develop a model of aberrant behavior by Bitcoin miners and test it with a new 2017-2025 dataset. Miners' rewards, comprised partly of user fees, exhibit variability across blocks of transactions. When large reward disparities exist between adjacent blocks, miners have incentives to attempt alternative versions of prior blocks and claim other miners' rewards for themselves. Regression analysis shows that fee differentials are associated with these attacks and longer waiting times between blocks. These patterns imply potential destabilization of the Bitcoin blockchain as future mining rewards become more volatile due to gradual withdrawal of fixed block subsidies.

Fabian Schär
University of Basel
f.schaer@unibas.ch

Dario Thürkauf
University of Basel
Faculty of Business and Economics
dario.thuerkauf@unibas.ch

David Yermack
New York University
Leonard N. Stern School of Business
Finance
and NBER
dyermack@stern.nyu.edu

1 Introduction

Since 2009 Bitcoin has conducted a remarkably successful payments and money transfer system that moves funds more rapidly, more securely, more transparently, and at least for larger transactions, more cheaply than the legacy financial system. Bitcoin operates continuously over a global, decentralized computer network with no barriers to participation other than an Internet connection, and without any leadership, regulation, or even a user manual. Governance occurs fully by algorithms embedded in open-source code, and any disputes or ambiguities about transactions are resolved by execution of the software across the network of nodes, a governance model known as "code is law."

Our paper investigates a type of aberrant behavior on the part of Bitcoin miners, the agents who earn rewards for doing the work of adding new blocks of transactions to Bitcoin's blockchain ledger. Because miners' rewards are comprised partly of transaction fees paid by individual senders, certain blocks on the blockchain will contain significantly greater rewards than others. When a large disparity exists between the rewards in the prior block and the potential rewards available for mining the next block, it may pay miners to try to create a competing version of the prior block. If successful, the multiple versions of the same block would represent a fork on the blockchain, and other miners would have to decide which branch of the fork to extend, launching a so-called block race.

The proclamation of Nakamoto (2008), a short document published about two months before the launch of the Bitcoin network, represents the closest thing to authoritative guidance for the behavior of miners. The anonymous author introduces the concepts of "honest mining" and mining "attacks" and conjectures that Bitcoin will achieve a stable, self-enforcing equilibrium, in which miners will find it more profitable to attempt extending the blockchain rather than attacking previous blocks. However, Nakamoto's discussion focuses on double-spending attacks, in which an agent exchanges a coin for valuable consideration and then attempts to overwrite the relevant block to lay the groundwork for spending the coin

again. Nakamoto does not appear to have anticipated predatory behavior by rival miners eyeing the rewards in one another's blocks, and the Nakamoto (2008) document provides no clear definition of what represents honest behavior or an attack. Even if Nakamoto's manifesto offered clear guidance, under the code-is-law paradigm there are no administrative, regulatory, or reputational penalties to constrain miners from attacking existing blocks when incentives make this behavior attractive.

Successful Bitcoin miners receive compensation from two sources: a block subsidy, consisting of newly minted bitcoins, and voluntary user transaction fees. The block subsidy is effectively financed by all bitcoin holders through dilution, whereas transaction fees are paid by the thousands of spenders whose transactions the miner selects to include in the block. Because Bitcoin's algorithm reduces the block subsidy by half every 210,000 blocks, roughly every four years, over time transaction fees will become the dominant source of miner compensation (Figure 1).

This structural shift from subsidy-based to fee-based miner compensation has sparked an active economic debate about Bitcoin's long-run security. Prior research focuses mainly on whether average transaction fees will be sufficiently high to sustain the amount of mining investment required to secure the blockchain as the block subsidy tapers off toward zero (Budish, 2025; Garratt and van Oordt, 2023; Harvey et al., 2025; Hinzen et al., 2022). If long-run transaction fee income proves too low, miners' incentive to provide computational resources will weaken, reducing the cost of attacking the network and jeopardizing settlement integrity. Budish (2025) formalizes the central constraint, namely that the recurring flow cost of honest mining must remain large at all times relative to the one-off gain from a majority attack, a condition that tightens as the subsidy, the most predictable component of that flow, declines. Hinzen et al. (2022) show that fork risk under network delay endogenously caps the throughput, and hence the fee revenue, that the system can generate, while Garratt and van Oordt (2023) argue that sunk costs in mining hardware anchor hashrate and partly offset the declining subsidy.

This literature evaluates blockchain security through the expected level of

fee revenue and largely abstracts from its variance. However, transaction fees differ from the block subsidy due to their volatility, which reflects the short-term supply and demand for limited transaction space. In practice, user fees depend on the state of the mempool, the buffer of unconfirmed transactions awaiting inclusion, and they can vary sharply from one block to the next.

A second strand of research studies the potential distortion in miners' incentives induced by reward volatility, recognizing that block-to-block variations in user fees may cause miners to deviate from honest chain extension. The first to identify such incentive-compatibility issues were Eyal and Sirer (2013), who show in a theoretical model that miners who strategically withhold blocks ("selfish mining") can earn more than their proportional share of rewards. Carlsten et al. (2016) tie this concern to fee variability, showing that in a fee-only regime the high variance of fee income makes it attractive to fork a wealthy block rather than extend the chain. Subsequent work refines this insight under block-size limits (Gong et al., 2022), with reinforcement-learning solvers (Bar-Zur et al., 2022; Sarenche et al., 2025), and under general stochastic reward functions (Bahrani et al., 2025), consistently finding that higher reward variance lowers the hash-power threshold at which strategic behavior becomes profitable. This work is almost entirely theoretical or simulation-based and stops short of testing whether the predicted distortions appear in realized mining behavior.

Our paper studies miners' behavior using a new dataset with more than seven years of transactions, a far greater sample with more detail than has been available to prior authors. We examine how miners' incentives evolve as Bitcoin transitions from a stable subsidy regime toward a fee-dominant environment characterized by high reward variance. We develop theoretical and empirical evidence that fee volatility can induce strategic behavior that undermines settlement finality and weakens coordination on the canonical chain. When the fee differential between the most recently mined block and the next high-value set of pending transactions becomes large, miners may rationally deviate from honest chain

extension and instead attempt to replace recent blocks to capture those blocks' fees. In economic terms, such conduct represents short-horizon rent seeking, and it could destabilize Bitcoin by increasing the frequency of chain reorganizations, eroding confidence in transaction finality, and generating inefficient investment in and allocation of mining resources.

To formalize this mechanism, we develop a model of miner incentives in which the profitability of continuation is compared with the profitability of attempting to re-mine a recent block. The model highlights a simple but important force: large fee differentials across adjacent blocks raise the private return to deviation. It also yields a sharp dynamic implication. The incentive to attack should be strongest immediately after a high-fee block is mined, when the reward gap between replacing that block and extending the chain is largest, and it should diminish over time as the mempool refills and the reward differential narrows. The framework implies that settlement fragility should become increasingly salient as the fixed block subsidy declines and fee revenue accounts for a greater fraction of miner income.

We test these predictions using a novel empirical design that combines blockchain records with mempool observations from multiple nodes. This allows us to estimate, at the block level, the rewards available under two competing strategies: continuation mining on the current chain tip and deviation in which a miner attempts to replace the most recent block and capture its fees. Using these counterfactual payoffs, we construct a measure of the incentive to attack rather than extend the chain.

The empirical evidence indicates that larger fee differentials are associated with a higher incidence of block races. This finding is robust to a wide range of specifications and an extensive battery of robustness checks. To corroborate this finding with a complementary design that does not rely on the incompletely observed block race outcome, we examine settlement dynamics in a survival analysis framework, asking whether fee differentials predict the time until the next block is mined. We find that a larger spread between the payoff to continuation and the payoff to attacking is associated with a significant increase in the duration until the

next block arrives. Consistent with the model, this effect attenuates over an interval of about 30 seconds as new transactions refill the mempool and the potential fee differential declines.

In studying the intertemporal variance of user fee rewards across blocks, our model derives a closed-form hurdle rate that links the fee differential between adjacent blocks to the profitability of strategic deviation; the hurdle rate yields testable predictions and identifies the block subsidy as a security buffer whose erosion lowers the threshold for profitable attacks. We then provide block-level empirical evidence that fee-differential incentives already shape miner behavior under the current subsidy regime, using a design that reconstructs the competing-strategy payoffs each miner faces from blockchain and mempool data. These findings have direct implications for the settlement finality on which Bitcoin-based financial products increasingly rely, and they highlight the risk of network destabilization as the fixed block subsidy gradually declines toward zero.

The remainder of the paper proceeds as follows. Section 2 provides institutional and technical background on Bitcoin mining and the determination of block-level rewards. Section 3 introduces the theoretical framework and derives the core empirical predictions that guide the analysis. Section 4 describes the underlying blockchain and mempool data, outlines the construction of the raw sample, and explains the computation of the key variables used in the empirical analysis. Section 5 presents the main empirical results. Section 6 discusses the broader implications of the findings for Bitcoin's long-run settlement security, and concludes.

2 Bitcoin Mining

Bitcoin mining refers to the process by which agents append new blocks to the blockchain and earn rewards in return. Mechanically, mining means attempting to create a Bitcoin block header that has a hash value less than or equal to a specific target defined by the network difficulty. Because cryptographic hashes are effectively unpredictable, a candidate

block header value cannot be computed analytically and must instead emerge from a large number of trial and error attempts, during which the miner repeatedly varies the header until one yields a hash below the target. Miners across the network conduct this search in parallel, and when one miner succeeds, that miner appends the new block to the tip of the chain and claims its reward. Ultimately the reward requires the implicit consensus of other miners on the network, which they demonstrate if they discontinue working at the height of the newly discovered block and shift their efforts to extend the blockchain on top of it.

Miners construct block headers by sequencing the Merkle root[1] of the included transactions along with a 32-bit nonce[2] and several other fields.[3] As hardware efficiency has increased over time and the Bitcoin algorithm has correspondingly increased the difficulty of mining, trial and error attempts based on using every possible value in the nonce space (2^{32} possibilities) have become insufficient as a source of variation in the block header, as miners will easily exhaust every possible nonce in less than a second. To provide additional entropy, miners generate an "extra-nonce" located in the coinbase transaction.[4] Altering the coinbase transaction changes the Merkle root in the block header, effectively allowing miners to re-use the nonce space.

Along with the fixed block subsidy, a miner also collects the fees attached to the transactions it chooses to include in its block. Pending transactions are broadcast to the network by users and accumulate in the mempool, a buffer set of transactions that have been announced but not yet confirmed in a block. Because each block can hold only a limited amount of data,

[1]The final hash value produced by recursively hashing pairs of transactions in a block until a single root hash remains.

[2]The nonce, short for "number used only once," is a source of entropy that allows miners to iteratively modify the block header hash without changing the underlying transaction set or other predetermined header fields.

[3]The remaining header fields are the version, the hash of the previous block, a timestamp, and the difficulty target (nBits).

[4]The coinbase transaction is the special first transaction in a block through which the miner claims the block reward, including newly issued bitcoins and the transaction fees paid by the transactions included in the block. It should not be confused with the company Coinbase, Inc.

a miner cannot include every pending transaction; it should generally prioritize those offering the highest fee rate, which is paid per unit of block space. Competition between users for the limited space available in each block determines the level of fees, which rises and falls with transaction demand and can therefore differ substantially from one block to the next, in contrast to the fixed block subsidy.

The stochastic nature of the mining process introduces significant reward variance for individual participants, particularly those with a small share of the total hashrate (the rate of hashes computed per second, which determines a miner's probability of winning the next block). To mitigate this income volatility, miners typically aggregate their computational power into mining pools, where rewards are distributed proportionally to each member's contributed work. Mining pool operators coordinate the work of affiliated miners via the Stratum protocol, the de facto industry standard. The pool operator constructs the block template that workers mine on, including the choice of which transactions to include, so individual miners within a pool do not select their own transaction set and effectively share the same candidate block.[5] Because pool operators determine both the transactions and the block height that workers mine on, individual miners have limited autonomy. We therefore assume that all miners within a given pool work on the same block template.

In recent years, the mining landscape has transitioned to a highly consolidated environment dominated by a few large pools, such as Antpool and Foundry USA (Figure 2). This trend is also illustrated quantitatively through the Herfindahl-Hirschman Index (HHI), which shows a significant rise in market concentration starting around 2022.

[5]Stratum V1 (Palatinus, 2012) distributes work without requiring each miner to maintain a full mempool: workers receive the metadata needed to construct a valid header and a coinbase split into two parts, between which they insert their assigned `ExtraNonce1` and chosen `ExtraNonce2` to produce a unique Merkle root. Pools maintain different update intervals and "clean job" policies, the latter instructing workers to abort current work and switch templates (typically when a new block is found on the network). Stratum V2, designed to decentralize transaction selection, remains limited in adoption as of 2026, with V1 still governing the majority of network hashrate.

3 Model

We develop a theoretical framework to study the choice faced by a miner who receives a signal from the network that a different miner has successfully mined a new block. Our miner must decide whether to accept the new block and attempt to extend the chain one block further, or alternatively, whether to continue mining at the previous blockchain height and attempt to create a rival block. If successful in the latter case, our miner would create a fork in the blockchain, and other miners would have to choose which version of the chain they would attempt to extend. If the set of miners divides itself between the two branches of the chain, a so-called block race ensues, and the chain that extends first will almost certainly become the canonical chain or main chain.

Our model identifies conditions under which a rational, profit maximizing miner will ignore the implicit intent of the Bitcoin protocol and not attempt to build upon the new block, but rather create a competing block at the same height. We will refer to this as an "attack." Building on elements of Carlsten et al. (2016) and Gong et al. (2022), our model endogenizes three main features: (i) time-varying mempool dynamics that erode the attacking incentive, (ii) block propagation frictions, and (iii) the attacker's ability to construct a competing block that differs from the original. The model yields a closed-form hurdle rate for the fee differential that may trigger an attack, expressed as a function of the miner's hashrate, the block subsidy, and the propagation state. This hurdle rate generates directly testable hypotheses for our empirical analysis.

3.1 Environment and Notation

Consider a proof of work blockchain operating within a single difficulty epoch, so that the difficulty target D and the block subsidy S are constant. We normalize total hashrate to unity ($H = 1$) and distribute it among three agents: the strategic attacker, with share β_A; the miner who proposed the latest block, with share β_0; and the rational fringe of

remaining miners, with share $\beta_R = 1 - \beta_A - \beta_0$. We assume no single miner controls a majority, that all miners are rational, and that they seek to maximize short-term expected profits. The protocol does not exclude the possibility for a miner to mine on top of any valid parent block, so attacking the chain tip is permissible. An attack leads to no regulatory or reputational consequences for the attacker due to the decentralized nature of the protocol. We denote the chain, extending the tip at height $h-1$ through a block B, by C^0, and a potential alternative chain proposed by an attacker at the same height by C^A.

At height h, another miner discovers block B_h and broadcasts it to the network; we set $t = 0$ at this moment of discovery, so that t measures elapsed time since discovery. Our miner observes B_h during the propagation window that follows (Section 3.3) at an instant on this clock that we denote $\tau \geq 0$, and must then decide whether to attack. Let F_h denote the transaction fees actually included in B_h by the other miner, and let $F_h^*(t)$ denote the maximum transaction fees attainable for a block at height h given the mempool state at time t. The other miner's total reward for B_h is therefore $R_h = S + F_h$.

Notably, the miner of B_h need not have built a fee-optimal block: a stale template or omitted high-fee transactions leave $F_h < F_h^*(0)$. An attacker constructing a competing block B_h^A at the same height may therefore assemble a more profitable set of transactions than the miner of B_h. The attacker chooses transactions yielding a fee total F_h^A at construction time τ, subject to the mempool ceiling $F_h^A \leq F_h^*(\tau)$.

3.2 Mempool Dynamics and the Fee Differential

Transactions arrive quasi continuously. For any time $t > 0$ after the discovery of B_h, define the mempool fee differential:

$$\Delta(t) \;\equiv\; F_h^*(t) \;-\; F_{h+1}^*(t) \tag{1}$$

where $F^*_{h+1}(t)$ represents the maximum fee revenue for an extension at height $h+1$, conditional on the optimal block at height h. That is, the transactions removed from the mempool when computing $F^*_{h+1}(t)$ are those that would appear in $F^*_h(t)$, not the transactions actually included by the other miner. Since the optimal block at h is a subset of the mempool, and these transactions cannot also appear at $h+1$, the set available to an attacker at height h is a superset of those available for $h+1$. Hence, a non-negative fee differential exists between the fee-maximizing blocks at height h compared to $h+1$, and we express this differential as $\Delta(t) \geq 0$.

Further, we define the realized fee differential as:

$$\tilde{\Delta}(t) \;\equiv\; F^*_h(t) \;-\; \tilde{F}^*_{h+1}(t) \tag{2}$$

where $\tilde{F}^*_{h+1}(t)$ equals the maximum fee revenue for an extension at height $h+1$ conditional on the actual transactions in B_h having been removed from the mempool. Unlike $\Delta(t)$, this measure depends on the initial miner's transaction selection: if the miner omits certain high-fee transactions, then they remove fewer valuable transactions from the mempool, raising $\tilde{F}^*_{h+1}$ and compressing $\tilde{\Delta}$.[6] Since that miner's behavior may itself have responded to fee conditions, $\tilde{\Delta}(t)$ is potentially endogenous. By contrast, $\Delta(t)$ is determined entirely by the mempool state and the protocol's block-weight limit.

As new high-fee transactions enter the mempool, $F^*_h(t)$ and $F^*_{h+1}(t)$ (equivalently $\tilde{F}^*_{h+1}(t)$) increase, but the latter benefits more because fresh transactions are equally available to either strategy while the fee advantage from B_h's confirmed transactions erodes. We model this delta decay as:

$$\Delta(t) \;\approx\; \Delta(0) \cdot e^{-\delta t}, \qquad \delta > 0 \tag{3}$$

The fee differential decay rate δ depends on the transaction arrival intensity, the transaction fee rates paid by users, and the share of mempool

[6] A miner that does not claim the maximum available fees may be engaging in so-called "undercutting avoidance" (Gong et al., 2022).

fees locked in B_h. This temporal erosion implies that the attacking incentive is strongest immediately after B_h arrives and dissipates as the mempool refills. Figure 3 illustrates the mechanism conceptually.

3.3 Block Propagation

Let $\gamma(t)$ denote the fraction of network hashrate that has received and validated B_h by time t:

$$\gamma(t) \;=\; 1 - e^{-\lambda t}, \qquad \gamma(0) = 0, \quad \lim_{t\to\infty} \gamma(t) = 1 \tag{4}$$

where $\lambda > 0$ governs propagation speed; its reciprocal $1/\lambda$, the relay time constant, is the time by which a fraction $1 - e^{-1} \approx 63\%$ of hashrate has received the block. In practice, $1/\lambda$ is on the order of 2–12 seconds, so propagation is largely complete within tens of seconds, but the initial ramp-up creates a window during which a significant fraction of miners continue to mine on top of the predecessor of B_h, as most will learn of the new block after a minor time delay due to latency in transmission of network signals. The propagation function is critical because it determines the fraction of the network that is available to mine B_{h+1}, directly affecting the probability that the incumbent chain extends before any potential attacker finds a competing block.

A useful property of modern Bitcoin block relay is that λ is largely orthogonal to the fee composition of B_h (or any block). Under the Compact Blocks enhancement to the Bitcoin protocol (BIP 152) that took effect in August 2016, nodes propagate a fixed-format header plus 6-byte short transaction identifiers; the message body is dominated by a count of transactions rather than by their cumulative weight or fee, and receivers reconstruct the block from their own mempools without further round trips except for transactions that were not previously seen. Under FIBRE (Fast Internet Bitcoin Relay Engine), in effect since December 2017, headers and forward-error-corrected transaction chunks propagate with weight-independent latency once peering links are warm. Both protocols imply that propagation latency is driven by network topology, peer

connectivity, and the count of unknown-to-receiver transactions, not by the magnitude or composition of fees in B_h. The fee differential $\Delta(t)$, by contrast, is a function of mempool composition that evolves through transaction arrivals on a timescale orthogonal to relay topology. We exploit this separation in the empirical analysis: residual variation in propagation speed cannot be a confounder for the fee differential effect except through observable channels (block size, miner hashrate share, and private transaction count), each of which we control for explicitly.

3.4 The Attack Game

Our miner chooses between two strategies, which we will call *Continue* and *Attack*. Under the *Attack* strategy the game proceeds in two phases. In **Phase 1**, the attacker races to find a competing block B_h^A before the original chain extends to $h+1$. If successful, B_h^A is broadcast, initiating **Phase 2**: a block race between B_h on C^0 and B_h^A on C^A, resolved when one chain is extended first.

Phase 1: The discovery race. Before B_h is broadcast, all miners are searching for a valid block header at height h. At time τ, when miners learn of B_h, they choose between switching to mining a potential extension block at $h+1$ and continuing to mine at h on a competing or attacking template B_h^A. Under the attack strategy, the miner commits hashrate β_A to B_h^A from τ onward. The original proposing miner (β_0) already knows about B_h (they mined it) and immediately starts mining at height $h+1$. Of the remaining fringe miners (β_R), only the fraction $\gamma(\tau)$ that has received B_h by time τ *may* also mine B_{h+1} on the incumbent chain. The remaining $(1-\gamma(\tau))\beta_R$ are unaware of B_h and continue mining at height h on their own templates.[7]

[7]These are not the strategic attacker β_A we model but the uninformed fraction of the fringe, $(1-\gamma(\tau))\beta_R$, still mining at height h because B_h has not yet reached them; if one finds a block at h, a three-way race can emerge. Their omission from $P_1(\tau)$ is exact rather than approximate: by the memorylessness of the competing exponential clocks, the probability that the attacker's block precedes the original chain's extension to $h+1$ does not depend on their clock. The only real simplification is treating Phase 2

The aggregate hashrate mining at B_{h+1} is therefore $\beta_0+\gamma(\tau)\beta_R$. The discovery race is an exponential competition. Treating γ as approximately constant over the short duration of Phase 1,[8] the probability the attacker finds B_h^A first is:

$$P_1(\tau) \;=\; \frac{\beta_A}{\beta_A+\beta_0+\gamma(\tau)\beta_R} \tag{5}$$

When the attack begins early ($\tau \to 0$, $\gamma \approx 0$), $P_1 \to \beta_A/(\beta_A+\beta_0)$, the attacker races only against the miner of the incumbent block, and is favored if $\beta_A > \beta_0$.[9] As $\gamma \to 1$, $P_1 \to \beta_A$, the attacker's odds reduce to their hashrate share. The continuous propagation function $\gamma(t)$ captures this transition in expectation.

Phase 2: The block race. Conditional on finding B_h^A, both B_h (the original block) and B_h^A coexist. Miners must choose which tip to extend. The original miner (β_0) remains on C^0, and a fraction $\alpha \in [0,1]$ of the rational fringe β_R mines on C^A. The effective hashrate on the attacker's chain is $\beta_{\text{eff}}(\alpha) \;=\; \beta_A + \alpha\beta_R$. Under the lead-of-one-block resolution rule, the next block found by any miner determines the winner. The probability the attacker's chain is extended first is:

$$P_2(\alpha) \;=\; \beta_{\text{eff}}(\alpha) \;=\; \beta_A + \alpha\beta_R \tag{6}$$

We adopt the lead-of-one-block rule as our baseline for three reasons: (i) it yields clean closed-form results, (ii) the empirical evidence suggests a short-horizon phenomenon consistent with single-block resolution, and (iii) most nodes follow a first-seen rule, so whichever chain gets extended first is likely to become canonical. The rule also matches the data: in our 2017–2025 sample all but two block races settle within a

as a two-way race between B_h and B_h^A; ignoring the third tip can only overstate the attacker's win probability, and is negligible once $1-\gamma(\tau)$ is small, within a few seconds of broadcast.

[8] The exact Phase 1 win probability integrates over the time s at which the first block is found: $P_1^{\text{exact}}(\tau) = \int_0^\infty \beta_A \exp\!\big(-\int_0^s [\beta_A+\beta_0+\gamma(\tau+u)\beta_R]\,du\big)\,ds$. The static formula (5) fixes γ at $\gamma(\tau)$ for the race; since relay completes within seconds ($1/\lambda \approx 2$–$12\,$s), the approximation is tight.

[9] The limiting case $\gamma = 0$ is a theoretical abstraction that assumes only the attacker and the original miner are aware of B_h.

single block, and the two exceptions (the consecutive-height stale pairs 656,477–656,478 and 788,686–788,687, fewer than one percent of races) are genuine two-block forks. These rare cases do not undermine the baseline. Appendix 7.1 generalizes to a lead-of-two rule, in which the winning probability from state $(1,1)$ becomes $P_2^{(2)} = \beta_{\text{eff}}^2/(\beta_{\text{eff}}^2 + (1-\beta_{\text{eff}})^2)$, and shows that all comparative statics are qualitatively unchanged.

3.5 The Solo Attack Hurdle Rate

Consider the baseline case with no fringe support for the attacker ($\alpha = 0$, $P_2 = \beta_A$). We compare the attacker's expected payoff over two slots, blocks h and $h+1$. Rewards from $h+2$ onward cancel from this comparison: however the contest at height h resolves, the network settles on a single canonical chain at $h+1$ and all miners resume with the same hashrate shares $\{\beta_A, \beta_0, \beta_R\}$. The two outcomes leave slightly different mempools, since B_h and B_h^A confirm different transactions, but this gap erodes as fresh transactions arrive, so continuation values are approximately equal across the attack-success and attack-failure paths. This cancellation is a deliberate simplification that affects the level of the threshold but not its comparative statics. We now compute the payoff for each strategy.[10]

Continue (accept the new block h and mine on top of it at $h+1$): B_h is already claimed by its miner. The attacker contributes β_A to mining at $h+1$ and earns, in expectation, $\beta_A \cdot R_{h+1}$, where $R_{h+1} = S + \tilde{F}_{h+1}^*$.

Attack (mine a competing block at h): Three mutually exclusive outcomes arise, conditional on Phase 1 success (probability P_1).

(a) *Attacker finds block $h+1$ on C^A* (probability β_A, conditional on P_1). The attacker's chain is extended, C^A wins, and the attacker earns both R_h^A and R_{h+1}^A.

[10]Throughout the derivation we suppress the decision-time argument τ on the reward and fee levels (R and F), which are all evaluated at τ; we retain it on the propagation function $\gamma(\tau)$ and the fee differential $\Delta(\tau)$, whose time-variation is the object of interest (the win probability P_1 inherits this dependence through γ).

(b) *A fringe miner finds block* $h{+}1$ *on* C^A (probability $\alpha\beta_R$, conditional on P_1; zero for the solo case). The attacker earns R_h^A only.

(c) *A miner on* C^0 *finds block* $h+1$ (probability $1-\beta_{\text{eff}}$, conditional on P_1). The original chain wins, the attacker's block is orphaned, and the attacker earns zero.

If Phase 1 fails (probability $1-P_1$), the main chain extends to $h+1$ while the attacker has been mining at h; the attacker misses both blocks and earns zero.

Collecting terms, the expected payoff from attacking (over blocks h and $h+1$) is:

$$U_A(\text{att}) \;=\; P_1\big[\,P_2\cdot R_h^A \;+\; \beta_A\cdot R_{h+1}^A\,\big] \tag{7}$$

where the first term reflects the block h reward weighted by the probability the attack succeeds in both phases, and the second term reflects the attacker's probability of also mining block $h{+}1$ on their own chain.[11]

The *Continue* payoff is simply:

$$U_A(\text{cont}) \;=\; \beta_A\cdot R_{h+1} \tag{8}$$

The attack is therefore profitable when $U_A(\text{att}) \geq U_A(\text{cont})$:

$$P_1(\tau)\big[\,P_2\cdot R_h^A \;+\; \beta_A\cdot R_{h+1}^A\,\big] \;\geq\; \beta_A\cdot R_{h+1} \tag{9}$$

In the solo case ($P_2=\beta_A$), the β_A terms in (9) cancel and the condition reduces to $P_1(\tau)(R_h^A+R_{h+1}^A)\geq R_{h+1}$. We derive the baseline hurdle under the substantive assumption that the attacker does not undercut, in the sense of Appendix 7.2: $F_h^A=F_h^*$, i.e., the attacker constructs a fee-optimal block at h from the time-τ mempool. Under this assumption, $R_{h+1}^A\approx R_{h+1}$ to leading order. The two diverge when (i) the incumbent at h was not fee-optimal ($F_h<F_h^*$), or (ii) new high-fee transactions

[11] If the attacker mines block $h{+}1$ on C^A, that event simultaneously resolves Phase 2 in the attacker's favor. Hence the probability β_A appears outside P_2: with probability β_A the attacker finds $h+1$ (guaranteeing C^A wins), earning both R_h^A and R_{h+1}^A.

arrived during $[0, \tau]$: in both cases the post-B_h mempool retains high-fee transactions that the attacker's fee-optimal B_h^A consumes, so $\tilde{F}_{h+1}^* \geq F_{h+1}^*$. We treat the gap as negligible for the derivation; it is bounded by the protocol's block-size limit, it is small relative to the fee differential of interest, and the empirical analysis uses the realized $\tilde{\Delta}(\tau)$, which absorbs this residual directly. The undercutting extension in Appendix 7.2 relaxes this assumption. Substituting $R_{h+1}^A = R_{h+1}$:

$$P_1(\tau) \cdot (R_h^A + R_{h+1}) \;\geq\; R_{h+1} \tag{10}$$

Solving for the *hurdle rate*:

$$\boxed{R_h^{A*} \;=\; R_{h+1}\left(\frac{1}{P_1(\tau)} - 1\right) \;=\; R_{h+1} \cdot \frac{1 - P_1(\tau)}{P_1(\tau)}} \tag{11}$$

From hurdle rate to fee differential threshold. Writing $R_h^{A*} = S + F_h^{A*}$ and $R_{h+1} = S + F_{h+1}^*$ in the hurdle rate (11), subtracting $S + F_{h+1}^*$ to express the condition in terms of the fee gap, and substituting $P_1 = \beta_A / (\beta_A + \beta_0 + \gamma(\tau)\beta_R)$ from (5) so that $(1 - 2P_1)/P_1 = (\beta_0 + \gamma(\tau)\beta_R - \beta_A)/\beta_A$, yields the *fee differential threshold*:

$$F_h^{A*} - F_{h+1}^* \;\geq\; (S + F_{h+1}^*) \cdot \frac{\beta_0 + \gamma(\tau)\beta_R - \beta_A}{\beta_A} \tag{12}$$

Interpretation. The fee differential threshold (12) has a transparent economic structure:

1. **Block subsidy.** The threshold is increasing in S: a higher subsidy raises the opportunity cost of diverting hashrate, because the attacker forgoes a proportional share of $S + F_{h+1}^*$ when attacking, so as S halves the threshold falls and attacks become viable at lower fee differentials. Section 3.6 develops this security-buffer role.

2. **Propagation as a defense.** As $\gamma(\tau)$ increases and more miners may join the incumbent chain, the threshold rises monotonically. Whether the threshold ever falls below zero depends on the

hashrate ranking between the attacker and the original miner: for $\beta_A > \beta_0$, the numerator $\beta_0 + \gamma(\tau)\beta_R - \beta_A$ is negative when $\gamma(\tau)$ is sufficiently small, so any realized $\Delta \geq 0$ satisfies the attack condition in the early propagation window; for $\beta_A \leq \beta_0$, the threshold is positive throughout propagation. At full propagation ($\gamma = 1$, $\beta_R = 1 - \beta_A - \beta_0$):

$$\Delta^* \;=\; (S + F^*_{h+1}) \cdot \frac{1 - 2\beta_A}{\beta_A} \tag{13}$$

For any $\beta_A < 1/2$, this is strictly positive. At $\beta_A = 1/2$, the threshold vanishes: a miner with half the hashrate can always profitably attack, consistent with the classical 51% threshold.

3. **Miner size.** The threshold is decreasing in β_A. A miner controlling 20% of hashrate faces a threshold of $3 \cdot R_{h+1}$, while a 5% miner faces $18 \cdot R_{h+1}$. This monotonicity holds across candidate attackers, since raising a miner's own share lowers its threshold even as it shrinks the fringe $\beta_R = 1 - \beta_A - \beta_0$. The largest pool other than the miner of B_h therefore faces the lowest hurdle and is the *marginal attacker*, so whether *any* miner finds an attack profitable is governed by this single pool, and race incidence does not require modeling simultaneous attempts by several miners.

4. **Time dependence.** Both $\gamma(\tau)$ and $\Delta(\tau)$ evolve over time: the former increases the threshold, the latter decreases the available fee differential. The attacking incentive is therefore maximized in a narrow window shortly after block h is discovered, when the fee differential is large and propagation is incomplete.

Reward-normalized fee differential. Dividing both sides of the threshold (12) by the next-block reward $R_{h+1} = S + F^*_{h+1}$ isolates the structural component of the attack condition:

$$\frac{\Delta^*(\tau)}{S + F^*_{h+1}} \;=\; \frac{\beta_0 + \gamma(\tau)\beta_R - \beta_A}{\beta_A} \tag{14}$$

The right-hand side depends only on hashrate (β_A, β_0, β_R) and the propagation function $\gamma(\tau)$; the dependence on fees and on the subsidy vanishes. Equation (14) therefore identifies the natural empirical counterpart of the fee differential incentive: attacking becomes profitable, and a block race may follow, when the fee differential, expressed as a fraction of the opportunity cost of ordinary extension, exceeds a threshold determined entirely by miner heterogeneity and propagation speed. This reward-normalized fee differential, $\Delta/(S + F^*_{h+1})$, serves as the principal regressor in our analysis of block-race incidence.[12]

3.6 The Block Subsidy as a Security Buffer

The hurdle rate (12) illuminates the fundamental role played by the block subsidy in effectuating consensus stability. Using the full-propagation threshold (13):

$$\Delta^* = \underbrace{S \cdot \frac{1-2\beta_A}{\beta_A}}_{\text{subsidy buffer}} + \underbrace{F^*_{h+1} \cdot \frac{1-2\beta_A}{\beta_A}}_{\text{fee-based friction}} \tag{15}$$

The subsidy buffer provides a fixed floor for the fee differential needed to trigger an attack, independent of mempool conditions. At full propagation and with the current subsidy of 3.125 BTC, even a large miner with $\beta_A = 0.25$ requires a fee differential of at least $\Delta^* \geq 6.25$ BTC plus the fee-based friction for a solo attack, a threshold that binds only rarely under current fee levels. Incomplete propagation, however, lowers the relevant threshold substantially: Section 3.7 shows that for the typical block the attack condition holds during a brief window after block ar-

[12]The derivation uses $F^*_{h+1}(\tau)$ throughout for notational economy: it denotes the continuation fees on the main chain conditional on a fee-optimal block at h. The empirically relevant quantity, and the one a miner actually faces when deciding at time τ, is $\tilde{F}^*_{h+1}(\tau)$, the continuation fee given the realized transactions in B_h. The two coincide when $F_h = F^*_h(\tau)$ (the incumbent was fee-optimal); otherwise $\tilde{F}^*_{h+1}(\tau) \geq F^*_{h+1}(\tau)$, reflecting the high-fee transactions the incumbent may have left in the mempool. Our survival analysis below uses $\tilde{\Delta}(\tau) = F^*_h(\tau) - \tilde{F}^*_{h+1}(\tau)$, the fee differential actually faced by the marginal miner at the decision time.

rival, and that the length of this window, rather than the unconditional threshold, is the margin along which the fee differential operates today. After the next halving the subsidy floor will drop by half to $S = 1.5625$ BTC. In the long run $S \to 0$, the subsidy buffer vanishes entirely:

$$\lim_{S \to 0} \Delta^* \;=\; F^*_{h+1} \cdot \frac{1 - 2\beta_A}{\beta_A} \tag{16}$$

and the stability of the system will depend exclusively on the level and variance of transaction fees relative to miner concentration.

3.7 The Attack Window

The full-propagation threshold (13) is the relevant benchmark only in the limit $\gamma = 1$. At the decision times that matter empirically, propagation is incomplete and the threshold (14) is state-dependent through $\gamma(\tau)$. This distinction is quantitatively important. For the typical block in our sample, the marginal attacker is *larger* than the incumbent: the largest pools control roughly 25–35% of network hashrate, while the mean incumbent share is 13% (Table 1). Whenever $\beta_A > \beta_0$, the numerator $\beta_0 + \gamma(\tau)\beta_R - \beta_A$ is negative at small γ, so the attack condition holds for *any* non-negative fee differential early in propagation. What separates high- from low-differential blocks is therefore not whether the attack condition ever binds, but for how long it remains satisfied.

To formalize this margin, let $d \equiv \Delta(\tau)/(S + F^*_{h+1}(\tau))$ denote the reward-normalized fee differential, treated as constant over the few seconds relevant here: the mempool refills on the fee-decay timescale $1/\delta$ (minutes), which is an order of magnitude slower than the relay timescale $1/\lambda$ (seconds). The attack condition $d \geq (\beta_0 + \gamma(\tau)\beta_R - \beta_A)/\beta_A$ inverts to a condition on the propagation state:

$$\gamma(\tau) \;\leq\; \gamma^*(d) \;\equiv\; \frac{\beta_A(1+d) - \beta_0}{\beta_R} \tag{17}$$

Under the propagation function (4), the condition holds for all $\tau \leq \tau^*(d)$,

where

$$\tau^*(d) \;=\; -\frac{1}{\lambda}\,\ln\big(1-\gamma^*(d)\big), \qquad 0 \le \gamma^*(d) < 1 \tag{18}$$

defines the **attack window**: the interval after the arrival of B_h during which deviation is profitable for the marginal attacker. The boundary cases are instructive. If $\gamma^*(d) \le 0$, the attacker is sufficiently smaller than the incumbent that the window is empty. If $\gamma^*(d) \ge 1$, which rearranges exactly to $d \ge (1-2\beta_A)/\beta_A$, the attack condition holds at *all* propagation states: the full-propagation threshold (13) is the differential at which the window becomes unbounded. Between these extremes, $\tau^*(d)$ is increasing in d and β_A, decreasing in β_0, and proportional to the relay time constant $1/\lambda$.

A block race materializes only if the attacker finds a competing block while the window is open. With difficulty normalized so that the network finds blocks every 600 seconds in expectation, the attacker's discovery rate is $\beta_A/600$ per second, so the probability that a race emerges at a block with differential d is

$$G(d) \;\equiv\; \Pr(\text{race} \mid d) \;=\; 1 - e^{-\beta_A \tau^*(d)/600} \;\approx\; \frac{\beta_A\,\tau^*(d)}{600} \tag{19}$$

ignoring the order-$\tau^*/600$ probability that the chain extends to $h+1$ within the window. Equation (19) maps the deterministic hurdle rate into a race probability that is smooth and increasing in d, with slope $\partial G/\partial d = \beta_A^2/\big[600\,\lambda\,\beta_R\,(1-\gamma^*(d))\big]$.

The implied magnitudes are consistent with the data. Evaluated at $\beta_A = 0.30$ (a large pool), $\beta_0 = 0.13$ (the sample mean incumbent share), a mid-range relay time constant of $1/\lambda = 4$ seconds, and the sample median differential $d = 0.016$, the window is $\tau^* \approx 1.5$ seconds and $G \approx 0.073\%$, against an observed race incidence of 0.069% (Section 5.1). The implied slope $\partial G/\partial d \approx 0.0015$ is of the same order of magnitude as the linear probability estimates reported below (0.0033 in the full specification). Varying $1/\lambda$ over the empirically relevant 2–12 second range moves G between roughly 0.04% and 0.22%, bracketing the observed incidence.[13]

[13]Two offsetting approximations bound this calibration. Equation (19) assumes the

The attack window also disciplines how the model's predictions should vary across relay regimes. Because both the level and the slope of $G(d)$ are proportional to $1/\lambda$, the model predicts that the sensitivity of race incidence to the fee differential *falls* as block relay accelerates. Relay latency declined by roughly an order of magnitude over our sample period with the deployment of Compact Blocks and FIBRE, so the cross-sectional slope should be largest in the earliest subperiod and attenuate thereafter, a prediction we revisit when interpreting the era subsamples in Section 5.1.

3.8 Testable Hypotheses

The model yields three directional predictions, each tied to a distinct empirical object (race incidence, the block-arrival hazard, and its time profile).

The attack window of Section 3.7 maps the deterministic hurdle rate (14) into a race probability $G(d) \approx \beta_A \tau^*(d)/600$ that is smooth and increasing in the reward-normalized differential d, with a level and slope consistent with the observed rarity of races at the current subsidy. The linear probability models in Section 5.1 approximate G locally. The same logic underlies H2 and H3: while the window is open, the hashrate that finds deviation profitable is subtracted from the canonical tip, depressing the block-arrival hazard, and the window closes within seconds as propagation completes and the fee differential erodes.

1. **H1 (Block race incidence).** The probability of a block race at height h is increasing in the initial fee differential $\Delta(t = 0)$ between

attacker observes B_h at $\tau = 0$; integrating instead over the attacker's own awareness time, drawn from the same propagation process, shortens the expected window to $\tau^*(d) - \gamma^*(d)/\lambda \approx 0.24$ seconds and lowers G to about 0.012% at the same parameter values. The truth lies toward the upper bound, since the largest pools maintain dedicated peering and relay-network connections and are systematically early in the propagation order. Conversely, the observed incidence of 0.069% includes latency races unrelated to fee incentives and excludes races that never reached archival nodes. The calibration should therefore be read as an order-of-magnitude consistency check rather than a point prediction.

block h and the block available to a miner who would follow the *Continue* strategy of attempting to extend the main chain.

2. **H2 (Hashrate diversion).** The hazard of the next canonical block arrival is decreasing in Δ: when Δ is large, a share of the network's hashrate continues to mine at height h rather than extending the main chain, slowing the arrival of block $h + 1$.

3. **H3 (Temporal concentration).** The effect of Δ on the block-arrival hazard is concentrated in the early propagation window, and this effect declines toward zero as propagation completes and the fee differential erodes.

The solo attack hurdle rate underlying H1–H3 is a conservative benchmark: undercutting to attract the rational fringe lowers it by a factor of $\beta_A/(1 - \beta_0)$ (Appendix 7.2), so support for these predictions against it understates the system's true sensitivity to fee incentives.

4 Data

4.1 Sample and Sources

Our sample covers 420,103 Bitcoin blocks, spanning block heights 463,500 to 883,602. The observation period begins in April 2017 and concludes in February 2025, providing a comprehensive view of the network's evolution across various market cycles.

We obtain transaction-level data from Chainalysis, the blockchain analytics provider. For each confirmed transaction in our sample window, our dataset includes its first-seen timestamp, size and weight, fee (in satoshis), fee rate (satoshis/vByte), and parent-child linkage indicators. These fields allow us to reconstruct transaction dependency clusters, specifically child-pays-for-parent (CPFP) structures, and compute effective fee rates, the primary economic metric for miner prioritization.

For each block, we collect the height, header timestamp, total weight and size, transaction count, fee reward, and coinbase reward from a Bitcoin full node maintained by the authors. Mining pool identities are parsed from coinbase transaction inputs and cross-validated via the *mempool.space* API.[14]

We supplement the blockchain data with information from two additional repositories to address protocol-level reporting gaps. First, we collect high-resolution arrival timestamps from the *block-arrival-times* repository[15] to mitigate the imprecision of on-chain block header timestamps. The Bitcoin protocol only requires that header timestamps exceed the median of the previous 11 blocks' timestamps and remain within a two-hour future window.[16] To ensure temporal accuracy, we record the earliest arrival timestamp across all reporting nodes for each block. Second, we identify block races by recording instances where competing stale blocks were observed at the same height, using data from the *stale-blocks* repository,[17] which provides a lower bound on the true number of races, as the repository relies on voluntary contributions and cannot guarantee completeness. Figure 4 plots the frequency of these races by difficulty epoch.

4.2 Variable Construction

To reconstruct mempool snapshots at regular intervals, we retain all transactions first observed before each snapshot time that had not yet been included in a confirmed block. We then identify parent-child clusters and compute their effective fee rates. We order transactions by effective fee rate to simulate optimal block construction subject to the block size constraints. Since we cannot observe the mempools of other nodes, we assume that miners share approximately the same transaction

[14] `https://mempool.space/docs/api/rest`

[15] `https://github.com/bitcoin-data/block-arrival-times`

[16] This loose constraint means on-chain timestamps are not strictly monotonic and can deviate substantially from the actual times that blocks propagate through the network.

[17] `https://github.com/bitcoin-data/stale-blocks`

set. This assumption is supported both by the strong economic incentive for large pools to maintain comprehensive mempools and by the data: the overwhelming majority of confirmed transactions are observed in our mempool before inclusion, with privately relayed transactions constituting roughly 0.23% of confirmed transactions over the sample and rising modestly only in the most recent period (Appendix Figure 10).

Using these mempool snapshots, we compute the mempool fee differential $\Delta(t)$, the realized fee differential $\tilde{\Delta}(t)$, and their reward-normalized counterparts $\Delta(t)/(S + F^*_{h+1}(t))$ and $\tilde{\Delta}(t)/(S + \tilde{F}^*_{h+1}(t))$ as defined in Section 3.2. We evaluate these metrics at the block's network arrival ($t = 0$, the earliest arrival timestamp across reporting nodes; Section 4) and at one-minute snapshots through $t = 30$ minutes. The cross-sectional analysis (Section 5.1) uses the arrival snapshot; the survival analysis additionally uses the later snapshots to update its time-varying covariates (Section 5.2). Figure 5 plots the resulting time profile of the realized fee differential. Consistent with the decay assumption in Section 3.2 (Equation (3)), $\tilde{\Delta}(t)$ is largest at block arrival and declines monotonically as the mempool refills and the fee advantage of the attacking strategy erodes.

For each block h, we compute its miner's hashrate share as the fraction of blocks mined in the preceding difficulty epoch and private transaction count as the number of transactions in block h that were not observed in the public mempool prior to the block's arrival. We also construct indicators for the prevailing block-relay regime, which shifts the baseline rate of stale blocks independent of fee incentives (Section 3.3). The relay regime appears in our regressions as a single three-level control (pre-FIBRE, post-FIBRE, and the relay bug week), with the pre-FIBRE period as the omitted baseline; the regime boundaries are detailed in the note to Table 3. Table 1 reports descriptive statistics for the main variables.

5 Analysis

We test the three hypotheses developed in Section 3. We examine H1, the incidence of block races, with linear probability models (Subsection 5.1), and H2 and H3, the effect of the fee differential on the block-arrival hazard and its temporal profile, with hazard models (Subsections 5.2 and 5.3).

The two designs rely on different fee differentials for identification. The block-race regression uses the mempool differential $\Delta(t)$, which depends only on the mempool state and the protocol's block-weight limit and is therefore predetermined with respect to the race. The realized differential $\tilde{\Delta}(t)$ cannot serve here: for the blocks that experience a race, which are the very outcomes the regression predicts, the canonical block at h is the race winner, so $\tilde{F}^*_{h+1}$, and hence $\tilde{\Delta}(t)$, is a function of which competitor prevailed. The survival design instead uses $\tilde{\Delta}(t)$, because each spell follows an already-observed canonical B_h: there, $\tilde{F}^*_{h+1}$ is the actual reward gap a would-be attacker faces, measured before the spell's outcome and free of post-outcome conditioning.

5.1 Block Races

We predict that the probability of a block race increases with the difference between the user fees for transactions that could be included in an additional block mined on top of the newly arrived block, compared to the transaction fees available to a miner who challenges a newly arrived block by attempting to create a valid competing block. As a preliminary test, we partition the sample of blocks into quartiles based on the reward-normalized mempool fee differentials that prevail in each block at the moment of arrival. We then perform a Pearson χ^2 test for independence of the block race frequencies across these quartiles. The test strongly rejects the null hypothesis of independence ($\chi^2(3) = 55.4$, $p < 0.001$), with the top quartile (the blocks with the highest fee differentials) containing a disproportionate share of observed races (Table 2).

This simple test provides initial support for our hypothesis.

We estimate linear probability models with heteroskedasticity-consistent standard errors (HC1) as our primary specification.[18] Table 3 reports estimates for five models with an increasingly complex set of regression controls. In the left column, a univariate regression yields a positive and significant coefficient on the reward-normalized mempool fee differential ($\hat{\beta} = 0.0051$, $p < 0.001$). Adding the relay-regime indicators (Section 4.2) and block-level controls attenuates the estimate ($\hat{\beta} = 0.0033$, $p < 0.01$), and the coefficient remains significant at the one percent level under epoch-clustered standard errors ($p = 0.0077$). A one-unit increase in $\Delta(0)/(S + F^*_{h+1}(0))$ corresponds to a fee differential equal to the next-block reward, well outside the empirical support; a one-standard-deviation increase ($\sigma \approx 0.0467$) raises predicted block race probability by about 0.015 percentage points, an increment equal to about 22% of the sample mean race frequency (0.069%).

Among the block-level attributes of the canonical block at height h, only block size is uniformly positive and significant ($\hat{\beta} \approx 2.4$–2.6×10^{-4} per megabyte, $p < 0.01$ in columns 3–5); miner hashrate share loads positively but is never distinguishable from zero, and private transaction count is essentially zero. We do not assign a causal interpretation to these coefficients because they are observed only for the surviving branch and may therefore capture selection on the race outcome. Their role is simply to absorb observable heterogeneity in block construction that is correlated with the fee differential. The post-FIBRE indicator is negative and significant in columns (2)–(4) ($\hat{\beta} \approx -9.6 \times 10^{-4}$ in column 2 and $\approx -1.09 \times 10^{-3}$ in columns 3–4, $p < 0.01$), consistent with block relay upgrades structurally reducing stale-block incidence. A post-FIBRE-only

[18] In our sample, block races occur in approximately 0.069% of blocks (284 out of 412,804). At this base rate, logit and probit coefficients that appear statistically significant on the latent scale can correspond to economically negligible probability changes (King and Zeng, 2001). The linear probability model coefficients directly measure marginal effects on the probability scale and are robust to the distributional misspecification that afflicts nonlinear estimators under latent heteroskedasticity (Angrist and Pischke, 2009). A heteroskedastic probit likelihood-ratio test rejects the constant-variance assumption ($\chi^2(1) = 8.74$, $p = 0.003$), providing additional justification for the linear probability model as the primary specification.

specification (column 5) yields a similar estimate for the mempool fee differential variable ($\hat{\beta} = 0.0030$, $p \approx 0.011$), indicating that the result is not driven by pre-FIBRE blocks.

We attempt to validate the results with an extensive set of robustness tests. Table 4 re-estimates the full specification as logit, probit, complementary log-log,[19] and Firth penalized logistic[20] models. The fee-differential coefficient stays positive and significant throughout, at the 1% level except under Firth ($p = 0.014$); the nonlinear average marginal effects are smaller than the linear probability slope (order 3–5 $\times 10^{-4}$ versus 3×10^{-3}), as expected in rare-event settings. Sign and significance also survive alternative transforms of the regressor (winsorizing at the 95th through 99.9th percentiles, logging, and leaving it unnormalized) as well as a permutation test with 1,000 reassignments of the race indicator ($p = 0.007$). We split the sample by halving eras, and we find the effect is positive and significant before the third halving ($\hat{\beta} = 0.0063$, $p < 0.01$) and positive but imprecise thereafter; the post-fourth-halving window is the least informative in our sample, as it spans only ten months with 36 races and little high-incentive variation (median differential 0.010 versus 0.019 earlier).

These results establish a robust positive association between the mempool fee differential and block race incidence, consistent with H1. We interpret these estimates causally under the identifying assumption that propagation latency is conditionally independent of the mempool fee differential given the observable propagation drivers (Section 3.3). Under Compact Blocks (BIP 152) and FIBRE, propagation latency is determined by network topology and the count of unknown-to-receiver transactions rather than by the fee composition of block h. The fee differential, by contrast, is a function of mempool composition that evolves on transaction-arrival timescales. The most direct link between them runs

[19]The complementary log-log link is appropriate when the binary outcome indicates whether an event arrives within an observation interval and the underlying arrival process is Poisson.

[20]Firth penalized logit (Firth, 1993) reduces small-sample bias and is robust to separation, which is useful when the outcome is rare.

through block size and the count of receiver-unknown transactions, which we control for.

A subtler version of the concern nonetheless survives this argument: during high-demand episodes the mempool churns rapidly and nodes' transaction sets diverge, which can slow compact-block reconstruction, and hence propagation, precisely when fee differentials are large, even though no single observable driver fully captures it. Because this congestion channel is correlated with Δ yet carries no strategic content, we confront it directly in two ways (Table 5). First, we add congestion controls to the full specification: the mempool backlog, the count of block transactions first seen within ten seconds of arrival (a direct proxy for the reconstruction round-trips that slow relay), and the transaction arrival rate. The fee-differential coefficient barely moves ($0.0033 \rightarrow 0.0030$, $p = 0.013$ with epoch-clustered errors). Second, and more sharply, we run a placebo. We measure the mempool backlog of *low-fee* transactions, a high-variance proxy for congestion intensity whose per-transaction value is negligible, so an attacker re-mining the tip captures essentially nothing from it. If races were a by-product of congestion in general, this backlog, which spikes in the same demand surges that inflate Δ, would predict them; the strategic mechanism predicts it should not. The latter holds: the low-fee backlog is insignificant ($p = 0.67$) while Δ retains its effect. A powerful, high-variance congestion measure thus fails to predict races exactly where the capturable fee differential succeeds, isolating the strategic channel from the propagation one.

Two residual concerns remain: stale-block records are an incomplete census of competing blocks, since forks that fail to propagate widely are unobservable, so our race counts are a lower bound; and block-level covariates are observed only for the winning block, raising standard concerns about post-outcome conditioning. The linear probability model also uses only the fee differential at block arrival and does not exploit its intra-spell dynamics. These constraints motivate the continuous-time analysis in the following section.

5.2 Time to Next Block

In proof of work, block interval times follow an exponential distribution, conditional on the instantaneous rate $\lambda(t)$. All hash attempts across the network form a homogeneous Poisson process with rate $\lambda_{\text{total}}(t) = \frac{H(t)}{D(t)}$, where $H(t)$ is the global hashrate and $D(t)$ is the difficulty. Since we can only reliably observe the subset of blocks that are accepted into the canonical chain, this observed sequence represents a thinned Poisson process.

If miners do not all mine on the same predecessor block, thereby splitting their hashrate across competing branches, this dispersion reduces the effective rate or hazard of a block extending the current chain head. Consequently, the reduction in the canonical chain's effective hazard leads to longer expected interval times for blocks on the canonical chain.

$$\lambda_{\text{canonical}}(t) = \frac{(1 - q(\Delta(t), \text{covariates}))\, H(t)}{D(t)} \tag{20}$$

We examine whether the fee differential influences the fraction of hashrate mining on the canonical tip versus an alternative fork (i.e., the preceding block), and consequently, how this affects the hazard of a block extending the current chain head.

We utilize survival analysis to study the time until a new block extends the chain's tip. Our outcome is the length of the block interval, from the network arrival of one block to the arrival of its successor. We measure arrivals as the earliest time any monitoring node reports seeing a block (the *block-arrival-times* data of Section 4), not the block header timestamp, which is chosen by the miner and can deviate from true propagation time by minutes. Every duration, event time, and hazard below is therefore defined on the network-arrival clock, with each spell beginning at a block's arrival ($t = 0$). The time-varying fee covariates are updated within the spell from mempool snapshots, the first at arrival and the rest on a one-minute grid pinned, for implementation reasons, to the header timestamp (Section 4.2); the header timestamp thus affects only when

covariates are re-sampled, never the measured durations, and the header-to-arrival offset (a median of about 18 seconds) is immaterial relative to the snapshot spacing and the fee-decay timescale. Figure 6 shows the distribution of these inter-arrival times; the mean is 586.4 seconds and the median is 407 seconds.

Let T denote time in seconds from arrival of block h until arrival of the next block $h{+}1$. The survival function $S(t) = P(T > t)$ gives the probability that no successor has arrived by time t, and the *hazard function*

$$h(t) \;=\; \lim_{\Delta t \to 0} \frac{1}{\Delta t}\, P\big(t \le T < t + \Delta t \,\big|\, T \ge t\big) \tag{21}$$

gives the instantaneous arrival rate conditional on survival to t.

5.2.1 Non-parametric estimation of the block-arrival hazard

We estimate the block-arrival hazard using a penalized B-spline smoother (Rebora et al., 2014).[21] Figure 7 plots the smoothed hazard in the early period after block arrival. The pooled estimate (Panel 7a) rises steeply over the first thirty seconds and stabilizes thereafter, indicating that block discovery is slower immediately after h arrives and accelerates as the network aligns on the new tip. Two non-exclusive mechanisms can generate this pattern: propagation frictions, under which miners who have not yet received block h cannot extend it, and strategic hashrate diversion, under which miners who have received h continue to mine at the same height while the fee differential is large. The Cox specifications that follow are designed to disentangle these channels.

Panel 7b overlays smoothed hazards for two subsamples, which we create by partitioning our observations based on the realized fee differential at block arrival. We assign observations to *Low* if the fee differential lies in the bottom three quartiles of its sample distribution and to *High* if it

[21] A B-spline smoother approximates the hazard as a weighted sum of piecewise polynomial basis functions; the penalized variant adds a roughness penalty, with the penalty parameter selected by cross-validation, yielding a smooth non-parametric hazard estimate with no assumed functional form.

lies in the upper quartile (split at the third quartile). H2 predicts that the upper-quartile stratum should exhibit a lower hazard in the early propagation window; the figure is consistent with this prediction.

These non-parametric estimates are univariate and do not adjust for confounders such as difficulty, mempool congestion, or miner characteristics. To estimate the conditional association between fee incentives and the instantaneous hazard, we turn to Cox proportional hazards regression.

5.2.2 Cox Regression

The Cox (1972) model is a semi-parametric method relating the hazard to a set of predictors: it leaves the baseline hazard $h_0(t)$ unrestricted while imposing a log-linear relationship between the predictors and the hazard,

$$h(t) = h_0(t) \exp(\beta_1 X_1 + ... + \beta_k X_k). \tag{22}$$

We report coefficients as hazard ratios, HR $= \exp(\beta)$; for a continuous covariate, $\exp(\beta_k)$ is the factor by which the hazard changes for a one-unit increase in X_k, holding the other predictors fixed.

Controls. All covariates are measured for block h or at its arrival time, which initiates the interval we model. *Continuing block fee* ($\tilde{F}^*_{h+1}/S$) enters directly from the hurdle-rate condition (12): the threshold depends on both the fee differential and the opportunity cost of forgoing the extension reward $R_{h+1} = S + \tilde{F}^*_{h+1}$, so conditional on this level the fee differential coefficient captures the relative incentive to deviate, and higher continuation fees should speed arrival (HR > 1). We normalize both fee covariates by the block subsidy S rather than the total reward, which keeps the scale epoch-invariant and comparable to (12) and avoids the mechanical correlation that reward normalization would induce between $\tilde{\Delta}$ and $\tilde{F}^*_{h+1}$. *Miner hashrate share* proxies the incumbent share β_0 and predicts faster arrival (HR > 1) through two channels: mechanically, since large pools propagate faster through dedicated peering, and through the hurdle rate, which is increasing in β_0. It is also a confounder

of the fee differential: large pools build more fee-optimal blocks, raising $\tilde{\Delta}$ while propagating faster, so omitting it would bias the fee differential coefficient upward. *Block size* closes a mechanical path: larger blocks remove more high-fee transactions from the mempool and so raise $\tilde{\Delta}$, while any residual block-size effect on propagation (Decker and Wattenhofer, 2013) could correlate with the outcome; without the control this combination could bias $\tilde{\Delta}$ toward our finding. *Private transaction count* captures transactions unknown to receivers, which require an extra compact-block round-trip that slows propagation, reducing the effective hashrate on the canonical tip (HR < 1).

Model Stratification. We stratify by difficulty epoch and calendar day, fitting a separate non-parametric baseline hazard for each *epoch* $\times$ *day* stratum while constraining the covariate coefficients to be common across strata. Epoch stratification absorbs epoch-specific factors such as changes in mining difficulty; day stratification absorbs day-to-day variation in network hashrate driven by Bitcoin price fluctuations, electricity costs, or mining-pool dynamics. Our coefficients therefore reflect the within-day, within-epoch association between the predictors and block timing. A further implication is that the block subsidy S is constant within each stratum, so the subsidy-normalized and raw-BTC specifications are equivalent up to an epoch-specific scalar, and all identification comes from within-cell variation in the fee quantities.

Ties and censoring. Block arrival intervals are measured in seconds, which leads to tied event times. In all Cox regressions we handle ties using the Efron approximation (Efron, 1977), which is standard and performs well when ties are present but not overwhelmingly large. Regarding censoring, our dependent variable is the realized time between consecutive blocks (measured using block arrival timestamps), so each spell ends with an observed event (the next block arrival). In the time-varying counting-process specification, each block contributes multiple at-risk intervals but exactly one event at the terminal interval.

Time-Varying Covariates. We estimate the Cox model on a *counting-process* representation, the standard device for letting covariates vary

within a spell. Each block defines one spell, or *risk period*: the open interval that begins when block h is broadcast and ends the instant its successor is found, during which the chain tip is "at risk" of the event, namely the next block's arrival. We divide each spell into consecutive sub-intervals at the times the fee covariates are re-measured, namely the mempool snapshots taken at roughly one-minute increments after arrival (Section 4.2), and enter each sub-interval as its own row, carrying a start time, a stop time, the covariate values prevailing over it, and an event indicator. Refreshing the fee differential and continuing block fee at each snapshot is what we mean by updating the covariates throughout the risk period: a block whose mempool refills over its spell contributes a sequence of rows with a declining fee differential. The event indicator equals one only on the terminal sub-interval, when the successor arrives, and zero on all earlier ones, so each spell contributes many at-risk rows but exactly one event. The 412,804 spells therefore expand to about 4.2 million rows in the static specification of Table 6, roughly ten one-minute rows per spell. Because successors arrive between snapshots rather than on the grid, the terminal sub-interval is a "stub" shorter than one minute, of length equal to the time elapsed since the last snapshot (from a second to nearly a minute). Stubs raise no inferential issue: the Cox partial likelihood uses only the ordering of event times and the composition of the risk sets, not the length of any interval, and each spell still contributes a single event. The full model is specified as

$$h_i(t \,|\, \text{epoch}_i, \text{day}_i) = h_{0,\,\text{epoch}_i,\,\text{day}_i}(t) \cdot \exp\Big[\, \beta_1 \, \tilde{\Delta}_i(t)/S_i + \beta_2 \, \tilde{F}^{*}_{h+1,i}(t)/S_i + \beta_3 \,\text{miner share}_i + \beta_4 \,\text{block size}_i + \beta_5 \,\text{private tx count}_i \Big]. \tag{23}$$

Static specification. Throughout all Cox specifications we report cluster-robust standard errors at the difficulty-epoch level, which subsumes clustering at the block-height level since each block's spell lies within a single epoch; inference is similar, if anything more conservative, than under block-height clustering. A hazard ratio below one corresponds to a lower hazard of next-block arrival (a longer expected inter-block time), and a hazard ratio above one to a higher hazard. Column (1) of Table 6 reports

estimates for the static specification defined in Equation (23).

The static realized fee differential coefficient is statistically indistinguishable from one (HR = 1.0156, $p = 0.81$), consistent with no average effect of the fee differential over the full spell. Estimated coefficients for the control variables support the model's mechanism. The continuing block fee enters with HR = 1.190 ($p < 0.001$), consistent with an opportunity-cost channel in which higher available rewards on the canonical tip accelerate block discovery. The coefficient for miner hashrate share is likewise positive (HR = 1.04, $p = 0.051$), consistent with propagation advantages for large pools (or, equivalently, fewer attacks given the higher hurdle rate rivals face). Block size enters with a positive and significant coefficient (HR = 1.0153, $p < 0.01$) but is economically negligible: a one-standard-deviation increase (about 0.48 MB) raises the hazard by under one percent. The sign is opposite to the classical prediction that larger blocks propagate more slowly and thus depress the hazard, but that channel is largely closed over our sample period by Compact Blocks and FIBRE, under which relay latency has become approximately independent of block size (Section 3.3). As with the other block-level controls, we treat this coefficient as absorbing heterogeneity rather than identifying a propagation effect. Private transaction count is not statistically distinguishable from one (HR = 1.000, $p = 0.95$).

The null fee differential coefficient in Column (1) is not evidence against H2. The static Cox model estimates a spell-weighted average effect, and if the effect is concentrated in the early propagation window, the averaging will attenuate it toward zero. With a mean spell of roughly $\bar{T} \approx 600$ seconds and an early window of 30 seconds, a true early-window effect of size β enters the static coefficient with a weight of roughly $30/\bar{T} \approx 5\%$, so even a large early-window effect will appear statistically insignificant in a static specification. The non-parametric hazard (Figure 7) already suggests this time profile. We therefore allow the coefficient on the fee differential to vary across the spell.

5.3 Strategic Timing and the Coordination Window

We allow the realized fee differential coefficient to differ between an early window (0–30 seconds) and the rest of the spell, implementing the split as a counting-process piecewise-coefficient Cox model via `survSplit`.[22] H3 predicts that the early-window coefficient is below one and the later coefficient is indistinguishable from one. The model is specified as

$$h_i(t \mid \text{epoch}_i, \text{day}_i) = h_{0,\,\text{epoch}_i,\,\text{day}_i}(t) \cdot \exp\Big[\beta_1^{\text{early}}\, \tilde{\Delta}_i(t)/S_i \cdot \mathbf{1}\{t \le 30\text{s}\} + \beta_1^{\text{late}}\, \tilde{\Delta}_i(t)/S_i \cdot \mathbf{1}\{t > 30\text{s}\} + \beta_2\, \tilde{F}^*_{h+1,i}(t)/S_i + \beta_3\, \text{miner share}_i + \beta_4\, \text{block size}_i + \beta_5\, \text{private tx count}_i \Big]. \tag{24}$$

Column (2) of Table 6 reports the piecewise estimates. The early-window fee differential hazard ratio is significantly below one ($\text{HR}_{0\text{–}30\text{s}} = 0.5461$, $p = 0.015$), implying that a one-unit increase in the subsidy-normalized realized fee differential, i.e., a fee differential equal to the block subsidy, is associated with a 45.4% reduction in the instantaneous arrival rate of the next canonical block within the first 30 seconds of a spell, controlling for continuing block fee, miner hashrate share, block size, and private transaction count. The late-window hazard ratio for this variable is statistically indistinguishable from one ($\text{HR}_{30\text{s}+} \approx 1.06$, $p = 0.381$). The piecewise specification improves global fit relative to the static model (LR gain ≈ 7 on 1 df) and is consistent with H3: the fee differential effect operates within the early window after arrival of block h and vanishes once propagation completes and new transactions arrive. The control coefficients are essentially unchanged between Columns (1) and (2), and applying the same early/late split to each control yields no significant early/late difference (Appendix Table 12), indicating that the temporal heterogeneity is specific to the fee differential rather than a generic feature of the spell. Removing the day stratum (i.e., stratifying on `epoch_id` alone) yields

[22] `survSplit` is a utility in the R *survival* package that cuts each spell at specified times into consecutive counting-process records, each carrying a *start* time, a *stop* time, and an event indicator. Interacting the fee differential with an indicator for the resulting interval lets its coefficient differ before and after the cut, while leaving the estimation sample and the number of events unchanged.

a qualitatively unchanged but less precise estimate ($HR_{0–30s} = 0.623$, $p = 0.14$); both the attenuation of the point estimate toward one and the widening of the standard error are consistent with day strata absorbing within-epoch hashrate variation correlated with the fee differential through the BTC price channel, so their identifying role is substantive rather than mere conservatism.

The same congestion confound that we addressed for block-race incidence (Section 5.1) applies here, since a congestion slowdown in propagation would depress the early-window hazard exactly as strategic diversion does, and we control for it the same way (Table 7). Adding the congestion controls, the mempool backlog, the count of block transactions first seen within ten seconds of arrival, and the transaction arrival rate, leaves the early-window fee-differential hazard ratio essentially unchanged ($0.55 \rightarrow 0.57$, $p = 0.03$). The low-fee-backlog placebo then provides a sharper test: a propagation confound would have this non-capturable churn depress the early-window hazard, yet it loads with a hazard ratio above one (a faster, not slower, next block), while the fee differential retains its early-window slowdown ($HR_{0–30s} = 0.53$, $p = 0.02$). As in the cross section, the non-capturable congestion measure carries the wrong sign for the confound exactly where the capturable differential reproduces the result, isolating the strategic channel from the propagation one.

5.4 Robustness of the Early-Window Effect

We conduct five complementary robustness checks of our finding that the next block arrives unusually slowly in the first 30 seconds after a new block. Our tests address three concerns: that the finding may occur as an artifact of (i) how the spell is split, (ii) the choice of a 30-second cutoff, or (iii) the binary step functional form. Table 8 summarizes the early- and late-window hazard ratios across all five specifications.

Implementation. We replicate the piecewise split using a `tt()` formulation in place of `survSplit`:[23] the fee differential enters with a main term plus a time-transformed increment active only for $t > 30$ seconds, so the early-window loading is the main coefficient and the late-window loading adds the increment. The early-window hazard ratio is nearly identical to Column (2) of Table 6. We also estimate an *early-only* variant in which the fee differential is active only for $t \leq 30$ seconds with no main term, constraining the late-window coefficient to zero; results are again unchanged, confirming that the late-period coefficient in the unconstrained model is not spuriously absorbing part of the early effect. Finally, we estimate a three-interval `survSplit` with boundaries at 5 and 30 seconds, which gives the fee differential a separate absolute loading in each of $(0, 5]$s, $(5, 30]$s, and $t > 30$s: the early column reports the $(5, 30]$s hazard ratio and the late column the $t > 30$s ratio, isolating the early-window effect from the very first seconds after block arrival.

Cutoff. Figure 8 shows estimates for the piecewise Cox model with the window varied from 10 to 90 seconds in five-second steps. The early-window hazard ratio is significantly below one throughout and attenuates toward one as the window widens, confirming that the effect is concentrated in the early window and not driven by the 30-second choice.

Functional form. We replace the binary step function with a continuous exponential-decay interaction, $\Delta \cdot e^{-t/10}$, so that the fee differential loading decays smoothly from its block-arrival value with a ten-second time constant. The coefficient is negative and significant, consistent in sign and significance with the step-function estimates; the implied early-window hazard ratio is steeper, as the exponential weighting concentrates the entire loading at the moment of block arrival. Across all five checks the qualitative conclusion is unchanged: the fee differential effect on

[23] `tt()` ("time transform") is the R *survival* package's facility for time-varying effects in `coxph`: a covariate wrapped in `tt()` is re-evaluated as a specified function of elapsed time at each event time, fitting the same time heterogeneity as `survSplit` but without expanding the data into counting-process records.

block-arrival times is currently a short-horizon phenomenon.

Proportional hazards and linearity. We assess the proportional hazards assumption for both the static (Table 9) and piecewise (Table 10) specification using scaled Schoenfeld residuals (Grambsch and Therneau, 1994). Both tables report covariate-specific tests and the global test. None of the individual tests rejects proportionality, and the global test likewise does not indicate a violation. The piecewise specification therefore adequately captures the time heterogeneity in the fee-differential effect without introducing residual proportional hazards violations. Functional-form diagnostics based on martingale residuals (Figure 9) do not indicate departures from linearity for the fee-differential measures. Table 11 examines the functional form of the fee differential. The early-window hazard ratio is below one and of similar magnitude across all four specifications (point estimates between 0.49 and 0.55), and it remains significant under the log and inverse-hyperbolic-sine transforms, which compress the right tail while preserving the ordering of high-fee-differential blocks. Winsorizing the top one percent of $\tilde{\Delta}/S$ leaves the point estimate essentially unchanged (HR $= 0.54$) but inflates its standard error and renders it insignificant, reflecting that identification draws on variation in the upper tail of the fee differential. The late-window ratio remains indistinguishable from one throughout.

6 Conclusion

This paper studies Bitcoin miners' incentives to not recognize their competitors' newly mined blocks and instead try to create competing blocks that capture the same user transaction fees. These so-called attacks on the Bitcoin blockchain, which can lead to forks known as block races, occur relatively infrequently today. However, they seem likely to increase significantly as Bitcoin transitions away from a reward system based mainly on a fixed subsidy per block toward a regime that places increasing weight on market-based user transaction fees. While theory papers

have recognized this potential problem for a number of years, our work presents the first empirical evidence connecting blockchain attacks to the ever-changing magnitude of available user fee rewards in the Bitcoin mempool. We use a far more detailed dataset than has been available to other Bitcoin researchers up to now, drawing upon nearly eight years of precise block and transaction data from the forensics firm Chainalysis.

We develop a hurdle rate model based upon the difference between the mempool fees available from attempting to extend the blockchain compared to the fees that could be earned by attempting to fork the blockchain with a competing block. Our model links the reward-normalized fee differential to the probability and timing of block races, and three findings emerge. The incidence of block races increases in the fee differential at the moment of block arrival. The block-arrival hazard decreases in the fee differential, consistent with a temporary diversion of hashrate away from the canonical chain tip and toward a possible attacking block. This hazard effect is concentrated in the first 30 seconds of the inter-arrival spell and attenuates toward zero thereafter, as new transactions enter the mempool and make the attacking opportunity less and less attractive.

Our paper makes two contributions. We provide a theoretical framework for how fee differentials affect miner behavior and derive a closed-form attack threshold, and we present the first empirical evidence that these fee-related incentives already shape dysfunctional miner behavior. The economic magnitudes we estimate today are modest because fee differentials in our sample rarely approach the unconditional hurdle.

The block subsidy operates as a security buffer: profitable deviation at full propagation requires a fee differential proportional to S, and at $S = 3.125$ BTC that threshold binds for only a small share of blocks, while the attack window that opens after each block arrival (Section 3.7) closes within a few seconds. The mechanism is therefore brief rather than absent: current settlement security rests on the shortness of the attack window, and the declining subsidy raises the normalized differential at every block and thereby lengthens it. Holding the empirical 2020–2024 fee differential fixed, the share of blocks for which the hurdle rate binds

for a miner with 25% hashrate share at full propagation rises from roughly 0.003% at the fifth halving ($S = 1.5625$) to 43% in the limit ($S \to 0$). While these projections are out of sample and transaction fees may co-evolve with the subsidy, the qualitative direction is uncontroversial: as the subsidy declines, its buffer erodes, and a given fee differential clears the hurdle at progressively more blocks.

Beyond this erosion of the subsidy buffer, the fee differential itself is likely to widen from two structural shifts unrelated to subsidy decay. Continued growth of layer-2 protocols and additional applications creates transactions whose on-chain footprint is small but whose contestable value is large. A miner reorganizing a block containing settlement transactions of L2 systems can, in principle, substitute or invalidate these transactions and capture value beyond the displayed fee, converting reordering rents (MEV) into an additional component of the effective fee differential. Looking further ahead, a quantum adversary running Shor's algorithm (1997) could recover private keys from exposed public keys, rendering many outputs effectively claimable by anyone with access to the technology. In a post-quantum regime where the migration to quantum-resistant scripts is incomplete, vulnerable UTXOs become miner-extractable rents, and the implicit value of replacing a block extends from its transaction fees to its entire transaction value. Both channels imply that the fee differential as observed in the mempool may understate the true incentive to deviate and that the variance of this incentive will likely grow.

The instability this variance can induce matters beyond the protocol itself. Bitcoin now sits inside the regulated financial system as a settlement asset for spot ETFs, futures and options markets, and a growing set of collateralized credit and custody arrangements. Each of these instruments embeds an assumption of probabilistic settlement finality whose horizon lengthens as the fee-to-subsidy ratio rises, since attacks and reorganizations become more likely. Settlement-risk parameters used in margining, custody, and clearing should therefore be benchmarked to a forward-looking distribution of fee variance rather than historical data.

References

Angrist, J. D. and Pischke, J.-S. (2009), *Mostly Harmless Econometrics: An Empiricist's Companion*, Princeton University Press.

Bahrani, M., Neuder, M. and Weinberg, S. M. (2025), Selfish mining under general stochastic rewards, *in* '7th Conference on Advances in Financial Technologies (AFT 2025)', Vol. 354 of *Leibniz International Proceedings in Informatics (LIPIcs)*.
URL: *https://arxiv.org/abs/2502.20360*

Bar-Zur, R., Abu-Hanna, A., Eyal, I. and Tamar, A. (2022), 'WeRLman: To tackle whale (transactions), go deep (RL)', Cryptology ePrint Archive, Paper 2022/175.
URL: *https://eprint.iacr.org/2022/175*

Budish, E. (2025), 'Trust at scale: The economic limits of cryptocurrencies and blockchains', *The Quarterly Journal of Economics* **140**(1), 1–62.
URL: *https://academic.oup.com/qje/article/140/1/1/7824430*

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

Cox, D. R. (1972), 'Regression models and life-tables', *Journal of the Royal Statistical Society. Series B (Methodological)* **34**(2), 187–220.
URL: *http://www.jstor.org/stable/2985181*

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

Efron, B. (1977), 'The efficiency of cox's likelihood function for censored data', *Journal of the American Statistical Association* **72**(359), 557–565.
URL: *http://www.jstor.org/stable/2286217*

Eyal, I. and Sirer, E. G. (2013), 'Majority is not enough: Bitcoin mining is vulnerable'.
URL: *https://arxiv.org/abs/1311.0243*

Firth, D. (1993), 'Bias reduction of maximum likelihood estimates', *Biometrika* **80**(1), 27–38.
URL: *https://doi.org/10.1093/biomet/80.1.27*

Garratt, R. J. and van Oordt, M. R. C. (2023), 'Why fixed costs matter for proof-of-work–based cryptocurrencies', *Management Science* **69**(11), 6482–6507.
URL: *https://doi.org/10.1287/mnsc.2023.4901*

Gong, T., Minaei, M., Sun, W. and Kate, A. (2022), Towards overcoming the undercutting problem, *in* I. Eyal and J. Garay, eds, 'Financial Cryptography and Data Security', Springer International Publishing, Cham, pp. 444–463.

Grambsch, P. M. and Therneau, T. M. (1994), 'Proportional hazards tests and diagnostics based on weighted residuals', *Biometrika* **81**(3), 515–526.
URL: *http://www.jstor.org/stable/2337123*

Harvey, C. R., John, K. and Saleh, F. (2025), 'Productivity enables security: The economics of blockchain settlement'. SSRN Working Paper.
URL: *https://ssrn.com/abstract=5868863*

Hinzen, F. J., John, K. and Saleh, F. (2022), 'Bitcoin's limited adoption problem', *Journal of Financial Economics* **144**(2), 347–369.
URL: *https://www.sciencedirect.com/science/article/pii/S0304405X22000198*

King, G. and Zeng, L. (2001), 'Logistic regression in rare events data', *Political Analysis* **9**(2), 137–163.

Nakamoto, S. (2008), 'Bitcoin: A peer-to-peer electronic cash system', https://bitcoin.org/bitcoin.pdf.

Palatinus, M. (2012), 'Stratum mining protocol'. Original specification, Slush Pool (mining.bitcoin.cz).
URL: *https://stratumprotocol.org/specification*

Rebora, P., Salim, A. and Reilly, M. (2014), 'bshazard: A flexible tool for nonparametric smoothing of the hazard function', *The R Journal* **6**(2), 114–122.

Sarenche, R., Aghabagherloo, A., Nikova, S. and Preneel, B. (2025), 'Bitcoin under volatile block rewards: How mempool statistics can influence bitcoin mining'.
URL: *https://arxiv.org/abs/2411.11702*

Shor, P. W. (1997), 'Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer', *SIAM Journal on Computing* **26**(5), 1484–1509.
URL: *http://dx.doi.org/10.1137/S0097539795293172*

7 Appendix

7.1 Lead-of-Two Extension

Under the lead-of-two resolution rule, the block race at state $(1,1)$ continues until one chain obtains a two-block advantage. This generates an infinite-state Markov chain (the "deuce game"). Let $\beta_{\text{eff}} = \beta_A + \alpha\beta_R$ denote the effective hashrate on the attacker's chain. At each state where the chains are tied, the next block is found by C^A with probability β_{eff} and by C^0 with probability $1-\beta_{\text{eff}}$, assuming the miners, including the attacker, stay loyal to the branch they first chose and do not abandon their chain when temporarily trailing.[24] The probability that C^A ultimately wins from state $(1,1)$ is:

$$P_2^{(2)}(\alpha) \;=\; \frac{\beta_{\text{eff}}^2}{\beta_{\text{eff}}^2 + (1-\beta_{\text{eff}})^2} \tag{25}$$

Equation (25) follows from solving a three-state Markov system. Let $P(k)$ denote the probability that C^A ultimately wins when its lead is $k \in \{-1, 0, +1\}$ blocks, with absorbing barriers at $k = \pm 2$:

$$\begin{aligned} P(1) &= \beta_{\text{eff}} + (1-\beta_{\text{eff}})P(0) \\ P(0) &= \beta_{\text{eff}}P(1) + (1-\beta_{\text{eff}})P(-1) \\ P(-1) &= \beta_{\text{eff}}P(0) \end{aligned}$$

Substituting $P(-1) = \beta_{\text{eff}}P(0)$ into the middle equation and $P(1) = \beta_{\text{eff}} + (1-\beta_{\text{eff}})P(0)$ into the result yields $P(0)\left[\beta_{\text{eff}}^2 + (1-\beta_{\text{eff}})^2\right] = \beta_{\text{eff}}^2$, which gives (25) as $P_2^{(2)} \equiv P(0)$. The attacker mines at rate β_A in both the winning and losing scenarios, so expected rewards from block $h+2$ onward are identical under both outcomes and cancel from the

[24] This is a simplifying assumption. A richer model would allow fringe miners to re-optimize at each intermediate state k, making transition probabilities state-dependent and reintroducing strategic complementarity at every node of the race tree. The closed-form derivation below relies on stationary transition probabilities and therefore requires this commitment assumption.

comparison. The lead-of-two attack condition is therefore:

$$P_1(\tau) \cdot P_2^{(2)} \cdot R_h^A \;\geq\; \beta_A \cdot R_{h+1} \tag{26}$$

Under lead-of-two, the switching condition for fringe miners reintroduces *strategic complementarity*. A fringe miner's payoff now depends on the *probability of their chain ultimately winning*, which is a function of β_{eff}. The expected reward for miner j on C^A is proportional to $P_2^{(2)}(\alpha)$, so the switching condition becomes:

$$R_{h+1}^A \cdot P_2^{(2)}(\alpha) \;\geq\; R_{h+1}^0 \cdot (1 - P_2^{(2)}(\alpha)) \tag{27}$$

Because $P_2^{(2)}$ is increasing in α (more support raises β_{eff}, raising the winning probability), this generates two stable pure-strategy Nash equilibria: no coordination ($\alpha = 0$) and full coordination ($\alpha = 1$). The threshold reward share for full coordination, $\rho \equiv R_{h+1}^A/(R_{h+1}^A + R_{h+1}^0)$, must satisfy $\rho \geq \beta_0^2/[(1-\beta_0)^2 + \beta_0^2]$, which is easily satisfied when β_0 is small.

The comparative statics with respect to S, $\gamma(\tau)$, and β_A are qualitatively identical to the lead-of-one case: the threshold is increasing in S and γ, and decreasing in β_A.

7.2 Coordination and Undercutting

The solo attack threshold assumes no support from other miners ($\alpha = 0$). An attacker can improve their odds by strategically constructing B^A to attract the rational fringe. The key instrument is *undercutting*: the attacker includes fewer fees in B_h^A than in the incumbent block B_h ($F_h^A < F_h$), leaving those transactions unclaimed in $h{+}1$ on the attacker's chain. This increases the fees available for block $h+1$:

$$F_{h+1}^A \;\approx\; \tilde{F}_{h+1}^* + (F_h - F_h^A) \tag{28}$$

The approximation holds when the mempool is sufficiently thin that the omitted transactions fit within the block size limit at height $h{+}1$. Under

congestion, the block size constraint binds and only the highest-fee subset of the omitted transactions displaces lower-fee transactions in the $h+1$ candidate block, so the effective increase in F^A_{h+1} is smaller than $F^*_h - F^A_h$.

The switching condition under lead-of-one. Under the lead-of-one rule, the resolution block at height $h+1$ is the single prize that determines the race outcome. A fringe miner with hashrate β_j who mines on chain C earns R^C_{h+1} if *they* are the one to find block $h+1$, an event of probability β_j, independent of how many other miners are on the same chain. Their expected reward is therefore $\beta_j \cdot R^C_{h+1}$, regardless of α.[25] The switching condition is:

$$R^A_{h+1} \;\geq\; R^0_{h+1} \qquad \Longleftrightarrow \qquad F^A_{h+1} \;\geq\; F^0_{h+1} \tag{29}$$

This comparison requires that fringe miners can observe the content of both competing blocks. Observability holds if B^A_h is broadcast at the onset of the race: once both blocks are visible, fringe miners can directly compare F^A_{h+1} and F^0_{h+1} before committing hashrate. If the attacker undercuts by even a marginal amount ($F^A_h < F_h$), then $F^A_{h+1} > F^0_{h+1}$ and all rational fringe miners strictly prefer C^A. The coordination problem under lead-of-one is therefore trivially resolved: any undercutting, however small, attracts the entire fringe ($\alpha = 1$, $\beta_{\text{eff}} = 1 - \beta_0$). Under the lead-of-two rule, individual payoffs depend on the probability of ultimately winning the race, which is a function of β_{eff}; this reintroduces strategic complementarity and generates multiple equilibria, as shown in Appendix 7.1.

What determines α. Whether rational miners actually join the attacker depends on three factors. First, *network propagation*: a fringe miner can only switch to C^A if they have received B^A_h; the fraction who do so before committing hashrate to C^0 depends on the attacker's net-

[25] In the Poisson mining race, the probability that any specific miner finds the next block is their hashrate share. If miner j finds $h+1$ on C^A, that block simultaneously extends C^A and resolves the race, so the reward is guaranteed.

work position and broadcast timing, and is closely related to the propagation function $\gamma(\tau)$ from Phase 1. Second, *behavioral rules*: some miners follow a first-seen heuristic (mining on whichever valid block arrived first) regardless of the fee comparison in (29); these miners contribute to α only if B_h^A arrived before the incumbent block. Third, *own-attack competition*: some miners may be mining their own competing block at height h, joining neither C^A nor C^0. If fraction α_{own} of β_R is occupied on a third chain, they are unavailable to either side, imposing a ceiling $\alpha \leq 1 - \alpha_{\text{own}}$ and reducing the attacker's maximum Phase 2 win probability to $\beta_A + (1 - \alpha_{\text{own}})\beta_R$. In general, $\alpha \in [0, 1]$ denotes the fraction of rational miners joining C^A.

The general hurdle rate. With fraction α of rational miners joining C^A, the effective Phase 2 win probability is $\beta_{\text{eff}}(\alpha) = \beta_A + \alpha\beta_R$. With marginal undercutting ($x \to 0^+$, so $R_{h+1}^A \approx R_{h+1}$), the attack condition yields the parametric hurdle rate:

$$R^{A*}(\alpha) \;=\; \frac{\beta_A(1 - P_1(\tau)) \cdot R_{h+1}}{P_1(\tau) \cdot (\beta_A + \alpha\beta_R)} \tag{30}$$

This is strictly decreasing in α: each additional unit of fringe support lowers the threshold. The two limiting cases recover earlier results: $\alpha = 0$ gives the solo rate (11), and $\alpha = 1$ gives the fully coordinated rate $R_{\text{coord}}^{A*} = \beta_A(1 - P_1)R_{h+1}/[P_1(1 - \beta_0)]$, which is lower than the solo rate by a factor of $\beta_A/(1 - \beta_0)$. The empirical analysis uses the solo rate as a conservative benchmark; the true hurdle rate is lower by a factor of $\beta_A/(\beta_A + \alpha\beta_R)$ for any $\alpha > 0$.

The attacker's tradeoff and optimal undercutting. Let $x = F_h - F_h^A \geq 0$ denote the amount undercut. With fraction α of rational miners joining C^A, the attack payoff is:

$$U_A(x, \alpha) \;=\; P_1\big[(\beta_A + \alpha\beta_R)(S + F_h - x) \;+\; \beta_A(S + \tilde{F}_{h+1}^* + x)\big] \tag{31}$$

Differentiating with respect to x:

$$\frac{\partial U_A}{\partial x} \;=\; P_1\big[\beta_A - (\beta_A + \alpha\beta_R)\big] \;=\; -P_1 \cdot \alpha\beta_R \;\leq\; 0 \tag{32}$$

For any $\alpha > 0$, the payoff is strictly decreasing in x: every additional unit of undercutting transfers reward from the higher-probability winning term $(\beta_A + \alpha\beta_R)$ to the lower-probability own-mining term β_A, reducing total expected payoff. The attacker therefore sets $x \to 0^+$: marginal undercutting is sufficient to attract the rational fringe while preserving nearly all of block h's reward.

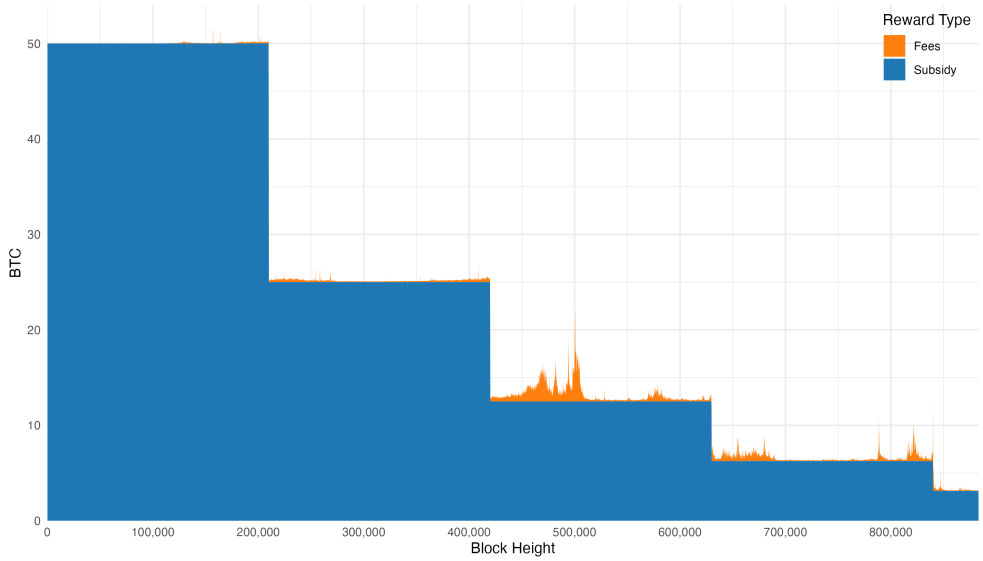
7.3 Transaction Visibility

Our counterfactual fee differentials are constructed from the transactions we observe in the public mempool before each block arrives, which requires that our mempool approximate the transaction set available to miners (Section 4.2). Figure 10 assesses this by splitting confirmed transactions into those observed in our mempool before the block arrived (*public*) and those first seen only at inclusion (*private*). Private transactions are a small share of total volume throughout the sample, rising modestly only above height ∼880,000.

7.4 Cox PH Checks

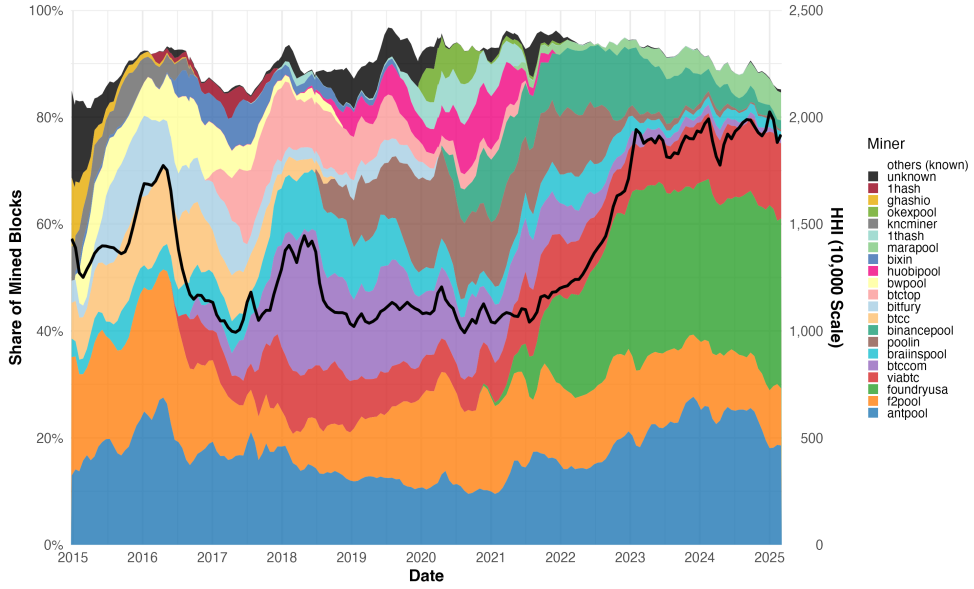
Scaled Schoenfeld residuals tests for the static and piecewise Cox models are reported in Tables 9 and 10 in Section 5. Table 12 reports a specificity check in which the early/late period split is applied to every covariate, not only the fee differential, and tests for each whether its early-window and late-window coefficients differ. The realized fee differential is the only covariate with a statistically significant early-versus-late difference, confirming that the temporal heterogeneity documented in Section 5 is specific to the fee differential rather than a generic feature of the counting-process split.

Figure 1: Bitcoin block rewards: subsidy and transaction fees, 2009–2025.



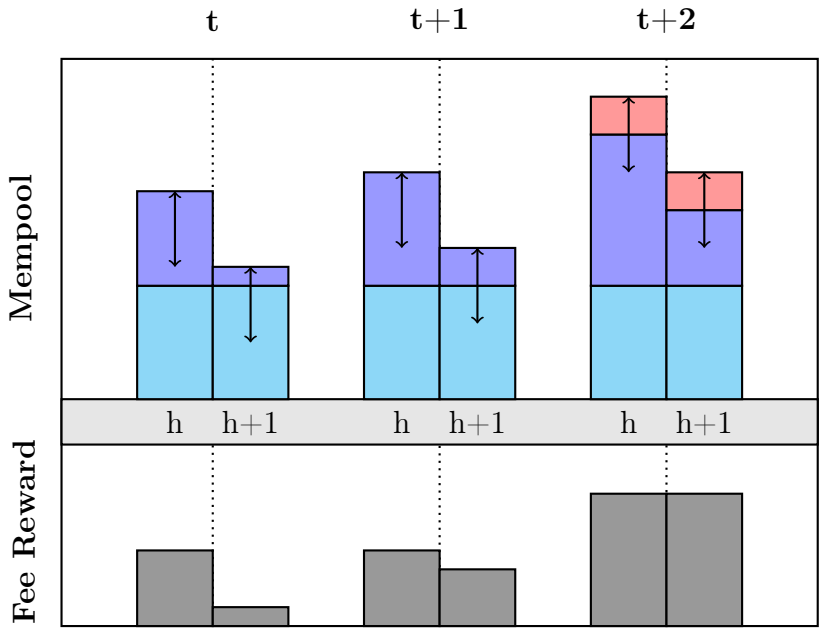
Note: For each calendar day, we average per-block subsidy (new issuance) and per-block transaction fees across all blocks mined that day; both are in BTC per block (vertical axis) and stack to total mean block revenue. The horizontal axis is block height, recorded as the smallest height among blocks on that day. Sample: January 2009–February 2025 through height 883,602. The subsidy halves every 210,000 blocks.

Figure 2: Evolution of Bitcoin mining power distribution and market concentration, 2015–2025.



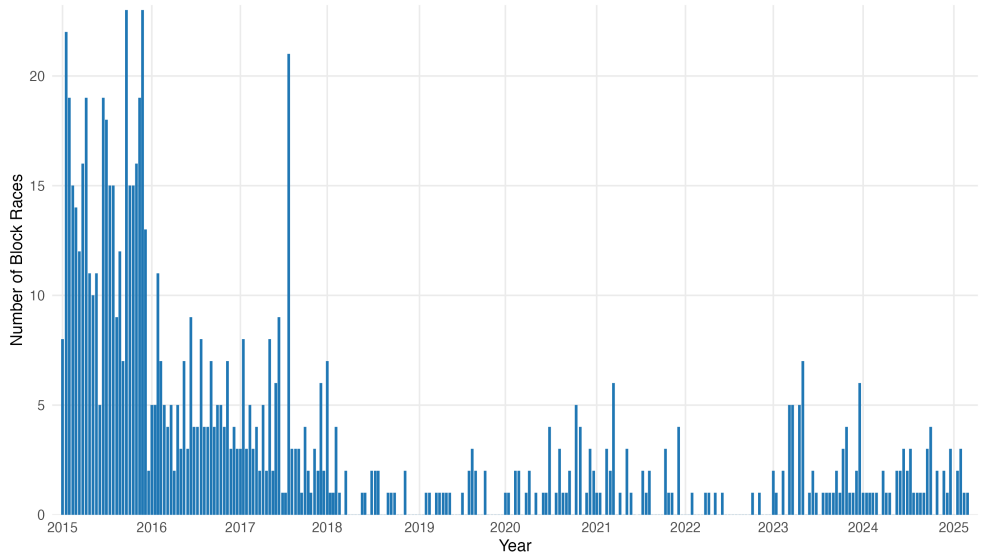
Note: Stacked areas show each pool's share of discovered blocks (left axis, percent of blocks), January 2015–early 2025, computed in a 40-day rolling window advanced in 20-day steps; shares proxy relative hashrate contribution. Pools with at least a 5% share are labeled individually; remaining mass is grouped as "others (known)" or "unknown." The solid line plots the Herfindahl-Hirschman index $\text{HHI} = \sum_i s_i^2$ on the conventional 0–10,000 scale (right axis). The HHI trace is offset vertically from the stacked series for legibility; tick labels report true index values. An HHI value below 1,500 typically indicates a competitive market.

Figure 3: Delta decay: temporal convergence of fee incentives between competing mining strategies.



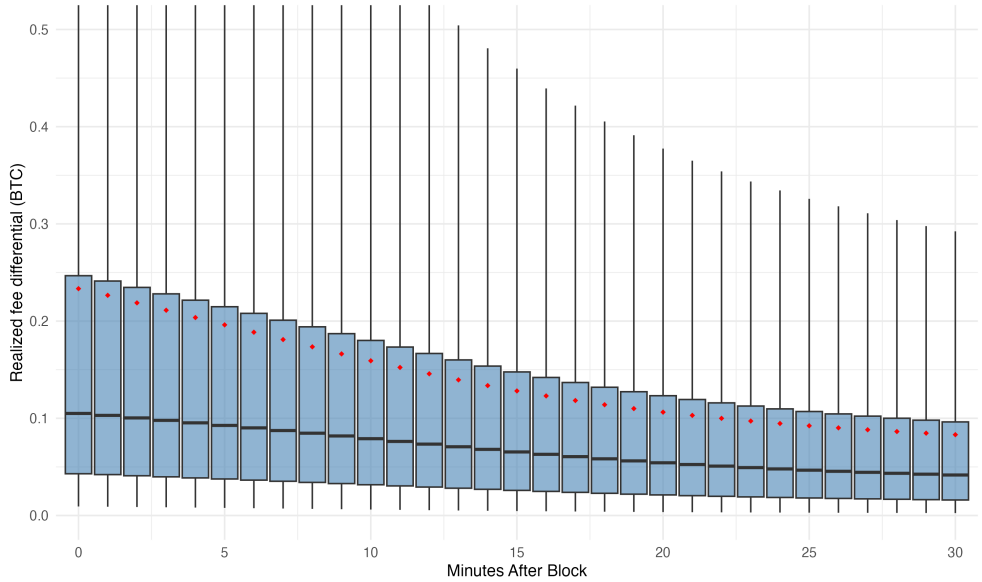
Note: Conceptual illustration of *delta decay*. At each time step t, the top panel shows the mempool composition for strategies h (re-mining the current block) and $h+1$ (honest extension). The double-headed arrows indicate the block size limit. As new transactions enter the mempool, both strategies can include them, eroding the fee differential $\Delta(t)$ shown in the bottom panel.

Figure 4: Stale block frequency by difficulty epoch, 2015–2025.



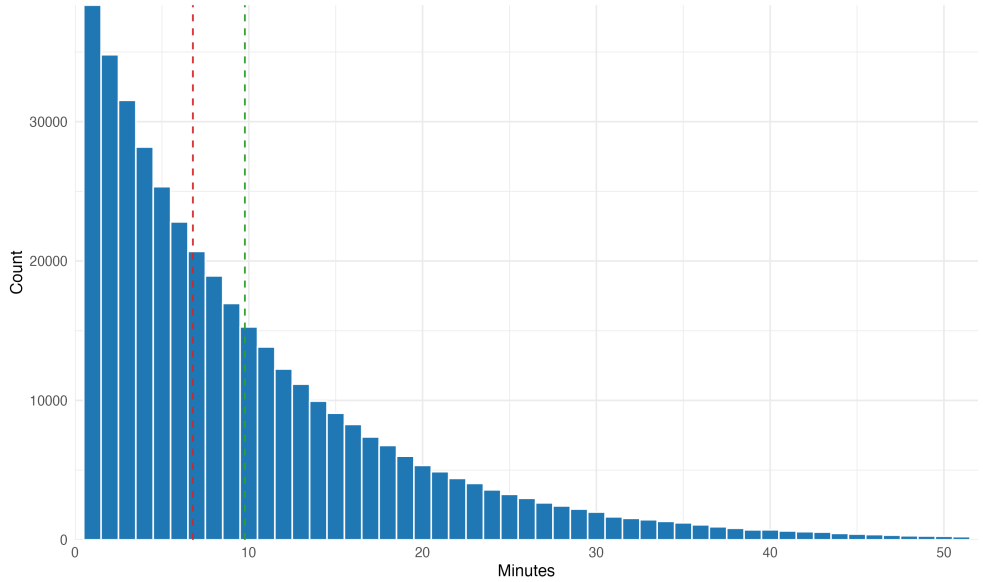
Note: Count of observed stale blocks per difficulty epoch ($\approx$2,016 blocks). Sample: 273 epochs spanning January 2015 through March 2025 (859 total race events). The sharp decline after 2018 coincides with the deployment of *Compact Blocks* (BIP 152) and high-speed relay networks such as *FIBRE*. The August 2017 spike of 21 stale blocks corresponds to the Bitcoin Cash hard fork at heights 478,559–478,576, during which competing implementations produced blocks at the same heights as a consequence of the chain split; we recode these heights as non-races in the regression sample.

Figure 5: Empirical delta decay: realized fee differential, 2017–2025.



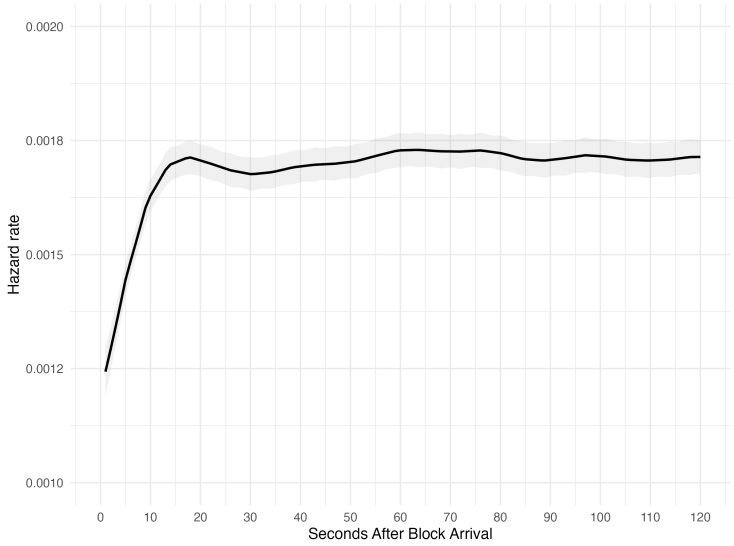
Note: Boxplots of the *realized* fee differential $\tilde{\Delta}(t) = F_h^*(t) - \tilde{F}_{h+1}^*(t)$ at one-minute snapshots after block arrival, computed over 420,103 block intervals (heights 463,500–883,602). Boxes denote the interquartile range with the median indicated by the central line; red diamonds mark the mean at each snapshot; whiskers extend from the 5th to the 95th percentile of $\tilde{\Delta}(t)$. The vertical axis is on a linear scale clipped to 0–0.5 BTC; observations outside this range enter the quantile and mean calculations but are not displayed.

Figure 6: Distribution of Bitcoin block inter-arrival times, 2017–2025.

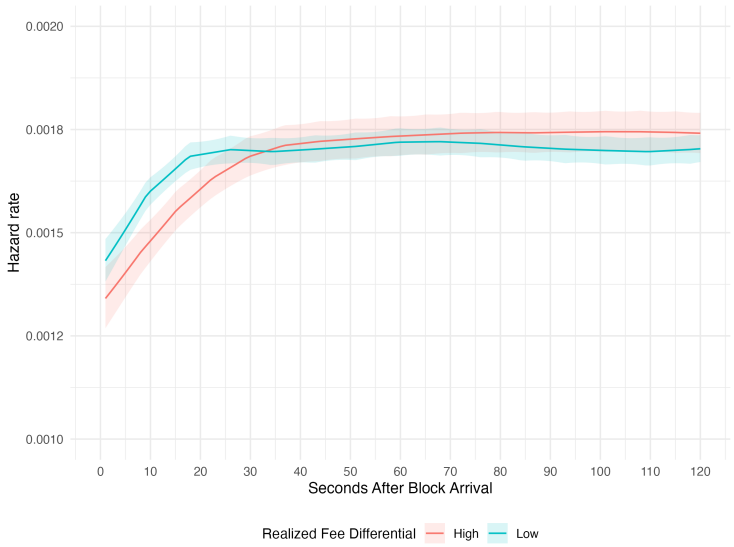


Note: Histogram of consecutive canonical-chain block inter-arrival times (April 2017–February 2025; $n = 412{,}804$). Each spell is the difference between successive blocks' network arrival timestamps (equivalently, the waiting time until height h's successor extends the chain). The horizontal axis measures spell length in minutes using one-minute bins; the vertical axis reports bin counts. The green and red dashed vertical lines mark the sample mean (586.4 seconds) and median (407 seconds), respectively. Constant-intensity hashing implies exponentially distributed inter-arrival times under idealized conditions.

Figure 7: Non-parametric hazard estimates for block inter-arrival times, 2017–2025.



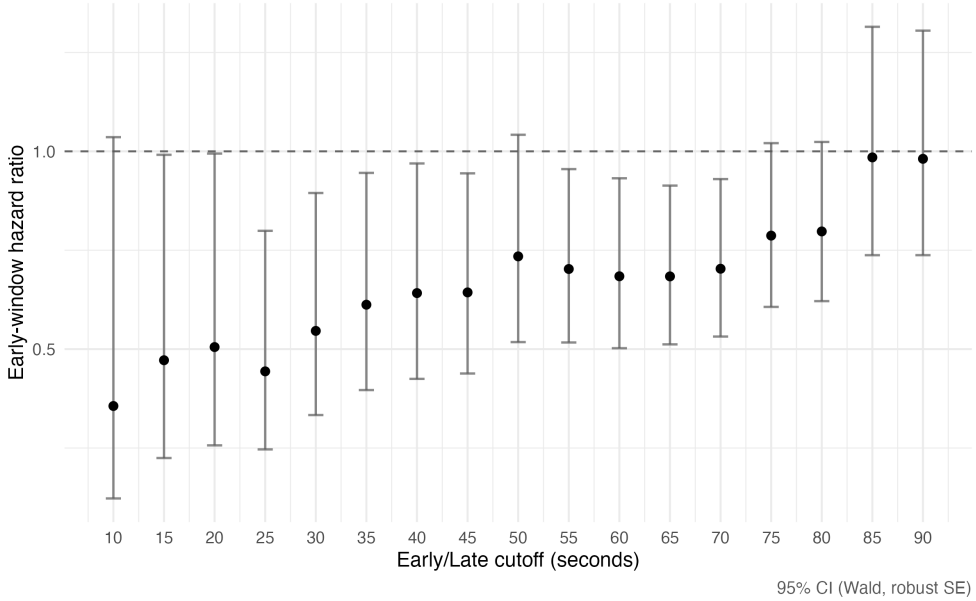
(a) Pooled hazard function



(b) Hazard function stratified by realized fee gap at block arrival (Low: bottom 75%; High: top 25%, split at the third quartile)

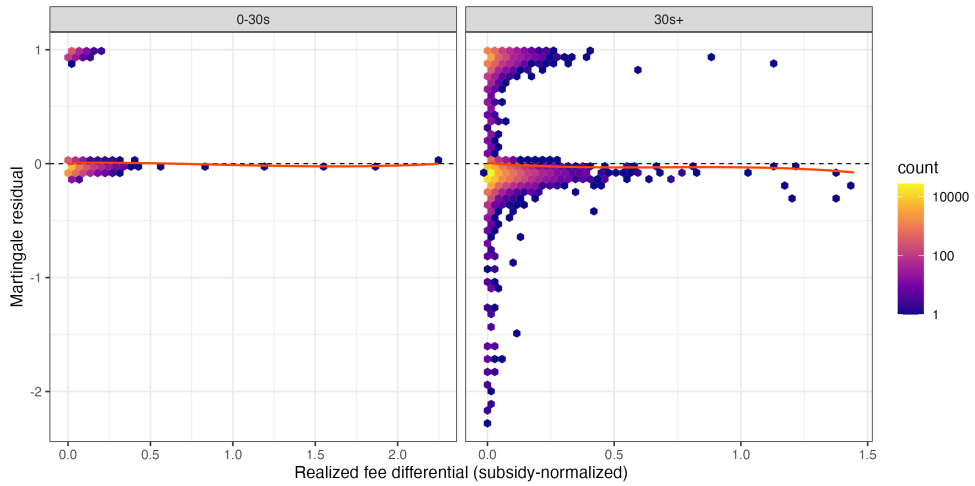
Note: Both panels plot smoothed hazard rates estimated using a penalized B-spline smoother (Rebora et al., 2014); the horizontal axis measures seconds since the previous block arrival. The initial ramp-up during the first ∼30 seconds reflects reduced effective hashrate on the canonical chain, attributable to propagation delay, strategic mining on the previous height, or both. Panel (b) stratifies on realized fee differential $\tilde{\Delta}(0)$ at block arrival: *Low* comprises the bottom three quartiles of its distribution and *High* the upper quartile (75th percentile cutoff). Sample: 412,804 block intervals (April 2017–February 2025).

Figure 8: Sensitivity of the early-window fee-differential hazard ratio to the cutoff window, 2017–2025.



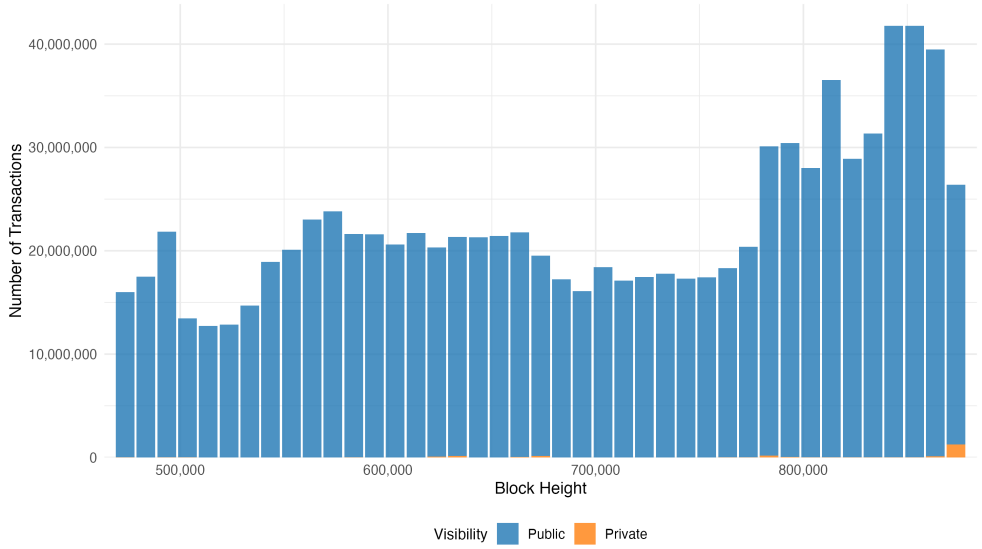
Note: Each point plots the hazard ratio for the *realized fee differential* in the early window of a piecewise Cox model in which the cutoff is swept from 10 to 90 seconds in five-second steps. Shaded band is the 95% confidence interval (delta-method SE, clustered at the difficulty epoch). The sample and controls are identical to Column (2) of Table 6. A hazard ratio below one indicates that a larger fee differential lowers the arrival rate of the next canonical block within the early window.

Figure 9: Martingale residuals against the realized fee differential, 2017–2025.



Note: Each panel plots martingale residuals from the piecewise Cox model (Column (2) of Table 6) against the realized fee differential $\tilde{\Delta}/S$, separately for the early window ($t \leq 30$s, left) and the late window ($t > 30$s, right). Hex bins show the local density of observations ($\log_{10}$ scale); the orange curve is a loess smoother. A flat smoother centered at zero indicates that the log-hazard is linear in $\tilde{\Delta}/S$ within each window. The early window smoother is approximately flat across the range of the fee differential, supporting the linearity assumption where the significant effect is identified. The vertical mass of negative residuals near $\tilde{\Delta}/S \approx 0$ in the late window reflects long spells with near-zero fee differential and no event, which is expected under the model. Random sample of 100,000 counting-process rows.

Figure 10: Transaction visibility: public versus private transactions by block height.



Note: Aggregate count of confirmed transactions by block-height bin, split by visibility. *Public* transactions were observed in our mempool before the block arrived; *private* transactions were first seen only at block inclusion. Across the full sample, private transactions account for 2,224,833 of 956,103,073 confirmed transactions (0.23%); they remain a small share throughout and rise modestly only in the most recent period (heights above ∼880,000).

Table 1: Summary statistics.

Statistic	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
Attacking block fee (BTC)	0.587	1.038	0.0001	0.087	0.212	0.625	85.251
Continuing block fee (BTC)	0.354	0.766	0.000	0.020	0.075	0.308	20.174
Realized fee differential (BTC)	0.233	0.434	0.000	0.043	0.104	0.246	84.029
Subsidy-normalized realized fee differential	0.027	0.052	0.000	0.006	0.014	0.030	13.445
Reward-normalized realized fee differential	0.025	0.041	0.000	0.006	0.014	0.028	11.247
Mempool fee differential (BTC)	0.289	0.519	0.000	0.054	0.126	0.307	84.062
Subsidy-normalized mempool fee differential	0.032	0.058	0.000	0.008	0.017	0.036	13.450
Reward-normalized mempool fee differential	0.030	0.047	0.000	0.008	0.016	0.034	11.301
Miner share	0.130	0.083	0.000	0.072	0.120	0.173	0.361
Block size (MB)	1.257	0.479	0.0002	1.049	1.291	1.544	3.994
Block private transactions	5.389	80.639	0	0	0	0	7,395
Block header timestamp interval (seconds)	586.439	586.669	0	170	407	810	8,354
Block arrival timestamp interval (seconds)	586.442	587.590	1	170	407	810	8,346
Block race (dummy)	0.001	0.026	0	0	0	0	1

Note: The sample is Bitcoin main-chain blocks at heights 463,500–883,602 (April 2017–February 2025; $N = 412{,}804$) with non-missing arrival timestamps. *Attacking block fee* is total transaction fees in block h (BTC); *continuing block fee* is the mempool-optimal fee total for honest extension at height $h{+}1$ from the arrival-time mempool snapshot (BTC). *Realized fee differential* is the difference between the two; *mempool fee differential* is the theoretical counterpart based on mempool-optimal packing at both heights. Subsidy- and reward-normalized differentials divide by the contemporaneous block subsidy and by subsidy plus continuation fee reward, respectively. *Miner share* is B_h miner's share of total hashrate computed as the share of blocks mined in the preceding difficulty epoch. *Block race* equals one when a competing block at height h is observed. Header and arrival intervals are consecutive block timestamps in seconds. *Private transactions* are transactions in B_h not observed in the public mempool prior to that block's arrival.

Table 2: Block race frequency by reward-normalized fee differential quartile, 2017–2025.

Fee differential quartile	Blocks	Block races	Races per 10,000 blocks
Q1 (lowest)	103,201	51	4.94
Q2	103,201	45	4.36
Q3	103,201	64	6.20
Q4 (highest)	103,201	124	12.02
Total	412,804	284	6.88

Note: Cross-tabulation of block-race incidence by quartile of the reward-normalized mempool fee differential $\Delta(0)/(S + F^*_{h+1}(0))$ at block arrival. Quartiles are formed over the 412,804 main-chain blocks (April 2017–February 2025); the final column expresses race incidence as the number of observed races per 10,000 blocks. The race rate is essentially flat across the bottom three quartiles and roughly doubles in the top quartile. A Pearson χ^2 test rejects independence of race frequency across quartiles ($\chi^2(3) = 55.4$, $p < 0.001$).

Table 3: Block race incidence: linear probability estimates.

	Block race indicator				
	(1)	(2)	(3)	(4)	(5)
Mempool fee differential (norm.)	0.00514*** (0.00135)	0.00339*** (0.00117)	0.00331*** (0.00117)	0.00331*** (0.00124)	0.00295** (0.00116)
Post FIBRE upgrade		−0.00096*** (0.00026)	−0.00109*** (0.00026)	−0.00109*** (0.00032)	
Relay bug week		0.00417* (0.00246)	0.00391 (0.00246)	0.00391*** (0.00033)	0.00503** (0.00245)
Miner share			0.00064 (0.00050)	0.00064 (0.00050)	0.00075 (0.00050)
Block size			0.00026*** (0.00008)	0.00026*** (0.00009)	0.00024*** (0.00008)
Private tx count			0.0000001 (0.0000004)	0.0000001 (0.0000004)	0.0000001 (0.0000004)
Constant	0.00053*** (0.00005)	0.00148*** (0.00026)	0.00119*** (0.00027)	0.00119*** (0.00033)	0.00012 (0.00011)
Estimator	LPM	LPM	LPM	LPM	LPM
Standard errors	HC1	HC1	HC1	CL (epoch)	HC1
Sample	Full	Full	Full	Full	Fibre
Block races	284	284	284	284	238
Mean dep. var.	0.00069	0.00069	0.00069	0.00069	0.00062
Observations	412,804	412,804	412,804	412,804	386,304
R^2	0.00008	0.00025	0.00028	0.00028	0.00018
Note:					*p<0.1; **p<0.05; ***p<0.01

Note: Linear probability models of block-race incidence. The dependent variable equals one if a competing block is observed at height h and zero otherwise (sample mean $\approx 0.069\%$). The regressor of interest is the reward-normalized fee differential at block arrival, $\Delta(0)/(S+F^*_{h+1}(0))$. Column (1) is univariate; column (2) adds indicators for the prevailing block-relay regime: a post-FIBRE indicator (height $\geq$ 490,000, December 2017, when FIBRE was deployed; Compact Blocks (BIP 152) predates the sample and is in force throughout) and a relay-bug-week indicator (roughly 1,000 blocks around height 789,000, early May 2023), both relative to the pre-FIBRE baseline; column (3) adds block-level controls (miner hashrate share, block size in megabytes, and private-transaction count); column (4) repeats column (3) with standard errors clustered at the difficulty epoch; and column (5) restricts the sample to the post-FIBRE period. Standard errors in parentheses are heteroskedasticity-consistent (HC1) except in column (4), as indicated. The sample is 412,804 main-chain blocks (April 2017–February 2025), or 386,304 in the post-FIBRE subsample.

Table 4: Block race robustness.

	Block race indicator				
	LPM	Logit	Probit	Firth Logit	Cloglog
Mempool fee differential (norm.)	0.0033*** (0.0012)	0.4888*** (0.0865)	0.2310*** (0.0546)	0.5666** (0.1281)	0.4758*** (0.0809)
Post FIBRE upgrade	−0.0011*** (0.0003)	−1.2544*** (0.1728)	−0.3668*** (0.0527)	−1.2559*** (0.1730)	−1.2545*** (0.1726)
Relay bug week	0.0039 (0.0025)	0.8123* (0.4469)	0.2830* (0.1537)	0.8845* (0.4341)	0.8107* (0.4457)
Miner share	0.0006 (0.0005)	0.7528 (0.7027)	0.2186 (0.2045)	0.7442 (0.7187)	0.7519 (0.7023)
Block size	0.0003*** (0.0001)	0.4520*** (0.1180)	0.1317*** (0.0344)	0.4513*** (0.1360)	0.4518*** (0.1179)
Private tx count	0.000000 (0.000000)	0.0001 (0.0005)	0.00003 (0.0001)	0.0005 (0.0003)	0.0001 (0.0005)
Constant	0.0012*** (0.0003)	−6.8817*** (0.1890)	−3.0833*** (0.0572)	−6.8766*** (0.1957)	−6.8813*** (0.1889)
Standard Errors	HC1	HC1	HC1	Profile Lik.	HC1
Mean of Dep. Var.	0.00069	0.00069	0.00069	0.00069	0.00069
Observations	412,804	412,804	412,804	412,804	412,804
Note:					*p<0.1; **p<0.05; ***p<0.01

Note: Robustness of the block race incidence result to the choice of binary-outcome estimator. Each column re-estimates the full specification (column (3) of Table 3) under a different model: a linear probability, logit, probit, Firth penalized logit, and complementary log-log. The dependent variable, the reward-normalized fee differential, and the controls (relay regime, miner hashrate share, block size, private-transaction count) are identical across columns. Only the linear probability coefficients lie on the probability scale; the logit, probit, Firth, and complementary log-log coefficients are on their respective latent or link scales and are not directly comparable in magnitude to the linear probability model; the corresponding average marginal effects are discussed in the text. Standard errors in parentheses are heteroskedasticity-consistent (HC1), except the Firth specification, which reports profile-likelihood standard errors. Sample: 412,804 main-chain blocks (April 2017–February 2025).

Table 5: Block races: congestion controls and placebo.

	(1) Baseline	(2) Congestion	(3) Placebo
Fee differential Δ	3.31^{***}	3.02^{**}	3.14^{**}
	(1.24)	(1.22)	(1.24)
Low-fee backlog (placebo)			-0.03
			(0.06)

Note: Robustness of the block-race result to congestion confounds. Column (1) is the full linear probability specification (column (3) of Table 3). Column (2) adds to it a set of *congestion controls*: the mempool backlog, the count of block transactions first seen within ten seconds of arrival (a proxy for compact-block reconstruction round-trips), and the transaction arrival rate. Column (3) instead adds the *placebo*, the mempool backlog of low-fee transactions ($\leq$ 30 sat/vB, with the mempool backlog also controlled), alongside the fee differential, contrasting the *capturable* top of the mempool (Δ) with the *non-capturable* low-fee tail, which churns the mempool but carries negligible capturable value, so the strategic mechanism predicts no effect from it. Columns (2) and (3) are parallel extensions of column (1), not a nested sequence. Coefficients are scaled by 10^3, with epoch-clustered standard errors in parentheses. The fee differential is stable across columns, while the placebo is insignificant. *, **, *** denote $p < 0.1$, 0.05, 0.01. Sample: 412,804 main-chain blocks.

Table 6: Cox model estimates with time-varying covariates.

	Hazard of next-block arrival	
	Hazard Ratio	
	Static	Piecewise
realized fee differential (norm.)	1.0156 (0.0641)	
continuing block fee (norm.)	1.1902*** (0.0556)	1.1906*** (0.0555)
miner share	1.0416* (0.0224)	1.0416* (0.0224)
block private transactions	1.0000 (0.00002)	1.0000 (0.00002)
block size	1.0153*** (0.0047)	1.0153*** (0.0047)
realized fee differential (norm.): period 0–30s		0.5461** (0.1376)
realized fee differential (norm.): period 30s+		1.0597 (0.0683)
Events	412,804	412,804
Concordance	0.505	0.505
LR test vs null	55.52 (df=5, p<1e-10)	62.55 (df=6, p<1.4e-11)
Strata	Epoch × Day	Epoch × Day
Cluster SE	Difficulty epoch	Difficulty epoch
Ties	Efron	Efron
Observations	4,198,335	4,582,736
Note:		*p<0.1; **p<0.05; ***p<0.01

Note: Cox proportional hazards models on the counting-process representation of block inter-arrival spells. Column (1) ("Static") estimates a single coefficient on the realized fee differential; Column (2) ("Piecewise") allows the fee differential to load separately within the first 30 seconds of a spell and thereafter. Both columns include the time-varying continuing block fee and block-level controls. Coefficients are reported as hazard ratios ($\exp(\hat{\beta})$); standard errors are delta-method standard errors on the hazard-ratio scale and are clustered at the difficulty-epoch level (*Cluster SE*). Baseline hazards are stratified by difficulty epoch × calendar day; ties are handled with the Efron approximation (Efron, 1977). Under piecewise estimation, *period 0–30s* is the fee differential coefficient active during the first 30 seconds of each spell, and *period 30s+* is the coefficient active afterward. The piecewise column adds one boundary at 30 seconds via `survSplit`, splitting the sub-interval that straddles 30 seconds for every spell longer than that; it therefore has about 384,000 more row-level observations than the static column (4,582,736 versus 4,198,335, the difference being the spells that exceed 30 seconds) but the same number of events (412,804). *LR test vs null* reports the Cox partial-likelihood ratio test relative to a model with no covariates. *, **, and *** denote $p < 0.1$, $p < 0.05$, and $p < 0.01$, respectively.

Table 7: Early-window hazard: congestion controls and placebo

	(1) Baseline	(2) Congestion	(3) Placebo
Fee differential Δ (HR)	0.55^{**}	0.57^{**}	0.53^{**}
	[0.016]	[0.030]	[0.019]
Low-fee backlog (placebo, HR)			1.03^{***}
			[0.004]

Note: Robustness of the early-window hazard result to congestion confounds. Column (1) is the piecewise Cox specification of Table 6. Column (2) adds *congestion controls*: the mempool backlog, the count of block transactions first seen within ten seconds of arrival, and the transaction arrival rate. Column (3) instead adds the *placebo*, the early-window loading of the low-fee mempool backlog ($\leq$ 30 sat/vB), a non-capturable churn measure, alongside the fee differential. Columns (2) and (3) are parallel extensions of column (1). Entries are early-window (0–30 s) hazard ratios with p-values in brackets; standard errors are clustered at the difficulty epoch. The fee differential's early-window slowdown (hazard ratio below one) is stable across columns, whereas the placebo carries the wrong sign for a propagation confound (a hazard ratio above one, i.e. faster arrival). *, **, *** denote $p < 0.1$, 0.05, 0.01. Sample: 412,804 block spells.

Table 8: Robustness of the early-window fee differential effect.

Specification	Early-window HR	Late-window HR	Late definition
survSplit piecewise (baseline)	0.546** (0.138)	1.060 (0.068)	$t > 30$s
survSplit three-way (omit $t < 5$s from early slice)	0.586** (0.143)	1.060 (0.068)	$t > 30$s
`tt()` step function	0.546** (0.138)	1.060 (0.068)	$t > 30$s
`tt()`, late constrained to zero	0.545** (0.137)	—	constrained
`tt()` exponential decay $\Delta \cdot e^{-t/10s}$	0.143** (0.124)	0.908** (0.039)	$t = 10s$ ($w = e^{-1}$)

HR = $\exp(\hat{\beta})$; delta-method SE in parentheses; $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Note: Each row re-estimates the piecewise Cox specification of Column (2) of Table 6 under a different implementation of the early window interaction: (i) `survSplit` counting-process expansion with separate coefficients for $t \leq 30$s and $t > 30$s (early vs. late as in the main text); (ii) three-interval `survSplit` with cuts at 5 and 30 seconds (early-column HR for $(5, 30]$s; late-column HR for $t > 30$s); (iii) `tt()` step in which the fee differential enters with a main coefficient plus a time-transformed increment active for $t > 30$s, so the early-column HR is the main coefficient and the late-column HR adds the increment; (iv) a `tt()` step with the fee differential active only for $t \leq 30$s and no main term, which constrains the late-window HR to unity; (v) exponential-decay loading $\Delta \cdot e^{-t/\tau}$ with $\tau = 10$s, so the weight on Δ is $e^{-t/10}$ within a spell, with τ set so the loading is roughly 95% decayed by the 30-second propagation window and the smooth form thus mirrors the binary 30-second split (early-column HR at $t = 0$, where $w = 1$; late-column HR at $t = 10$s, where the weight is e^{-1}). Columns report the hazard ratio on the fee differential ($\Delta(0)/S$) in the respective window, with delta-method standard errors clustered at the difficulty-epoch level. All specifications include the time-varying continuing block fee, miner hashrate share, block size, private transaction count, and stratification by difficulty epoch × calendar day.

Table 9: Proportional hazards test for the static Cox model.

	χ^2	df	p
Realized fee differential ($\tilde{\Delta}/S$)	0.121	1	0.728
Continuing block fee ($\tilde{F}^*_{h+1}/S$)	0.478	1	0.490
Miner hashrate share	1.542	1	0.214
Private transaction count	0.777	1	0.378
Block size	0.684	1	0.408
Global	3.518	5	0.621

Note: Scaled Schoenfeld residuals test (Grambsch and Therneau, 1994) for the static Cox model (Column (1) of Table 6). Each row reports the test statistic and p-value for the null hypothesis that the covariate's log-hazard coefficient is constant over spell time. The global row reports the omnibus test across all covariates.

Table 10: Proportional hazards test for the piecewise Cox model.

	χ^2	df	p
Continuing block fee ($\tilde{F}^*_{h+1}/S$)	0.818	1	0.366
Miner hashrate share	1.540	1	0.215
Private transaction count	0.825	1	0.364
Block size	0.836	1	0.361
Realized fee diff. $\times$ period (joint)	3.464	2	0.177
Global	7.112	6	0.311

Note: Scaled Schoenfeld residuals test (Grambsch and Therneau, 1994) for the piecewise Cox model (Column (2) of Table 6). The realized fee differential enters as an interaction with the early/late period indicator; its 2-df test jointly assesses whether the two period-specific hazard ratios are stable over spell time. No test rejects proportionality at conventional significance levels ($p > 0.1$ throughout).

Table 11: Functional-form robustness, piecewise Cox model.

Specification	Early-window HR ($t \leq 30$s)	Late-window HR ($t > 30$s)
Linear $\tilde{\Delta}/S$ (baseline)	0.546** (0.138)	1.060 (0.068)
Winsorized at 99th percentile	0.539 (0.210)	0.985 (0.079)
$\log(1 + \tilde{\Delta}/S)$	0.486** (0.153)	0.999 (0.079)
$\text{asinh}(\tilde{\Delta}/S)$	0.518** (0.142)	0.999 (0.071)

Note: Each row re-estimates the piecewise Cox model under a different transform of the realized fee differential. The baseline is the linear specification from Column (2) of Table 6. The winsorized specification caps $\tilde{\Delta}/S$ at its 99th sample percentile and so preserves the units of the baseline; the log and asinh transforms compress the right tail while preserving the sign, but rescale the regressor, so their hazard ratios are not directly comparable in magnitude to the linear baseline and corroborate the sign and significance of the effect. Hazard ratios with delta-method standard errors in parentheses; $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Table 12: Specificity of the early-window effect: period split applied to all covariates.

Covariate	HR (0–30s)	HR (30s+)	χ^2	p (early=late)
Realized fee differential	0.594	1.058	3.54	0.060
Continuing block fee	0.945	1.197	1.18	0.278
Miner hashrate share	1.109	1.038	0.45	0.503
Private transaction count	1.000	1.000	2.82	0.093
Block size	1.026	1.015	0.31	0.580

Note: Each covariate is interacted with the early/late period indicator (the 30-second split of Table 6), yielding a separate hazard ratio within the first 30 seconds of a spell (`0-30s`) and thereafter (`30s+`). The final columns report a Wald test of equality of the two period-specific coefficients (H_0: no early-versus-late difference), using difficulty-epoch-clustered standard errors.