The DeFi Dilemma*

Aron Bodisz Nikolaus Hautsch Stefan Voigt

Blockchain technology enables financial interactions without central counterparties by distributing pending transactions in an open network and offering compensation fees to validators in order to ensure a reliable record of transaction histories. We show that these essential building blocks of decentralized settlement make decentralized finance (DeFi) applications inherently inefficient: Front-running risk causes a wasteful arms race for transaction fees and renders arbitrage prohibitively costly. Exploiting a novel, granular dataset of prices and transactions on decentralized exchanges, we show that cross-exchange price differences are substantial and persistent, while transaction fees almost entirely consume arbitrage profits. Overcoming these inefficiencies is possible only by reinstalling central counterparties, thus undermining the fundamental principle of DeFi.

JEL Codes: G00, G10, G14

Keywords: Arbitrage, Blockchain, Market Frictions

*Bodisz is at the University of Vienna and Vienna Graduate School of Finance (VGSF). Hautsch is at the University of Vienna – Department of Statistics and Operations Research, Research Platform Data Science @ Uni Vienna, Vienna Graduate School of Finance (VGSF), and Center for Financial Studies (CFS). Voigt is at the University of Copenhagen – Department of Economics. Support from the Danish Finance Institute (DFI) is gratefully acknowledged. Corresponding author: stefan.voigt@econ.ku.dk. 0045 44 11 61 04. Øster Farimagsgade 5, 1353 København.

1. Introduction

Blockchain technology promises to render trusted intermediaries obsolete. Instead, a blockchain records transactions by an open network of validators without centralized control. To establish consensus about transaction histories without a third party requires collaboration among validators. A consensus protocol specifies *how* validators reach consensus and defines an incentive scheme to ensure that validators' incentive-compatible behavior aligns with a reliable record of transactions on the blockchain (Saleh, 2020; Biais et al., 2021).

The consensus protocol design can take many different forms, but two building blocks are fundamental to enabling blockchain-based settlement. First, unlike traditional stock markets, blockchain technology requires *pre-trade transparency*. More specifically, transactions waiting for validation are held in a memory pool (mempool), which is open for network participants to retrieve information on pending activity. With dispersed validators, achieving consensus naturally requires sufficient distribution of information for verification (Cong and He, 2019). However, to ensure incentives for collaboration and, thus, a reliable transaction history, *transaction fees* are necessary as the second major building block for a blockchain-based system. In a decentralized system, where pending transactions become public information, however, such fees are only paid if fast settlement is a scarce resource. Consequently, blockchain-based settlement possesses capacity constraints creating incentives to pay rewards to validators in order to exploit the rents of public transactions (Chiu and Koeppl, 2019; Easley et al., 2019; Hinzen et al., 2022). This causes a natural tension for fully decentralized (blockchain-based) settlement, making pre-trade transparency and transaction fees indispensable.

Blockchain technology is the backbone for the fast-growing field of Decentralized Finance (DeFi) applications to enable seamless, frictionless trading. DeFi largely rests on smart contracts, self-enforcing computer code based on terms contingent on the state of the blockchain. For example, a decentralized exchange (DEX) is a set of smart contracts that interact with each other to facilitate the trading of tokens created on the same blockchain (Harvey et al., 2021; Lehar and Parlour, 2024). Smart contracts render DEXs pure matchmakers and promise the exchange of assets without any exposure to counterparty risk, execution risk, and limitations to arbitrage capital (Gromb and Vayanos, 2010).

In this paper, we show that pre-trade transparency paired with transaction fees for validators imply substantial friction for *any* DeFi application. Counter the widespread belief that blockchain-based settlement has the potential to disrupt financial services, we show that there is a fundamental dilemma that causes market inefficiencies in the spirit of Grossman and Stiglitz (1980). In essence, the dilemma arises due to the fact that pending transactions reveal rents that other market participants could realize. This implies front-running risk and mitigates cross-DEX arbitrage. Consider a cross-DEX arbitrage trade. The cross-DEX arbitrageur aims to exploit a concurrent mismatch in prices. A smart contract-based transaction allows the joint submission of the resulting buy-side and sell-side transaction for validation to the mempool. The transaction is valid contingent on *both* transactions being executed simultaneously and fails otherwise. However, network participants can leverage the pre-trade transparency and can extract the profits from cross-DEX arbitrage transactions by copying the arbitrage transaction. Such strategy is only profitable, however, if participants can front-run the cross-DEX arbitrageur, i.e., getting validated first. This causes them to pay higher transaction fees in order to increase the incentive for validators to prioritize a transaction. Consequently, there is a wasteful arms race until transaction fees comprise the entire cross-DEX profit. In such a situation, anticipating front-running risk, the cross-DEX arbitrageur becomes indifferent between staying idle and arbitrage activity. As a result, the inner workings of blockchain-based settlement harm price informativeness by reducing the incentives to exploit smart contract-based cross-DEX arbitrage.

The empirical analysis in this paper utilizes novel, granular cross-DEX arbitrage data to shed light on the extent of this friction. The data yields block-level data on DEX prices, liquidity, and trading fees. We document three main findings: First, we report substantial and persistent price differences across DEXs. With the possibility of smart contract-based execution of cross-DEX arbitrage transactions, such an observation may seem, at first sight, puzzling. However, we characterize the optimal decision of a cross-DEX arbitrageur and identify substantial periods of time where, *in the absence of front-running risk*, potential arbitrage gains remain unexploited. This finding cannot be attributed to limited arbitrage capital: Given that buy and sell-side transactions occur within the same block, execution risk (i.e., the risk that just one leg of the transaction is executed) is eliminated and, as a result, such risk-free transactions can rely on virtually unconstrained arbitrage capital via so-called flash loans. Substantial and persistent cross-DEX price differences are thus attributed to front-running risk rendering arbitrage unprofitable.

Second, we provide evidence that network participants exploit the pre-trade transparency of cross-DEX arbitrage transactions. Based on our data, we document excess transaction fees for transactions with extractable rents. The publicly available information that cross-DEX transactions wait for verification in the mempool yields a wasteful arms race for validation capacities. In our sample, executed cross-DEX arbitrage transactions forego on average 64 percent of the arbitrage profits to validators, effectively yielding median transaction fees of 78 USD, 81 percent higher than the median transaction settled on the Ethereum blockchain.

Third, we quantify the magnitudes of price differences that would prevail if front-running risk did not exist. That is, we trace price differences in a counterfactual world where cross-DEX arbitrageurs exploit them whenever profitable without being faced with front-running risk. We find that 90.8 percent of the documented price differences can be directly attributed to front-running risk, rendering the frictions associated with blockchain-based settlement economically highly relevant.

Our results spotlight a fundamental dilemma for DeFi, which is clearly beyond limited price informativeness due to the absence of cross-DEX arbitrage via smart contracts: Transparency of transactions waiting for verification alone may not harm market efficiency if execution is guaranteed based on time priority. The need for transaction fees to compensate validators solely may neither harm market efficiency as long as arms races could be prevented, as pending transactions would not be publicly available. However, the idea of blockchain-based settlement relies on *both* pillars simultaneously: pre-trade transparency *and* transaction fees are necessary conditions to enable financial interactions without trusted intermediaries.

With publicly available transactions waiting for verification and the possibility to front-run such transactions, it is questionable how DeFi can sustain itself. While this paper focuses on impediments to cross-DEX arbitrage, a core component of market efficiency, Capponi and Jia (2021) and **?** identify similar issues for liquidity providers: Front-running risk renders liquidity provision costly such that liquidity may evaporate.

It should be noted, that our results do not indicate that arbitrage is impossible, though. Costly alternatives exist that seem preferable choices for cross-DEX arbitrageurs. We show that order splitting can be a costly alternative to smart contract-based cross-DEX trading. Hence, instead of relying on smart contract technology to submit buy- and sell-transactions, which are contingent on each other to switch off execution risks, transactions can be submitted separately. In fact, we identify a substantial number of such cases where cross-DEX arbitrageurs accept substantial execution risks in exchange for substantially reduced front-running risk. The second alternative is to focus on price differences between DEXs and centralized exchanges (CEX). A CEX facilitates trades by running a limit order book maintained internally and settling transactions off-chain. Here, front-running risk is less of a concern because smart contract-based execution against the internal order book of CEXes is impossible. However, CEXes act as largely unregulated custodians of their customers' funds, thus undermining the core principle of DeFi. Interacting with trusted intermediaries such as CEXes reincarnates counterparty risks, which can be substantial (Makarov and Schoar, 2020; Hautsch et al., 2024)

Hence, the core message of this paper is that DeFi faces a dilemma. A (seeming) solution is to overcome front-running risk by limiting transparency, e.g., by establishing dark pools instead of an open mempool. However, this requires a trusted *central* authority (intermediary), thus undermining the DeFi principle. Another (seeming) solution is to alter the consensus protocol to reduce the incentive for front-running either by introducing a closed, trusted network of validators (a private blockchain) or direct regulation via a central authority (Auer, 2019). Again, this undermines the DeFi principle since, in both cases, trusted central counterparties are required.

2. Pre-trade transparency and the risk of front-running

Blockchain transactions require validation. Every transaction has to be propagated to the (peer-to-peer) network, where transactions are gathered and wait for execution in the so-called memory pool. Transparency is a fundamental feature of blockchains, encompassing pre-trade transparency. On blockchain-based markets it manifests as the ability to easily inspect the content of all transactions that are waiting in the mempool to be picked for validation. Users have to attach a transaction fee to their transactions, which is often called a gas fee (that compensates the validator for the computationally demanding validation service). Ordering the transactions in the block is at the discretion of the validator. Although, validators can pick and order transactions in the block in any manner they prefer, they have an incentive to prioritize transactions based on the transaction fees. The intuitive reason behind this is that the block space is limited, and the validator prioritizes those who pay a higher transaction fee for a certain block space. Following this scheme, the validator maximizes her revenue from the block. Transaction fees effectively reflect the waiting costs of users, who can gain immediacy by increasing the transaction fee attached to their transaction (Huberman et al., 2021). The most widely used transaction fee mechanism requires blockchain users to submit a transaction fee above a certain threshold (base fee) that dynamically changes based on the utilization of the blockchain. This transaction fee mechanism is effectively an English-type auction combined with a minimum bid (Roughgarden, 2023).

The name transaction originally referred to a single transfer between two wallets on the blockchain (e.g., on the Bitcoin blockchain). Still, with the emergence of smart contracts, transactions became a more sophisticated tool through which users can interact with blockchain-based applications such as DEXs. Users can include multiple actions (e.g., taking a loan and exchanging tokens) in one transaction. Upon validation, every action is executed in the order that was pre-specified by the user, otherwise the transaction fails entirely. This feature of blockchain-based transactions is called atomicity. It enables users to execute a relatively sophisticated series of actions across platforms without worrying about other transactions modifying the state of the blockchain. A relevant example of such a transaction is a cross-DEX arbitrage transaction. A cross-DEX arbitrage transaction requires (at least) a buy and a sell order that is placed after each other. Figure 1 shows how cross-DEX arbitrage works. First, the cross-DEX arbitrageur searches for price differences across DEXs in the latest validated block. Conditionally on finding a profitable arbitrage opportunity, she submits an arbitrage transaction to the network with a transaction fee sufficiently high to guarantee execution. The transaction is placed in the mempool, where all transactions are waiting to be picked for validation. Finally, the validator of the next block validates the transaction, and the cross-DEX arbitrageur claims the arbitrage rent.

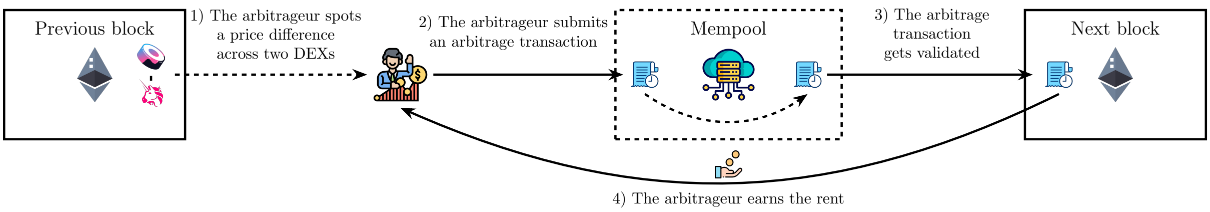


Figure 1: Arbitrage in the absence of front-running risk

As the previous example shows, arbitrage transactions can generate a private rent for the cross-DEX arbitrageur. However, by leveraging the pre-trade transparency, other participants can monitor the mempool and front-run the original arbitrage transaction by copying its content (the smart contract code) and submitting it with a higher transaction fee to the network (Qin et al., 2022, 2023). The higher transaction fee-paying transaction enjoys a priority over the original arbitrage transaction, thus it will be validated first, which renders the value of the original arbitrage transaction to zero. Figure 2 depicts such a scenario. After the arbitrage transaction is propagated to the network and moves into the mempool, another market participant infers its content and claims its rent by submitting the same transaction with a higher transaction fee. The original arbitrage transaction fails as it is not profitable anymore and the cross-DEX arbitrageur pays a reversion fee.

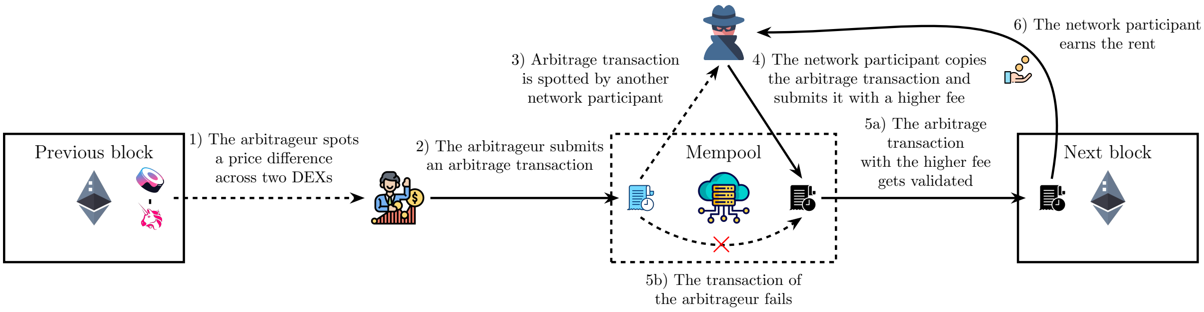


Figure 2: Arbitrage in the presence of front-running

Traditionally front-running refers to the practice when a market maker uses its proprietary information on incoming orders to place a trade before a large incoming trade.[1] However, in a DeFi context, transactions do not necessarily have to include a large trade to be subject to the risk of front-running. Due to the atomicity of transactions, rent-generating opportunities do not come solely in the form of large trades, but also in the form of smart-contract code that interacts with the financial applications on the blockchain. An arbitrageur, for instance, has to exert effort to find profitable arbitrage opportunities, optimally select the amounts of tokens to trade across pools and assemble these in the right order within a transaction. Albeit, the transaction is produced based on public information, the value of the transaction lies in the smart-contract code that generates a positive rent from it. As we noted earlier, cross-DEX arbitrage is a prominent example of this practice due to its importance in financial markets, but there exist other (and even more sophisticated) ways to extract rents utilizing public information. Therefore, in contrast to the 'traditional' case, where front-running refers to exploiting the information from the incoming order flow of single trades, in DeFi the definition is broader due to the atomicity of transactions and it also refers to exploiting information from the content (smart-contract code) of transactions.

[1]Malinova and Park (2017) consider an environment where agents send single trade transactions to each other via distributed ledger technology. In their model, an investor faces front-running risk when trading with a large trader.

3. The decision problem of the cross-DEX arbitrageur

In this section, we describe the decision problem of a cross-DEX arbitrageur who optimally executes an arbitrage across two DEXs. We derive our results under a hypothetical market environment where front-running is forbidden or impossible. Our intention is to derive the formulas used in our empirical analysis and by presenting comparative statics to enhance the understanding of our empirical results.

3.1 Setup

The cross-DEX arbitrageur follows the steps:

1. Monitoring of the (latest) state of the blockchain to exploit price differences across the DEXs.

2. Every time a new block is released, calculate the maximal achievable gross arbitrage profit that can be earned by trading away the price differences across the two DEXs.

3. Choosing a profit-maximizing amount of tokens to trade, given the pool liquidity and trading fees on DEXs.

4. Choosing a transaction fee based on the current validation demand (that can be inferred from the mempool), which guarantees the execution of the arbitrage transaction in the next block.

Figure 3 depicts the actions of the cross-DEX arbitrageur. The top part of the figure shows a block on the left that is already part of the blockchain and a block to the right that is about to be appended. The dashed line indicates the time that elapses between two consecutive blocks. The bottom part of the figure shows the mempool. The green area indicates that, in contrast to the blockchain state, the mempool state evolves continuously in time.

3.2 Description of the market

Consider a blockchain with two DEXs indexed by i and j. There are liquidity pools on both DEXs, where token X is traded against a numéraire token (denoted by Y). Each pool employs a constant product market maker (CPMM), where prices are determined by the amount of liquidity (k) available in the pool and the trading fee charged on trading

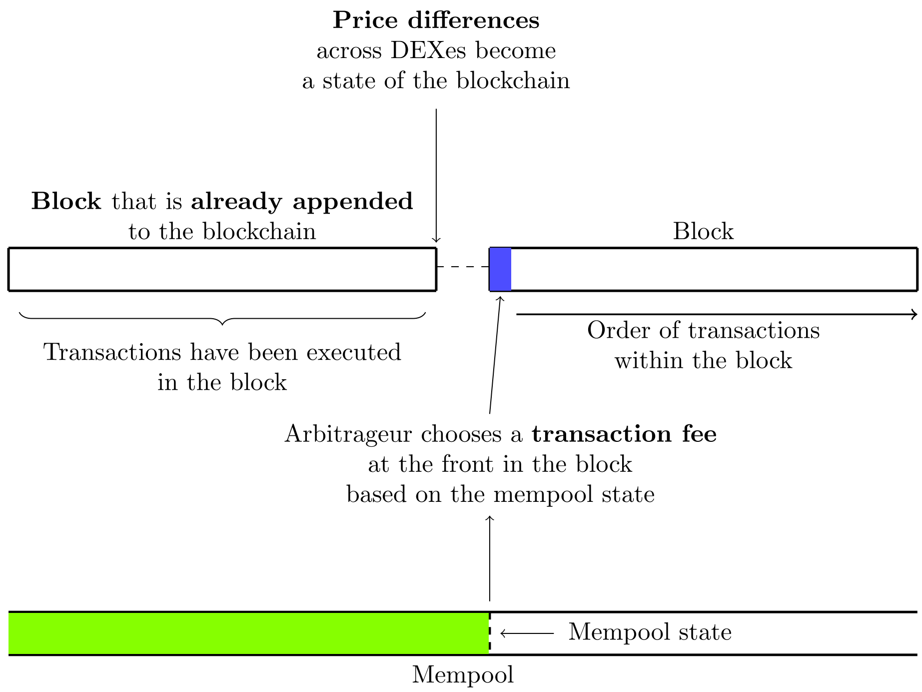


Figure 3: The decision problem of the cross-DEX arbitrageur

volume (τ). Such types of automated market makers are widely adopted across DEXs. The liquidity parameter (or invariant) is equal to the product of the actual reserves of token X and the numéraire token in the pool. Thus, the liquidity on DEX i equals:

$$\text{reserves of token } X \times \text{reserves of numéraire token } Y = x_i \times y_i = k_i.$$

The mid-price of token X relative to the numéraire on DEX i is determined by the ratio of the reserves and equals $p_i = x_i/y_i$. When a liquidity taker arrives at the pool on DEX i, and exchanges Δx_i tokens of X for Δy_i numéraire tokens, the liquidity in the pool must remain the same by the pricing rule used by the CPMM:

$$\underbrace{x_i y_i}_{\text{Before trading}} = \underbrace{\big(x_i + (1-\tau_i)\Delta x_i\big)\big(y_i - \Delta y_i\big)}_{\text{After trading}} = k_i.$$

The trading fee is charged on the incoming volume and accrues to the liquidity providers. The price impact of a trade can be directly computed from the formula above, with its magnitude depending on the trade size relative to the pool liquidity. Figure 4 gives an example of how the price and token reserves change in the pool when a liquidity taker exchanges USD for ETH.

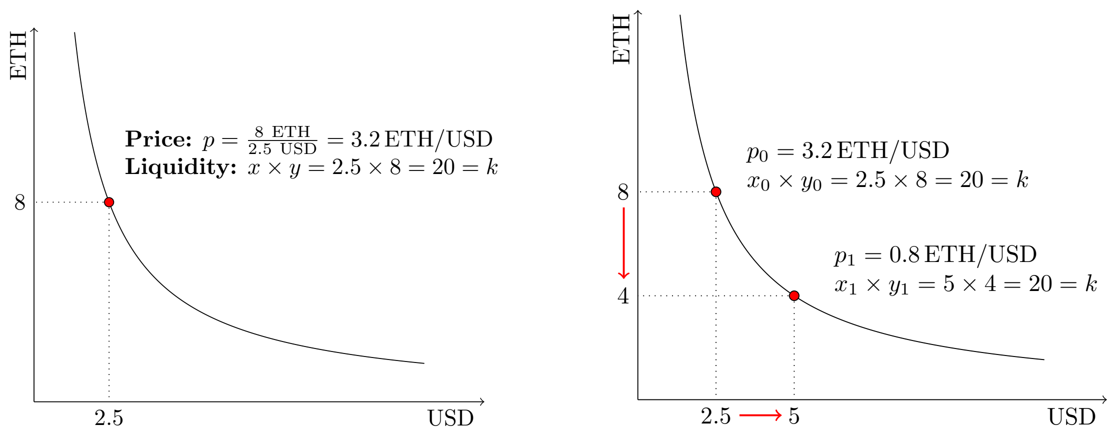


Figure 4: Trade example on a constant product market maker (CPMM)

3.3 Determining the maximal gross profit from arbitrage

Assume that the cross-DEX arbitrageur prefers to pocket the profit in the numéraire token.[2] For brevity, we refer to token X as 'token' and the numéraire token as 'numéraire'. Then, arbitrage requires two consecutive trades: (i) a buy trade to acquire tokens X for numéraire tokens in the pool where the price of the token relative to the numéraire is higher (i.e., the price of the numéraire relative to the token is lower) and (ii) a sell trade using the acquired tokens X for the numéraire tokens in the other pool, where the price of the token relative to the numéraire is lower (i.e., the price of the numéraire relative to the token is higher).

[2]Our empirical analysis and several other empirical papers (e.g., McLaughlin et al. (2023)) show that cross-DEX arbitrageurs on the Ethereum blockchain use the native token for arbitrage in a clear majority of all cases.

The token reserves in the pools are given by the tuples: (x_i, y_i) and (x_j, y_j) (which also determine k_i and k_j), and the pools charge trading fees τ_i and τ_j on the incoming trading volume. Furthermore, we assume that the mid-price of the token in the pool on DEX i is higher than in the pool on DEX j, i.e., $p_i = x_i/y_i > p_j = x_j/y_j$. In a first step, the cross-DEX arbitrageur buys Δx tokens for Δy_{init} numéraire in the pool on DEX i:

$$\Delta x = \frac{(1-\tau_i)x_i \Delta y_{init}}{y_i + (1-\tau_i)\Delta y_{init}} \tag{1}$$

Then, the cross-DEX arbitrageur exchanges the acquired tokens Δx for Δy_{final} numéraire in the pool on DEX j:

$$\Delta y_{final} = \frac{(1-\tau_j)y_j \Delta x}{x_j + (1-\tau_j)\Delta x} \tag{2}$$

The cross-DEX arbitrageur maximizes the difference between the final amount she receives (Δy_{final}) and the amount she initiates through the series of trades (Δy_{init}). By referring Δy_{init} to as the initial capital, the gross arbitrage profit equals the difference between the final and initial amount of numéraires:

$$\Pi = \Delta y_{final} - \Delta y_{init} \tag{3}$$

Δy_{final} can be expressed as a function of Δy_{init} by substituting (1) into (2). After replacing Δy_{final} in (3) with the former expression in the gross arbitrage profit function, it becomes the function of the initial capital Δy_{init}:

$$\Pi = \underbrace{\frac{(1-\tau_i)(1-\tau_j)x_j y_i \Delta y_{init}}{x_i y_j + (1-\tau_j)\big(x_i + (1-\tau_i)x_j\big)\Delta y_{init}}}_{\Delta y_{final}} - \Delta y_{init} \tag{4}$$

The cross-DEX arbitrageur chooses the optimal initial capital in the numéraire Δy^*_{init}, maximizing the gross arbitrage profit. The optimal Δy^*_{init} also determines the optimal amount of tokens to be purchased in the first pool Δx^* and the optimal final amount of numéraire Δy^*_{final}. For simplicity, assume that the trading fees are identical across pools, i.e., $\tau_1 = \tau_2 = \tau$. This is a realistic assumption, as on most DEXs, either there is only a pre-specified flat trading fee (e.g., Uniswap V2) or there is a limited number of trading fee tiers (e.g., Uniswap V3) that pools can charge. By plugging in Δy^*_{init} into the profit equation, the maximal gross arbitrage profit can be expressed as the function of (i) the liquidity in the pools and (ii) the trading fee:

$$\Pi^* = \frac{x_j y_i + (1-\tau)\Big((1-\tau)x_i y_j - 2\sqrt{x_i y_i x_j y_j}\Big)}{(1-\tau)\Big((1-\tau)x_i + x_j\Big)} \tag{5}$$

3.4 Comparative statics of the maximal gross arbitrage profit

In this subsection, we show how the change of the trading fee and the liquidity in the pools affects the maximal arbitrage gross profit. Throughout the calculations, we assume that an arbitrage opportunity exists across the pools $\Pi^* > 0$.

Trading fee. By differentiating (5) with respect to τ gives the expression for the marginal change in the maximal gross arbitrage profit:

$$\frac{\partial \Pi^*}{\partial \tau} = \frac{x_j^2 y_i + (1-\tau)x_i\Big(2(1-\tau)\sqrt{x_i y_i x_j y_j} - 2x_j y_i + (1-\tau)x_j y_j\Big)}{(1-\tau)^2\Big((1-\tau)x_i - x_j\Big)} \tag{6}$$

Figure 5 plots the gross profit as a function of Δy_{init} under different trading fees. The trading fee tiers shown are the most common ones used by DEXs. Not surprisingly, the maximal gross arbitrage profit increases as the trading fee decreases, and at the same time, the initial capital required for the arbitrage increases. Consistent with Lehar and Parlour (2024), the fixed spread that the cross-DEX arbitrageur has to pay increases with the increase of the trading fee.

Liquidity. As the liquidity grows in the pool, the depth of the market increases, and trades, on average, exert a lower price impact. The reserves of the tokens enter the profit equation separately, so a derivative with respect to $k = x \cdot y$ cannot be evaluated, but the marginal effects can be studied numerically. First, we consider the case when the liquidity in one pool increases by the multiples of 10, i.e., $10^m \cdot k_{init}$, $m \in \{0, 1, 2\}$, while keeping the liquidity in the other pool fixed. Second, we keep the size of the larger pool fixed and increase the size of the other pool to a comparable level. Panel (a) and (b) in Figure 6 plot these two cases, respectively. Our numerical exercise shows that for a fixed ratio of pool reserves, (i) the arbitrage profit increases in liquidity, and (ii) the size of the smaller pool limits the maximal gross arbitrage profit. Intuitively, as one pool becomes larger, the price impact of the buy trade reduces, thus the cross-DEX arbitrageur purchases more

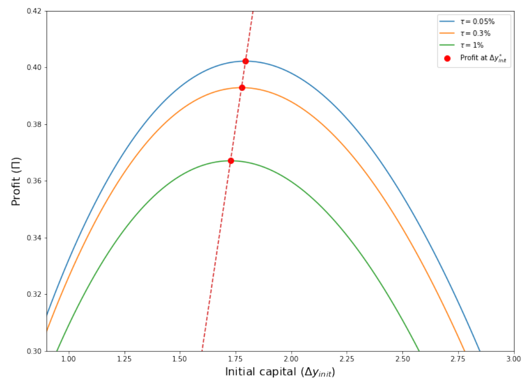
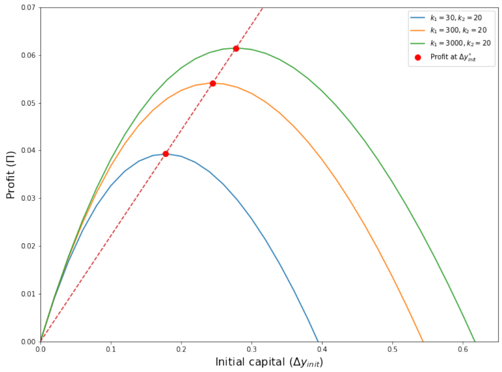


Figure 5: Comparative statics: trading fee. The figure shows the values of the gross arbitrage profit as the function of the initial capital (Δy_{init}) under different trading fees (τ) charged by liquidity pools. The liquidity parameters used for the calculations are $(150, 20)$ and $(100, 20)$.

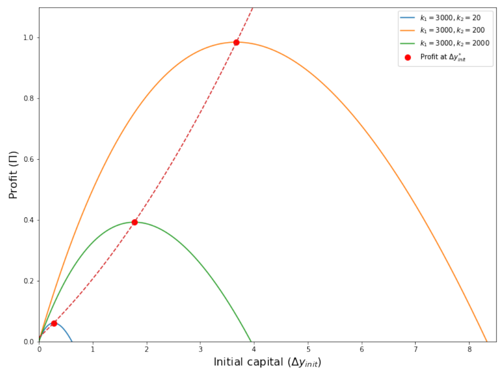
tokens. However, as a countervailing force, the price impact of the sell trade increases in the smaller pool, where the liquidity is kept constant. Overall, the maximal achievable profit increases but is limited due to these two counteracting effects. The dashed line on Panel (a) indicates that whenever one of the pools is small, the maximal arbitrage profit grows linearly with the price impact in the smaller pool. In Panel (b), the effect of the price impact reduces as the small pool also grows in size and we observe an exponential growth of the maximal achievable arbitrage profit.

3.5 Determining the optimal hypothetical transaction fee and the maximal net profit from arbitrage

Transactions including trades, liquidity additions, or liquidity withdrawals, modify the state of the pool. Thus, to avoid price slippage, the cross-DEX arbitrageur has to place its transaction in front of the queue in the next block. Denote by q_a the queue position in the upcoming block where the cross-DEX arbitrageur aims to place the transaction. We assume that no transactions modifying the state of the pool in front of the targeted position in the block occur. By choosing a transaction fee that exceeds the transaction fee of the transaction in the queue at position q_m in the mempool (right before the block is added), the cross-DEX arbitrageur guarantees that the transaction will be validated



(a) Two initial small pools



(b) One initial large and one initial small pool

Figure 6: Comparative statics: liquidity. The figure shows the values of the gross arbitrage profit as the function of the initial capital (Δy_{init}) under different pool liquidity values. On panel a) one pool remains relatively small in terms of liquidity compared to the other. On panel b) one pool has an initial large size (the large pool from Panel (a)) and the other smaller pool grows to a comparable size.

before the transaction at position q_m.[3] On the other hand, the cross-DEX arbitrageur wants to maximize her net profit, so she chooses the lowest transaction fee that places her in front of her intended queue position.[4] The cross-DEX arbitrageur thus chooses a transaction fee f maximizing the net profit, which is simply the difference between the maximal gross arbitrage profit and the transaction fee:

$$\max_f \pi = \max_f (\Pi^* - f) \text{ s.t. } p_{\text{execution}}(f | q_a > q_m) = 1, \tag{7}$$

where $p_{\text{execution}}$ denotes the execution probability of the transaction given the distribution of transaction fees in the mempool. The probability on the right-hand side is one, meaning that the arbitrage transaction gets validated with certainty in the next block in front of the 'target' position. Then, the optimal hypothetical transaction fee equals the transaction fee that is attached to the transaction at the position $q_a - 1$ (or, in case there are multiple m transactions waiting at the same transaction fee level, $\{q_a - 1, \ldots, q_a - m\}$), i.e.,

[3]Transaction fees comprise of gas prices multiplied by the gas used. Gas in Ethereum is the unit for measuring the computational effort required to execute a transaction. More complex transactions consume more gas. For simplicity, we refer to transaction fees throughout. We control for differences in gas usage in our empirical results.

[4]Our approach is mechanical because we assume that the cross-DEX arbitrageur a priori aims to place the transaction before a certain position. In real practice, however, she might try to predict the target position based on recent data.

$f^* = f_{q_a - 1}$.

3.6 Other costs

Smart contract based arbitrage alleviates the costs associated with arbitrage on CEXs. There are three essential aspects. First, flash loans allow anyone to borrow and repay tokens within one atomic transaction. Cross-DEX arbitrageurs can borrow even millions of dollars, thus cross-DEX arbitrage does not require (substantial) initial capital. To justify the assumption that interest rates on flash loans, r_{fl}, are approximately zero, we analyzed interest rates charged on every flash loan issued by the lending platforms frequently used by arbitrageurs, namely Aave, dydx, and Bancor, until March 2024. Out of $750,000$ flash loans, $200,000$ borrowers paid an interest of 0.02%, $300,000$ paid 0.05%, and $250,000$ paid 0.09%. Given this evidence, the assumption of zero interest rates is widely realistic. Otherwise, our calculations above need to be modified, with Δy_{init} in the gross arbitrage profit function replaced by $(1 + r_{fl})\Delta y_{init}$, where r_{fl} is the interest paid on the flash loan.

Second, the atomicity of transactions guarantees that trades are executed either in the pre-specified order or fail completely, implying that the cross-DEX arbitrageur does not face any costs associated with inventory risk. Third, by utilizing smart contract code, the cross-DEX arbitrageur can impose a condition in the arbitrage transaction that reverts the transaction if it generates a negative profit, eliminating execution risk.

3.7 The optimal strategy of the cross-DEX arbitrageur under front-running risk

Up until now, we assumed that front-running is forbidden or impossible on the blockchain. We relax this assumption, which means that the arbitrageur faces competition. This alters the arbitrageur's optimal choice for the transaction fee because she has to take into account the actions of other market participants.

After a block is added to the blockchain, the cross-DEX arbitrageur faces a short time period (on the Ethereum blockchain, around 13 seconds) to submit or even re-submit the same arbitrage transaction with a different transaction fee to the mempool. Upon submission of the transaction, its content becomes public information for other market participants, who may try to front-run the transaction by copying and submitting it with a higher transaction fee. Such bidding wars among cross-DEX arbitrageurs are discussed and empirically documented in the seminal paper by Daian et al. (2020).

What is the optimal transaction fee bidding strategy of the cross-DEX arbitrageur in such a market setup and information structure? Easley and Tenorio (2004) and Daniel and Hirshleifer (2018) analyze sequential bidding games with agents with distinct private values of an item. Cross-DEX arbitrageurs in our market setup know the profit the opportunity generates, so they value it equally. Under zero bidding costs $c = 0$, all market participants follow the "ratchet" strategy, i.e., they sequentially bid up the transaction fee to the value of the arbitrage opportunity. The auction winner earns zero profit, and the transactions of the other market participants revert.[5] Hence, under front-running risk the validator earns the arbitrage profit through transaction fees.

Suppose the cross-DEX arbitrageur deviates from the optimal strategy and – by hoping to earn a positive profit – submits the transaction with a transaction fee less than the arbitrage profit without updating it. Then, it is optimal for another market participant to front-run the cross-DEX arbitrageur. As a result, the cross-DEX arbitrageur's transaction reverts and yields a negative profit $-r$. Hence, the arbitrageur's expected profit is negative, so it is not optimal for cross-DEX arbitrageurs to submit arbitrage transactions under front-running risk.

Note that an important ingredient of the argumentation above relies on the assumption that the reversion fee is positive, i.e., $r > 0$. This assumption is easily justified since it is impossible to design a (secure) blockchain with reversion fees being zero. In fact, due to security reasons a positive reversion fee must be charged on reverting transactions, otherwise, a hacker could perform a denial-of-service (DoS) attack, which would disrupt the functioning of the blockchain. In case of a zero reversion fee, a hacker could submit a transaction with a high transaction fee that consumes all the block space, which is reverted before execution. This would prevent other transactions from being included in the block, rendering it impractical for use.

[5] Under non-zero bidding costs $c > 0$, the cross-DEX arbitrageur submits the transaction with a transaction fee $\Pi^* - c$. With such a high initial bid, the cross-DEX arbitrageur avoids paying the bidding costs once more and, at the same time, deters other market participants from front-running the transaction as submitting the transaction with a higher transaction fee would result in a negative profit. However, the cross-DEX arbitrageur still makes a zero profit.

4. Data

4.1 Data sources

We utilize several data sources to conduct our empirical analysis. Below we briefly introduce the datasets and corresponding variables used in the analysis.

Ethereum block and transaction data. We acquired block and transaction level data via the Ethereum-etl tool. This dataset covers a large range of variables, where from the block-level data we utilize the block number and the time when the block was added to the blockchain. From the transaction level data, we obtain the hashes of every transaction, the gas transactions consumed, and the gas price transactions paid.

DEX data. Moreover, we utilize trading and pool reserves data using Dune Analytics' API.On Dune, the decoded blockchain data is available in a tabular format. We have extracted Sync events, which is a function automatically called whenever a trade (swap), a liquidity addition (mint), or liquidity withdrawal (burn) changes the token reserves in the pool and posts the new reserves. Furthermore, we have data on Swap events that include how much token X has been exchanged for token Y for every trade.

Arbitrage data. For our analysis, we use the Flashbots MEV dataset, which contains, among others, the transaction hashes of identified arbitrage transactions. It is important to note that, unlike on continuous-time markets, arbitrages across DEXs can be identified with a higher precision as every transaction of a blockchain account can be inspected. It means that when a cross-DEX arbitrageur submits a transaction, the money flow can be precisely tracked as it moves through the two (or more) DEXs and finally returns to the cross-DEX arbitrageur's account. From this dataset, we use the hashes of identified arbitrage transactions, the gross arbitrage profit, the net arbitrage profit, and the amount paid to the validator.

Mempool data. We use the mempool data from Jochen Hoenicke's website that reports the number of pending transactions between certain gas price levels. The transactions are categorized into 31 unevenly spaced bins based on gas prices, with updates occurring roughly every 3 minutes. Data at this frequency allows us to capture aggregate demand movements.

Other. To express the value of the tokens in USD, we gathered minute-level data from Coinpaprika. This allows us to convert any token value to USD at the actual price. In our filtering process, we also utilize the transaction hashes of non-arbitrage transactions from the Flashbots MEV dataset. These transactions, individually (or in combination) generate rents and are commonly classified as MEV (Maximal Extractable Value) transactions. Furthermore, to identify blockchain-based trading algorithms, commonly referred to as MEV bots, we collected addresses labeled as 'MEV Bot' from Etherscan. We use an additional Flashbots dataset containing all the dark pool blocks and transaction information. Although the Flashbots dark pool was the most popular option among users, to be able to paint a full picture in our analysis we collected transaction-level data from another significant dark pool service provider, Eden Network. Combined, these two providers accounted for more than 90% of the dark pool transactions.

4.2 Descriptive statistics

Our sample covers the period from December 7, 2020, to August, 31, 2022. We consider arbitrage across the pools of 3 DEXs that use CPMMs and account for $80 - 85\%$ of the trading volume on the Ethereum blockchain at the beginning of the sample period. These DEXs are Uniswap V2, Sushiswap and Shibaswap. Shibaswap was created after the starting date of our sample period and the liquidity was very low in its pools after the deployment, therefore we exclude the observations of the first week from the Shibaswap sample. Furthermore, we focus on arbitrage across the economically most significant pools that trade the token pairs WETH-USDC, WETH-USDT, WETH-DAI, and WETH-WBTC. USDC (USD Coin), USDT (Tether), and DAI are stablecoins, which are pegged to the USD, while WETH (Wrapped Ethereum) and WBTC (Wrapped Bitcoin) are non-stable ones.[6] The trading fees in each pool are identical and equal to 0.3%. Table 1 reports the overall trading volume and shares of trading volume pools on the DEXs within the sample period.

To identify two-legged arbitrage transactions, we merge the hashes of the arbitrage transactions from the Flashbots MEV dataset with the trading data and select transactions that contain only two trades across the pool pairs. Note that several large-scale empirical arbitrage studies show that nearly half of the arbitrage transactions include only two legs (e.g., Qin et al. (2022), McLaughlin et al. (2023)). This ratio increases even

[6] 'Wrapping' simply enables these tokens to be traded across DEXs, and they are worth the same as ETH (Ethereum) and BTC (Bitcoin), respectively.

more when within-DEX arbitrages (arbitrage across the pools of one DEX) are excluded. Therefore, we restrict our analysis to two-legged arbitrage disregarding multi-legged arbitrage transactions.

DEX	WETH-USDT pool	WETH-WBTC pool	WETH-USDC pool	WETH-DAI pool	Overall trading volume (billion USD)
Uniswap v2	9.12%	2.09%	9.96%	3.71%	425.10
Sushiswap	9.37%	6.23%	15.94%	8.47%	189.14
Shibaswap	7.08%	2.34%	6.41%	3.35%	13.85

Table 1: Trading volumes captured by the sample pools. The table reports the overall trading volume captured by the DEXs during our sample period. The percentages show the share of this volume attributed to each respective pool within the DEXs.

Table 2 and Table 3 report the daily aggregate statistics on trading and pool sizes. Trading activity is the highest on Uniswap V2, where the number of trades is around 3 million for the WETH-USDT and WETH-USDC pools. The average daily trading volumes are similar and the largest in these pools, which, compared to the respective average pool sizes, imply a 25% daily turnover. The median and average pool sizes of Uniswap V2 and Sushiwap are of comparable magnitudes, while Shibaswap pools are significantly smaller.

5. Measuring price differences and transaction fees

5.1 Measuring price differences

As discussed in Section 3, we measure price differences and calculate arbitrage profits after every block that modifies the state of the pools. By picking the last state of the pool (pool reserves) from each block we calculate the maximal gross arbitrage profits (Π^*), the optimal initial capital (Δy^*_{init}) needed for the arbitrage transaction, the optimal token purchase (Δx^*) and the optimal final amount received (Δy^*_{final}). Measuring price differences solely based on mid-prices across pools is misleading as price impacts and trading fees would be neglected, meaning that mid-price differences might prevail across pools because they are not profitable to be arbitraged away. Therefore, to calculate the percent of the mid-price difference across pools that could have been arbitraged away we derive a formula that is the function of the maximal gross (and net) arbitrage profit, which value has already been corrected for trading fees and the price impact (and the transaction fee). Accordingly, the problem can be reformulated as follows. After

Pool	Median daily trade size (million USD)	Average daily trade size (million USD)	Std. of daily trade size (million USD)	Minimum daily trade size (million USD)	Maximum daily trade size (million USD)	Number of trades (thousand)
Uniswap v2 WETH-USDT	31.373	51.06	60.004	2.025	658.747	3166.107
Uniswap v2 WETH-WBTC	2.871	11.241	19.233	0.037	177.47	284.368
Uniswap v2 WETH-USDC	35.669	57.638	63.187	3.853	736.383	2948.681
Uniswap v2 WETH-DAI	6.817	19.897	34.766	0.432	351.062	833.628
Sushiswap WETH-USDT	20.0	26.951	34.1	0.607	428.507	643.121
Sushiswap WETH-WBTC	8.046	18.342	30.186	0.106	383.463	167.023
Sushiswap WETH-USDC	33.8	46.46	65.969	0.546	967.448	874.604
Sushiswap WETH-DAI	14.723	24.625	39.726	0.199	575.296	425.414
Shibaswap WETH-USDT	1.249	2.323	3.556	0.056	34.261	90.212
Shibaswap WETH-WBTC	0.357	0.771	1.208	0.001	9.367	11.405
Shibaswap WETH-USDC	0.971	2.104	3.011	0.014	20.059	77.911
Shibaswap WETH-DAI	0.697	1.098	1.322	0.015	11.02	32.222

Table 2: Summary statistics on trading activity

the cross-DEX arbitrageur calculates the optimal amounts of tokens to trade with, she effectively buys Δx^* tokens at the effective price p_e for Δy^*_{init} numéraire. Then she sells Δx^* tokens at the effective price $1/(p_e - \Delta p_e)$ to acquire Δy^*_{final} numéraire. Formally, the two arbitrage trades are:

$$\Delta x^* = p_e \Delta y^*_{init}, \tag{8}$$

$$\Delta y^*_{final} = \frac{1}{p_e - \Delta p_e} \Delta x^*. \tag{9}$$

Using equations (8) and (9), the effective price difference can be expressed as

$$\Delta p_e = p_e - \frac{\Delta x^*}{\Delta y^*_{final}} = \frac{\Delta x^*}{\Delta y^*_{init}} - \frac{\Delta x^*}{\Delta y^*_{final}} = \Delta x^* \Big(\frac{1}{\Delta y^*_{init}} - \frac{1}{\Delta y^*_{final}} \Big). \tag{10}$$

Furthermore, noting that the maximal gross arbitrage profit equals $\Pi^* = \Delta y^*_{final} - \Delta y^*_{init}$, the effective price difference can be expressed as a function of the maximal gross arbitrage

Pool	Median pool size (million USD)	Average pool size (million USD)	Std. of pool size (million USD)	Maximum pool size (million USD)
Uniswap v2 WETH-USDT	207.681	206.815	73.047	390.569
Uniswap v2 WETH-WBTC	223.007	232.596	106.586	449.685
Uniswap v2 WETH-USDC	237.174	237.598	75.282	462.069
Uniswap v2 WETH-DAI	126.194	121.456	55.870	232.383
Sushiswap WETH-USDT	199.391	185.146	99.499	416.670
Sushiswap WETH-WBTC	534.111	590.211	310.034	1475.027
Sushiswap WETH-USDC	313.917	285.570	131.831	516.577
Sushiswap WETH-DAI	209.194	199.305	102.155	461.507
Shibaswap WETH-USDT	20.900	36.099	57.934	260.730
Shibaswap WETH-WBTC	69.101	103.790	121.606	455.805
Shibaswap WETH-USDC	24.205	50.215	73.629	277.141
Shibaswap WETH-DAI	17.607	23.033	21.145	107.938

Table 3: Summary statistics on pool sizes

profit,

$$\Delta p_e = \Delta x^* \Big(\frac{\Pi^*}{\Delta y^*_{init} \Delta y^*_{final}} \Big). \tag{11}$$

These price differences, however, do not account for the transaction fee that the cross-DEX arbitrageur has to pay to guarantee the execution of the transaction. Using the definition of the maximal net arbitrage profit from equation (7), the transaction fee-corrected effective price difference is given by

$$\Delta p_{e,net} = \Delta x^* \Big(\frac{\Pi^* - f^*}{\Delta y^*_{init} \Delta y^*_{final}} \Big) = \Delta x^* \Big(\frac{\pi^*}{\Delta y^*_{init} \Delta y^*_{final}} \Big). \tag{12}$$

The effective price differences in equations (11) and (12) are expressed in terms of tokens per numéraire.

To ensure comparability of the price differences across liquidity pools, we calculate the percentage of the mid-price difference that could have been arbitraged away by dividing the effective price difference by the mid-price difference,

$$\Delta p_{e,pct} = \frac{p_e - (p_e - \Delta p_e)}{\max\{p_i, p_j\} - \min\{p_i, p_j\}} = \frac{\Delta p_e}{\max\{p_i, p_j\} - \min\{p_i, p_j\}} \in [0, 1] \qquad (13)$$

Note that the effective prices at which the arbitrageur buys and sells are: $p_e - \Delta p_e \geq \min\{p_i, p_j\}$ and $p_e \leq \max\{p_i, p_j\}$, which means that price improvement is bounded between zero and one. Furthermore, notice that a mid-price difference can be only completely arbitraged away in case the trading fees are zero ($\tau = 0$) and the pools have infinite liquidity ($k_i = k_j = \infty$), implying a zero fixed spread and no price impact for any trade.

5.2 Measuring transaction fees

As discussed in Section 3, the optimal transaction fee is the lowest transaction fee that guarantees a queue position in the next block such that the cross-DEX arbitrageur's transaction is executed before the states of the pools change. Empirically, we evaluate three transaction fee levels corresponding to 'target' positions to secure the arbitrage transaction's placement in the next block: (1) $q_a = 25$ (before the 25th position), (2) $q_a = 10$ (before the 10th position), and (3) $q_a = 1$ the first position.

On average, a block in our sample contains 190 transactions. To assess the robustness of our transaction fee level choices, we compute how frequently transactions modify the pool states before the queue positions 10 and 25. We find that within our sample period, 10% and 15% of the time pool states are modified within those queue positions. Furthermore, we check how many of those transactions generate rents. For this purpose, we use the identified MEV transaction hashes from the Flashbots MEV summary dataset. We find that around 25% of these transactions are indeed MEV transactions that pay a high transaction fee due to front-running.

The transaction fee that has to be paid by the cross-DEX arbitrageur is determined by the gas (the computational complexity of the transaction) times the gas price that is paid after each unit of gas. The gas required for an arbitrage transaction is stable over time, given that a two-legged arbitrage always entails the computational cost of two trades and a flash loan. Based on historical arbitrage data, we find that the average gas cost of such a transaction is slightly less than 250, 000. Gas prices fluctuate with validation demand and are selected based on the 'target' positions. Thus, transaction fees are calculated as:

$$f^* = \text{Gas}_{arb} \times \text{GasPrice}^* = \text{Gas}_{arb} \times \text{GasPrice}_{q_a-1} = 250,000 \times \text{GasPrice}_{q_a-1},$$

where $GasPrice^*$ denotes the optimal gas price. Note that because the gas required for arbitrage is the same over time, the optimal transaction fee effectively requires choosing an optimal gas price. Figure 7 displays the one-day moving average of the USD value of the transaction fees at the three levels.

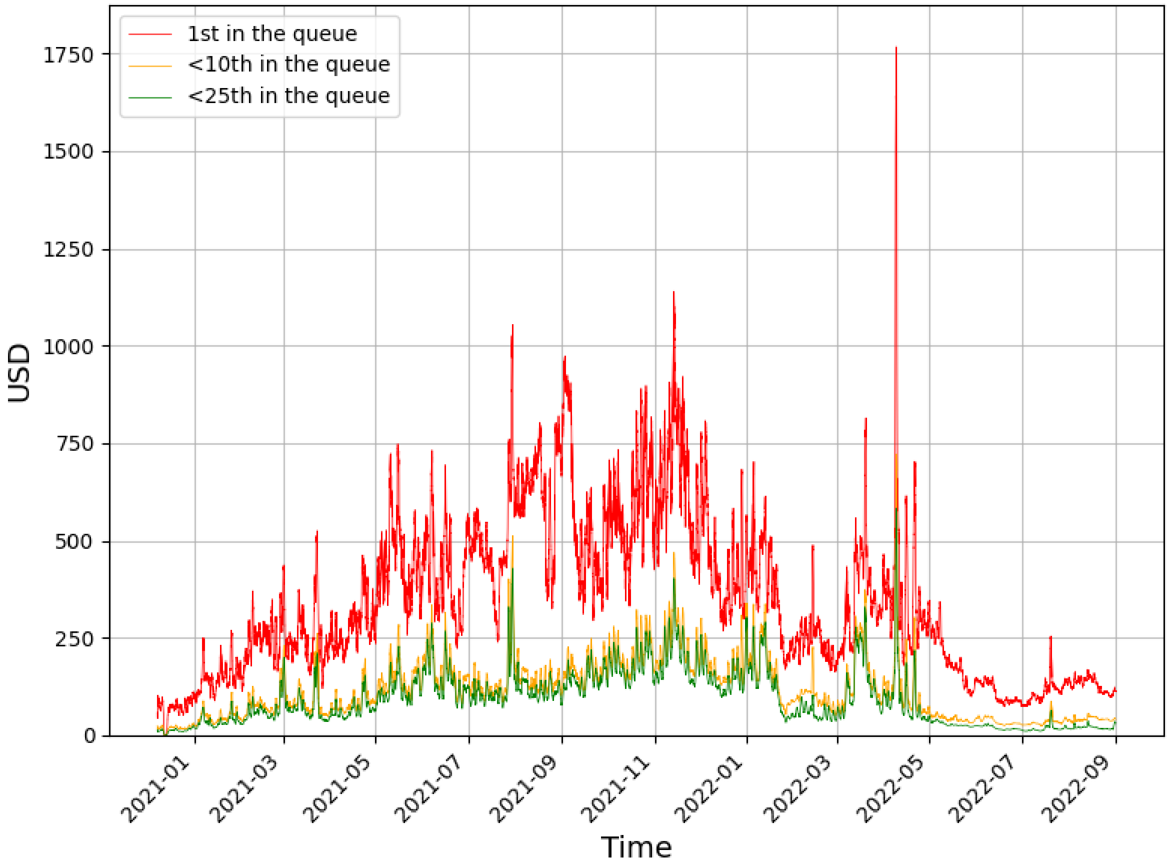


Figure 7: Transaction fees required for a two-legged arbitrage (one-day moving average). The figure presents the one-day moving average of optimal transaction fees of a two-legged arbitrage transaction, that guarantees a place in the queue at positions: $q_a = 25$ (before the 25th), $q_a = 10$ (before the 10th) and $q_a = 1$ (the first position).

6. Empirical results

6.1 Effective price differences and price improvements

Table 4 reports the averages of the price improvements and the arbitrageur's share (from which she earns a net profit) from the price improvements across the pool pairs. The price improvements are calculated based on (13), which is the ratio of the effective price difference that could have been closed by an arbitrage transaction and the mid-price difference. Whenever there is a positive effective price difference across pools, there is an arbitrage opportunity, so the hypothetical price improvement is also positive. In the

Pool pair	Average price improvement	Average arbitrageur's share from price improvement (transaction placed < 25th in the queue)	Average arbitrageur's share from price improvement (transaction placed < 10th in the queue)	Average arbitrageur's share from price improvement (transaction placed 1st in the queue)	Share of blocks with positive effective price differences
Uniswap v2-Sushiswap WETH-USDT	7.77	2.37	2.12	1.43	0.34
Uniswap v2-Shibaswap WETH-USDT	10.09	0.26	0.18	0.07	2.58
Sushiswap-Shibaswap WETH-USDT	10.98	0.42	0.30	0.14	3.61
Uniswap v2-Sushiswap WETH-WBTC	7.88	2.60	2.30	1.57	0.39
Uniswap v2-Shibaswap WETH-WBTC	9.87	0.73	0.59	0.24	1.58
Sushiswap-Shibaswap WETH-WBTC	10.44	0.98	0.77	0.37	1.60
Uniswap v2-Sushiswap WETH-USDC	8.28	3.45	3.15	2.24	0.24
Uniswap v2-Shibaswap WETH-USDC	10.48	0.24	0.17	0.07	2.42
Sushiswap-Shibaswap WETH-USDC	10.94	0.34	0.23	0.09	3.14
Uniswap v2-Sushiswap WETH-DAI	9.35	2.49	2.20	1.36	0.71
Uniswap v2-Shibaswap WETH-DAI	10.40	0.63	0.47	0.19	3.25
Sushiswap-Shibaswap WETH-DAI	9.92	0.72	0.59	0.26	3.81

Table 4: Average price improvements and the arbitrageur's share from the price improvement (measured in percentage points). *Average price improvement* column reports the average of the mid-price differences closed by arbitrage transactions across pool pairs in percentage points. *Average arbitrageur's share from price improvement* columns report the share of the price improvement that generated a positive net profit for the arbitrageur (while the rest accrued to the validator in the form of transaction fees). The parentheses under the column names contain the fee levels corresponding to the 'target' positions. The price improvements were calculated according to equation (13). *Share of blocks with positive effective price differences* presents the percentage of blocks that contain a positive price difference.

absence of front-running and without controlling for hypothetical transaction fees, the mid-prices across pool pairs could have been moved closer to each other by 7.77% to

10.98% (second column). These price improvements also translate into gross arbitrage profits. However, the arbitrageur must pay a transaction fee that varies based on her desired queue position in the next block. In columns three to six, we report the magnitudes from the average price improvements that generate a profit for the arbitrageur. These numbers show that in the absence of front-running risk, even after controlling for relatively high (or even extreme) transaction fees, there are still hypothetical arbitrage opportunities that can generate profits for the arbitrageur.

The rightmost column shows the share of blocks containing positive effective price differences, where there is room for a price improvement across pools. The share is calculated relative to the number of blocks including at least one transaction that caused a change of any of the pool states. Positive effective price differences arise more frequently when one of the pairs includes a Shibaswap pool. In low-liquidity Shibaswap pools, trades create more pronounced shifts in token reserves, that cause more price discrepancies across pairs of pools. However, we observe low average arbitrageur shares from the price improvements after accounting for transaction fees for the same pool pairs. As discussed in Section 3, the liquidity of the smaller pool limits the maximal gross arbitrage profit attainable across pools. Consequently, when transaction fees are taken into account, they consume a significant portion of the profit.

6.2 Arbitrage profits

We compute the hypothetical profits that could have been realized by performing arbitrage to eliminate the price differences. Table 5 reports the actual and hypothetical cumulative net profits in millions of USD for each pool pair over our sample period. The second and third columns contain actual cumulative arbitrage net profits and cumulative payments to validators, which combined are equal to the gross profit from arbitrage. Columns four to six include the hypothetical net profit from arbitrage at various transaction fee levels. The parentheses in columns four to six indicate the percentage of the actual net arbitrage profit foregone at various transaction fee levels.

We find that the majority of arbitrage opportunities remain unexploited due to front-running risk, with 85% to 99% of the hypothetical net arbitrage profit not being realized. We get these numbers by dividing the actual cumulative net profits by the hypothetical cumulative net profits. There are two mechanisms in place: (1) due to front-running risk arbitrageurs are not submitting transactions, but (2) even when they submit, due to front-running they bid up the transaction fee and pay nearly the whole rent to the

Pool pair	Actual arbitrage: Cumulative net profit from actual arbitrage transactions	Actual arbitrage: Cumulative payments to validators	Hypothetical arbitrage: Cumulative net profit from hypothetical arbitrage transactions (transaction placed < 25th in the queue)	Hypothetical arbitrage: Cumulative net profit from hypothetical arbitrage transactions (transaction placed < 10th in the queue)	Hypothetical arbitrage: Cumulative net profit from hypothetical arbitrage transactions (transaction placed 1st in the queue)
Uniswap v2-Sushiswap WETH-USDT	0.072	0.112	1.262 (94%)	1.246 (94%)	1.160 (94%)
Uniswap v2-Shibaswap WETH-USDT	0.005	0.004	0.082 (94%)	0.077 (93%)	0.062 (92%)
Sushiswap-Shibaswap WETH-USDT	0.007	0.006	0.136 (95%)	0.130 (95%)	0.112 (94%)
Uniswap v2-Sushiswap WETH-WBTC	0.002	0.006	0.701 (99%)	0.697 (99%)	0.682 (99%)
Uniswap v2-Shibaswap WETH-WBTC	0.001	<0.001	0.042 (98%)	0.041 (98%)	0.037 (98%)
Sushiswap-Shibaswap WETH-WBTC	<0.001	<0.001	0.011 (99%)	0.011 (99%)	0.007 (98%)
Uniswap v2-Sushiswap WETH-USDC	0.041	0.110	1.343 (97%)	1.328 (97%)	1.239 (97%)
Uniswap v2-Shibaswap WETH-USDC	0.001	0.003	0.107 (99%)	0.103 (99%)	0.085 (99%)
Uniswap v2-Shibaswap WETH-USDC	0.004	0.003	0.113 (97%)	0.108 (96%)	0.093 (96%)
Uniswap v2-Sushiswap WETH-DAI	0.059	0.111	0.723 (92%)	0.710 (92%)	0.638 (91%)
Uniswap v2-Shibaswap WETH-DAI	0.004	0.002	0.067 (95%)	0.064 (94%)	0.053 (93%)
Sushiswap-Shibaswap WETH-DAI	0.008	0.003	0.078 (90%)	0.074 (89%)	0.055 (85%)

Table 5: Hypothetical and actual arbitrage profits (in million USD). *Cumulative net profit from actual arbitrage transactions* reports the net profit earned from actual arbitrage in USD. *Cumulative payments to validators* presents the USD value of the payments received by validators either in the form of fees or direct payments. *Cumulative net profit from hypothetical arbitrage transactions* columns report the net profit, in USD, that arbitrageurs could have hypothetically earned at the three transaction fee levels in a front-running risk-free market. The parentheses in columns four to six indicate the percentage of the actual net arbitrage profit foregone at various transaction fee levels.

validator in the form of fees. On average, validators capture approximately 64% of the profit from arbitrage, but to provide a more detailed picture of the shares claimed by validators, we plot the distribution of these shares in Figure 8. The concentration of validators' shares on the right side of the distribution indicates that validators secure over 90% of the profit in many instances.

6.3 Counterfactual analysis

Our results indicate that front-running risk impedes arbitrage activity, which in turn affects price informativeness on DEXs. For instance, the updated price of a token following a large informed trade on Uniswap V2 should be mirrored by the token price on Sushiswap. Compared to CEXs where quote updates are possible, price updates across

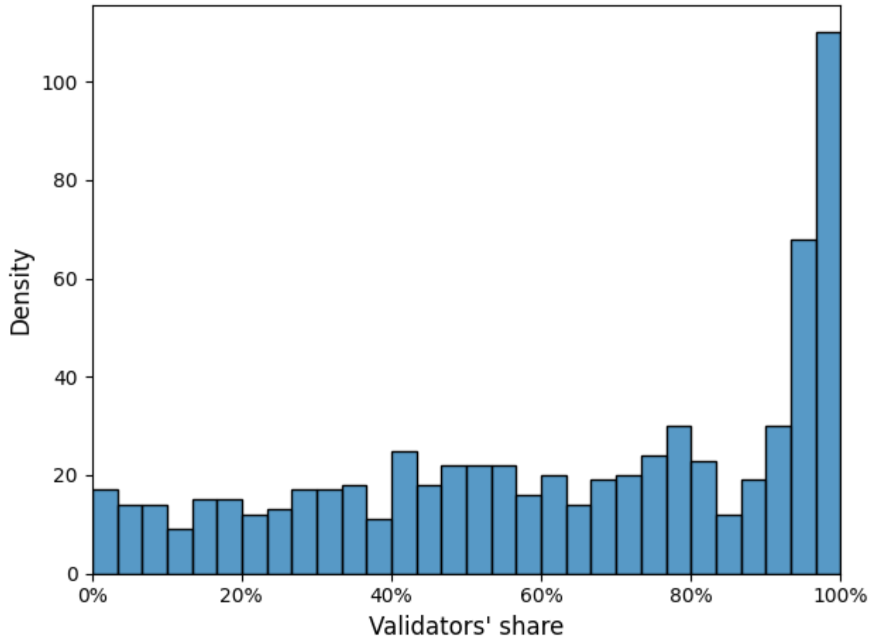


Figure 8: The distribution of arbitrage profit shares claimed by validators

DEXs happen solely via arbitrage as liquidity providers are passive.

To quantify this effect, we conduct a counterfactual analysis, introducing an arbitrage transaction at the beginning of the block whenever it results in a positive net profit. Subsequently, we assess the mid-price differences across exchanges. Whenever we detect an actual arbitrage transaction at the beginning of the subsequent block, we do not add a hypothetical arbitrage transaction. Our objective is to evaluate differences between mid-prices, that traders would face in the block after a hypothetical arbitrage transaction takes place in the first position of the block. Hence, our analysis is performed under the scenario where the cross-DEX arbitrageur's 'target' position is the first queue position ($q_a = 1$).

Table 6 presents the results of the counterfactual analysis. The second column contains the difference between the hypothetical post-arbitrage mid-price difference across pools ($\Delta p_{post-arb}$) and the original price differences (Δp), with t-statistics included in parentheses. Our findings suggest that in an environment free of front-running, prices encountered by traders would have been more closely aligned. Specifically, our analysis indicates that price disparities across pool pairs could have been reduced by 95 to 300 basis points, a change that is statistically (except for the Uniswap V2-Shibaswap WETH-WBTC pool pair) and economically significant.

Pool pair	Actual average mid-price difference	Hypothetical average mid-price difference (1st queue position)	Average difference of mid-price differences (1st queue position)	Number of arbitrage transactions (1st queue position)
Uniswap v2-Sushiswap WETH-USDT	158.21	63.73	94.48 (17.395***)	561
Uniswap v2-Shibaswap WETH-USDT	265.61	165.38	100.23 (2.305**)	141
Sushiswap-Shibaswap WETH-USDT	285.03	141.31	143.72 (4.121***)	149
Uniswap v2-Sushiswap WETH-WBTC	192.17	64.59	127.58 (2.866***)	154
Uniswap v2-Shibaswap WETH-WBTC	487.16	74.76	412.40 (1.419)	23
Sushiswap-Shibaswap WETH-WBTC	179.60	76.26	103.35 (5.425***)	31
Uniswap v2-Sushiswap WETH-DAI	164.09	64.89	99.21 (18.632***)	520
Uniswap v2-Shibaswap WETH-DAI	262.75	82.10	180.66 (7.577***)	94
Sushiswap-Shibaswap WETH-DAI	269.08	89.76	179.32 (10.277***)	133
Uniswap v2-Sushiswap WETH-USDC	148.27	62.29	85.98 (21.094***)	621
Uniswap v2-Shibaswap WETH-USDC	379.05	75.25	303.79 (5.343***)	111
Sushiswap-Shibaswap WETH-USDC	377.45	74.71	302.74 (4.802***)	98

Table 6: Counterfactual analysis (price differences are reported in basis points). *Actual average mid-price difference* is the average of the actual price differences prevailing across pools without an arbitrage transaction. *Hypothetical average mid-price difference* shows the average hypothetical mid-price differences after adding a hypothetical arbitrage transaction at the top of the block. *Average difference of price differences* is the difference between the actual and the hypothetical (or counterfactual) mid-price differences, which shows the hypothetical price improvement in a front-running free market. *Number of arbitrage transactions* reports the number of hypothetical arbitrage transactions executed across the pools. The cross-DEX arbitrageur's 'target' position is the first queue position ($q_a = 1$).

6.4 Infrastructure fees

Our findings indicate that front-running risk hampers arbitrage activity, and when arbitrage transactions do occur, validators capture the rents. Our second finding aligns with Capponi et al. (2023), who also find that conditional on an arbitrage transaction being submitted, arbitrageurs and other market participants engage in front-running each other's transactions until their net profits become nearly zero. They calculate a revenue-cost ratio, that represents the share of arbitrage profit accruing to the validators, and find that its median value is approximately 97%. In the case of CEXs, Budish et al. (2024) derive a theoretical model and show that given the current U.S. market structure (three CEXs and six high-frequency trading firms (TFs) being present on the market), the share of stale quote sniping rents accruing to CEXs in the form of exchange-specific speed-technology (ESST) fees is 30%.

	DEX		CEX
	In the absence of front-running risk (transaction placed 1st queue position)	In the presence of front-running risk (Capponi et al., 2023)	Exchanges' share from sniping rents (Budish et al., 2024)
Average revenue-cost ratio (%)	41.1	83.3	30
Median revenue-cost ratio (%)	36.1	96.6	30

Table 7: Infrastructure fees *The table presents the share of the arbitrage/sniping rents earned by validators/exchanges depending on their market type. The second column shows the share of the rent the validator would earn without front-running risk. We also calculate this share for the other transaction fee levels and find that the average and median are always around* 40%*. The third column shows the shares under front-running risk calculated by Capponi et al. (2023) i.e. when market participants are front-running each other. The fourth column shows the share of the arbitrage rent earned by CEXs based on the theoretical model of Budish et al. (2024), where they assume that there are 3 centralized exchanges (CEXs) and 6 trading firms (TFs) on the market. The former numbers are based on the actual US market structure.*

To investigate the potential revenue-cost ratio in the absence of front-running, we calculate its hypothetical value under each hypothetical fee level. The second column of Table 7 presents the average and median revenue-cost ratios under the most conservative hypothetical fee level. We also calculate these values under the less conservative hypothetical fee levels and find that, in the absence of front-running risk, the revenue-cost ratio remains stable around 40%. According to the formula in Proposition 3.2 of Budish et al. (2024), this value corresponds to the scenario where either two CEXs and six TFs or six CEXs and four TFs are present in the market. This revenue-cost ratio would only increase in either scenario if the market for CEXs or the market for TFs became more concentrated. This implies that, even in the absence of front-running risk, validators could, on average, expropriate a higher share of arbitrage rents than CEXs do in the current U.S. market. However, this does not imply that a CEX-based market is always preferable for arbitrage activity. A DEX-based market, free of front-running and characterized by generally low transaction fees, could result in a more favorable revenue-cost ratio, making it more attractive for arbitrage than the existing CEX-based U.S. market structure.

7. Deviations from pure DeFi arbitrage

We show that front-running discourages cross-DEX arbitrageurs from initiating atomic arbitrage transactions. However, empirically, we do not observe persistent price divergences across DEXs, contrary to what might be anticipated in an environment lacking arbitrage activity. In this section we provide evidence that shows how arbitrageurs deviating toward arbitrage practices that help them circumvent the front-running risk either by (1) not revealing the arbitrage transaction before it is validated in the next block (limited pre-trade transparency) or (2) splitting the arbitrage transaction into separate buy and sell transactions (reintroduction of costs).

7.1 Arbitrage activity in private pools

The introduction of trusted centralized entities called private relays or private pools by Flashbots in early 2021 enabled users (including arbitrageurs) to propagate their transactions directly to validators on a private channel without revealing their content to other market participants, similar to dark pools in centralized markets. The operators of private pools typically enforce specific rules to prevent validators from engaging in certain practices and violating these rules could result in the banning of validators from the private pool.[7] Furthermore, transactions sent to private pools are held under their custody until a validator within their service decides to include them in an upcoming block. Since these transactions do not land on-chain, i.e., in the public mempool, users do not have to pay a fee in the case of reversion.

During our sample period, the largest private pool was operated by Flashbots alongside with a smaller but still significant private pool operated by Eden Network. We use transaction-level data from Flashbots and gather similar data from the Eden Network website. During our sample period, these two private pool providers held a market share of over 90%. We label arbitrage transactions based on the venue to which they were propagated, designating Flashbots and Eden Network private pools as dark venues and the public mempool as the lit venue. Figure 9 shows the weekly number of arbitrage transactions by venue type. The Flashbots relay was adopted by validators on a larger scale starting in April 2021; before that, almost all arbitrage transactions were sent to

[7] For instance, Flashbots Fair Market Principles (https://hackmd.io/@flashbots/fair-market-principles) states that *'in the event of a breach of these principles by one of the block producers, the Flashbots core devs may act on behalf of stakeholders in disabling the access to the Flashbots network until the breach is rectified.'*

the lit public mempool. As the figure shows, after their introduction, dark venues were favored over the lit venue, though arbitrageurs did not completely stop propagating their transactions to the public mempool.

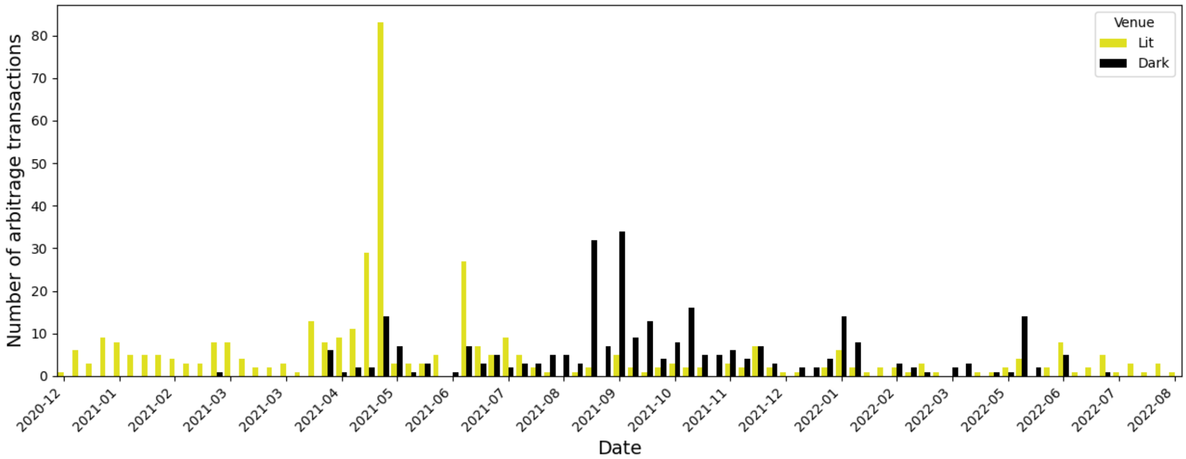


Figure 9: Weekly number of arbitrage transactions submitted to lit (public mempool) and dark (Flashbots and Eden Network private pools) venues

The reason dark venues might limit pre-trade transparency is that some market participants, such as other arbitrageurs, no longer see the arbitrage transaction in the public mempool. However, the validator of the next block who has access to the incoming private order flow can still screen the transactions. Validators can threaten the arbitrageur with front-running or leak information to other market participants, thus forcing the arbitrageur to pay the validator the value of the opportunity. It is important to emphasize that this type of front-running differs from situations where a validator front-runs or sandwiches[8] a transaction that includes trades, such as a large informed trade (see Capponi et al. (2023)). Monitoring front-running or sandwiching activity is a challenging task for private pool providers. For instance, Heimbach et al. (2023) finds that sandwich transactions have been validated in the blocks of private pool providers who explicitly claim to ban such transactions. Tracking arbitrage transactions that are being front-run is even challenging, given that it is hard to determine whether the same arbitrage transaction submitted to the private pool is the result of front-running or simply another arbitrageur submitting the same opportunity with a higher transaction fee. A key distinction from the lit market case is that, because the transaction is not directly sent to the mempool,

[8]Sandwich attacks involve front-running and back-running a large trade that moves the price in a liquidity pool, resulting in earning the rent within a block.

the reversion fee is zero ($r = 0$). This means that the expected net profit of the arbitrage opportunity for the arbitrageur is no longer negative.

	Dark (Flashbots)	Dark (Eden Network)	Dark (combined)	Lit (public mempool)
Number of arbitrage transactions	209	83	292	394
Average validator share (%)	75	59	70	63
Std. dev. of validator share (%)	26	34	30	43

Table 8: Arbitrage transactions by venue. *The firs row contains the number of actual arbitrage transactions by venue type. The second row shows the average share of the arbitrage (gross) profit paid to the validators in percentages, and the third row contains the standard deviation of the validator shares also in percentages. The second and third columns show separately the statistics for the dark pool. The last two columns contain the aggregate statistics for the dark and lit venues.*

Table 8 shows the number of actual arbitrage transactions by venue and the average share of arbitrage rent captured by validators (including its standard deviation) in percentages. The average validator share is 7 percentage points higher in dark venues[9] than in the lit venue. This indicates that, even though the arbitrage transaction is only visible to the validator before execution, arbitrageurs, on average, still pay a transaction fee roughly equal to the value of the arbitrage opportunity due to the threat of front-running.

The Proposer-Builder Separation (PBS) implemented on the Ethereum blockchain takes this concept further. In PBS, block building is done by builders, whose job is to assemble blocks and then compete with other builders to propose the highest value block to the validator (Buterin, 2021). During this process, builders can receive private order flow directly from users, including arbitrageurs, as well as access transactions in the public mempool. Essentially, under PBS, arbitrageurs have the same two options for submitting transactions as before: directly submitting to a builder, which corresponds to submitting to a private pool pre-PBS, or submitting to the public mempool, i.e., the lit venue. Capponi et al. (2024) shows that builders are incentivized to maximize the rent from transactions, as this determines whether they win the block proposal competition against other builders, even referring to the current scheme for Ethereum as 'Proof-of-MEV'.

[9]The average validator share in the dark pool operated by Eden Network is lower than in the lit market. However, Eden Network also offered a 'subscription' to their users (stakers), providing them with a priority spot in their blocks. Some arbitrageurs had this type of 'subscription', so the transaction fees might not always reflect the full cost of the arbitrage.

7.2 Statistical arbitrage activity

Cross-DEX arbitrageurs can circumvent front-running risk by splitting their arbitrage transaction into a buy- and sell-transaction. Nonetheless, this reintroduces other costs associated with arbitrage (e.g., Gromb and Vayanos (2010)). On the one hand, the cross-DEX arbitrageur cannot leverage flash loans to cover the capital needs of arbitrage, thus she must hold sufficient capital reserves. Second, the possibility of transactions landing between the buy- and sell-transactions and modifying the price exposes the cross-DEX arbitrageur to inventory risk. Statistical (or non-atomic) arbitrage can be implemented across DEXs on the same blockchain, across different blockchains (cross-chain arbitrage), or between a DEX and a CEX (DEX-CEX arbitrage). Recent literature has mainly focused on DEX-CEX arbitrage and has shown that arbitrageurs can profitably trade against liquidity pools employing CPMMs (Milionis et al., 2022, 2024), similar to those within our sample.

To investigate statistical arbitrage activity, we inspect the activities of blockchain-based trading algorithms, often referred to as MEV bots. We use similar heuristics as Heimbach et al. (2024) to identify statistical arbitrage activity. We use the addresses from the Etherscan label database to identify MEV bots and filter for their trading activity across the pools in our sample. Therefore in contrast to Heimbach et al. (2024) we do not require the transaction to be first propagated to a dark pool, as trading algorithms make all transactions. We use the hashes of MEV transactions to exclude atomic arbitrage transactions and sandwich transactions from the trading dataset. While sandwich transactions may appear similar to statistical arbitrage due to the presence of transactions containing a buy or a sell trade in the front of the block, they are always followed by another transaction that makes a trade in the opposite direction in the same pool at a lower position of the same block or in the next block. The transactions remaining are all submitted by MEV bots and consist solely of a single trade (identified as transactions using less than 250,000 gas), each of which would typically be identified as one of the legs of a statistical arbitrage. In line with our approach throughout the paper, we focus on transactions that change the pool states first within the block. Furthermore, we examine the inventory holdings of MEV bots, which serves as another indicative marker of their involvement in statistical arbitrage. The most active MEV bots possess inventories worth millions of dollars in various tokens, further suggesting their engagement in such arbitrage strategies.

We investigate whether arbitrageurs split their transactions by inspecting whether

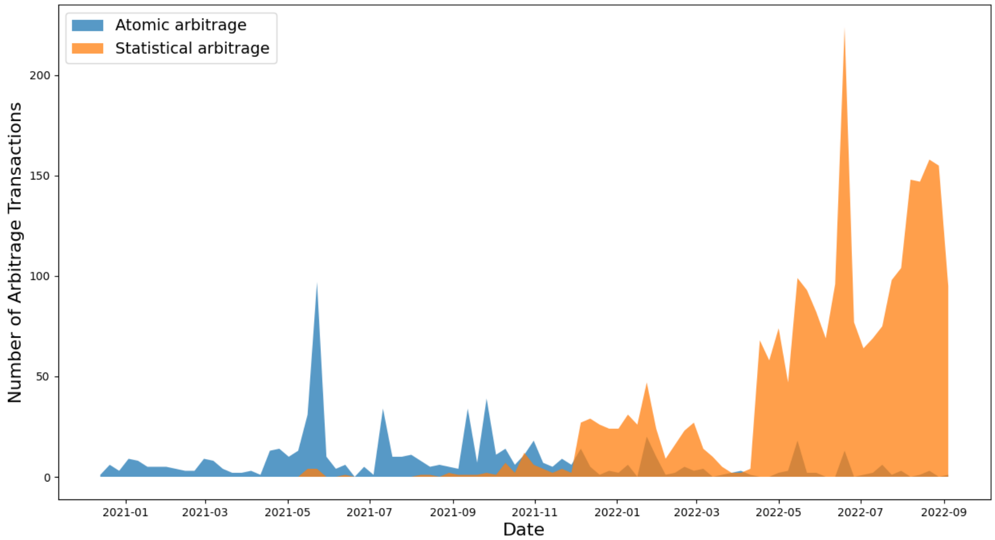


Figure 10: The weekly number of atomic (blue) and (potential) statistical (orange) arbitrage transactions.

they submit two consecutive single trade transactions, but we find no evidence of that. Any linking between blockchain addresses and CEX accounts is vague. Therefore, we cannot claim with certainty that a transaction observed on the blockchain represents one leg of an arbitrage transaction. It is also possible that MEV bots are implementing strategies that involve single trade transactions, which may not be related to statistical arbitrage. However, we are confident that our filtering process reasonably accurately identifies the components of statistical arbitrage transactions.

In Figure 10, we plot the weekly number of atomic and potential statistical arbitrage transactions observed in our sample pools. The time series plot shows that by the end of our sample period, the number of statistical arbitrage transactions greatly exceeded the number of atomic arbitrage transactions. This anecdotal evidence suggests that, despite the costs associated with statistical arbitrage, it remains a viable and profitable option compared to atomic arbitrage in the presence of front-running risk. Our results align with the large-scale study by Heimbach et al. (2024) who documents that DEX-CEX arbitrage might account for as much as one-fourth of the trading volume on the five largest exchanges on the Ethereum blockchain.

8. Conclusions

Many market participants believe that blockchain technologies have the potential to radically transform the transfer of financial assets. This paper questions whether Decentralized Finance (DeFi) can sustain itself. DeFi relies on blockchain-based settlement. We show that blockchain-based settlement implies a fundamental dilemma that causes market inefficiencies for *any* DeFi application. The dilemma is simple: Without pre-trade transparency of transactions waiting for verification, validators cannot reliably cooperate to verify transaction validity. However, without offering transaction fees to validators, there is no incentive for them to collude with the aim of blockchain-based settlement: establish consensus about transaction histories without a trusted third party.

Combining these two fundamental pillars of blockchain-based settlement – pre-trade transparency and transaction fees – however, triggers front-running feasible. This practice exploits that public information (such as transactions waiting for verification) is available in advance. For good reasons, such practices are considered illegal in traditional financial markets. Front-running risk renders, for example, arbitrage activities prohibitively costly as it triggers a wasteful arms race for validation services. The result is devastating: markets fail to work because it is not worthwhile to act on informational advantages anymore. We provide evidence for this pattern based on the novel, granular data for cross-decentralized exchange trading. Our analysis shows that price differences remain large, unexploited by arbitrageurs due to front-running risks.

The dilemma for DeFi unfolds as follows: The only way to overcome pre-trade transparency *and* transaction fees is by undermining the DeFi ideal, which sets off to render trusted intermediaries obsolete. Our paper shows that front-running risk renders DeFi applications inefficient without trusted intermediation. Reinstalling trusted third parties mitigates the problem.

References

Auer, R. (2019). Embedded supervision: how to build regulation into decentralised finance . *BIS Working Paper*.

Biais, B., C. Bisiere, M. Bouvard, and C. Casamatta (2021). The Blockchain Folk Theorem. *Review of Financial Studies 32*(5), 1662–1715.

Budish, E., R. S. Lee, and J. J. Shim (2024). A theory of stock exchange competition and innovation: Will the market fix the market? *Journal of Political Economy (forthcoming) 0*(0), 000–000.

Buterin, V. (2021). Proposer/block builder separation-friendly fee market designs.

Capponi, A. and R. Jia (2021). The Adoption of Blockchain-based Decentralized Exchanges. *Working paper*.

Capponi, A., R. Jia, and S. Olafsson (2024, February). Proposer-builder separation, payment for order flows, and centralization in blockchain. Available at SSRN: `https://ssrn.com/abstract=4723674`.

Capponi, A., R. Jia, and S. Yu (2023, November). Price discovery on decentralized exchanges. Available at SSRN: `https://ssrn.com/abstract=4236993`.

Chiu, J. and T. V. Koeppl (2019). Blockchain-Based Settlement for Asset Trading. *Review of Financial Studies 32*(5), 1716–1753.

Cong, L. W. and Z. He (2019). Blockchain Disruption and Smart Contracts. *Review of Financial Studies 32*(5), 1754–1797.

Daian, P., S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels (2020). Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 910–927.

Daniel, K. D. and D. Hirshleifer (2018). A theory of costly sequential bidding. *Review of Finance 22*(5), 1631–1665.

Easley, D., M. O'Hara, and S. Basu (2019). From Mining to Markets: The Evolution of Bitcoin Transaction Fees. *Journal of Financial Economics 134*(1), 91–109.

Easley, R. F. and R. Tenorio (2004). Jump bidding strategies in internet auctions. *Management Science 50*(10), 1407–1419.

Gromb, D. and D. Vayanos (2010). Limits of Arbitrage. *Annual Reviews of Financial Economics 2*(1), 251–275.

Grossman, S. J. and J. E. Stiglitz (1980). On the impossibility of informationally efficient markets. *The American economic review 70*(3), 393–408.

Harvey, C. R., A. Ramachandran, and J. Santoro (2021). *DeFi and the Future of Finance*. John Wiley & Sons.

Hautsch, N., C. Scheuch, and S. Voigt (2024). Building trust takes time: Limits to arbitrage for blockchain-based assets. *Review of Finance (Forthcoming)*.

Heimbach, L., L. Kiffer, C. F. Torres, and R. Wattenhofer (2023). Ethereum's proposer-builder separation: Promises and realities.

Heimbach, L., V. Pahari, and E. Schertenleib (2024). Non-atomic arbitrage in decentralized finance.

Hinzen, F. J., K. John, and F. Saleh (2022). Bitcoin's Limited Adoption Problem. *Journal of Financial Economics 144*(2), 347–369.

Huberman, G., J. D. Leshno, and C. Moallemi (2021). Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System . *The Review of Economic Studies 88*(6), 3011–3040.

Lehar, A. and C. A. Parlour (2024). Decentralized Exchange: The Uniswap Automated Market Maker. *The Journal of Finance (forthcoming)*.

Makarov, I. and A. Schoar (2020). Trading and Arbitrage in Cryptocurrency Markets. *Journal of Financial Economics 135*(2), 293–319.

Malinova, K. and A. Park (2017, July). Market design with blockchain technology. Available at SSRN: `https://ssrn.com/abstract=2785626`.

McLaughlin, R., C. Kruegel, and G. Vigna (2023). A large scale study of the ethereum arbitrage ecosystem. In *32nd USENIX Security Symposium (USENIX Security 23)*, Anaheim, CA, pp. 3295–3312. USENIX Association.

Milionis, J., C. C. Moallemi, and T. Roughgarden (2024). Extended abstract: The effect of trading fees on arbitrage profits in automated market makers. In A. Essex, S. Matsuo, O. Kulyk, L. Gudgeon, A. Klages-Mundt, D. Perez, S. Werner, A. Bracciali, and G. Goodell (Eds.), *Financial Cryptography and Data Security. FC 2023 International Workshops*, Cham, pp. 262–265. Springer Nature Switzerland.

Milionis, J., C. C. Moallemi, T. Roughgarden, and A. L. Zhang (2022). Quantifying loss in automated market makers. New York, NY, USA, pp. 71–74. Association for Computing Machinery.

Qin, K., S. Chaliasos, L. Zhou, B. Livshits, D. Song, and A. Gervais (2023). The blockchain imitation game. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 3961–3978. USENIX Association.

Qin, K., L. Zhou, and A. Gervais (2022). Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 198–214.

Roughgarden, T. (2023). Transaction fee mechanism design. *Working paper*.

Saleh, F. (2020). Blockchain Without Waste: Proof-of-Stake. *The Review of Financial Studies 34*(3), 1156–1190.