First-Spammed, First-Served: MEV Extraction on Fast-Finality Blockchains

Krzysztof M. Gogol[1], Manvir Schneider[2], and Claudio J. Tessone[1,3]

[1] University of Zurich UZH
[2] Cardano Foundation
[3] UZH Blockchain Center

Abstract. This research analyzes the economics of spam-based arbitrage strategies on fast-finality blockchains. We begin by theoretically demonstrating that, splitting a profitable MEV opportunity into multiple small transactions is the optimal strategy for CEX-DEX arbitrageurs. We then empirically validate these findings on major Ethereum rollups. To uncover the structure of reverted transactions, we construct execution graphs from transaction traces and systematically search them to identify DEX or router interactions and targeted liquidity pools. This analysis reveals that 80% of reverted transactions are swaps with approximately 50% targeting USDC-WETH pools on Uniswap v3/v4. These patterns intensified following the March 2024 Dencun upgrade, which lowered L2 gas costs and made spam-based arbitrage economically viable.
Counterintuitively, we find that these reverted MEV transactions rarely engage with Priority Fee Auctions (PFAs), preferring to submit duplicate transactions rather than bid for inclusion. Moreover, reverted transactions cluster at the very top of blocks on fast rollups like Arbitrum and ZKsync, indicating an intense latency race and revealing the fragility of fee-based ordering under sub-second block times.

Keywords: MEV, Rollup, Arbitrage

1 Introduction

The evolution of Maximum Extractable Value (MEV) has undergone significant transformations, beginning with Ethereum's early days. The introduction of MEV Boost mechanisms, followed by Proposer-Builder Separation (PBS), transformed how MEV was captured on Ethereum, and now, with the rise of rollups [24], Layer-2 (L2) scaling solutions for Ethereum, history appears to be repeating itself.

Rollups are gaining traction, reshaping transaction execution and settlement dynamics. On average, rollups process 30 times more transactions per second (TPS) than Ethereum [15], offer 55 times higher gas per second (GPS) [4] and, consequently, have attracted a growing number of Decentralized Finance (DeFi) users and protocols, further driving Total Value Locked (TVL) on L2s [15]. However, despite these advancements, MEV remains an inherent challenge, leading to

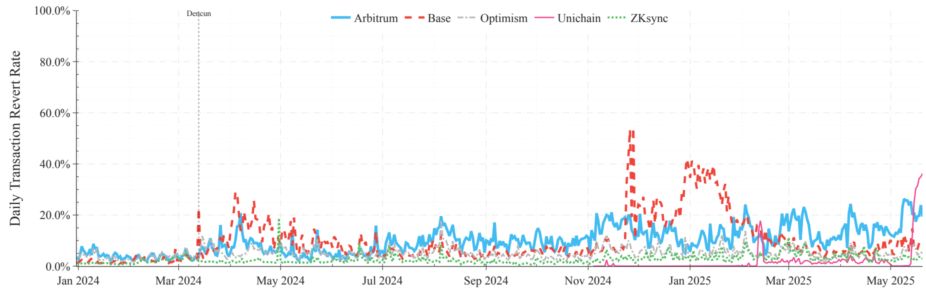


Fig. 1: Transaction revert rates across Ethereum L2s, highlighting the increase observed following the Dencun upgrade on March 13, 2024.

what is often described as the *MEV trilemma*—where avoiding MEV extraction entirely is impossible.

The centralization of sequencers significantly shapes MEV on rollups. A rollup sequencer is an entity responsible for ordering transactions, forming blocks, and submitting them to the Layer 1 chain [19]. As the sole actor capable of transaction ordering, the sequencer introduces implicit trust assumptions—namely, that transactions are processed on a First-Come, First-Served (FCFS) basis with Priority Fee Auctions (PFA). Moreover, most rollups (e.g., Arbitrum, Optimism, Base, Unichain, ZKsync) operate private mempools, rendering front-running infeasible. MEV opportunities on L2s are thus limited to back-running strategies such as arbitrage and liquidations. Due to the low latency of rollups (200ms–2s), MEV extraction becomes a race to be the first to submit transactions to the sequencer.

In March 2024, the Dencun upgrade [5] introduced *blobs* [6], a temporary data storage designed to optimize rollups' scalability, which resulted in a significant reduction in gas fees on L2 [12]. At the same time, we observed a surge in the transaction revert rate [7] that might indicate the spam-based MEV strategies on L2s. Although initially blob transactions were relatively inexpensive, their rising costs could pose new challenges for rollups' scalability and MEV extraction strategies.

Looking ahead, major developments such as Unichain [1] with the proposed sequencer-builder separation, MEV tax [21] and the revert protection mechanism, can introduce new paradigms for MEV extraction on L2s. This research explores these developments and analyzes the reverted transactions on L2s with the goal of attributing them to MEV extraction strategies.

Related Work. Zhu et al. [29] present a game-theoretic model of MEV auctions under revert protection, showing analytically that revert protection increases auction revenue, market efficiency, and blockspace usage by incentivizing deterministic participation. In contrast, our work focuses on empirically observed spam-based MEV strategies on fast-finality blockchains with FCFS ordering. We show that, absent revert protection, optimal arbitrage strategies involve splitting trades into many small, potentially failing transactions, a behavior our data confirms across multiple rollups.

TimeBoost is the transaction ordering policy for rollup sequencers incorporating transaction timestamps and bids [17]. Its performance simulation shows it can lead to similar or higher returns than FCFS for L2 MEV searchers [8], however the empirical validation is still pending. *MEV tax* is another mechanism that is proposed to enhance MEV extractions by allowing each smart contract to organize its priority fee auctions. This model, if implemented by DeFi protocols, would allow re-distributing MEV back to the DeFi users [21].

The empirical study of MEV, especially on L2 blockchains, remains relatively underexplored. Heimbach et al. (2024, January) [13] evaluated the non-atomic arbitrage on Ethereum. Gogol et al. (2024, March) [11] estimated the non-atomic arbitrage between CEX and L2 DEX, and Torres et al. (2024, April) [26] quantified the extracted atomic arbitrage on rollups. Subsequently, Gogol et al. (2024, June) [10] and Oz et al. (2025, January) [30] investigated non-atomic arbitrage strategies on L2s and estimated the potential size of cross-rollup MEV.

Contribution. The contributions of this research are threefold:

– **Theoretical insight into optimal MEV strategy.** We show that, on fast-finality blockchains with private mempools and first-come, first-served (FCFS) transaction ordering, the optimal strategy for CEX-DEX arbitrageurs is to split MEV transactions into multiple smaller trades, mitigating failure risk and maximizing expected profit.
– **Empirical evidence of spam-based MEV extraction.** We empirically find that over 80% of reverted transactions on Ethereum rollups are swap transactions attributable to spam-based arbitrage (MEV) by bots. To identify these patterns, we construct a graph from transaction traces for each reverted transaction and search it to detect DEX/router interactions and the targeted liquidity pools. This analysis reveals that roughly 50% of reverted swaps target USDC-WETH pools on Uniswap v3 and v4, underscoring a strong concentration of MEV activity on high-liquidity pairs.
– **Counterintuitive dynamics of priority fee.** Despite the presence of Priority Fee Auctions (PFAs), MEV bots rarely use priority fees, instead favoring duplicate transaction spam—likely due to unreliable fee-based ordering on fast-finality chains. Furthermore, we find that reverted transactions cluster at the top of blocks on faster rollups (e.g., Arbitrum, ZKsync), revealing an intense latency race and suggesting the ineffectiveness of current economic ordering mechanisms under minimal time margins.

Paper Organization. Section 2 provides an overview of roll-ups, Layer-2 (L2) blockchain mechanisms, and their implications for MEV dynamics. Section 3 builds the optimal MEV strategy on fast-finality blockchain for a MEV arbitrageur. Subsequently, sections 4 and 5 present an empirical analysis of reverted transactions on L2s and explain their link to MEV extraction. Finally, Sections 6 and 7 include the discussion and conclusions.

2 Preliminaries: MEV on Rollups

According to the blockchain scalability trilemma [18], a blockchain can prioritize at most two of the following three properties: decentralization, security, and scalability. Ethereum, the leading DeFi blockchain by TVL, prioritizes decentralization and security. This design choice has led to network congestion, high gas fees, and limited throughput—approximately 12 transactions per second (TPS). To address these limitations, scaling solutions have been developed at both Layer 1 (L1) and Layer 2 (L2). L1 scaling introduces new blockchains with alternative consensus mechanisms [16], sharding [27], and independent infrastructure. In contrast, L2 scaling executes intensive computations off-chain, posting compressed results to the underlying L1 [23,9].

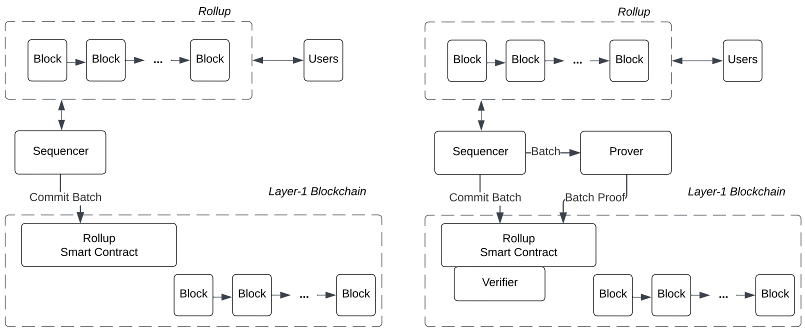
Rollups [25] are non-custodial L2 solutions that function as independent blockchains: they execute transactions, produce blocks, and periodically submit compressed transaction data to the L1. This design allows rollups to inherit the security guarantees of the underlying L1—e.g., Ethereum's staked ETH—making it infeasible to tamper with rollup data without compromising L1 security.

Optimistic rollups [14] assume transactions are valid unless challenged. This assumption simplifies implementation and accelerates compatibility with the Ethereum Virtual Machine (EVM), helping optimistic rollups become the first to attract DeFi adoption. However, their fraud-proof mechanism introduces a withdrawal delay, typically enforced through a 7-day challenge period.

ZK-rollups [3] use zero-knowledge proofs (ZKPs) to validate state transitions. After a proof is generated off-chain by a *prover*, it is verified on-chain by a smart contract (*verifier*). This architecture ensures rapid finality and enhances compression—only proofs, not raw transaction data, need to be posted to L1. The trade-off is increased computational overhead due to proof generation.

A *sequencer* [2] is a key component of rollup infrastructure, responsible for ordering transactions, forming blocks, and bundling them into batches submitted to L1. This improves gas efficiency relative to L1 execution. Despite cryptographic guarantees from optimistic and ZK rollups, sequencers are currently centralized. Major rollups like Arbitrum, Optimism, Base, Unichain, ZKsync, and StarkNet rely on centralized sequencers—and in the case of ZK-rollups, centralized provers as well. Efforts are underway to decentralize both [20,28].

Blockchain finality refers to the point at which a transaction becomes irreversible. On rollups, we distinguish between *soft* and *hard* finality. Soft finality occurs when a transaction is accepted by the sequencer and included in an L2 block, making it irreversible from the rollup's perspective. Hard finality is achieved once the corresponding batch is posted and confirmed on L1, securing the transaction with L1 guarantees. As rollups are currently centralized, users, bridges, and MEV searchers typically act upon soft finality, which occurs every 200ms–2s—much faster than Ethereum's 12s L1 block time.



(a) Optimistic rollup (b) ZK-rollup

Fig. 2: Most rollups operate with centralized sequencers that maintain private mempools. These sequencers have exclusive control over transaction ordering within the rollup.

3 Trade Splitting

Token pair. There is a single token pair (X, Y) traded on

- a **constant-product AMM** with reserves (x, y) and fee $f \in [0, 1)$;
- a **centralized exchange (CEX)** that quotes a *fixed* price $P_c > 0$ expressed in Y per X and always settles.

AMM swap mechanics. If an arbitrageur swaps $q > 0$ units of X with the AMM liquidity pool, she receives

$$\Delta y(q) \ = \ \frac{y(1-f)q}{x + (1-f)q}, \tag{1}$$

and the reserves in the AMM liquidity pool update to $(x + (1-f)q,\ y - \Delta y(q))$. Formula (1) is the standard CPMM payout. Note that $\Delta y(q)$ is *strictly concave* in q and satisfies $\Delta y'(q) > 0,\ \Delta y''(q) < 0$.

Swap–failure risk. The on-chain swap succeeds with probability $p(q) \in (0, 1]$, $p(0) = 1,\ p'(q) < 0$,[4]

Costs.

- A fixed per-swap overhead $c_g \geq 0$ (gas & latency).
- If the swap fails, the arbitrageur is left *long* q units of X; liquidating this inventory incurs a penalty $\phi \geq 0$ (possibly 0).

[4] In our empirical analysis, we show that on fast-finality blockchains, priority fees do not guarantee swap execution due to intense latency races. Larger swaps, which consume more gas, may miss priority or slippage tolerance if smaller swaps execute firsts.

Arbitrage objective. The trader wishes to arbitrage a *total* size $D > 0$ (in token X) by

1. buying D units of X on the CEX for $P_c D$ units of Y;
2. selling those units into the AMM, possibly split into $n \in \mathbb{N}$ equal chunks of size $q = D/n$.

3.1 Per–swap expected profit

For one attempted swap of size q (before any pool updates):

$$\pi(q) = p(q)\big[\Delta y(q) - P_c\, q\big] - \big(1 - p(q)\big)\big[P_c\, q + \phi\big] - c_g \tag{2}$$
$$= p(q)\big[\Delta y(q)\big] - P_c\, q - (1 - p(q))\phi - c_g. \tag{3}$$

Interpretation.

- *Success* ($p(q)$): receive $\Delta y(q)$ from AMM, offset the $P_c q$ spent on CEX.
- *Failure* ($1 - p(q)$): CEX trade clears, AMM trade reverts; trader still paid $P_c q$ on CEX and may liquidate at cost ϕ.
- *Overhead* c_g is incurred regardless.

3.2 Optimal trade-splitting

Executing n equal-sized swaps delivers expected profit $\Pi(n) = n\, \pi(D/n)$.

Ignoring the (concavity-improving) drift of (x, y) between slices only makes splitting *more* advantageous, so the sign of $\Pi(n) - \Pi(1)$ is decisive.

Proposition 1 (Trade–splitting threshold). *Define the* ***marginal–benefit*** *function:*

$$M(q) := p'(q)\big[\Delta y(q) + \phi\big] + p(q)\, \Delta y'(q) - P_c, \qquad q > 0. \tag{4}$$

(i) *$M(q)$ is strictly decreasing on $(0, \infty)$.*
(ii) *For any per–swap overhead $c_g \geq 0$ the optimal chunk size $q^\star \in (0, D]$ is the* unique *solution (if it exists) of*

$$q\,\big[M(q) + P_c\big] = p(q)\,\big[\Delta y(q) + \phi\big] - \big(c_g + \phi\big), \tag{5}$$

with the dichotomy

$$q^\star = \begin{cases} D, & \text{if } c_g \geq \theta, \\ \text{the unique root of (5) on } (0, D), & \text{if } c_g < \theta, \end{cases} \qquad \theta := p(D)\big[\Delta y(D) + \phi\big] - \phi - D[M(D) + P_c]\,.$$

(iii) *Putting $n^\star = \lceil D/q^\star \rceil$, the profit-maximizing (Nash equilibrium) strategy is to execute $n^\star$ equal-sized swaps when $q^\star < D$, and a single swap otherwise.*

Proof. **(i)** Because $p'(q) < 0$, $p''(q) \leq 0$ and $\Delta y''(q) < 0$, we have $M'(q) = p''(\Delta y + \phi) + 2p'\Delta y' + p\,\Delta y'' < 0$.

(ii) The expected profit of one swap of size q is

$$\pi(q) = p\,\Delta y - P_c q - (1-p)\phi - c_g,$$

and its derivative is $\pi'(q) = M(q)$ by construction (4). For n equal slices ($q = D/n$) the total profit is $\Pi(n) = n\,\pi(D/n)$. Treat n continuously; the first-order condition $\Pi'(n) = 0$ gives $\pi(q) = q\,\pi'(q)(q = D/n)$. Substituting π and π' yields

$$p\Delta y - P_c q - (1-p)\phi - c_g = q\bigl[M(q) + P_c\bigr],$$

i.e. equation (5). Since the left-hand side of (5) is *strictly increasing* in q while the right-hand side is *strictly decreasing* (by part (i)), there is at most one root on $(0, D)$. Evaluating both sides at $q = D$ defines the threshold θ; if $c_g \geq \theta$ no interior root exists and $q^\star = D$.

(iii) If $q^\star = D$ the arbitrageur makes a single swap ($n^\star = 1$). Otherwise, decreasing q below $q^\star$ (hence increasing n) lowers profit because the right-hand side of (5) falls faster than the left; rounding $n = D/q^\star$ upward gives the integer optimum $n^\star = \lceil D/q^\star \rceil$.

4 Data Collection and Methodology

Data Collection. A key challenge in our empirical analysis is that DeFi smart contracts do not emit event logs for reverted transactions. To overcome this, we analyze transaction traces directly. For each reverted transaction, we construct a trace-based execution graph and search it to identify DEX or router interactions, targeted liquidity pools (Uniswap v2/v3), and token addresses (Uniswap v4). Although all analyzed rollups are EVM-compatible, each liquidity pool, AMM DEX, router, and token contract has a different address across L2s, requiring chain-specific mapping. To support this, we build a labeling library for each rollup using event logs from successfully executed swaps, which we then use to match and annotate nodes in the trace graphs of reverted transactions. In this study, we analyze both successful and reverted transactions on leading EVM-compatible rollups: Arbitrum, Optimism, Base, ZKsync, and Unichain. The dataset spans from the Dencun upgrade (March 2023) to the present and is sourced from full archive nodes. A detailed overview of the analyzed rollups is provided in Table 1. Notably, all analyzed rollups have private mempools operated by centralized sequencers with block times ranging between 0.2s and 2s (compared to 12s in Ethereum). The default order flow is First-Come First-Served (FCFS) with Priority Fee Auctions (PFA). On April 17, 2025, Arbitrum introduced the TimeBoost mechanism (explained further).

Users interact with blockchains and smart contracts through atomic transactions. Smart contracts can trigger the revert option for the entire atomic transaction, if certain conditions are not met (e.g. swap price changed exceeding the slippage limit). The reverted transaction does not emit event logs, but registers its actions in the traces, which we analyze in this study. The exact methodology for on-chain data extractions follows.

Methodology. A key challenge in our empirical analysis is that DeFi smart contracts do not emit event logs for reverted transactions. To overcome this, we analyze transaction traces directly. For each reverted transaction, we construct a trace-based execution graph and search it to identify DEX or router interactions, targeted liquidity pools (Uniswap v2/v3), and token addresses (Uniswap v4). Although all analyzed rollups are EVM-compatible, each liquidity pool, AMM DEX, router, and token contract has a different address across L2s, requiring chain-specific mapping. To support this, we build a labeling library for each rollup using event logs from successfully executed swaps, which we then use to match and annotate nodes in the trace graphs of reverted transactions.

Successful transactions. We extract swap details from Uniswap v2, v3, and v4 by parsing their emitted event logs.

Reverted transactions. Since reverted transactions do not emit logs, we use on transaction *traces* to analyze their behavior.

- **Uniswap v2 and v3:** Traces are matched to specific liquidity pools using pool addresses. We use the `dex.trades` labeling dataset from Dune Analytics for pool classification.
- **Uniswap v4:** Given the architectural changes in Uniswap v4, we identify swap transactions by checking in the traces whether the transaction interacted with (i) a Uniswap v4 pool manager and (ii) known token contract addresses (e.g. USDC or WETH). In v4, both event logs and traces reference only the pool manager address, rather than individual pools (or hook).

Bot Identification.

MEV bots are typically smart contracts storing the strategy and logic for interacting with DeFi protocols or other smart contracts. Since these bots must be externally triggered to execute MEV strategies, we identify potential bots by analyzing the `to` addresses in reverted transactions. Specifically, we focus on those addresses that are frequently targeted, excluding known DEX routers and AMM pools. To refine this classification, we apply the following criteria:

- Addresses with a known label or owner name (e.g., wallets labeled by blockscan, centralized exchanges, or DAOs) are excluded, as they are unlikely to be bots. This is captured by the condition `owa.owner_key IS NOT NULL`.
- Addresses that contain on-chain bytecode are flagged as likely automated agents (i.e., bots or smart contract traders), corresponding to the condition `cm.contract_address IS NOT NULL`.

5 Empirical Analysis

The sudden increase in revert rates on Ethereum rollups occurred following the Dencun upgrade on March 13, 2024, as depicted in Figure 1. This upgrade led to a significant reduction in gas fees on L2s, bringing them below $0.01. While L2

Table 1: Technical characteristics of analyzed Ethereum rollups: L2s rely on private mempools operated by centralized sequencers [30] with Priority Fee Auctions (PFAs). Arbitrum introduced TimeBoost sequencing on April 17.

Rollup	**Type**	**Mempool**	**Block Time**	**Order Flow**	**Launch**
Arbitrum (ARB)	Optimistic	Private	0.25s	TimeBoost	Aug 21
Base (BASE)	Optimistic	Private	2.0s	PFA	Aug 23
Optimism (OP)	Optimistic	Private	2.0s	PFA	Dec 21
Unichain (UNI)	Optimistic	Private	0.25s	PFA	Apr 24
ZKsync (ZK)	ZK	Private	1.0s	PFA	Mar 23

revert rates oscillated between 5% prior to the upgrade, they rose to over 10% afterward—by contrast, the revert rate on Ethereum mainnet remains around 1%.

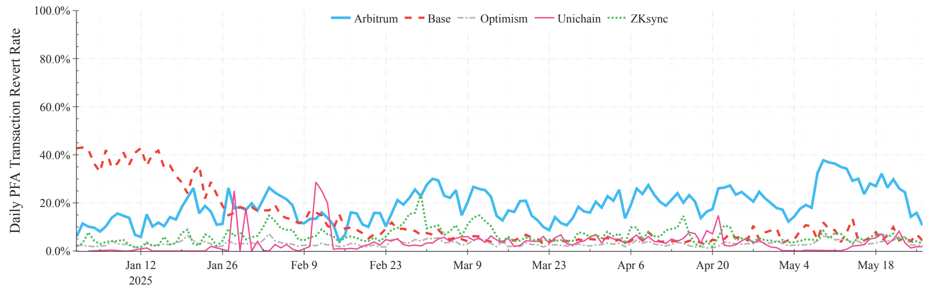
In the subsequent sections, we analyze the reverted transactions with respect to their use of priority fee auctions and their position within blocks. We classify reverted transactions into swap transactions and examine their key characteristics, including the targeted DEX and the token pair pool.

5.1 Priority-Fee Transactions

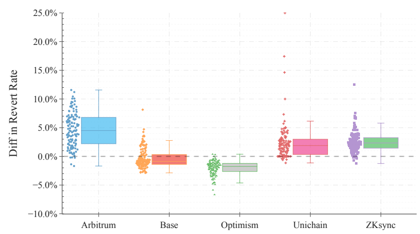
Users on Ethereum L2s can specify a *priority fee per gas* to incentivize the inclusion of their transactions in a block. Figure 3a illustrates the revert rate of the priority-fee transaction, whereas Figure 3b shows the difference in daily revert rates between transactions that utilize this mechanism and the overall transaction set. On Base and Optimism, the difference is predominantly negative, indicating that priority-fee transactions are less likely to revert. In contrast, on Arbitrum, Unichain, and ZKsync, the difference is mostly positive—suggesting that transactions with priority fees are more likely to revert than others.

This discrepancy can be partially explained by differences in block times across rollups. Both Base and Optimism have block times of approximately 2 seconds, whereas Arbitrum and Unichain produce blocks every 0.25 seconds, and ZKsync every 1 second, Table 1. On fast-finality chains with block times of 1 second or less, paying a higher priority fee does not necessarily guarantee inclusion. This is because two transactions—one with a priority fee and one without—may be initiated at nearly the same time, but the non-priority-fee transaction could reach the mempool slightly earlier and be executed first. In that case, the priority-fee transaction may still revert, despite offering a higher fee. Thus, the priority fee only guarantees execution if the non-priority transaction is still pending in the mempool, which is not always the case on fast-finality chains.

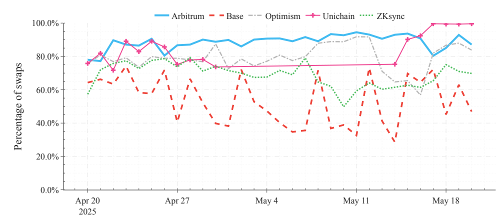
Furthermore, Figure 4 shows that many reverted transactions do not use the priority fee mechanism at all. On Arbitrum and Optimism, more than 50%



(a) Daily transaction revert rates for priority-fee transactions across Ethereum L2s.



(b) Difference in revert rates between priority-fee transactions and all transactions. Most L2s show a positive difference, suggesting that priority-fee transactions revert more often.



(c) Share of swaps among reverted transactions. Most of reverted transactions on L2s are classified as swaps, indicating strong MEV or DEX trading activity.

Fig. 3: Analysis of transaction reverts on Ethereum rollups: (a) shows revert rates for priority-fee transactions across L2s; (b) compares revert rates for priority-fee versus all transactions; (c) examines the composition of reverted transactions.

of reverted transactions omit the priority fee. On Unichain, nearly all reverted transactions set the priority fee to exactly 1 wei, indicating minimal fee competition and a lack of engagement with the auction mechanism.

Figure 4 presents the distribution of positions within a block for reverted transactions. On Arbitrum and ZKsync, reverts are heavily concentrated at the very beginning of the block, while on other rollups (Base, Optimism, and Unichain), reverted transactions typically appear around the fourth position—indicating differing MEV execution patterns across L2s.

Further analysis of gas fees paid by reverted transactions is provided in Appendix A, where we break down the total fee into three components: priority fee, base fee, and L1 data availability fee. Overall, although many reverted transactions either do not pay a priority fee or pay only a minimal one, the share of the priority fee in the total fee paid on L2s exceeds half across all analyzed L2 networks.

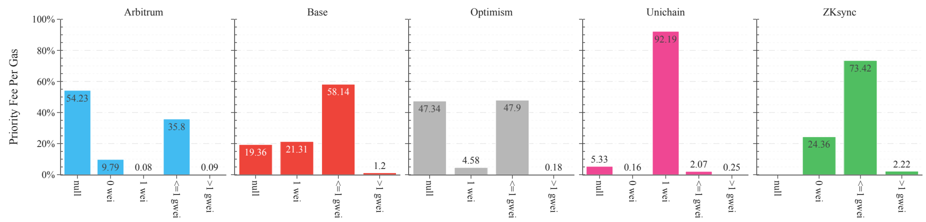


Fig. 4: Priority fee per gas used by the reverted transaction across analyzed blockchains. Only on Base and ZKsync the priority-fee was used by majority reverted transactions.

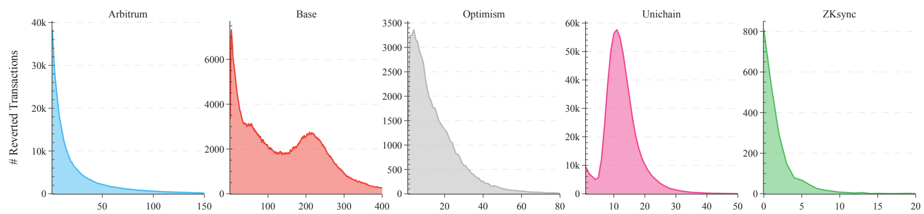


Fig. 5: Index of reverted transactions within blocks. On Arbitrum and ZKsync, they cluster at the block start, while on other rollups they peak around the fourth position.

5.2 Reverted Swaps

Figure 3c reveals that the majority of reverted transactions are swaps. On most chains—such as Arbitrum, Optimism, and Unichain—over 80% of reverted transactions fall into this category. The classification of transactions as swaps is based on transaction labeling heuristics; hence, the reported values should be interpreted as lower bounds.

A more detailed breakdown is presented in Figure 6. Overall, the majority of reverted swaps interact with USDC-WETH pools on Uniswap (v3) or other decentralized exchanges, suggesting that many of these transactions originate from MEV searchers executing arbitrage strategies on highly liquid token pairs.

Uniswap (v3) is the dominant DEX among reverted swaps on Arbitrum, accounting for over 60% of such transactions. On Unichain, Uniswap (v4) is the primary target, with more than 90% of reverted swaps involving its contracts. In contrast, Base exhibits a more diversified distribution across DEXs, with reverted swaps targeting Uniswap (v2), (v3), and (v4) each comprising approximately 20% of cases.

The most frequently targeted token pair is USDC-WETH, accounting for over 50% of reverted swaps on Arbitrum. This pair spans multiple liquidity pools, including USDC-WETH 5bps and 30bps fee tiers. Other commonly targeted pairs include USDC-USDT and WBTC-WETH.

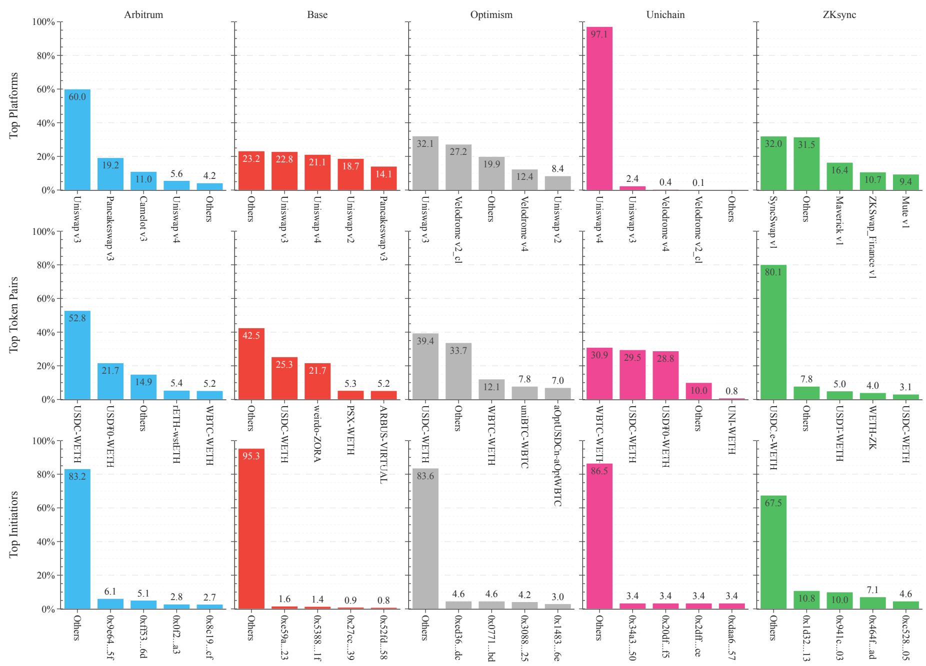


Fig. 6: Percentage breakdown of the top 3 attributes of reverted swaps — target DEX, token pair, transaction sender — for the analyzed blockchains: Arbitrum, Base, Optimism, Unichain and ZKsync.

Lastly, we analyzed the distribution of initiators of reverted swaps. The results indicate a broad and fragmented distribution, suggesting that no single actor dominates the reverted swap activity.

6 Discussion

Our theoretical model suggests when arbitrageurs should optimally split trades into multiple smaller chunks to mitigate the risk of transaction failure. The empirical findings show that on fast-finality chains priority fees do not guarantee swap execution due to intense latency races, and PFAs are not widely used. Consequently, swap success probabilities decline with trade size, which is formalized in the model by a decreasing function $p(q)$, where larger swaps are more likely to revert.

This reflects several empirical realities: larger swaps induce greater price impact and are more likely to breach slippage limits; they are also more exposed to liquidity fragmentation in AMMs like Uniswap v3, where large trades may span multiple ticks, any of which could be insufficiently funded or prone to revert. Additionally, large transactions consume more gas, increasing their vulnerability to eviction or delay in latency-sensitive environments.

The data confirms widespread trade slicing, but it remains an open question whether this slicing follows the optimal threshold q^* derived in Proposition 1. Future research could reverse-engineer trade sizes to estimate the implicit $p(q)$ function and compare observed chunking with theoretical optima.

Moreover, we observe that some addresses repeatedly submit the same transaction multiple times without variation in size, slippage, or timing—suggesting naive or scripted strategies rather than adaptive optimization. These transactions are more likely to fail, potentially incurring unnecessary gas costs and reducing arbitrage profitability.

The Dencun Effect. The Dencun upgrade introduced a significant shift in L2 gas economics. Pre-upgrade, the relatively high cost (often \$0.5–\$1 per transaction) disincentivized failed spam-based arbitrage. Post-Dencun, with average gas costs dropping below \$0.01, the marginal cost of failure became negligible. This cost asymmetry makes spam-based arbitrage economically rational post-upgrade and explains the observed spike in revert rates.

Importantly, the low gas regime reintroduces a Tragedy of the Commons dynamic: when all actors spam simultaneously to secure MEV, the revert rate increases system-wide, congesting sequencers and degrading execution quality. This suggests the need for economic mechanisms that internalize the externalities of spam.

Incentive Design and New Sequencing Mechanisms. The current status quo—FCFS ordering on centralized, private mempools—favors low-latency spamming and limits the scope for efficient, market-based prioritization. Several mechanisms have been proposed to address this:

- **TimeBoost**, adopted on Arbitrum, adds a temporal dimension to transaction bidding but lacks widespread adoption and has yet to demonstrate superior MEV allocation empirically.
- **Revert Protection**, as tested on Unichain, radically reduces transaction spam by disincentivizing loss-making MEV strategies. Our data shows near-zero reverts during its activation period, aligning with Zhu et al.'s theoretical model that revert protection enhances both revenue and efficiency [29].
- **MEV Tax** introduces a protocol-native method to auction execution priority and redistribute proceeds. However, it requires adoption at the DEX level, introducing coordination and governance challenges [21].

These emerging mechanisms represent a shift from *execution-based* to *market-based* MEV allocation. Instead of racing to be first (via spam), actors compete economically (via bids or auctions). Yet, as of today, adoption remains partial and fragmented. Arbitrum's TimeBoost is opt-in; Unichain's revert protection is under testing; and MEV Tax remains a proposal. Without broad adoption, spam-based extraction remains profitable.

Underreliance on Priority Fee Auctions. A notable empirical observation is the limited use of priority fee auctions (PFA) by MEV searchers on rollups. Despite the theoretical promise of PFAs to prioritize transaction inclusion, we find that searchers often prefer to duplicate their transactions—submitting the same swap multiple times—rather than bid for priority.

This raises a natural question: why are PFAs underutilized in practice? One explanation lies in the nature of fast-finality blockchains. On rollups with block times below one second, even small differences in message propagation can cause a non-priority-fee transaction to reach the sequencer before a higher-fee transaction. In such cases, the priority fee has no effect, and the MEV opportunity may be captured by the earlier-arriving, lower-fee transaction.

This renders the PFA mechanism unreliable in practice, especially in environments with high competition and minimal latency margins. Similar observations have been made and formalized on Solana. Solana's fee mechanism allows users to set a "priority fee" to improve their chances of transaction inclusion. However, combined with a static base fee, the mechanism essentially functions as a First-Price Auction (FPA), which is known to be not dominant-strategy incentive compatible (DSIC), meaning that paying a higher fee does not guarantee inclusion [22]. Our findings suggest that a comparable inefficiency may be present on fast-finality blockchains, explaining the rational preference for transaction duplication over bidding.

7 Conclusions

This research investigates the economics and execution patterns of spam-based arbitrage strategies on fast-finality Ethereum rollups, where transaction ordering follows a first-come, first-served (FCFS) policy and sequencers operate private mempools.

We begin with a theoretical model showing that CEX-DEX arbitrageurs can maximize expected profits by splitting large MEV opportunities into multiple smaller transactions. This result emerges under realistic assumptions of increasing swap failure probability with trade size and fixed per-swap overhead. We formally derive the optimal chunk size and prove that such trade-splitting dominates single-shot execution on fast-finality blockchains.

To validate this model empirically, we analyze reverted transactions on five major rollups—Arbitrum, BASE, Optimism, Unichain, and ZKsync—following the March 2024 Dencun upgrade. Our dataset, extracted from full archive nodes, spans both successful and reverted swaps. We observe a consistent rise in revert rates across L2s, from 5% to 10–20%, coinciding with the dramatic drop in transaction fees due to blob-based data availability. This cost reduction made spam-based MEV strategies economically viable.

Using trace-level analysis, we construct execution graphs for each reverted transaction and systematically identify their structure. Our graph-based search reveals that over 80% of reverts are swap transactions, and approximately 50% of these target USDC-WETH pools on Uniswap v3/v4—suggesting a high concentration of MEV activity on highly liquid token pairs.

We further highlight a set of counterintuitive findings. Despite the implementation of Priority Fee Auctions (PFAs) across rollups, MEV bots rarely use priority fees. Instead, they duplicate transactions, relying on sheer speed rather than economic bidding, likely because even small propagation delays can prevent fee-based prioritization from working effectively under sub-second finality. Moreover, on fast rollups like Arbitrum and ZKsync, reverted transactions cluster at the very beginning of blocks, indicating that MEV extraction has become a low-latency race rather than a fee-based competition.

In sum, this study offers both theoretical and empirical evidence that spam-based MEV is a rational and prevalent strategy on today's rollups. Without protocol-level changes to transaction ordering, pricing, or revert handling, these dynamics are likely to persist—and may worsen—as L2 fees continue to decline.

Acknowledgments

The first author gratefully acknowledges support from the Flashbots Research Grant. The authors would also like to thank all collaborators from Flashbots and other institutions involved in the discussions that shaped this research.

References

1. Adams, H., Toda, M., Karys, A., Wan, X., Gretzke, D., Zhong, E., Wong, Z., Marzec, D., Miller, R., Hasu, Floersch, K., Robinson, D.: Unichain (2024)
2. Chaliasos, S., Firsov, D., Livshits, B.: Towards a Formal Foundation for Blockchain Rollups. arXiv preprint arXiv:2406.16219 (2024)
3. Chaliasos, S., Reif, I., Torralba-Agell, A., Ernstberger, J., Kattis, A., Livshits, B.: Analyzing and Benchmarking ZK-Rollups (2024)
4. ConduitXYZ: Rollup Gas per Second (GPS) (2024), https://rollup.wtf/, accessed on February 7, 2025
5. Ethereum Foundation: Ethereum Roadmap (2024), https://ethereum.org/en/roadmap/, accessed on February 7, 2025
6. Ethereum Foundation: Proto-Danksharding FAQ. https://notes.ethereum.org/@vbuterin/proto_danksharding_faq (2024), accessed on Febraury 7, 2025
7. Flashbots: Flashbots – Multichain Transaction Reverts. https://dune.com/flashbots/multichain-transaction-reverts (2024), accessed on February 7, 2025
8. Fritsch, R., Silva, M.I., Mamageishvili, A., Livshits, B., Felten, E.W.: Mev capture through time-advantaged arbitrage (2025), https://arxiv.org/abs/2410.10797
9. Gangwal, A., Gangavalli, H.R., Thirupathi, A.: A Survey of Layer-Two Blockchain Protocols. Journal of Network and Computer Applications, Vol 209 (2022)
10. Gogol, K., Messias, J., Miori, D., Tessone, C., Livshits, B.: Cross-Rollup MEV: Non-Atomic Arbitrage Across L2 Blockchains. arXiv preprint arXiv:2406.02172 (2024)
11. Gogol, K., Messias, J., Miori, D., Tessone, C., Livshits, B.: Quantifying Arbitrage in Automated Market Makers: An Empirical Study of Ethereum ZK Rollups. In: The 5th International Conference on Mathematical Research for Blockchain Economy (MARBLE) (2024)
12. Growthepie: Layer-2 Transaction Costs. https://www.growthepie.xyz/fundamentals/transaction-costs (2024), accessed on February 7, 2025
13. Heimbach, L., Pahari, V., Schertenleib, E.: Non-Atomic Arbitrage in Decentralized Finance. arXiv preprint arXiv:2401.01622 (2024)

14. Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S.M., Felten, E.W.: Arbitrum: scalable, private smart contracts. In: Proceedings of the 27th USENIX Conference on Security Symposium (2018)
15. L2Beat: Scaling Actitivy (2024), `https://l2beat.com/scaling/activity`, accessed on February 7, 2025
16. Lashkari, B., Musilek, P.: A Comprehensive Review of Blockchain Consensus Mechanisms. IEEE Access (2021)
17. Mamageishvili, A., Kelkar, M., Schlegel, J.C., Felten, E.W.: Buying time: Latency racing vs. bidding in transaction ordering (2023), `https://arxiv.org/abs/2306.02179`
18. Monte, G.D., Pennino, D., Pizzonia, M.: Scaling blockchains without giving up decentralization and security. In: Proceedings of the Workshop on Cryptocurrencies and Blockchains for Distributed Systems (2020)
19. Motepalli, S., Freitas, L., Livshits, B.: Sok: Decentralized sequencers for rollups (2023), `https://arxiv.org/abs/2310.03616`
20. Motepalli, S., Freitas, L., Livshits, B.: SoK: Decentralized Sequencers for Rollups. arXiv preprint arXiv:2310.03616 (2023)
21. Robinson, D., White, D.: Priority Is All You Need. Paradigm Reseach (2024)
22. Roughgarden, T.: Transaction Fee Mechanism Design (2023), `https://arxiv.org/abs/2106.01340`
23. Sguanci, C., Spatafora, R., Vergani, A.M.: Layer 2 blockchain scaling: a survey. arXiv preprint arXiv:2107.10881 (2021)
24. Thibault, L.T., Sarry, T., Hafid, A.S.: Blockchain Scaling Using Rollups: A Comprehensive Survey. IEEE Access, Vol. 10 (2022)
25. Thibault, L.T., Sarry, T., Hafid, A.S.: Blockchain Scaling Using Rollups: A Comprehensive Survey. IEEE Access (2022)
26. Torres, C.F., Mamuti, A., Weintraub, B., Nita-Rotaru, C., Shinde, S.: Rolling in the Shadows: Analyzing the Extraction of MEV Across Layer-2 Rollups. arXiv preprint arXiv:2405.00138 (2024)
27. Wang, G., Shi, Z.J., Nixon, M., Han, S.: SoK: Sharding on Blockchain. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies (2019)
28. Wang, W., Zhou, L., Yaish, A., Zhang, F., Fisch, B., Livshits, B.: Mechanism Design for ZK-Rollup Prover Markets. arXiv preprint arXiv:2404.06495 (2024)
29. Zhu, B.Z., Wan, X., Moallemi, C.C., Robinson, D., Bachu, B.: Quantifying the Value of Revert Protection. arXiv prepritnt arXiv:2410.19106 (2024)
30. Öz, B., Torres, C.F., Gebele, J., Rezabek, F., Mazorra, B., Matthes, F.: Pandora's Box: Cross-Chain Arbitrages in the Realm of Blockchain Interoperability. arXiv preprint arXiv:2501.17335 (2025)

A Gas Fees of Reverted Transactions

This section provides a detailed breakdown of the gas fees incurred by the reverted transactions. Although reverted transactions do not alter the blockchain state, they are still included in the ledger and consume resources during execution. In general, on Ethereum rollups, the total fee paid by a user comprises two main components:

- **L2 Execution Fee.** The cost of executing the transaction on the rollup's virtual machine.

- **L1 Data Availability Fee.** The cost of posting transaction calldata to Ethereum L1, which ensures data availability and security.

The L2 execution cost itself can be further decomposed into a *base fee* and a *priority fee*, leading to the following structure:

$$\text{Gas Fee on L2} = \underbrace{\underbrace{\text{Base Fee} + \text{Priority Fee}}_{\text{L2 Execution Fee}} + \text{L1 Data Fee}}_{\text{Gas Fee}}$$

These components can be computed from on-chain data using the following formulas:

$$\begin{aligned}\text{Execution Fee} &= \texttt{gas\_price} \times \texttt{gas\_used}\\ \text{Priority Fee} &= \texttt{priority\_fee\_per\_gas} \times \texttt{gas\_used}\\ \text{Base Fee} &= \max(\texttt{gas\_price} - \texttt{priority\_fee\_per\_gas}, 0) \times \texttt{gas\_used}\end{aligned}$$

Figure 7 presents the empirical distribution of gas fees associated with reverted transactions on analyzed L2s. We analyze how these fees evolve over time, and how the relative share of base, priority, and calldata-related costs differ (L1 fees are presented for ZKsync).

A key observation is that the priority fee accounts for a significant portion of the total fees incurred by reverted transactions. These priority fees represent direct revenue for the (centralized) sequencer operating a rollup.

Another notable observation is the sharp drop in the L1 fee component at the beginning of May 2025. This change is attributed to the Ethereum network's Pectra upgrade, which occurred on May 7th 2025 and improved blob management for rollups.

Fig. 7: Daily total gas fees incurred by reverted transactions. The breakdown shows the L2 execution fee (split into base and priority components) and the L1 fee for calldata posting to Ethereum.