


Optimistic MEV in Ethereum Layer 2s: Why Blockspace Is Always in Demand

Ozan Solmaz ✉

ETH Zurich

Lioba Heimbach ✉ 

ETH Zurich

Yann Vonlanthen ✉ 

ETH Zurich

Roger Wattenhofer ✉ 

ETH Zurich

Abstract

Layer 2 rollups are rapidly absorbing DeFi activity, securing over \$40 billion and accounting for nearly half of Ethereum’s DEX volume by Q1 2025, yet their MEV dynamics remain understudied. We address this gap by defining and quantifying optimistic MEV, a form of speculative, on-chain MEV whose detection and execution logic reside largely on-chain in smart contracts. As a result of their speculative nature and lack of off-chain opportunity verification, optimistic MEV transactions frequently decide not to execute any trades.

In this work, we focus on cyclic arbitrage, which we find is predominantly executed as optimistic MEV on Layer 2s. Using our multi-stage identification pipeline on Arbitrum, Base, and Optimism, we show that in Q1 2025, transactions from cyclic arbitrage contracts account for over 50% of on-chain gas on Base and Optimism and 7% on Arbitrum, driven mainly by “interaction” probes (on-chain computations searching for arbitrage). This speculative probing indicates that cyclic arbitrage on Layer 2s is predominantly executed as optimistic MEV and contributes to generally keeping blocks on Base and Optimism persistently full. Despite consuming over half of on-chain gas, these optimistic MEV transactions pay less than one quarter of total gas fees. Cross-network comparison reveals divergent success rates, differing patterns of code reuse, and sensitivity to varying sequencer ordering and block production times. Finally, OLS regressions link optimistic MEV trade count to ETH volatility, retail trading activity, and DEX aggregator usage. Together, these findings show that optimistic MEV has become a major source of persistent spam-like transaction activity on Layer 2s, dominating blockspace with low-value probes and reshaping the composition of on-chain activity.

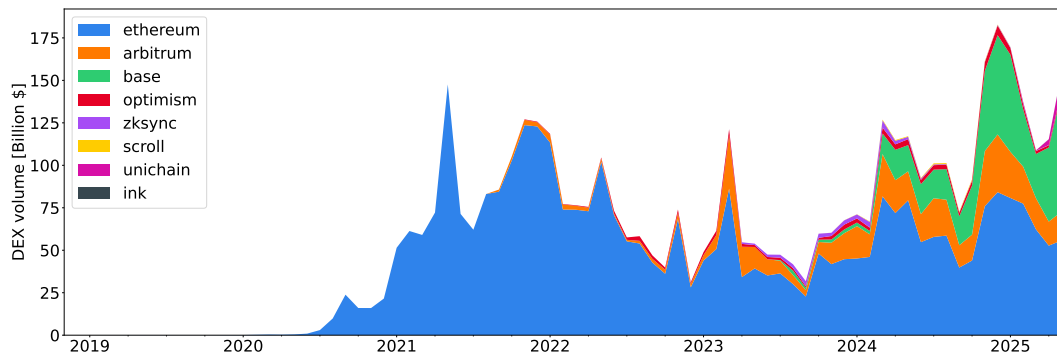
2012 ACM Subject Classification Computer systems organization → Distributed architectures; Applied computing → Electronic commerce; Networks → Network measurement

Keywords and phrases blockchain, MEV, Layer 2, Ethereum

Acknowledgements We thank Burak Öz, Vabuk Pahari and the anonymous AFT reviewers for their helpful feedback. Our gratitude extends to Patrick Züst for his exploratory work in his Master thesis supervised by the last three authors and Quentin Knip, which helped shape this work.

1 Introduction

Ethereum, a decentralized and programmable blockchain, features robust smart contract functionality that enables trust-minimized applications and value transfer without reliance on traditional intermediaries. This programmability has cultivated a dynamic ecosystem of decentralized applications, particularly within the sphere of *decentralized finance (DeFi)*, which includes *decentralized exchanges (DEXes)* and *lending protocols*. As DeFi and other on-chain activities have grown in scale and value, they have contributed to significant network



■ **Figure 1** Monthly DEX transaction volume by network from November 2018 to May 2025. The plot highlights the significant rise of Layer 2 networks — such as Arbitrum, Base, Optimism, and others — in total DEX activity, with Ethereum Layer 1 gradually representing a smaller share of the overall volume.

congestion on Ethereum Layer 1, driving transaction fees to levels that render smaller-scale operations economically unviable.

To address these critical scaling limitations, the Ethereum community has strategically embraced a rollup-centric development roadmap [20, 65, 47]. Layer 2 rollups scale Ethereum by moving transaction execution off-chain and periodically anchoring summarized results on-chain. This design preserves security while enabling higher throughput and lower costs [44, 50]. Adoption has been significant: as of April 2025, Layer 2 networks secure around \$40 B in assets and account for a growing share of on-chain transaction volume, highlighting their central role in the Ethereum ecosystem.

As illustrated in Figure 1, a substantial share of DeFi activity now occurs on Layer 2 networks. To be precise, in the first quarter of 2025, roughly 47% of DEX volume in the Ethereum ecosystem was happening on Layer 2 networks. This figure is up from 35% in the first quarter of 2024. This migration of DeFi activity to Layer 2 networks has concomitantly fostered a distinct *Maximal Extractable Value (MEV)* landscape.

The shift in the MEV landscape has been accelerated by the advent of Layer 2 rollups and upgrades such as EIP-4844 Proto-Danksharding [37, 38], which have significantly reduced data availability costs for Layer 2 networks and, in turn, lowered transaction fees on these networks. These lower costs have allowed for MEV extraction strategies that are less viable on the Ethereum mainnet. Beyond lower fees, Layer 2s introduce operational characteristics that further enable MEV strategies previously infeasible on Layer 1. These include differing transaction ordering policies implemented by sequencers — ranging from simple *First-Come, First-Served (FCFS)* to auction-based mechanisms such as *Priority Gas Auctions (PGA)* — as well as variations in mempool privacy, and short inter-block times.

The confluence of these features reduces the financial risk and increases the execution uncertainty of MEV-seeking transactions. As a result, cyclic arbitrage bots can adopt an optimistic approach: they issue high-frequency, speculative transactions without knowing in advance whether an arbitrage opportunity exists, relying instead on rapid on-chain state reads and being sufficiently close in time to the opportunity-creating transaction to back-run it successfully. This differs from traditional cyclic arbitrage on Ethereum Layer 1, where generally bots precompute a guaranteed profit path and bundle all trades into a single cyclic transaction. On Layer 2, optimistic strategies involve repeatedly probing liquidity pools for small gains, despite not knowing if an opportunity exists, and accepting a high failure rate in exchange for marginal profits.

We refer to this class of strategies that rely on speculative execution, where MEV bots submit transactions dependent on on-chain computation to identify profitable opportunities, often resulting in failed attempts, as *optimistic MEV*. To the best of our knowledge, this is the first work to formalize and systematically study optimistic MEV on Layer 2s.

Our contributions. We summarize our main contributions below.

- i. We provide the first definition of *optimistic MEV*, describing it as a class of MEV extraction strategies that rely heavily on on-chain logic for opportunity identification and speculative execution.
- ii. We design and implement a multi-stage pipeline to identify and classify MEV transactions. This includes the construction of a cyclic arbitrage detector, an on-chain behavior classifier, and a revert/success labeling system to capture speculative dynamics.
- iii. We apply our methodology to present the first large-scale measurement of optimistic MEV on major Layer 2s, showing that cyclic arbitrage accounts for roughly 7% of gas usage on Arbitrum, 51% on Base, and 55% on Optimism in Q1 2025 and is predominantly optimistic in nature. In particular, our analysis reveals that a large share of this gas is consumed by transactions that fail to execute profitable trades, supporting the hypothesis that these bots engage in speculative on-chain probing.
- iv. We compare optimistic MEV activity across Arbitrum, Base, and Optimism and find substantial differences in strategy execution patterns, success rates, and gas usage concentration. We attribute these to network-specific factors such as transaction ordering policies and inter-block times.
- v. Using regression analysis, we examine how optimistic MEV activity correlates with market conditions and user trade behavior. Our findings suggest that volatility, trade volume, and aggregator usage significantly impact optimistic MEV prevalence.
- vi. We discuss the broader implications of optimistic MEV for the MEV ecosystem and Layer 2 design, highlighting its economic impact and the challenges it poses for existing mitigation approaches.

2 Related Work

Maximal Extractable Value. The study of MEV was commenced by Eskandari et al. [36] and Daian et al. [26], who first defined MEV and documented phenomena like front-running and PGAs on Ethereum. In subsequent work, Qin et al. [67] and Torres et al. [74], quantified various forms of MEV on Ethereum mainnet, including sandwich attacks (i.e., a type of front-running attack on DEXes), liquidations of positions on lending protocols, and cyclic arbitrage, providing a baseline understanding of its scale and impact. A subsequent line of work analyzes various and evolving aspects of the MEV landscape on the Ethereum Layer 1 [83, 82, 74, 80, 64, 68, 22, 69, 57, 81, 46, 85, 48]. While these works have established a foundation for understanding MEV on Ethereum Layer 1, the Layer 2 landscape remains comparatively underexplored. Our work builds on this foundation by focusing on the unique characteristics of Layer 2s, providing a classification and analysis of MEV transactions in these emerging environments.

A closely related line of work examines MEV in alternative Layer 1 networks, which operate under distinct architectural and economic conditions. Öz et al. [84] analyze MEV on FCFS blockchains such as Algorand, identifying latency optimization — rather than fee bidding — as the dominant extraction mechanism. Further, work done by Umbra

Research [76] has provided valuable insights into the MEV on Solana. They point to the existence of optimistic MEV on Solana but provide no in-depth analysis of the phenomenon. In contrast, our work focuses on Layer 2 networks within the Ethereum ecosystem, where optimistic MEV is both prevalent and, to the best of our knowledge, has not been investigated in depth.

Maximal Extractable Value in Layer 2 Networks. Recent efforts have begun to examine MEV on Layer 2 networks. Torres et al. [75] conducted a comparative analysis of MEV across Ethereum and major Layer 2s (Arbitrum, Optimism, zkSync). Theoretical work has also addressed cross-domain MEV, with Obadia et al. [59] formalizing MEV across domains such as different Layer 2s. Complementing this, Gogol et al. [43] empirically analyzed non-atomic cross-rollup MEV and CEX-DEX arbitrage. Öz et al. [62] present the first systematic study of non-atomic cross-chain arbitrage strategies across nine blockchains, including several Layer 2s. Their work highlights the effects of these strategies on network congestion and the security implications of cross-chain MEV. Our work extends these Layer 2-focused studies by analyzing optimistic MEV, a form of MEV that involves on-chain detection of opportunities.

Decentralized Exchanges. Theoretical frameworks for routing on DEXes and profits of liquidity providers have provided important context for understanding MEV behavior. Angeris et al. [10] analyzed optimal routing and arbitrage in CFMMs. Milionis et al. [54, 55] modeled loss-versus-rebalancing (LVR) under Poisson-distributed interblock times; and Nezlobin et al. [58] extended this to deterministic block intervals, particularly relevant given the regular block production in many blockchain networks. Building on these theoretical foundations, our work provides empirical insights into gas usage, MEV bot behavior, and protocol-level design factors across Layer 2 networks, focusing on optimistic MEV: a speculative, high-frequency form of MEV.

3 Optimistic MEV

Next, we introduce optimistic MEV: a class of extraction strategies that (1) perform opportunity discovery and trade execution primarily through on-chain computation and (2) rely on speculative transaction submission without prior off-chain verification of profitability.

This behavior is most evident in *optimistic cyclic arbitrage*, the focus of this work. The MEV types under consideration are sandwich attacks, cyclic arbitrage (arbitrage between DEXes on the same chain), non-atomic arbitrage (e.g., CEX-DEX arbitrage), and liquidations of positions on lending protocols. Of these, cyclic arbitrage offers a practically viable case where opportunities can be identified and executed entirely on-chain in a single transaction through frequent probing. While liquidations are in principle detectable on-chain, they occur only when loans become undercollateralized and are likely too infrequent to support high-frequency probing [67]. Sandwich attacks require two separate transactions, and non-atomic arbitrage depends on external price data from off-chain venues. For this reason, we focus our analysis of optimistic MEV on cyclic arbitrage, as this MEV strategy can be detected and executed fully on-chain.

Empirically, we observe this form of MEV most prominently on low-fee Layer 2 networks. Its defining signature is speculative execution: bots submit transactions without certainty that an arbitrage opportunity exists, using cheap blockspace to make such probing economically viable. In our analysis, these transactions frequently begin with top-level `STATICCALL` operations directed at DEX liquidity pools to read real-time reserves and prices. The contract

then decides whether to execute a swap or terminate without trading based on these on-chain checks. This behavior, combined with the common use of minimal or repeated `calldata`, indicates that key parameters, i.e., the path of the arbitrage transaction and the swap amounts, are determined during execution rather than precomputed off-chain. The reliance on such runtime inspection also explains why optimistic MEV becomes infeasible when transaction fees are high: the preparatory on-chain calls consume significant gas. These observations motivate the following definition of optimistic MEV.

► **Definition 1 (Optimistic MEV).** *Optimistic MEV denotes MEV strategies in which the existence and parameters of a profitable opportunity are determined within the transaction itself through on-chain computation at execution time, rather than being verified off-chain prior to submission. This leads to speculative transactions where profitability is only resolved during execution, resulting in a significant share of transactions that terminate without performing a trade.*

Case Study. To concretely illustrate the observed Layer 2 MEV patterns, particularly the prevalent use of initial `STATICCALL` operations for on-chain reconnaissance, we present a comparative case study of two transactions interacting with the same contract `0xf5ff...a8c0`¹: `0x1d97...18cb` (henceforth Tx_A) and `0xb678...9bd7` (henceforth Tx_B) on Base. Both transactions were initiated with the same function selector, namely `0x00003748`, as evidenced by the identical first 4 bytes in the `calldata`. This common entry point suggests a shared execution pathway, likely designed for conditional arbitrage based on real-time market conditions. Despite this identical initiation, the on-chain outcomes of Tx_A and Tx_B diverged significantly. Tx_A resulted in no token swaps. In stark contrast, Tx_B successfully executed a 2-swap cyclic arbitrage, cycling value through the path $WETH \rightarrow TYBG \rightarrow WETH$ and realizing a profit.

```
{
  "calls": [
    {"input": "slot0()", "to": "UNIV3 ETH-USDC Pool", "type": "STATICCALL" },
    {"input": "slot0()", "to": "UNIV3 PLAY-USDC Pool", "type": "STATICCALL" },
    {"input": "slot0()", "to": "CL ETH-PLAY Pool", "type": "STATICCALL" }
    // ... further calls omitted for brevity
  ],
  "from": "0xdd57...a88",
  "to": "0xf5ff...8c0",
  "type": "CALL"
}
```

■ **Example 1** Condensed initial call sequence from Tx_A , illustrating `STATICCALL` probes to DEX pools. Input `slot0()` retrieves key state variables from Uniswap V3-compatible pools.

A detailed analysis of the execution trace for Tx_A reveals that the contract systematically probed multiple DEX liquidity pools via `STATICCALL` operations. These read-only calls occurred before any attempt to execute a trade, and the transaction subsequently terminated. This sequence of actions is strongly indicative of an on-chain reconnaissance strategy, where the contract assesses current pool states (e.g., token reserves, potential slippage, pool's fee structure, or tick data) to determine the viability of an arbitrage opportunity before

¹ We marked this contract as one of the top gas consuming cyclic arbitrage contracts on Base, which we will discuss in more detail in the later sections.

committing to an execution path. A condensed representation of the initial call sequence within Tx_A is provided in Example 1.

4 Data Classification

To study optimistic MEV, we start by collecting relevant data (see Appendix C) from the Ethereum Layer 1 as well as Arbitrum, Base, and Optimism (i.e., the biggest three Layer 2s in terms of DEX volume at the time of this writing). Then, we systematically identify cyclic arbitrage MEV activities on Layer 2 networks using a three-stage classification methodology. Recall that we focus on optimistic cyclic arbitrage in our work, as cyclic arbitrage opportunity detection can be fully resolved on-chain in a single transaction. In Section 5, we will then show that cyclic arbitrage on Layer 2 networks predominantly operates optimistically.

The implementation of our cyclic arbitrage classification pipeline, as well as the classification output (the full list of cyclic arbitrage contracts) are made openly accessible [63].

4.1 Cyclic Arbitrage Contract Detection

Building on the cyclic arbitrage detection heuristic of Wang et al. [80] and the toolkits behind EigenPhi and libmev [33, 52], we implement a cyclic arbitrage contract detection pipeline in Dune SQL, leveraging Dune Analytics’ data tables and the Torino engine. Inspired by the replay logic of mev-inspect-py [39] and Brontes [71] we adapt our methodology to the Layer 2 circumstances.

We start by computing the set of transactions that, with high likelihood, perform cyclic arbitrage:

- 1. Token Path & Balance Reconstruction.** We consider all transactions containing swap events by querying the swap logs from `dex.trades` [32]. Each transaction’s swap events are ordered to recover the exact token path and the initiator’s net token balance.
- 2. Router & Aggregator Filter.** Transactions that directly interact with labeled routers (captured through the metadata of the `dex.addresses` [28] table) or aggregators (captured through the `dex_aggregator.trades` [29] table) are dropped.
- 3. Cyclic Arbitrage & Profit Filter.** A transaction is kept only if it forms a sequence of at least two swaps, such that (i) the token bought in the j -th swap is the same as the token sold in the $(j + 1)$ -th swap, (ii) the sequence begins and ends with the same token and (iii) yields a strictly non-negative balance change in every token, with at least one positive gain.

The first callee in any such profitable, cyclic arbitrage transaction is tagged as a *candidate contract*, indicating it likely belongs to a cyclic arbitrage bot. A more detailed and formal description of this classification process can be found in Appendix D.

Next, we inspect the candidate contracts once again to avoid misclassifications. The few misclassified contracts we find during this stage are mainly due to popular mislabeled routers or public utility contracts. The details of this step can be found in Appendix E. Finally, the remaining contracts form our final set of cyclic arbitrage contracts \mathcal{C}_{bot} .

4.2 Transaction-Level Classification

We classify each transaction in our study window (obtained through the respective table `{{chain}}.transactions` [31]) according to three dimensions.

Transaction purpose (cyclicArb / other). The transaction is marked `CYCLICARB` when its first callee (the address in the `to` field) is a member of the validated set of cyclic arbitrage contracts \mathcal{C}_{bot} ; otherwise, it is marked `OTHER`. This choice ties the label to the entity that originates the on-chain action rather than to the presence of any particular pattern inside the trace.

DEX involvement (trade / interaction / residual). We scan swaps in `dex.trades` and consider the full call trace (`{{chain}}.traces`). If the transaction itself emits at least one swap event (e.g. `Swap` in Uniswap V2/V3 [7] or `TokenExchange` in Curve [3]) as a result of function calls such as Uniswap V2’s `swapExactTokensForTokens`, Uniswap V3 pool’s `swap` method [7], or a Curve pool’s `exchange` function [3]), we label it `TRADE`. When no swap is emitted but the trace touches a recognized DEX pool contract, typically through read-only calls such as `getReserves` or `slot0`, the label assigned is `INTERACTION`. (We use the table `dex.raw_pools` to detect the existence of subcalls interacting with DEX-related contracts.) All remaining transactions, which never enter a DEX contract, are labelled `RESIDUAL`. The three options are mutually exclusive and collectively exhaustive.

Execution outcome (success / revert). Finally, this dimension records the final on-chain status of the transaction, based on its receipt. The transaction is marked as `SUCCESS` when execution was successful (`status=1`) and `REVERT` otherwise (`status=0`). A transaction classified as a `TRADE` cannot subsequently be classified as a `REVERT`, as only transactions that do not revert can perform trades.

These three categories give each transaction a concise profile, which we use to group transactions for the analysis that follows. For example, we write `CYCLICARB - INTERACTION` to mean the subset of transactions that invoke a cyclic arbitrage contract as the first callee and also probe a DEX contract.² `OTHER - REVERT` defines the set of all reverted transactions that have a callee outside of the set \mathcal{C}_{bot} . Comparing such sets of transactions over time and across roll-ups enables us to quantify the prevalence of optimistic MEV, success rates, and failure modes under different Layer 2 design choices, such as private mempools, sequencer ordering, low fees, and sub-second blocks.

5 Optimistic MEV Landscape

We commence our analysis of optimistic MEV by providing a broad overview of the landscape. Our analysis focuses on the period from August 2023 to May 2025 and looks at three Layer 2 networks (i.e., Arbitrum, Base, and Optimism) in comparison to the Ethereum Layer 1.

5.1 Execution Outcomes and Success Rates

Recall that the data classification identified cyclic arbitrage contracts. The following analysis shows that on Layer 2 networks, these contracts operate predominantly in an optimistic manner, exhibiting the speculative on-chain behavior characteristic of optimistic MEV, in contrast to their counterparts on Ethereum Layer 1.

Figure 2 illustrates the relative share of transactions from identified cyclic arbitrage contracts that execute swaps (red) compared to those that only probe DEX pools and termin-

² One thing to note is that `CYCLICARB - INTERACTION` contains transactions from `CYCLICARB - INTERACTION - SUCCESS` and `CYCLICARB - INTERACTION - REVERT`.

Optimistic MEV in Ethereum Layer 2s: Why Blockspace Is Always in Demand

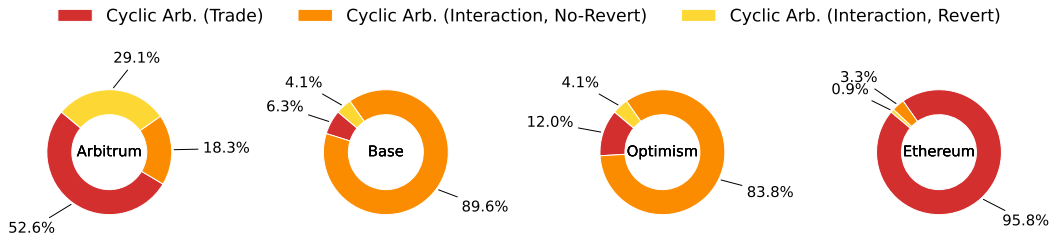


Figure 2 Distribution of cyclic arbitrage transaction outcomes across networks. Red indicates transactions from cyclic arbitrage contracts that execute trades, while the yellow and orange segments represent transactions that likely attempt arbitrage but do not execute any trade, either terminating without swaps (orange) or reverting entirely (yellow).

ate without a trade (orange) or revert (yellow), considering only the subset of transactions that interact with a DEX. The difference between networks is pronounced. On Base and Optimism, only 6.3% and 12% of such transactions, respectively, result in a successful trade, whereas on Arbitrum the share rises to 52.6% and on Ethereum Layer 1 to 95.8%. This distribution highlights a fundamental contrast: on Layer 1, cyclic arbitrage transactions appear largely precomputed and succeed overwhelmingly, while on Layer 2s, and particularly on OP-stack rollups, they display a speculative character with most attempts ending without executing a trade. This pattern of profitability being resolved only at execution and the prevalence of transactions that terminate without trading is exactly what defines optimistic MEV.

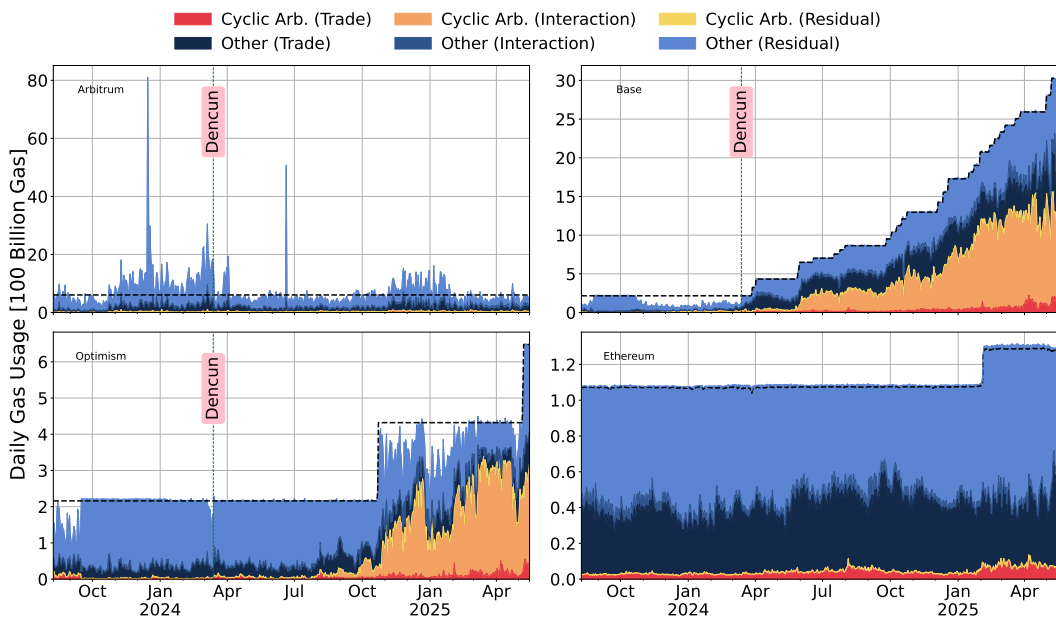
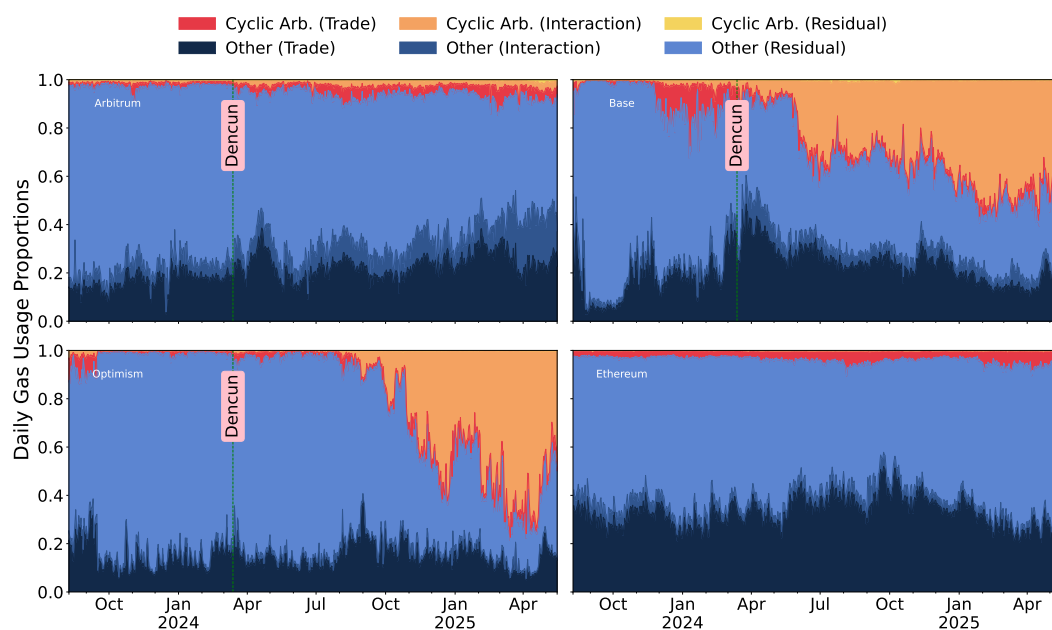


Figure 3 Daily gas usage by transaction category on Arbitrum, Base, Optimism, and Ethereum Layer 1. Each area shows gas consumed by cyclic arbitrage MEV bots (CYCLICARB) and all other activity (OTHER), subdivided into TRADE (transactions with executed swaps), INTERACTION (on-chain probing without swaps), and RESIDUAL segments (see Section 4.2). The vertical green dotted line marks the Dencun upgrade (EIP-4844), while the black dashed line indicates each network’s target gas limit (“Gas Target” on Base, Optimism, and Ethereum; “Speed Limit” on Arbitrum).

5.2 Prevalence Across Networks

We explore these differences further in Figure 3, showing the evolution of daily gas usage on Arbitrum, Base, and Optimism, broken down by activity from addresses performing CYCLICARB MEV and all OTHER activity. Each category is further subdivided into TRADE (transactions with executed token swaps), INTERACTION (on-chain probing or contract calls without token transfers), and RESIDUAL activity (see Section 4.2). Recall that the INTERACTION category includes both probing transactions that never execute a swap as well as reverted transactions. Notably, the share of gas consumed by CYCLICARB MEV (yellow, orange, and red), which on Layer 2 networks manifests predominantly as optimistic MEV, is very large on Base and Optimism in particular. The share of gas attributed to CYCLICARB - INTERACTION transactions rises sharply over time, indicating a surge in speculative, non-token-transfer, DEX contract probing activity associated with optimistic MEV. On Base, this category becomes a dominant contributor to gas usage by early 2025, accounting for 48% of total gas in the first quarter of the year. Overall, these optimistic MEV bots performing cyclic arbitrage are responsible for 51% of Base’s gas usage during this period. Further, gas usage from an optimistic MEV bot on Base exceeds the entire block capacity of Ethereum Layer 1 by an order of magnitude. Optimism exhibits a similar trend, though with a slightly delayed onset. By the first quarter of 2025, optimistic MEV bots performing cyclic arbitrage account for 55% of gas usage on Optimism, with CYCLICARB - INTERACTION transactions specifically contributing 52%. In contrast, on Arbitrum and Ethereum Layer 1, CYCLICARB MEV consumes a much smaller fraction of available blockspace.



■ **Figure 4** Relative daily gas usage by transaction category on Arbitrum, Base, Optimism, and Ethereum Layer 1. Each area shows gas consumed by cyclic arbitrage MEV bots (CYCLICARB) and all other activity (OTHER), subdivided into TRADE (transactions with executed swaps), INTERACTION (on-chain probing without swaps), and RESIDUAL segments (see Section 4.2). The vertical green dotted line marks the Dencun upgrade (EIP-4844).

Turning to Figure 4, where we show the relative change in daily gas usage on Arbitrum, Base, Optimism, and Ethereum Layer 1, we focus on the relative usage of the respective

categories. Observe that on Base and Optimism `CYCLICARB - INTERACTION` activity surges dramatically after Dencun, quickly outpacing executed arbitrages and consuming the majority of newly available blockspace in 2025. In contrast to the two other Layer 2s, Arbitrum shows no comparable increase in `CYCLICARB - INTERACTION` gas usage, suggesting differing strategic dynamics or protocol-level constraints (e.g., transaction ordering, fees, or shorter interblock times) that may discourage such speculative behavior. Note that Arbitrum displays the largest proportion of `OTHER - INTERACTION` gas consumption (see Appendix F). This gas usage is driven by multiple contracts, the largest of which is the Uniswap V3 Liquidity Position Manager, but we observe no evidence of cyclic arbitrage from contracts in this category. Finally, the trade share from probing on the Ethereum Layer 1 remains flat, with trades by non-MEV actors dominant.

Importantly, Figures 3 and 4 display the categorization without taking the last factor (see `SUCCESS / REVERT` in Section 4.2) in consideration. Since optimistic MEV searchers often fail to capture arbitrage opportunities, one might expect a high revert rate among their transactions. However, as shown in Appendix B, the gas consumed by reverted optimistic MEV transactions is proportionally smaller on Base and Optimism than their share of total gas usage. Thus, on Base and Optimism failed MEV attempts by `CYCLICARB` bots primarily succeed but do not perform trades (i.e., `CYCLICARB - INTERACTION - SUCCESS`). By contrast, on Arbitrum and Ethereum Layer 1, a substantial share of MEV-related gas goes to reverted transactions, reflecting a higher rate of outright failures (i.e., `CYCLICARB - INTERACTION - REVERT`).

Next, we discuss possible factors driving the distinct behavior of optimistic MEV searchers on OP-Stack rollups versus Arbitrum and Ethereum Layer 1. First, Arbitrum’s FCFS transaction ordering, especially before the Timeboost mechanism,³ turns MEV extraction into a latency race, disincentivizing on-chain speculative probing. For example, with FCFS ordering, an optimistic MEV bot cannot specify a low-priority fee to sit at the end of the block and capture price differences after other transactions execute. Instead, it must carefully time its transaction submission, resulting in less control over its block position and over the number of possible preceding transactions that create exploitable price differences.

Second, CEX-DEX arbitrage profits are known to scale with the square root of the mean interblock time [54, 58]. Given that Arbitrum has shorter block times (approximately 250ms) compared to Base and Optimism (both 2s), the expected profits from CEX-DEX arbitrage are lower on Arbitrum. Conversely, the higher expected profits on Base and Optimism from CEX-DEX arbitrage may also elevate profitability in DEX-DEX arbitrage, particularly for pairs not actively traded on centralized exchanges. Once CEX-DEX arbitrage opportunities are closed, residual price discrepancies can remain between DEX pairs, creating additional cyclic arbitrage opportunities within the same Layer 2. Thus, lower block times might discourage excessive transaction spam.

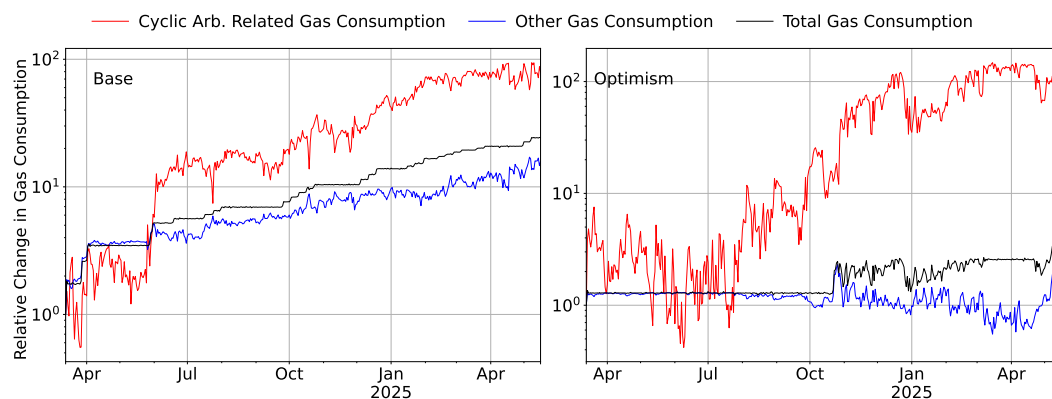
Finally, block capacity and transaction fees also appear to shape the prevalence of optimistic MEV. In Figure 3, we see that immediately following the Dencun hardfork, Base increased its gas target by roughly an order of magnitude (black dashed line), and Optimism similarly raised its gas target. In contrast, Arbitrum’s speed limit (the equivalent throughput cap) remained unchanged throughout our measurement window.

We conclude by highlighting that in contrast to all three Layer 2s, Ethereum Layer 1 shows almost no significant gas usage in the `CYCLICARB - INTERACTION` category. Instead,

³ TimeBoost, introduced on Arbitrum on April 17, 2025, adds a sealed-bid second-price auction “express lane” that allows users to submit transactions directly to the sequencer for prioritized inclusion [13].

gas usage related to DEXes remains dominated by OTHER - TRADE activity. Activities resembling speculative probing, so prominent on Layer 2s, are largely negligible on Layer 1, reinforcing the fundamental distinction in MEV dynamics between the two environments. Overall, these findings support our core premise: optimistic MEV strategies are uniquely fostered by the low-cost, high-throughput conditions of Layer 2 architectures.

5.3 Impact of Dencun on Growth



■ **Figure 5** Relative change in daily gas consumption for three categories — cyclic arbitrage related, other, and total — on Base and Optimism, normalized to 1 on the day of Dencun hardfork (i.e., 13 March 2024). Each curve shows how gas usage in each category evolves post-Dencun, highlighting the divergent growth rates of optimistic MEV activity (CYCLICARB) versus all other on-chain activity.

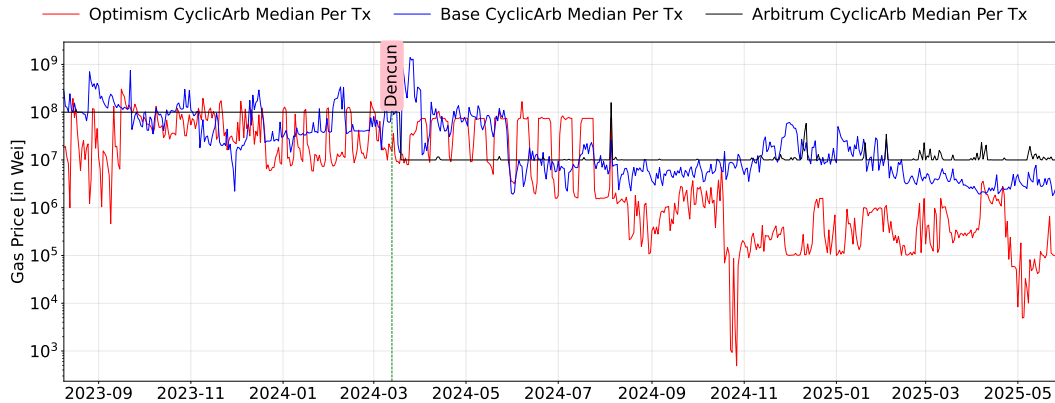
To examine this further, Figure 5 shows the relative change in daily gas consumption for three categories (i.e., CYCLICARB, OTHER, and total activity (CYCLICARB+OTHER)) on Base (left) and Optimism (right), each normalized to 1 on the day of the Dencun upgrade. In both networks, cyclic arbitrage MEV bots exhibit the largest increase, indicating that the additional block capacity is largely consumed by optimistic MEV activity. Notably, on Optimism, there is no consistent rise in gas usage by other transactions despite the higher gas limit. This suggests that the main beneficiary of the additional blockspace is optimistic MEV, with most of the new capacity consumed by on-chain computations that could be done off-chain.

5.4 Fee Dynamics

Turning to transaction fees, Figure 6 plots the daily median gas prices paid by optimistic MEV bots on each Layer 2. Notably, Arbitrum enforces a floor of 0.01 GWei on its base fee when demand falls below its speed limit [12]. In contrast, Base and Optimism follow the EIP-1559 model, allowing fees to decline arbitrarily when demand is under the gas target. This difference explains why Base and Optimism blocks appear consistently full (see Figure 3) in comparison to target utilization, whereas Arbitrum’s blocks often show unused headroom.

Median gas prices paid by optimistic MEV bot transactions (i.e., those labeled as CYCLICARB MEV) starting from Dencun, further underscore these distinctions: Optimism sees the lowest median price (0.0005 GWei), Base sits in the middle (0.0061 GWei), and Arbitrum commands the highest median (0.01 GWei). Thus, the median price paid by optimistic MEV bots on Optimism is a factor of 20 lower than on Arbitrum, while it is nearly

a factor of 2 lower on Base than on Arbitrum.⁴ Lower fees on Base and Optimism reduce the cost of speculative probing, making optimistic MEV more profitable, while Arbitrum’s relatively higher fee floor dampens such behavior. Thus, a final factor driving optimistic MEV is the comparatively low fees on Base and Optimism. These reduced costs make speculative probing more profitable on those networks, whereas Arbitrum’s higher fee floor suppresses such behavior.



■ **Figure 6** Median gas price paid by optimistic MEV bots (i.e., `CYCLICARB` MEV) on Optimism, Base, and Arbitrum. For Base and Optimism, the plot show `gas_price` which includes `base_fee` + `priority_fee` and for Arbitrum, it shows `effective_gas_price`.

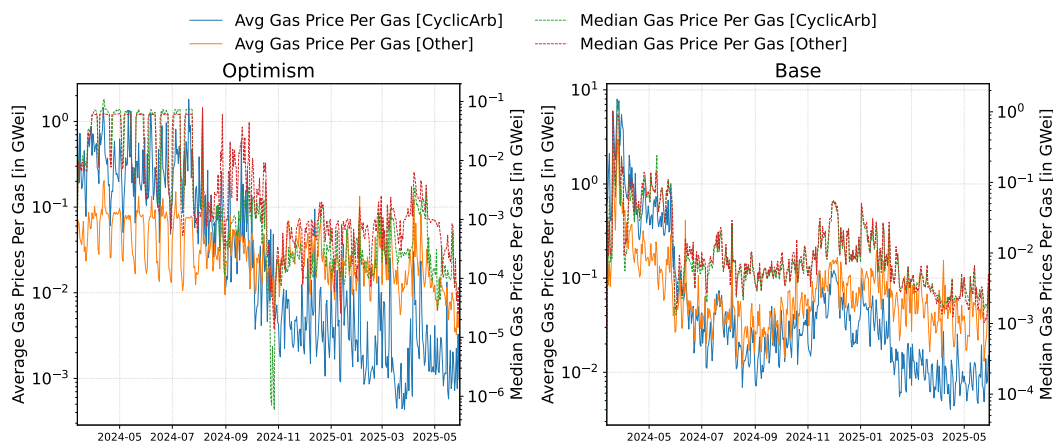
Importantly, even though transactions from optimistic MEV searchers on Base and Optimism account for more than half the gas usage in the first quarter of 2025, they only account for 23% and 17% of the transaction fees paid, respectively. Thus, there is a disconnect between the gas used and the fees paid. Optimistic MEV transactions pay less for their transactions.

Next, we examine this fee discrepancy in more detail in Figure 7, which plots median and mean gas price per unit gas paid by optimistic MEV bots (i.e., those classified as `CYCLICARB`) versus all other transactions on Base and Optimism over time.

On Base, in Q1 2025, the average gas price per gas paid by optimistic MEV bots is 0.0209 GWei, compared with 0.0710 GWei for all other transactions — a factor of 3.5 difference. When we consider the median instead of the mean, the gap narrows: MEV bots pay a median of 0.0047 GWei, while the rest pay 0.0057 GWei. Notably, between April and September 2024, there was a period when MEV bot transactions actually paid higher average fees than other users. This shows that, although optimistic MEV bots tend to pay less overall, the difference is smaller and more nuanced than one might expect.

On Optimism, the pattern is similar. In Q1 2025, the average gas price per gas for MEV bots is 0.0042 GWei, versus 0.0252 GWei for other actors — a factor of 5 difference. Median gas prices are 0.0003 GWei for MEV bots and 0.0007 GWei for the rest. Again, the smaller median gap and occasional fee spikes by MEV bots in mid-2024 confirm that on-chain probing does not always seek the absolute cheapest gas. Nonetheless, the high prevalence of

⁴ One reason the gap between Base and Arbitrum is smaller is that there is a short period where fees on Base are higher than those on Arbitrum. Between late November 2024 and early January 2025, we see the gas price paid by optimistic MEV bots on Base exceeding Arbitrum. The reason for this is the congestion in the network during those periods caused by the sniper bots (see Appendix B).



■ **Figure 7** Gas prices per gas unit from optimistic MEV bots (i.e., CYCLICARB MEV) versus other actors on Base (right) and Optimism (left) starting from the day of Dencun hardfork (i.e., 13 March 2024). Solid lines show the daily average gas price per gas unit paid by CYCLICARB MEV bots (blue) and other actors (orange), while dashed lines show the corresponding median fees. Both average and median gas prices paid per gas unit by CYCLICARB MEV bots are lower than those of others.

optimistic MEV correlates with periods of low gas prices after the Dencun hardfork, as we saw previously.

5.5 Characteristics of the Largest Cyclic Arbitrage Bots

We now turn our attention to the behavior and structure of the most active CYCLICARB bots on each network. Specifically, we analyze the top 10 bots by gas usage on Base, Optimism, and Arbitrum, focusing on execution outcomes, code similarity, and operational architecture.

The statistics relevant to the top 10 bots (based on gas usage) are presented in the following tables for Base (Table 1a), Optimism (Table 1b), and Arbitrum (Table 1c). The primary metrics in the tables are defined as follows:

Swaps The total number of individual swap operations executed as a result of calls to the given MEV bot contract. Notably, a single on-chain transaction may include multiple distinct swap operations.

Transactions With Trades The total number of unique transactions that called the MEV bot contract and included one or more swap operations. It counts transactions rather than individual swaps.

Non-Reverted Transactions The total number of transactions that called the MEV bot contract and were successfully executed and committed on-chain.

Reverted Transactions The total number of transactions that called the MEV bot contract but failed during execution (i.e., were reverted), and therefore did not result in the intended state changes, although they still consumed gas.

Transaction per Unique Calldata The ratio total number of transactions by the bot to the unique calldata entries used, where a value of 10 means that, on average, each calldata entry is reused across 10 transactions.

Median Calldata Length The median length of the calldata used by the bot (in hexadecimal characters).

Cumulative MEV Bot Gas (%) The cumulative percentage of total transaction gas consumed by MEV bot contracts. The percentage is calculated relative to the total gas used

by a defined set of contracts (i.e., all identified MEV bots within the dataset $\mathcal{C}'_{bots,chain}$) and is rounded to two decimal places.

Execution Outcomes and Bot Behavior Across Networks

A principal finding of our analysis concerns the execution outcomes of cyclic arbitrage attempts by MEV bots. Across all three Layer 2 networks, i.e., Base, Optimism, and Arbitrum, we observe that a large proportion of cyclic arbitrage attempts do not result in profitable outcomes as they do not even execute a swap. However, the way unsuccessful attempts are handled differs substantially between networks.

On OP-Stack-based networks (Base and Optimism), many unprofitable MEV transactions still conclude successfully from the perspective of the *Ethereum Virtual Machine (EVM)*; that is, they do not revert (on Optimism revert rate is 0.01%, on Base 0.005%), even though they fail to execute a profitable arbitrage. These transactions typically make no substantive state changes apart from incurring transaction fees. This behavior implies that the primary disincentive for speculative probing on these chains is economic (i.e., fees), not technical failure. Despite their technical success, our analysis shows that the actual success rate (defined as executing an arbitrage) is exceptionally low among the top contracts on Base (0.58%) and Optimism (1.49%). This indicates a high volume of ultimately unsuccessful, speculative execution.

In contrast, Arbitrum displays a different pattern. When transactions initiated by the top cyclic arbitrage bots do not revert, they are overwhelmingly likely to result in a successful arbitrage. There, the success rate of non-reverted transactions is 77%, while the overall success rate (i.e., including those transactions that revert) is 56%. This suggests a more selective or precise execution strategy. However, this high success rate for non-reverted transactions is accompanied by a significantly higher incidence of reverts, relative to Base and Optimism. Thus, Arbitrum-based bots appear to favor failing fast (via reverts) over speculative probing. These findings were further validated through cross-referencing with a curated dataset prepared by Entropy Advisors, an official partner of the Arbitrum Foundation [35, 34], strengthening our confidence in the observed execution dynamics across networks.

Further, our analysis of the largest cyclic arbitrage bots demonstrates again that they predominantly operate as optimistic MEV bots. To demonstrate this, we compute the number of transactions per unique calldata entry. On Base, the largest cyclic arbitrage bot reuses the same calldata more than 100 times on average, while on Optimism, one bot repeats identical calldata over 7 million times. Across both Base and Optimism, all major cyclic arbitrage contracts exhibit extensive calldata reuse, reflecting a speculative approach where little specific information such as exact paths or trade amounts is precomputed off-chain. This is indicative of optimistic MEV. Examining the calldata itself reinforces this: in many cases it is non-informative, as indicated by short median calldata lengths, or consists primarily of lists of DEXes to probe for potential opportunities.

When we turn to Arbitrum, the picture is somewhat different. While the majority of the largest cyclic arbitrage contracts exhibit extensive calldata reuse, several do not. In particular, the third, fourth, and fifth largest contracts show minimal reuse, resembling the pattern observed on Ethereum, where none of the major cyclic arbitrage bots have a transactions per unique calldata ratio higher than 1.006. Despite this, their transactions revert frequently, around 50%, suggesting they are not speculative in the sense of performing computation on-chain but are instead exposed to significant uncertainty about whether their transactions

contract	swaps	txs with trades	non-reverted txs	reverted txs	txs per unique calldata	median calldata length	cum. MEV bot gas (%)
0xf5ff...a8c0	175,301	60,315	23,173,896	1	107.08	3830	6.71
0xbff6...73e9	411,376	152,046	24,379,515	3700	90.95	2977	11.76
0xaa87...fb84	128,100	65,074	14,680,490	535	16.65	2722	15.16
0xddfa...abd3	42,715	15,948	9,032,099	84	3.14	1255	17.90
0xdade...6084	148,186	51,864	5,338,322	28	145.06	6	19.42
0xe91c...96af	56,531	18,637	6,880,362	379	70.72	1330	20.50
0x4e85...0cea	125,575	45,586	1,765,057	25	44.43	2	21.56
0x2b24...446f	65,548	22,627	3,542,945	9	129.45	6	22.58
0x826f...b5f5	343,999	106,016	5,009,861	3	1.51	2465	23.58
0xbba9...3cf9	68,352	25,976	2,737,139	14	76.01	2	24.56

(a) Base

contract	swaps	txs with trades	non-reverted txs	reverted txs	txs per unique calldata	median calldata length	cum. MEV bot gas (%)
0x8872...6324	676,753	156,634	11,966,203	0	498.11	6	33.06
0xabf4...017c	29,586	8251	7,979,548	0	7,979,548.00	10	41.94
0xcdcc...5c8f	247,308	68,040	6,382,847	0	3489.80	6	49.08
0xd3dc...f2ac	214,741	60,522	6,901,973	5	1674.42	6	55.96
0x4d43...987f	208,002	47,459	2,322,899	10	195.39	3213	62.34
0x3955...74c2	422,807	107,201	1,992,783	0	2.98	2	64.56
0x9d1b...f96f	37,920	9531	825,165	40	1650.41	6	66.62
0x0daf...a7dd	398,174	105,069	2,299,930	0	4.72	2	68.51
0x2642...39e6	73,322	26,656	1,387,723	242	775.83	2422	69.98
0xf261...c883	242,058	51,232	928,822	4004	100.14	6	71.44

(b) Optimism

contract	swaps	txs with trades	non-reverted txs	reverted txs	txs per unique calldata	median calldata length	cum. MEV bot gas (%)
0x0000...98bd	9,871,348	3,338,292	3,338,353	1,241,384	17.34	110	10.42
0x60ca...294b	5,106,002	1,836,869	4,355,290	26,968	1.12	1207	18.68
0x6893...151c	4,769,221	1,706,662	1,708,167	1,087,474	1.00	572	24.28
0x9e52...5867	3,055,148	1,065,722	1,067,478	1,317,339	1.00	336	29.22
0xa9ff...d82f	774,311	347,885	348,021	51,732	1.05	2353	32.67
0x0000...2e43	3,045,007	1,125,161	1,125,162	338,202	14.52	104	35.34
0x0000...fb51	1,527,676	502,973	503,320	290,686	1.95	163	37.53
0xf238...e145	320,724	116,712	585,063	2796	2.39	266	39.28
0xe98b...d87c	1,145,798	560,741	516,379	413,451	30.23	52	40.87
0x84f1...b7cf	522,016	248,810	252,743	251,681	17.31	46	42.46

(c) Arbitrum

■ **Table 1** This table reports statistics for the top 10 cyclic arbitrage MEV bot contracts (ranked by cumulative gas usage) on Base (Table 1a), Optimism (Table 1b), and Arbitrum (Table 1c). For each contract, we report: the total number of individual swap operations executed (swaps), the number of transactions that included at least one swap (txs with trades), the number of transactions that executed without reverting (non-reverted txs), and the number of transactions that reverted during execution (reverted txs). The column txs per unique calldata gives the ratio of total transactions to the number of unique calldata entries used by the bot and median calldata length reports the median size of calldata (in hexadecimal characters) observed for the bot. The last column (cum. MEV bot gas (%)) indicates the cumulative share of gas consumed by each contract relative to all identified MEV bots on that chain.

will land as intended. The remaining large cyclic arbitrage bots on Arbitrum, however, display frequent calldata reuse, consistent with the optimistic MEV pattern observed on Base and Optimism.

Concentration of Gas Usage

We also compared the concentration of MEV bot activity across networks in terms of gas usage. On Optimism, the top 10 MEV bots account for a disproportionately large share of gas consumed by atomic arbitrage, suggesting a more concentrated and potentially less competitive landscape. Arbitrum shows a slightly more distributed gas usage among top bots, while Base has the most diffuse distribution. These trends align with recent shifts in dominance across Layer 2s, with Base emerging as the most active MEV network, followed by Arbitrum and then Optimism.

Code Similarity and Cloning Behavior

To explore potential relationships among MEV-performing contracts, we conducted a bytecode similarity analysis inspired by prior work on Ethereum contract topology by Kiffer et al. [49]. This revealed meaningful differences in code reuse and cloning behavior across networks.

Contracts deployed on Optimism and Base show a moderate degree of bytecode similarity, suggesting some shared tooling or deployment patterns. In contrast, cyclic arbitrage contracts on Arbitrum appear more structurally distinct, with fewer instances of directly shared or closely related implementations. Among the three networks, Optimism exhibits the highest level of intra-network code similarity, pointing to a relatively concentrated MEV ecosystem with repeated use of identical or near-identical contracts — potentially indicating lower competitive diversity.

We also identified clear examples of contract cloning behavior, particularly on Base (see Appendix A). For instance, contracts deployed at addresses `0x2b24...446f` and `0xdade...6084` share identical bytecode not only with each other but also with 48 additional contracts. All 50 of these contracts engage in cyclic arbitrage and are classified as MEV bots in our dataset [16]. Although none of these clones individually ranked among the top 10 gas consumers, their collective presence underscores the widespread use of standardized MEV bot implementations. This pattern may reflect coordinated deployments by a single actor or group.

Further analysis of these cloned contracts reveals another striking pattern: the transaction `calldata` passed to them is often minimal or non-informative (e.g., `0x0001`), suggesting that key arbitrage parameters are not set off-chain. As an example, `0xdade...6084` executed 51,864 transactions with trades of which 50,861 had `0x0001` as `calldata`. Instead, the contract logic itself likely handles opportunity identification, path selection, and trade sizing through real-time on-chain computation. This architecture relies heavily on internal heuristics and on-chain state queries (e.g., via `STATICCALLS`), contrasting with more traditional MEV bots where critical parameters are calculated off-chain and passed in via `calldata`.

6 Drivers of Optimistic MEV

To move beyond descriptive patterns and develop a quantitative understanding of the drivers of MEV activity on Layer 2 networks, we employ an OLS regression framework. The goal is to assess the statistical significance and direction of influence of various market and network-level metrics on daily MEV fluctuations. By systematically analyzing these relationships, we

aim to identify the conditions under which MEV activity is amplified or suppressed across evolving Layer 2 ecosystems.

Our analysis focuses on two dependent variables: the daily change in the total number of transactions associated with `CYCLICARB` bots ($\Delta\text{CyclicArbTx}_t$) and the subset of those transactions that execute trades ($\Delta\text{CyclicArbTxWTrade}_t$), described in detail below:

$\Delta\text{CyclicArbTx}_t$ (Change in Cyclic Arbitrage Transaction Count) The total number of transactions classified as `CYCLICARB` (see Section 4.2) within our dataset is measured by this variable.

$\Delta\text{CyclicArbTxWTrade}_t$ (Change in Cyclic Arbitrage Trade Transaction Count) The daily change in the total number of transactions that are classified as `CYCLICARB-TRADE`.

This dual perspective allows us to distinguish between general `CYCLICARB` activity and the subset of transactions involving explicit on-chain value extraction through token trades, `CYCLICARB-TRADE`. Understanding these dynamics is crucial for assessing the efficiency of Layer 2 markets and informing the design of potential mitigation strategies.

The regression equation is then specified as:

$$\Delta y_t = \beta_0 + \beta_1 \cdot \Delta\text{Price}_t + \beta_2 \cdot \Delta\text{Volatility}_t + \beta_3 \cdot \Delta\text{RetailTxs}_t + \beta_4 \cdot \Delta\text{RetailAggFrac}_t + \epsilon_t$$

where $\Delta x_t = x_t - x_{t-1}$ denotes the first difference of variable x at time t , and ϵ_t is the error term. The independent variables are defined as follows:

ΔPrice_t (Change in ETH Price) The daily change in the price of ETH (denominated in US\$).

$\Delta\text{Volatility}_t$ (Change in ETH Volatility) The daily change in intraday volatility of ETH. Intraday volatility is computed using the Garman-Klass estimator [42, 51], defined for a given day as:

$$\sigma_{\text{OHLC}} = \sqrt{0.5 \cdot (\ln H - \ln L)^2 - (2 \ln 2 - 1) \cdot (\ln C - \ln O)^2}$$

where O, H, L, C are the Open, High, Low, and Close prices for ETH within the day, respectively.

$\Delta\text{RetailTxs}_t$ (Change in Retail Trade Count) The daily change in the total number of on-chain trades — defined as transactions involving at least one swap — initiated by entities not classified as `CYCLICARB` bots in our curated dataset, i.e., `OTHER-TRADE`.

$\Delta\text{RetailAggFrac}_t$ (Change in Retail Aggregator Usage) The daily change in the fraction of trades by non-cyclic arbitrage bot addresses that are routed through DEX aggregators:

$$\text{RetailAggFrac}_t = \frac{\text{RetailAggregatorTrades}_t}{\text{RetailTxs}_t}$$

A lower value indicates a greater share of direct-to-pool trades, which may reflect routing inefficiencies and create arbitrage opportunities [10, 73, 72]. However, not all direct trades are necessarily exploitable, some may avoid introducing new arbitrage opportunities and thus remain economically efficient, even without using an aggregator.

This metric excludes trades routed via standard Uniswap routers, which differ in function and behavior from third-party DEX aggregators [8, 53, 40, 77, 78].

We now turn to the empirical results of our regression analysis, presented in Table 2. The models examine how the previously defined variables relate to daily fluctuations in `CYCLICARB`

	$\Delta\text{CyclicArbTx}$ (1, Base)	$\Delta\text{CyclicArbTxWTrade}$ (2, Base)	$\Delta\text{CyclicArbTx}$ (1, Optimism)	$\Delta\text{CyclicArbTxWTrade}$ (2, Optimism)	$\Delta\text{CyclicArbTx}$ (1, Arbitrum)	$\Delta\text{CyclicArbTxWTrade}$ (2, Arbitrum)
const	2445.7691 (6290.7125)	173.5872 (582.2378)	372.2486 (1066.6453)	44.9657 (152.2307)	82.2414 (437.4133)	32.3800 (177.3206)
ΔPrice	-12608.7057 (10629.4760)	-1881.3146** (897.7109)	-1156.6053 (1421.6869)	-454.8404 (298.1963)	-25.9810 (706.7666)	-357.1149 (264.1559)
$\Delta\text{Volatility}$	34112.7533*** (11474.9164)	8919.1607*** (913.0034)	1323.4766 (1722.3345)	1289.5335*** (375.0010)	2265.7813*** (707.9527)	299.0799 (307.2508)
$\Delta\text{RetailTxs}$	23175.6525** (10917.7060)	4172.5237*** (1245.3633)	2956.2109 (1837.3891)	3276.7001*** (532.7932)	16024.0033*** (904.0916)	8097.8011*** (360.9759)
$\Delta\text{RetailAggFrac}$	12230.8890** (4756.1076)	1883.1927*** (442.2920)	-1785.4280 (1253.4087)	-1264.8155*** (369.1399)	-1524.6186*** (475.3803)	-1319.3665*** (289.9207)
Obs	654	654	700	700	700	700
Adj. R^2	0.0736	0.3399	0.0240	0.5818	0.7143	0.7898
F-stat	6.8461	31.7334	2.7295	57.8104	208.3965	265.6485

■ **Table 2** OLS regression results for Base, Optimism, and Arbitrum. The dependent variables are the daily change in CYCLICARB-related transactions ($\Delta\text{CyclicArbTx}$) and the daily change in CYCLICARB-related transactions involving DEX trades ($\Delta\text{CyclicArbTxWTrade}$). Independent variables include changes in ETH price, intraday volatility, the count of non-CYCLICARB (OTHER, see Section 4.2) trades, and the fraction of those routed via DEX aggregators. Robust standard errors are reported in parentheses. Significance levels: * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

activity across Layer 2 networks. Specifically, we report results for two dependent variables: the daily change in overall CYCLICARB-related transactions ($\Delta\text{CyclicArbTx}$) and the daily change in CYCLICARB-related transactions involving DEX trades ($\Delta\text{CyclicArbTxWTrade}$).

A consistent pattern across all three networks is that the model explains daily changes in **cyclicArb-trade transaction count** (Model 2) substantially better than changes in the broader **cyclicArb transaction count** (Model 1), as indicated by higher adjusted R^2 values. This suggests that the chosen market variables are more directly associated with CYCLICARB behaviors involving on-chain value extraction (i.e., swaps, TRADE), rather than speculative probing.

Arbitrum exhibits the best overall model fit, particularly for $\Delta\text{CyclicArbTx}$ category. This likely reflects a tighter correlation between CYCLICARB-TRADE transaction count and total CYCLICARB transaction count on that chain.

Comparative Analysis of Independent Variables

ΔPrice (Daily Change in ETH Price). The effect of ETH price fluctuations on CYCLICARB activity appears weak and inconsistent across networks:

- **Model 1:** No statistically significant association between daily ETH price changes and overall CYCLICARB transaction count on any network.
- **Model 2:** A statistically significant negative effect is observed on Base (-1881.31 , $p < 0.05$), suggesting that a sharp decrease in ETH price may increase the number of CYCLICARB-TRADE executions. The coefficients for Optimism and Arbitrum are also negative but fall short of significance thresholds.

These findings suggest a higher prevalence of arbitrage opportunities on days marked by ETH price declines. We propose two primary hypotheses for this observation. Firstly, diminished market liquidity during price downturns [23] may lead to increased slippage and suboptimal trade execution, thereby fostering greater price discrepancies [9]. Secondly, the predominance of long leverage in DeFi protocols [45] could result in a higher frequency of liquidations on days with negative price movements. Such liquidations can trigger significant price swings, which in turn may create further price disparities [66].

Δ Volatility (Daily Change in ETH Volatility). Volatility, by contrast, plays a more significant role:

- **Model 1:** A strong and significant positive effect on Base (34112.75, $p < 0.01$) and Arbitrum (2265.78, $p < 0.01$), indicating that volatility increases overall CYCLICARB transaction activity. No significant effect is observed on Optimism.
- **Model 2:** A similarly strong positive effect is found for CYCLICARB-TRADE transactions on Base (8919.16, $p < 0.01$) and Optimism (1289.53, $p < 0.01$). On Arbitrum, however, the coefficient is small and not statistically significant.

The widely held expectation that increased volatility fuels MEV activity is supported by our results for CYCLICARB-TRADE transactions on Base and Optimism [54, 48], as well as for overall CYCLICARB counts on Base and Arbitrum. However, the absence of a significant volatility effect on the change of CYCLICARB-TRADE on Arbitrum deserves special attention. In a high-throughput setting with very short block times, an efficient CEX-DEX arbitrage layer can continually realign each DEX price to its CEX counterpart, effectively erasing intra-DEX cyclic arbitrage opportunities, regardless of volatility. Under such a “hierarchical arbitrage” regime, volatility may drive cross-venue trades but leave purely on-chain cycles unprofitable. This provides a possible explanation for why we observe little volatility-driven CYCLICARB trading on Arbitrum.

Δ RetailTx (Change in Retail Trade Count). Retail activity (used here as a proxy for organic user flow) is consistently important for MEV trade behavior:

- **Model 1:** Significant positive effects on Base (23175.65, $p < 0.05$) and Arbitrum (16024.00, $p < 0.01$); not significant on Optimism.
- **Model 2:** Strong and significant on all three networks (i.e., Base, Optimism, and Arbitrum) all at $p < 0.01$.

These findings reinforce the idea that user-driven flow is the foundation for MEV extraction via arbitrage. This is because user transactions, such as trades on DEXs, are the primary actions that perturb market prices. These perturbations create transient price discrepancies across different venues or asset pairs. CYCLICARB bots then capitalize on these temporary imbalances by executing arbitrage trades, effectively profiting from the price impact of the initial user-driven activity. Thus, without the initial flow from users, the opportunities for this form of MEV extraction would be significantly diminished. On the other hand, the lack of significance for total CYCLICARB transaction count on Optimism suggests that probing or spam-like transactions may be more prevalent there, reducing the signal from genuine trade-driven activity.

Δ RetailAggFrac (Change in Aggregator Usage by Retail Users). This variable reveals the most striking network divergence:

- **Model 1:** Base shows a positive and significant effect (12230.89, $p < 0.05$); Arbitrum, a significant negative effect (-1524.62 , $p < 0.01$); Optimism shows no significance.
- **Model 2:** Base again shows a significant positive effect (1883.19, $p < 0.01$), while both Optimism (-1264.82 , $p < 0.01$) and Arbitrum (-1319.37 , $p < 0.01$) show significant negative effects.

Increased aggregator usage is generally indicative of more efficient transaction routing, which we hypothesized would lead to fewer price discrepancies. However, the Base results present a counterintuitive finding, as higher aggregator utilization did not correspond with a reduction in naive arbitrage opportunities. Several hypotheses may explain this observation:

- Many tokens on Base may have only one active pool, making direct-to-pool trades effectively MEV-optimal (i.e., they do not open up an arbitrage opportunity) — undermining the aggregator efficiency signal.
- Aggregator usage may correlate with increased activity in major token pairs (e.g., ETH-stables), especially during volatile periods, making `RetailAggFrac` a latent proxy for volatility.

Thus, our use of `RetailAggFrac` as a proxy for the ratio between MEV-optimal Trades and Total Trades, where MEV-optimal trades are those that do not introduce new arbitrage opportunities, may be misaligned on Base due to its unique market structure.

On Optimism and Arbitrum, the expected pattern emerges: increased aggregator usage correlates with reduced `CYCLICARB-TRADE` activity, consistent with the role of aggregators in mitigating simple arbitrage opportunities by improving trade routing.

Summary of Insights

The regression results demonstrate that some drivers of `CYCLICARB` (such as user trade count) are robust across networks. In contrast, other factors like volatility and aggregator usage show strong network-specific effects, reflecting differences in market structure and protocol design. These findings underscore that `CYCLICARB` dynamics are not uniform across Layer 2s, but are shaped by their unique configurations, including block times, mempool behavior, and execution environments.

7 Discussion

Optimistic MEV fits into the broader MEV landscape as a new variant of an old problem that was thought to have been largely solved. Its optimistic execution model is novel because profitability is determined on-chain during transaction execution, but the externality it creates in the form of large volumes of low-value or failed transactions mirrors and even outpaces the spam problem that plagued early Ethereum MEV. On Ethereum, that earlier spam arose from bots flooding the mempool with precomputed transactions and engaging in priority gas auctions without knowing whether their transactions would land in the right place in the block to secure the opportunity, a dynamic that private orderflow and MEV auctions were introduced to mitigate.

This behavior has significant consequences. It generates large volumes of optimistic MEV transactions on Layer 2s, creating spam that consumes blockspace without delivering corresponding value to users. This spam constitutes a negative externality for the network and must still be stored by nodes. The resulting data burden increases costs for infrastructure providers and risks concentrating node operation among better-resourced actors. It also appears that optimistic MEV creates a workload that very often fills up blocks, which should be taken into account when designing blockchains.

For searchers, this environment reshapes the underlying economics. The focus shifts away from high-certainty execution toward balancing the cost of repeated speculative probes compared to the profit they generate. Capturing value becomes less of a latency race; instead, the transactions are submitted before the opportunities even exist. Further, optimistic MEV requires paying far lower priority fees compared to traditional MEV on Ethereum Layer 1, which fundamentally changes how bots compete and extract value.

MEV auctions are the main mechanism used in practice to prevent the negative externalities of MEV by reducing spam. Optimistic MEV makes this harder to apply. Its speculative, high-frequency nature could produce a flood of bids and overwhelm an auction with noise,

while the short block intervals on Layer 2s leave little room to run the process at all. Revert protection, as implemented by Unichain, offers an alternative mitigation but risks generating a similar kind of spam directed at the sequencer. In addition, the presence of centralized sequencers on Layer 2s creates incentive dynamics very different from Layer 1.

Addressing optimistic MEV will likely require rethinking how, or whether, MEV auctions and related mechanisms can work in rollup environments. Our results suggest that network design choices can influence the prevalence of optimistic MEV: the significantly lower rate of optimistic MEV we observe on Arbitrum compared to OP-stack rollups indicates that factors such as gas pricing or block intervals can meaningfully affect these strategies. This points toward protocol-level levers for reducing optimistic MEV as Layer 2 designs evolve.

8 Outlook and Conclusion


In this work, we investigate the significant and large-scale impact of *optimistic MEV* on Layer 2 networks, focusing on cyclic arbitrage, which on these networks is predominantly executed in this optimistic form and accounts for more than 50% of gas usage in OP-Stack ecosystems. Our findings reveal that a confluence of factors, namely low transaction fees, extended interblock times, and PGA ordering mechanisms prevalent on OP-Stack chains, contributes to this extensive MEV activity.

The ramifications of this activity are substantial, manifesting as a phenomenon akin to network spam. This spam-like behavior inundates the network with low-value transactions, leading to inefficient resource allocation, wasted chain space, and potential degradation of user experience. Critically, this also limits the network's capacity to scale for higher-value or user-driven activity, as blockspace is increasingly consumed by speculative probing.

Addressing this challenge is essential for the long-term sustainability of these networks. A shift in network dynamics is necessary, either through the emergence of organic demand willing to pay higher fees, thereby crowding out low-value MEV attempts, or through direct interventions such as raising minimum transaction costs. Arbitrum, where different design choices have demonstrated some efficacy in curbing similar behaviors, serves as a pertinent case study for the latter approach.

Further differentiating the network dynamics, our analysis uncovered that most unsuccessful MEV attempts do not revert on OP-Stack Layer 2s, a stark contrast to Arbitrum, where such reversions are common. This divergence is likely attributable to Arbitrum's FCFS transaction ordering, approximately tenfold shorter interblock times, and a distinct fee market. These elements collectively appear to incentivize more deterministic behavior from MEV bots on Arbitrum, as evidenced when comparing the strategies of top bots across platforms, unlike the more speculative attempts observed on Optimism and Base. This underscores how architectural and fee-market designs can significantly influence the strategies of MEV bots and overall network congestion.

References

- 1 Arbiscan. <https://arbiscan.io>. Accessed: 2025-05-19.
 - 2 Basescan. <https://basescan.org>. Accessed: 2025-05-19.
 - 3 Curve finance github repository. <https://github.com/curvefi>. Accessed: 2025-05-19.
 - 4 Intel arkm. <https://intel.arkm.com>. Accessed: 2025-05-19.
 - 5 Optimistic etherscan. <https://optimistic.etherscan.io>. Accessed: 2025-05-19.
 - 6 Unichain block explorer. <https://unichain.blockscout.com>. Accessed: 2025-05-19.
 - 7 Uniswap github repository. <https://github.com/Uniswap>. Accessed: 2025-05-19.
- 

- 8 0x534154. Uniswap bug causes loss of eth. https://hackmd.io/@0x534154/Hy6_66gT5, 2022. HackMD post, accessed 2025-05-09.
- 9 Austin Adams, Benjamin Y Chan, Sarit Markovich, and Xin Wan. Don't let mev slip: The costs of swapping on the uniswap protocol, 2024. URL: <https://arxiv.org/abs/2309.13648>, arXiv:2309.13648.
- 10 Guillermo Angeris, Tarun Chitra, Alex Evans, and Stephen Boyd. Optimal routing for constant function market makers, 2022. URL: <https://arxiv.org/abs/2204.05238>, arXiv:2204.05238.
- 11 Arbitrum. Arbitrum. <https://arbitrum.io>. Accessed: 2025-05-18.
- 12 Arbitrum. Gas fees: Child chain gas pricing. URL: <https://docs.arbitrum.io/how-arbitrum-works/gas-fees#child-chain-gas-pricing>.
- 13 Arbitrum Docs. Timeboost: A gentle introduction. <https://docs.arbitrum.io/how-arbitrum-works/timeboost/gentle-introduction>, April 2025. Accessed: 2025-06-17.
- 14 bantg. interesting point, here is a distribution of backrun transaction position in the block relative to the initial buy. <https://x.com/bantg/status/1694271714991943879>, August 2023. Accessed: 2025-05-12.
- 15 Base. Base. <https://www.base.org>. Accessed: 2025-05-18.
- 16 BaseScan. Find similar contracts: Contract 0x22009730d6ad8a90574ea826603906dc81766fb1. <https://basescan.org/find-similar-contracts?a=0x22009730d6ad8a90574ea826603906dc81766fb1&m=exact>, 2025. Accessed: May 21, 2025.
- 17 Jon Becker. heimdall-rs: A Rust-based blockchain analysis tool. <https://jbecker.dev/repo/Jon-Becker/heimdall-rs>, 2025. Accessed: 2025-05-18.
- 18 bertcmiller. MEV Searcher Strategy on Base with Friend.tech. <https://x.com/bertcmiller/status/1693734693592608795>, 2023. Accessed: 2025-05-12.
- 19 bertcmiller. My guess: if 2 txs have the same gas price then the sequencer orders the one it saw earlier first. X post, 2023. Accessed: 2025-05-28. URL: <https://x.com/bertcmiller/status/1693734701503012914>.
- 20 Vitalik Buterin. Possible futures of the ethereum protocol, part 2: The surge, 2024. Accessed on March 26, 2025. URL: <https://vitalik.eth.limo/general/2024/10/17/futures2.html>.
- 21 Federico Certera, Massimo La Morgia, Alessandro Mei, Alberto Maria Mongardini, and Francesco Sassi. Ready, aim, snipe! analysis of sniper bots and their impact on the defi ecosystem. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 1093–1102, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3543873.3587612.
- 22 Sam Chepal. The hidden world of mev. <https://www.samchepal.com/the-hidden-world-of/>, 2023. Accessed: 2025-05-25.
- 23 Tarun Chordia, Avaniidhar Subrahmanyam, and Richard W. Roll. Market liquidity and trading activity. *SSRN Electronic Journal*, July 2000. URL: <https://ssrn.com/abstract=237674>, doi:10.2139/ssrn.237674.
- 24 Clanker. Clanker world, 2025. Accessed: 2025-05-29. URL: <https://www.clanker.world>.
- 25 Doug Colkitt. Sniper bots and base reverts: A thread on clanker deployment issues. <https://x.com/0xdoug/status/1861662510023123030>, 2024. Accessed: 2025-05-03.
- 26 Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges, 2019. URL: <https://arxiv.org/abs/1904.05234>, arXiv:1904.05234.
- 27 Dune. Data freshness in the Dune data catalog. <https://docs.dune.com/data-catalog/data-freshness>, 2025. Accessed: 2025-05-18.
- 28 Dune Analytics. Address labels. <https://docs.dune.com/data-catalog/curated/labels/address-labels>, 2025. Accessed: 2025-05-25.

- 29 Dune Analytics. Dex aggregator trades. <https://docs.dune.com/data-catalog/curated/evm/DEX/dex-aggregator-trades>, 2025. Accessed: 2025-05-25.
- 30 Dune Analytics. Ethereum raw traces. <https://docs.dune.com/data-catalog/evm/ethereum/raw/traces>, 2025. Accessed: 2025-05-25.
- 31 Dune Analytics. Ethereum raw transactions. <https://docs.dune.com/data-catalog/evm/ethereum/raw/transactions>, 2025. Accessed: 2025-05-25.
- 32 Dune Docs. DEX Trades. <https://docs.dune.com/data-catalog/curated/evm/DEX/dex-trades>. Accessed: 2025-05-18.
- 33 EigenPhi. Eigenphi, 2025. Accessed: 2025-05-23. URL: <https://eigenphi.io>.
- 34 Entropy Advisors. Arbitrum mev arbitrage dashboard. https://dune.com/entropy_advisors/arbitrum-mev-arbitrage, 2025. Accessed: May 22, 2025.
- 35 Entropy Advisors. Entropy advisors. <https://entropyadvisors.com/>, 2025. Accessed: May 22, 2025.
- 36 Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. SoK: Transparent Dishonesty: Front-Running Attacks on Blockchain. In *Financial Cryptography and Data Security (FC), Kota Kinabalu, Sabah, Malaysia*, 2020.
- 37 Ethereum Improvement Proposals. EIP-4844: Shard Blob Transactions. <https://eips.ethereum.org/EIPS/eip-4844>, 2025. Accessed: 2025-05-08.
- 38 Ethereum Improvement Proposals. EIP-4844: Shard Blob Transactions (GitHub Markdown). <https://github.com/ethereum/EIPs/blob/master/EIPS/eip-4844.md>, 2025. Accessed: 2025-05-08.
- 39 Flashbots. mev-inspect-py: A tool for inspecting MEV transactions. <https://github.com/flashbots/mev-inspect-py>, 2025. Accessed: 2025-05-18.
- 40 foldfinance. Tweet posted 2022-10-09 20:43 utc. Tweet. URL: <https://x.com/foldfinance/status/1579211031855108096>.
- 41 Friend.tech. Friend.tech: Decentralized Social Platform. <https://www.friend.tech>, 2024. Accessed: 2025-05-12.
- 42 Mark B. Garman and Michael J. Klass. On the estimation of security price volatilities from historical data. *The Journal of Business*, 53(1):67–78, 1980. URL: <http://www.jstor.org/stable/2352358>.
- 43 Krzysztof Gogol, Johnnatan Messias, Deborah Miori, Claudio Tessone, and Benjamin Livshits. Cross-rollup mev: Non-atomic arbitrage across 12 blockchains, 2024. URL: <https://arxiv.org/abs/2406.02172>, arXiv:2406.02172.
- 44 growthepie. growthepie: Layer 2 analytics and education. Accessed on March 27, 2025. URL: <https://www.growthepie.xyz>.
- 45 Lioba Heimbach and Wenqian Huang. Defi leverage. *Available at SSRN*, May 2023. URL: <https://ssrn.com/abstract=4459384>, doi:10.2139/ssrn.4459384.
- 46 Lioba Heimbach, Lucianna Kiffer, Christof Ferreira Torres, and Roger Wattenhofer. Ethereum’s Proposer-Builder Separation: Promises and Realities. In *ACM Internet Measurement Conference (IMC), Montreal, QC, Canada*, 2023.
- 47 Lioba Heimbach and Jason Milionis. The early days of the ethereum blob fee market and lessons learnt, 2025. URL: <https://arxiv.org/abs/2502.12966>, arXiv:2502.12966.
- 48 Lioba Heimbach, Vabuk Pahari, and Eric Schertenleib. Non-atomic arbitrage in decentralized finance, 2024. URL: <https://arxiv.org/abs/2401.01622>, arXiv:2401.01622.
- 49 Lucianna Kiffer, Dave Levin, and Alan Mislove. Analyzing ethereum’s contract topology. In *Proceedings of the Internet Measurement Conference 2018*, IMC ’18, page 494–499, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3278532.3278575.
- 50 L2BEAT. Layer 2 scaling: Total value secured. Accessed on March 27, 2025. URL: <https://l2beat.com/scaling/tvs>.
- 51 Young Li. A practical model for prediction of intraday volatility. https://assets.bhub.io/professional/sites/10/intraday_volatility-3.pdf, 2021. Bloomberg Enterprise Quants, accessed 28 Apr. 2025.

- 52 LibMEV. Libmev, 2025. Accessed: 2025-05-23. URL: <https://libmev.com>.
- 53 MevRefund. “@uniswap what you doing here? poor man just wants to plow 0.36 eth into some shitcoin ...”. Tweet. URL: <https://x.com/MevRefund/status/1643048619547971584>.
- 54 Jason Milionis, Ciamac C. Moallemi, and Tim Roughgarden. Automated market making and arbitrage profits in the presence of fees, 2023. URL: <https://arxiv.org/abs/2305.14604>, arXiv:2305.14604.
- 55 Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing, 2024. URL: <https://arxiv.org/abs/2208.06046>, arXiv:2208.06046.
- 56 Bert C. Miller. Right now a lot of l2s are congested because demand for gas is higher than supply... <https://x.com/bertcmiller/status/1831797928756687292>, September 2024. Accessed: 2025-05-28.
- 57 Misaka. Mev research during wormhole incident. <https://misaka.substack.com/p/mev-research-during-wormhole-incident>, 2022. Accessed: 2025-05-25.
- 58 Alex Nezlobin and Martin Tassy. Loss-versus-rebalancing under deterministic and generalized block-times, 2025. URL: <https://arxiv.org/abs/2505.05113>, arXiv:2505.05113.
- 59 Alexandre Obadia, Alejo Salles, Lakshman Sankar, Tarun Chitra, Vaibhav Chellani, and Philip Daian. Unity is strength: A formalization of cross-domain maximal extractable value, 2021. URL: <https://arxiv.org/abs/2112.01472>, arXiv:2112.01472.
- 60 Optimism. Optimism. <https://www.optimism.io>. Accessed: 2025-05-18.
- 61 Ethereum Optimism. Add initial devnet op-geth changes. <https://github.com/ethereum-optimism/op-geth/pull/118>, 2023. Accessed: 2025-05-12.
- 62 Burak Öz, Christof Ferreira Torres, Jonas Gebele, Filip Rezabek, Bruno Mazonra, and Florian Matthes. Pandora’s box: Cross-chain arbitrages in the realm of blockchain interoperability. *arXiv preprint arXiv:2501.17335*, 2025.
- 63 Ozan Solmaz. Optimistic MEV. <https://github.com/kingbaldwin-iv/Optimistic-MEV>, 2025.
- 64 Paradigm. Ethereum is a dark forest. <https://www.paradigm.xyz/2020/08/ethereum-is-a-dark-forest>, 2020. Accessed: 2025-05-25.
- 65 Polynya. Ethereum l1 scaling roadmap, 2023. Accessed on March 26, 2025. URL: https://polynya.mirror.xyz/epju72rsymfB-JK52_uYI7HuhJ-W_zM735NdP7alkAQ.
- 66 Kaihua Qin, Liyi Zhou, Pablo Gamito, Philipp Jovanovic, and Arthur Gervais. An empirical study of defi liquidations: incentives, risks, and instabilities. In *Proceedings of the 21st ACM Internet Measurement Conference, IMC ’21*, page 336–350. ACM, November 2021. URL: <http://dx.doi.org/10.1145/3487552.3487811>, doi:10.1145/3487552.3487811.
- 67 Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest?, 2021. URL: <https://arxiv.org/abs/2101.05511>, arXiv:2101.05511.
- 68 Rekt News. Return to the dark forest. <https://rekt.news/return-to-the-dark-forest/>, 2021. Accessed: 2025-05-25.
- 69 Samczsun. Escaping the dark forest. <https://samczsun.com/escaping-the-dark-forest/>, 2020. Accessed: 2025-05-25.
- 70 Ozan Solmaz. Dune analytics query: 5534551. <https://dune.com/queries/5534551>, 2024. Accessed on 2024-07-25.
- 71 Sorella Labs. Brontes: A blockchain analysis tool. <https://github.com/SorellaLabs/brontes>, 2025. Accessed: 2025-05-18.
- 72 ThogardPvP. Tweet posted 2022-10-09 20:28 utc. Tweet. URL: <https://x.com/ThogardPvP/status/1579207229059715072>.
- 73 ThogardPvP. Tweet posted 2022-10-09 20:35 utc. Tweet. URL: <https://x.com/ThogardPvP/status/1579208974812909568>.
- 74 Christof Ferreira Torres, Ramiro Camino, and Radu State. Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain. In *30th USENIX Security Symposium, Virtual Event*, 2021.

- 75 Christof Ferreira Torres, Albin Mamuti, Ben Weintraub, Cristina Nita-Rotaru, and Shweta Shinde. Rolling in the shadows: Analyzing the extraction of mev across layer-2 rollups, 2024. URL: <https://arxiv.org/abs/2405.00138>, arXiv:2405.00138.
- 76 Umbra Research. MEV on Solana. <https://www.umbraresearch.xyz/writings/mev-on-solana>, 2023. Accessed: 2025-05-25.
- 77 Uniswap. Universal router repository. <https://github.com/Uniswap/universal-router>, 2022. GitHub repository, accessed 2025-05-09.
- 78 Uniswap Labs. Universal router – overview. URL: <https://docs.uniswap.org/contracts/universal-router/overview>.
- 79 Virtuals.io. Virtuals app, 2025. Accessed: 2025-05-29. URL: <https://app.virtuals.io>.
- 80 Ye Wang, Yan Chen, Haotian Wu, Liyi Zhou, Shuiguang Deng, and Roger Wattenhofer. Cyclic arbitrage in decentralized exchanges, 2022. URL: <https://arxiv.org/abs/2105.02784>, arXiv:2105.02784.
- 81 Ben Weintraub, Christof Ferreira Torres, Cristina Nita-Rotaru, and Radu State. A flash (bot) in the pan: Measuring maximal extractable value in private pools. In *ACM Internet Measurement Conference (IMC), Nice, France, 2022*.
- 82 Liyi Zhou, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais. On the Just-In-Time Discovery of Profit-Generating Transactions in DeFi Protocols. In *IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021*.
- 83 Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. High-Frequency Trading on Decentralized On-Chain Exchanges. In *IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021*.
- 84 Burak Öz, Jonas Gebele, Parshant Singh, Filip Rezabek, and Florian Matthes. Playing the mev game on a first-come-first-served blockchain, 2024. URL: <https://arxiv.org/abs/2401.07992>, arXiv:2401.07992.
- 85 Burak Öz, Danning Sui, Thomas Thiery, and Florian Matthes. Who wins ethereum block building auctions and why?, 2024. URL: <https://arxiv.org/abs/2407.13931>, arXiv:2407.13931.

A Contract Similarity

For every address⁵ that survives the manual audit, we retrieve the bytecode directly from the nodes using the JSON-RPC call `eth_getCode(<address>, "latest")`. Next, we use the Heimdall disassembler [17] to convert this bytecode into a sequence of EVM opcodes. Following the method used by Kiffer et al. [49], we remove all operand data so that only the opcode mnemonics remain. We then slide a five-opcode window over the cleaned stream and count the occurrences of each unique chunk, producing a high-dimensional frequency vector for each contract. Finally, we compute the cosine similarity between any two vectors to quantify how closely their opcode patterns match. The key steps are summarized in the following pseudocode:

Algorithm 1 Compute Contract Similarity Between Two Contracts

```

1: function SIMILARITY(bytecode1, bytecode2)
2:   dis1, dis2 ← disassemble(bytecode1), disassemble(bytecode2)
3:   opcs1, opcs2 ← strip_opcodes(dis1), strip_opcodes(dis2)
4:   freq_vec1, freq_vec2 ← compute_freq_vecs(c1=opcs1, c2=opcs2, N=5)
5:   return  $\frac{\text{freq\_vec}_1 \cdot \text{freq\_vec}_2}{\|\text{freq\_vec}_1\| \|\text{freq\_vec}_2\|}$ 
6: end function

```

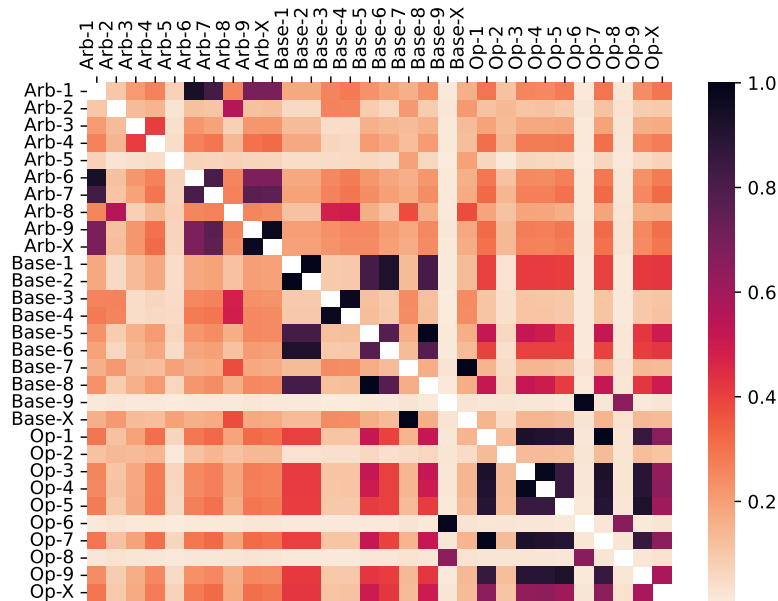


Figure 8 Cosine similarity scores between MEV bot contracts on Arbitrum, Base, and Optimism. Each block along the diagonal (e.g., Arbitrum-1 through Arbitrum-X, Base-1 through Base-X, Op-1 through Op-X) shows intra-network similarity, while off-diagonal blocks reveal inter-network code reuse. High similarity values (closer to 1.0) indicate shared or cloned implementations, with particularly tight clusters visible among Base contracts.

⁵ Although we focus only on the primary contracts, many arbitrage transactions also invoke auxiliary, non-DEX “helper” contracts during execution.

Figure 8 visualizes the pairwise cosine similarity between MEV bot contracts on Arbitrum, Base, and Optimism using 5-opcode frequency vectors derived from each contract’s disassembled bytecode. Along the diagonal, intra-network comparisons reveal that Base contracts form tight clusters, indicating many bots share nearly identical implementations, while Optimism shows a moderate level of code reuse and Arbitrum exhibits the greatest internal diversity. Off-diagonal blocks show inter-network similarity: Base and Optimism share some common code patterns, likely reflecting shared bot frameworks, but Arbitrum contracts remain largely distinct from those on the other two chains. The prominent high-similarity bands on Base confirm widespread cloning or redeployment of identical bot logic, whereas the lighter, more sporadic similarities elsewhere suggest more heterogeneous or independently developed MEV implementations.

B Reverts

Contrary to common assumptions [56], we find that most transaction reverts on Base are not driven by failed cyclic arbitrage attempts but by event-driven “liquidity-sniping” strategies [21, 25]. In these cases, MEV bots monitor the on-chain pool for new token listings and submit purchase transactions in the same block that liquidity is added. The first significant surge in revert rates on Base coincided with bots back-running FriendTech share listings as soon as new accounts launched [41]. Although Base’s mempool is private, a transient transaction-pool leak enabled MEV bots to execute same-block back-runs [14] until the vulnerability was patched [61]. MEV bots exploited the sequencer’s ordering by matching user gas bids and spamming identical transactions; any duplicate that landed before the user’s own transaction reverted, yet the low gas prices on Base made this spray-and-pray approach profitable, as illustrated in block 2930614 [18, 19].

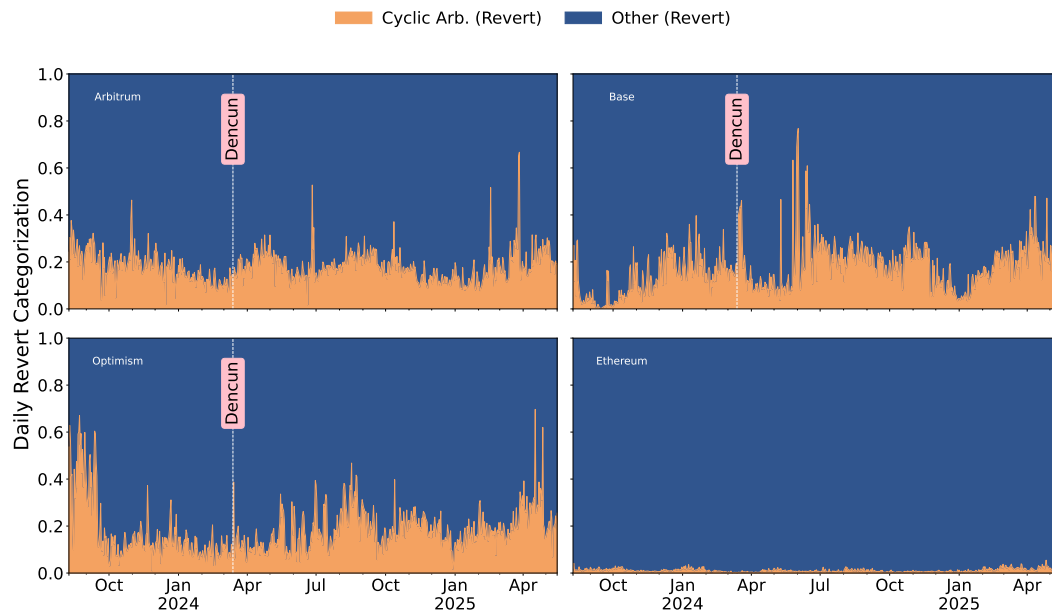


Figure 9 Daily share of reverted transactions on Arbitrum, Base, Optimism, and Ethereum Layer 1, split between cyclic arbitrage MEV bots (“Cyclic Arb. (Revert)”) and all other activity (“Other (Revert)”).

Figure 9 shows that, although optimistic-MEV bots account for a large share of total gas usage on Base and Optimism, they represent a substantially smaller fraction of reverted

transactions. This disparity indicates that most reverts on these networks stem from non-optimistic-MEV activity, chiefly liquidity-sniping, rather than failed arbitrage attempts.

Additionally, around late November 2024 and early January 2025, we see the most increase, in absolute terms, in reverts on Base, which causes network fees to increase. These are once again caused by sniper bots interacting with applications such as Clanker [24, 25] and Virtuals [79]. Importantly, we note that these drive the median gas price paid by MEV bots in our dataset during those periods as can be seen in the Figure 6.

C Data Collection

We collect data for Base [15], Optimism [60], Arbitrum [11] and Ethereum. At the time of writing, Base, Optimism, and Arbitrum are the three largest Layer 2s based on *total value locked (TVL)* according to L2BEAT [50, 44]. For comparison, we collect the same data for Ethereum too, enabling us to characterize the differences between Layer 1 and Layer 2, especially regarding the prevalence of optimistic MEV.

Our empirical analysis relies on a comprehensive dataset primarily sourced from Dune Analytics [27], a platform offering comprehensive access to raw and decoded blockchain data across various networks, including all networks of interest for this study. We collect the respective data spanning from August 2023 to May 2025. This dataset encompasses several critical types of data sources:

- 1. Transactional Data:** We retrieve high-level transaction details including, but not limited to, sender and recipient addresses, gas utilized, transaction value, input data, block numbers, and timestamps. Crucially, we also collect detailed transaction traces, which include internal transactions, emitted logs (events), and relevant state changes. This granular data was primarily sourced by querying Dune Analytics' tables such as `{{chains}}.transactions` (e.g. [31]) and `{{chains}}.traces` (e.g. [30]) for Ethereum Layer 1 and each respective Layer 2 network.
- 2. Decentralized Exchange Activity Data:** To analyze MEV transactions, we collect extensive data on trades and liquidity pools from DEX protocols operating on the target Layer 2s. This includes detailed swap event data and liquidity pool contract addresses, leveraging curated datasets such as those described in [32].
- 3. Contract Address Identification:** A curated list of smart contract addresses relevant to DEX activities, such as prominent DEX routers and aggregator contracts on the studied Layer 2s, is compiled. This is achieved through a combination of leveraging existing tagged address lists within Dune Analytics [27], cross-referencing them with on-chain explorers [1, 5, 2, 4], and preliminary heuristic-based identification from our dataset.
- 4. Contract Bytecodes:** For the contracts that we marked on Layer 2s, we fetch their bytecodes to be utilized in future steps to measure contract similarity (see Appendix A). We run our own archive nodes for Base, Arbitrum, and Optimism networks to fetch the bytecodes using JSON-RPC method `eth_getCode`.
- 5. Price Data:** We additionally obtain daily Open-High-Low-Close (OHLC) data for the ETH price on Ethereum Layer 1 using Dune Analytics [27], which we use for regression analysis and volatility calculation.

This multi-faceted data collection strategy was designed to provide the necessary inputs for our MEV and transaction classification pipeline, enabling a robust identification and empirical analysis of various MEV activities on the selected Layer 2 networks. We acknowledge Dune

Analytics' data ingestion processes, including considerations for data freshness (up to a day of delay) and the support of all major DEXs [32]. The few missing DEXs (such as those involving linch OTC trades) do not impact our results, as they are not relevant to MEV activity.

D Cyclic Arbitrage Contract Detection (Extended)

This section explains the full cyclic arbitrage contract detection process shown in Section 4.1 in more detail. This classification stage aims to identify smart contracts exhibiting strong on-chain signs consistent with cyclic arbitrage activities.

The classifier was implemented entirely in Dune SQL, leveraging Dune Analytics' data tables and Torino engine. The core tables utilized are described in the following way:

dex_aggregator.trades [29] Utilized to identify and subsequently filter out known DEX aggregator and router contract addresses, as these primarily act as intermediaries rather than originating MEV actors, thereby reducing noise.

dex.trades [32] Provided swap events from DEXs, including essential fields such as **tx_hash**, event index within the transaction (**evt_index**), transacted tokens, and amounts involved in each swap.

dex.addresses [28] Served as a label database, crucial for identifying contract types (e.g., routers, liquidity pools, factories) and distinguishing them from potential MEV bot contracts.

This process is formalized as follows:

Let \mathbb{A} be the set of all *externally owned accounts (EOAs)*, \mathbb{K} the set of contract addresses, a **swap event**, denoted as s , is defined as a 5-tuple:

$$s = (\text{token}_{\text{sold}}, \text{token}_{\text{bought}}, \text{amount}_{\text{sold}}, \text{amount}_{\text{bought}}, \text{idx}) \in \mathbb{K} \times \mathbb{K} \times \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{N}$$

where:

- $\text{token}_{\text{sold}}$ is the contract address of the token sold.
- $\text{token}_{\text{bought}}$ is the contract address of the token bought.
- $\text{amount}_{\text{sold}}$ is the quantity of $\text{token}_{\text{sold}}$.
- $\text{amount}_{\text{bought}}$ is the quantity of $\text{token}_{\text{bought}}$.
- idx is the intra-transaction index of the emitted swap log, preserving execution order.

Let \mathcal{S} be the set of all possible swap events. A **trade transaction**, denoted as tr , associated with a single blockchain transaction, is the set of all swap events $s \in \mathcal{S}$ that occurred within that transaction. Thus, $tr \in \mathcal{P}(\mathcal{S})$.

A **transaction** is represented by a 4-tuple:

$$(\text{hash}, \text{to}, \text{from}, tr),$$

where

- hash is the transaction hash.
- $\text{to} \in \mathbb{K}$ is the contract address that the transaction invoked first.
- $\text{from} \in \mathbb{A}$ is the EOA that initiated the transaction.
- tr is the trade (set of swap events) executed within this transaction.

Let \mathcal{T}_{raw} be the set of all such transactions containing at least one swap event. We define a feature extraction function $f : \mathcal{P}(\mathcal{S}) \rightarrow \text{Seq}(\mathbb{K}) \times \text{Map}(\mathbb{K}, \mathbb{R})$, where $\text{Seq}(\mathbb{K})$ is the space of sequences of token addresses and $\text{Map}(\mathbb{K}, \mathbb{R})$ is the space of mappings from token addresses to real-valued balance changes. For a given trade $tr = \{s_1, \dots, s_k\}$, these swaps are first ordered by their `idx` value, yielding an ordered sequence $\tilde{tr} = \langle \acute{s}_1, \acute{s}_2, \dots, \acute{s}_k \rangle$, where $\acute{s}_j = (\text{token}_{\text{sold},j}, \text{token}_{\text{bought},j})$. The function $f(tr)$ then produces a pair (Π, Δ_B) :

1. $\Pi = \langle \text{token}_{\text{sold},1}, \text{token}_{\text{bought},1}, \dots, \text{token}_{\text{sold},k}, \text{token}_{\text{bought},k} \rangle$. This is an ordered sequence of token addresses reflecting the actual path of token conversions in the transaction.
2. $\Delta_B = \{(\kappa, \delta_\kappa) \mid \kappa \in \mathbb{K}, \delta_\kappa \in \mathbb{R}\}$. This is a map representing the net balance changes after the transaction from the initiator's perspective across all tokens involved in the trade tr . Specifically, for each token κ , $\delta_\kappa = \sum_{j=1}^k (\text{amount}_{\text{bought},j} \cdot \mathbb{I}(\text{token}_{\text{bought},j} = \kappa) - \text{amount}_{\text{sold},j} \cdot \mathbb{I}(\text{token}_{\text{sold},j} = \kappa))$, where $\mathbb{I}(\cdot)$ is the indicator function.

Let \mathcal{T}' be the set of transactions transformed by f :

$$\mathcal{T}' = \{(\text{hash}, \text{to}, \text{from}, f(tr)) \mid (\text{hash}, \text{to}, \text{from}, tr) \in \mathcal{T}_{\text{raw}}\}$$

An element in \mathcal{T}' is of the form $(\text{hash}, \text{to}, \text{from}, (\Pi, \Delta_B))$. Then, we apply three sequential filters $(\alpha_1, \alpha_2, \alpha_3)$ to \mathcal{T}' :

1. **Filter α_1 (Router/Aggregator Exclusion):** Let $\mathcal{R}_{\text{contracts}}$ and $\mathcal{A}_{\text{contracts}}$ be the sets of known router and aggregator contract addresses, respectively. This filter removes transactions directly interacting with these intermediary contracts.

$$\mathcal{T}^{(1)} = \alpha_1(\mathcal{T}') = \{(\text{hash}, \text{to}, \text{from}, (\Pi, \Delta_B)) \in \mathcal{T}' \mid \text{to} \notin (\mathcal{R}_{\text{contracts}} \cup \mathcal{A}_{\text{contracts}})\}$$

2. **Filter α_2 (Cyclic Swap Detection):** This filter identifies transactions whose sequence of token swaps $\Pi = \langle \pi_1, \pi_2, \dots, \pi_{2k} \rangle$ forms a cycle. The predicate `isCyclic(Π)` holds true if:

- $k \geq 1$ (i.e., there is at least one swap).
- $\pi_1 = \pi_{2k}$ (the first token sold is the same as the last token bought).
- For all $j \in \{1, \dots, k-1\}$, $\pi_{2j} = \pi_{2j+1}$ (the token bought in the j -th swap is the same as the token sold in the $(j+1)$ -th swap, ensuring path continuity).

$$\mathcal{T}^{(2)} = \alpha_2(\mathcal{T}^{(1)}) = \{(\text{hash}, \text{to}, \text{from}, (\Pi, \Delta_B)) \in \mathcal{T}^{(1)} \mid \text{isCyclic}(\Pi)\}$$

3. **Filter α_3 (Profitability Assessment):** This filter retains transactions that result in a net profit for a token without incurring losses in any other token. It evaluates the balance changes map Δ_B . The predicate `isProfitable(Δ_B)` holds true if:

- There exists at least one token κ such that its balance change δ_κ is strictly positive ($\delta_\kappa > 0$).
- For all token balance changes $\delta_{\kappa'}, \delta_{\kappa'} \geq 0$ (i.e., there are no negative balance changes).

$$\mathcal{T}^{(3)} = \alpha_3(\mathcal{T}^{(2)}) = \{(\text{hash}, \text{to}, \text{from}, (\Pi, \Delta_B)) \in \mathcal{T}^{(2)} \mid \text{isProfitable}(\Delta_B)\}$$

Combining these filters, the set of transactions identified as high-probability cyclic arbitrage is $\mathcal{T}_{\text{final}} = \mathcal{T}^{(3)}$. As the final step of the algorithmic pre-filter, the set of contract addresses marked as high-probability cyclic arbitrage bots, \mathcal{C}_{bot} , is derived from these transactions:

$$\mathcal{C}_{\text{bot}} = \{\text{to} \mid (\text{hash}, \text{to}, \text{from}, (\Pi, \Delta_B)) \in \mathcal{T}_{\text{final}}\}$$

These contracts are then passed to the subsequent validation stage.

E Cyclic Arbitrage Contract Validation

To verify that the data set does not contain false positives, we inspect the candidate contracts. We perform these validation steps for each chain and the set of respective candidate contracts until more than 80% of all the gas used in transactions involving these contracts originates from validated contracts. Our validation process consists of the following four steps:

- **Code Verification and Labeling:** We investigate whether the contract’s source code was verified on public block explorers (e.g., Arbiscan and its forks [6, 1, 5, 2]). Verified contracts that were not proxies and had clear, non-MEV related functionalities, such as standard token contracts or well-known application logic, were excluded. We further query Arkham [4] and community-curated datasets [34, 52], and exclude contracts with labels that are inconsistent with MEV-bot activity.
- **Transaction Trace Analysis for Non-Trading Activity:** For transactions initiated by a candidate contract that did not result in direct swaps or token transfers, we inspect their execution traces to see if they interacted with DEX-related contracts (e.g., calling functions such as `getReserves` or `slot0` on DEXs to read pool reserves, or interacting with periphery contracts). We exclude all candidate contracts that do not have a clear majority (60% and more) of such DEX-related interaction. Note that the few candidates removed all had very low interaction numbers (between 0% and 30%), while the large majority of candidates have a very high DEX-interaction rate (85% or more).
- **Caller Profile Analysis:** We examined the diversity of callers interacting with the candidate contract and their frequency. If a candidate contract had (i) more than 3 distinct, and unrelated EOAs interacting with it and (ii) the frequency of interaction could credibly have been human, we consider the contract to not belong to a cyclic arbitrage bot. With this validation heuristic, we err on the side of caution, as there are cases where specialized MEV bots are triggered by multiple EOAs.
- **Swaps per Transaction:** We analyze the distribution of swaps per transaction (of those transactions that involve at least one swap) by observing the 10th, 25th, 50th, 75th, and 90th percentiles. A 10th or 25th percentile value of one ($P_{10} = 1$ or $P_{25} = 1$) would serve as a strong indicator that the contract does not belong to a cyclic arbitrage bot, as a cyclic arbitrage requires at least two transactions. For $P_{10}, P_{25}, P_{50}, P_{75}, P_{90}$ values of the top 10 contracts on each analyzed Layer 2, consult the Table 3. All candidate contracts pass this validation stage.

The few misclassified contracts we find during validation are mainly due to popular mislabeled routers or public utility contracts. The full list of removed contracts is found in [63]. These contracts are removed and the remaining contracts form our final set of cyclic arbitrage contracts \mathcal{C}_{bot} .

F Arbitrum: Other (Interaction)

The relative plot in Figure 4 reveals that the OTHER - INTERACTION category accounts for a significantly larger share of gas consumption on **Arbitrum** compared to other chains. An analysis of these transactions identified the primary gas-consuming contracts. Uniswap’s **V3 Position Manager** (0xC364...FE88) was the single largest consumer, responsible for nearly **10%** of the gas in this category. It was followed by a variety of other contracts, including Uniswap’s V3 Routers (0xe592...1564, 0x68b3...fc45), contracts associated with on-

contract	P_{10}	P_{25}	P_{50}	P_{75}	P_{90}
0xf5ff...a8c0	2	2	3	3	5
0xbff6...73e9	2	2	2	3	4
0xaa87...fb84	2	2	2	2	2
0xddfa...abd3	2	2	2	3	4
0xdade...6084	2	2	2	3	5
0xe91c...96af	2	2	3	3	6
0x4e85...0cea	2	2	3	3	5
0x2b24...446f	2	2	3	3	5
0x826f...b5f5	2	2	3	3	6
0xbba9...3cf9	2	2	3	3	3

(a) Base

contract	P_{10}	P_{25}	P_{50}	P_{75}	P_{90}
0x8872...6324	2	2	3	4	8
0xabf4...017c	2	2	2	3	6
0xcdcc...5c8f	2	3	3	3	6
0xd3dc...f2ac	2	2	3	3	6
0x4d43...987f	2	2	3	4	7
0x3955...74c2	2	2	3	3	6
0x9d1b...f96f	2	2	3	4	7
0x0daf...a7dd	2	2	3	3	6
0x2642...39e6	2	2	2	3	4
0xf261...c883	2	2	3	5	9

(b) Optimism

contract	P_{10}	P_{25}	P_{50}	P_{75}	P_{90}
0x0000...98bd	2	2	3	3	4
0x60ca...294b	2	3	3	3	3
0x6893...151c	2	2	3	3	4
0x9e52...5867	2	2	3	3	4
0xa9ff...d82f	2	2	2	2	3
0x0000...2e43	2	2	3	3	4
0x0000...fb51	2	2	3	3	4
0xf238...e145	2	2	3	3	3
0xe98b...d87c	2	2	2	2	2
0x84f1...b7cf	2	2	2	2	2

(c) Arbitrum

■ **Table 3** This table reports statistics for the top 10 MEV bot contracts (ranked by cumulative gas usage) on Base (Table 3a), Optimism (Table 3b), and Arbitrum (Table 3c). For each contract, we report 10th, 25th, 50th, 75th and 90th percentile of the number of swaps in a transaction which include at least one swap. The query associated with the tables can be viewed at [70].

chain games (0x2cfc...05c6), and suspected CEX-DEX bots such as 0xf3ae...dDaB.⁶ The

⁶ It is unlikely to be a DEX-DEX bot since the swap count equals the trade count.

category also includes contracts from other DEXs and aggregators like Ramses, Camelot, PancakeSwap, SushiSwap, Odos, Dopex (now Stryke), 1inch, and many more. To ensure that CYCLICARB contracts were not incorrectly placed in the OTHER category, we examined the transaction patterns of the identified contracts that were not labeled. We found that for all of them in the top 50% of the gas consumers, $\geq 98\%$ of transactions that included at least one swap operation contained **exactly one swap**. This confirms that they are predominantly simple DEX interactions rather than complex cyclic arbitrage activities, justifying their classification.

