# The Gold Digger in the Dark Forest: Industrial-Scale MEV Analysis in Ethereum

Ningyu He[1], Tianyang Chi[2], Xiaohui Hu[3], and Haoyu Wang[3]
[1]*The Hong Kong Polytechnic University*, Hong Kong SAR, China
[2]*Beijing University of Posts and Telecommunications*, Beijing, China
[3]*Huazhong University of Science and Technology*, Wuhan, China
ningyu.he@polyu.edu.hk, chitianyang@bupt.edu.cn, {xiaohui_hu, haoyuwang}@hust.edu.cn

***Abstract*—Maximal Extractable Value (MEV) activities pose critical operational challenges for blockchain enterprises, requiring automated detection systems to maintain platform integrity and regulatory compliance. Current industrial practices rely on heuristic rule-based methods with substantial accuracy limitations and inability to adapt to evolving MEV strategies. This paper presents an automated software engineering solution for large-scale MEV detection, introducing a novel graph-based profitability identification algorithm that replaces inflexible heuristic rules with adaptive mechanisms. Our automated system achieves 0.6% false positive rates for arbitrage detection and 2.4% false negative rates, significant improvements over existing methods with much higher error rates. We validate our approach on 21 million Ethereum blocks containing 2.5 billion transactions, covering critical infrastructure transitions including The Merge and Proposer-Builder Separation. Our automated pipeline identifies 12.1 million MEV activities, including 1.2 million previously undetectable advanced variants that pose emerging risks to platform operators. Key findings provide actionable insights for blockchain enterprises: private transaction architectures protect 71.4% of low-yield MEV opportunities rather than harming participants, contradicting previous assumptions. However, we identify concerning builder-searcher collusion involving 2,000+ transactions worth 350 ETH, highlighting compliance risks. Additionally, intensifying centralization trends show a single oligopoly controlling 43.1% of MEV activities in 2024, presenting systemic risks. Our automated detection framework provides blockchain enterprises with production-ready tools for MEV monitoring, risk assessment, and compliance management while offering critical insights for infrastructure design decisions in rapidly evolving DeFi environments.**

***Index Terms*—Blockchain, Ethereum, Maximal Extractable Value, MEV**

## I. Introduction

Decentralized Finance (DeFi) has emerged as a transformative force in the financial technology landscape, with Ethereum reaching a peak Total Value Locked (TVL) of $109 billion in May 2024 [1]. This explosive growth has created new operational challenges for blockchain enterprises and DeFi platform operators, particularly in managing Maximal Extractable Value (MEV) activities that can significantly impact platform fairness, user experience, and regulatory compliance.

Enterprise blockchain platforms face critical operational challenges in detecting and analyzing MEV activities. Attracted by profitability opportunities, numerous MEV searchers actively participate in extraction through two dominant strategies: arbitrage and sandwich attacks. According to previous industry statistics, these strategies account for over 99% of all MEV activities, with over $675 million extracted before September 2022 alone [2].

Current industrial detection systems suffer from two fundamental limitations impacting operational effectiveness. 1) *Inadequate Detection Accuracy.* Existing methods rely on heuristic rule-based patterns causing numerous false negatives and positives. These inflexible approaches fail to identify emerging advanced MEV types, such as conjoined sandwich attacks, and often ignore whether MEV initiators achieve profitability, a critical factor for risk assessment and compliance monitoring. 2) *Limited Adaptability to Infrastructure Evolution.* Most industrial analyses use early-stage Ethereum data and cannot effectively guide operations after major infrastructure changes. The Merge in 2022 transitioned Ethereum from Proof of Work to Proof of Stake, while Proposer-Builder Separation (PBS) fundamentally altered block construction processes. These changes directly impact MEV dynamics but existing detection systems lack adaptability to evolving architectures.

To address these industrial challenges, this work presents a comprehensive automated approach for MEV detection and analysis. Our solution introduces several key innovations for enterprise blockchain operations. 1) *Graph-Based Profitability Identification.* We develop a novel algorithm that imports token exchange rate concepts, calculating exchange rates between any two tokens to avoid inaccuracies from direct token amount comparisons. This approach provides more reliable profitability assessment for enterprise risk management. 2) *Adaptive Detection Algorithms.* Based on our profitability identification framework, we design robust algorithms to identify arbitrage and sandwich attacks, including their advanced variants that traditional methods cannot detect. 3) *Comprehensive Infrastructure Analysis.* Our approach covers all three critical MEV evolution stages, providing actionable insights for enterprise platform design and operational strategies in the post-Merge, PBS-enabled environment.

**This work.** We compiled the largest MEV-related dataset to date for comprehensive validation, encompassing all transactions and blocks up to October 2024. We also collected 118.5 million private transactions through Blocknative and Flashbots APIs to ensure comprehensive coverage of modern MEV architectures. We address three critical topics relevant to blockchain enterprise operations: *automated MEV detection ef-*

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00303

*fectiveness* (§VI), *private transaction architecture implications* (§VII), and *MEV ecosystem centralization trends* (§VIII).

**Contribution.** Our work provides several key contributions for blockchain enterprises and platform operators:

- **Enhanced Detection Capabilities.** Automated algorithms achieving 0.6% false positive and 2.4% false negative rates, significantly outperforming existing industrial methods across 2.5B analyzed transactions.
- **Comprehensive Infrastructure Insights.** Analysis covering all major Ethereum evolution phases from 2020-2024, providing guidance for enterprise platform design and risk management strategies during infrastructure transitions.
- **Advanced Threat Detection.** Identification of sophisticated MEV variants including multi-layered burger attacks and conjoined sandwich attacks that collectively extracted 349.4 ETH through private architectures.
- **Operational Intelligence.** Quantitative analysis of private transaction architecture impacts and centralization trends across 118.5M private transactions, informing enterprise strategic decisions for MEV mitigation.
- **Open-Source Resources.** Complete datasets and automated detection scripts are publicly released at link to accelerate research advancement and enable adoption by blockchain enterprises and the broader industrial community.

## II. Background

### A. Ethereum

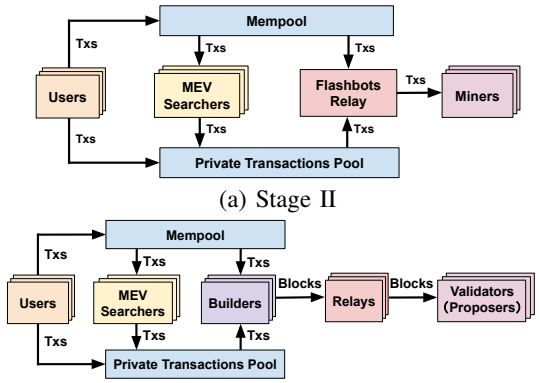
Ethereum is a blockchain platform where smart contracts enable decentralized applications (DApps). All interactions are encoded in transactions, which are packed into blocks by block creators who collect transactions from the public mempool. To avoid public visibility, some accounts use private transactions sent directly to private transaction pools. In September 2022, Ethereum implemented Proposer-Builder Separation (PBS), introducing three roles: *block builders*, *relays*, and *block proposers*. Specifically, block builders are responsible for collecting and assembling transactions into a *block body* and then sending it to relays, while block proposers propose the block after signing the *block header*, sending it to relays and accepting the full block constructed by relays [3]. Such a separation of duty improves the decentralization of Ethereum.

### B. Maximal Extractable Value Extraction

Maximal extractable value (MEV) refers to value extracted during block production beyond standard rewards by selecting or reordering transactions [4]. Accounts pursuing such opportunities are called MEV searchers.

**Front-running & Back-running.** Transaction reordering enables MEV extraction through front-running (placing MEV transactions before target transactions to anticipate effects) and back-running (placing MEV transactions after target transactions to capitalize on market changes).

**Common Types of MEV.** Two mainstream MEV activities dominate: arbitrage and sandwich attacks [2], [5]. Arbitrage exploits price differences between DEXes by swapping tokens across multiple exchanges for profit. Sandwich attacks exploit price slippage by front-running a victim's transaction to alter exchange rates unfavorably, then back-running to profit from the manipulation.




(a) Stage II

(b) Stage III

Fig. 1: Typical MEV architectures.

TABLE I: Overview of our datasets.

| Source | Data Type | Covered Stages * | Time Span | Number |
|---|---|---|---|---|
| Geth | Tx<br>Event<br>Trace<br>Block | Stage I/II/III | 2015.07<br>-2024.10 | 2,568,826,217<br>4,303,522,088<br>8,028,843,264<br>21,000,000 |
| Blocknative API | Private Tx | Stage II/III | 2021.01<br>-2024.10 | 118,518,850 |
| Flahbots API | | | 2021.02<br>-2022.09 | |

* Please refer to §II-B

**MEV Evolution Stages.** MEV activities evolved through three stages: Stage I (Early-stage) featured direct mempool broadcasting leading to Priority Gas Auctions; Stage II (Flashbots-stage) introduced centralized private transaction pools to reduce competition and improve privacy (Fig. 1(a)); Stage III (PBS-stage) decentralized the architecture with multiple relays handling blocks from builders to proposers (Fig. 1(b)).

## III. Data Overview

A comprehensive investigation of MEV activities requires diverse data. Therefore, we made our best effort to collect data from different data sources, as shown in Table I. Firstly, we use the most well-known Ethereum client node, Geth, to sync transactions, emitted events, traces, and blocks. In total, from block #0 to #21,000,000 (packed in Oct. 2024), we parsed over 2.5 billion Ethereum transactions, 4.3 billion emitted events, and 8.0 billion traces. Then, to filter out private transactions, we take advantage of *Blocknative* [6], which offers the most exhaustive historical archive of the time each transaction stays in the mempool. We take the ones with 0 seconds in mempool recorded by Blocknative as private transactions, which strategy has also been adopted by related works [2], [5]. Moreover, for a more complete dataset of private transactions, we also query the *Flashbots API* [7], a public data source that records private transaction bundles passed through the Flashbots relays (see Fig. 1(a)). Consequently, 118.5 million private transactions are acquired. We underline that *this is the largest ever dataset in*

*characterizing MEV activities for now* and *all data collected in this work is publicly available, and no ethical issues should be raised.*

## IV. Methodologies

Our study focuses on two types of MEV activities, *i.e.,* arbitrage and sandwich attack. Previous works [5], [8], [9] widely adopt heuristic methods, which can lead to both false positives and false negatives according to our investigation. We first demonstrate the limitations of previous methods with case studies in §IV-A. Then we propose our robust methods in §IV-B and §IV-C.

### A. Limitations of Existing Methods

According to our investigation on existing heuristic methods [5], [8], [9], two main limitations on detecting arbitrages and sandwich attacks are discovered.

**L1: Ineffectiveness in identifying profitable transactions.** Existing methods directly compare input and output token amounts in DEX swaps, causing false positives. For arbitrages, one of the adopted heuristic rules require positive net token flow where the first swap's input must be less than the final swap's output, but ignore that arbitrageur token balances may decrease. Current methods cannot confirm whether traders achieve real-world profits.

**L2: Insufficient flexibility of adopted rules.** Traditional rules lack flexibility for evolving arbitrage and sandwich attacks. Arbitrage swaps may not execute chronologically, causing false negatives in previous detection methods. Additionally, sandwich attacks may consist of more than two attack transactions, which existing methods cannot detect.

We illustrate existing limitations with two case studies.

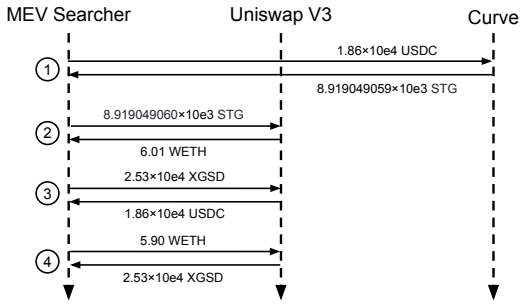
**Case 1: Arbitrage.** Current heuristic methods identify arbitrages using two rules: (1) swaps must be in sequential order, and (2) output token quantity of each swap should equal the input of the next. However, these rules fail in practice. As shown in Fig. 2(a)[1], flashloan services enable non-sequential execution (swaps ③ and ④ are reversed), and token amounts may slightly differ between consecutive swaps (② input exceeds ① output). Despite violating both heuristics, this remains a profitable arbitrage that existing methods incorrectly classify as false negative.

**Case 2: Sandwich Attack.** Traditional sandwich attacks involve two attack transactions surrounding one victim transaction with matching token quantities. However, Fig. 2(b) shows a complex variant with three attack transactions (①, ④, ⑥) and three victims (②, ③, ⑤). The WETH amounts between ① and ④ and SILKROAD amounts between ④ and ⑥ differ significantly. Current methods cannot detect this pattern, yet it generates profit in both SILKROAD and WETH tokens, demonstrating successful sandwich attack execution.

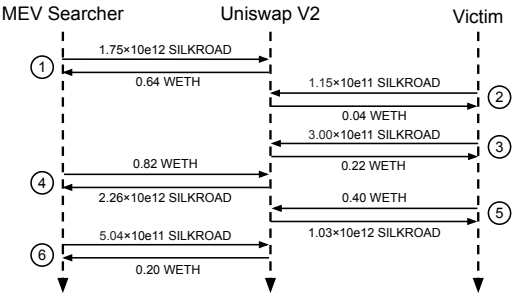
### B. Profitability Identification

Evaluating the profitability for an MEV activities is the gold standard to determine if is successful or not. However, previous methods are clumsy in determining whether a transaction is profitable, as shown in §IV-A. To this end, we propose a *profitability identification algorithm*, as shown in Algorithm 1.

[1]Transaction: Link




(a) Arbitrage




(b) Sandwich attack

Fig. 2: Real-world MEV activities that cannot be identified by existing methods.

**Algorithm 1** Profitability Identification Algorithm

**Input** $tx$ - a transaction or transactions; $g$ - a directed graph where nodes represent tokens and edges represent swaps.
**Output** A boolean value indicating if $tx$ is profitable for the trader.

```
addrBalChange ← getBalChange(tx)
addrBalChange ← rmvIrrAddr(addrBalChange)
tknChange ← aggrTknChange(addrBalChange)
for (tkn_i, tkn_k) in {(tkn_i, tkn_k) | tkn_i, tkn_k ∈ tknChange ∧ tknChange.tkn_i < 0 ∧ tknChange.tkn_k > 0 } do
    ratio ← (tkn_i.out − tkn_i.in)/tkn_i.in
    if ratio < ϵ then
        exInput ← −tknChange.tkn_i
        exAmt ← exchangeToken(tkn_i, tkn_k, g, exInput)
        if tknChange.tkn_k − exAmt > 0 then
            tknChange.tkn_k ← tknChange.tkn_k − exAmt
            tknChange.tkn_i ← 0
if ∀tkn_i ∈ tknChange(tkn_i ≥ 0) then return true
else return false
```

Algorithm 1 examines transaction profitability by assessing whether losses in one token type can be covered by gains in another. The algorithm operates through several key steps to determine overall profitability.

Initially, at L1, `getBalChange` extracts balance changes for involved addresses by parsing ERC-20 transfer events and transaction traces, storing results in `addrBalChange`, a two-

**Algorithm 2** Exchange Token

**Input** *inToken* - exchanged token; *outToken* - output token; $g$ - the directed graph in Algorithm 1; *exInput* - exchange amount.
**Output** *exOutput* - output amount.

```
1: routeList ← DFS(inToken, outToken, g)
2: exOutList ← []
3: for route_i in routeList do
4:   exOutput ← exInput
5:   for swap_i in route_i do
6:     exRate ← swap_i.outputAmt/swap_i.inputAmt
7:     exOutput ← exOutput * exRate
8:   exOutList.append(exOutput)
9: return MAX(exOutList)
```

level dictionary. At L2, `rmvIrrAddr` filters out irrelevant addresses including token contracts, DEX addresses, and the black hole address. Subsequently, L3 aggregates token changes for remaining addresses into `tknChange`, a dictionary where positive amounts represent trader profits and negative amounts represent losses.

The algorithm's core logic begins at L4 with a loop that evaluates whether token gains can offset token losses. This loop iterates through `tknChange` to identify tuples where $tkn_i < 0$ and $tkn_k > 0$, meaning the trader loses $tkn_i$ but gains $tkn_k$. At L5, the algorithm calculates a $ratio$ to determine if the loss is sufficiently small (below threshold $\epsilon$) to be covered by $tkn_k$ profits. When this condition is met, $tkn_i$ is converted to an equivalent value of $tkn_k$, with corresponding balance updates (L9–L11). The `exchangeToken` function at L8 performs this conversion, as detailed in Algorithm 2. After multiple iterations, if all tokens in `tknChange` become non-negative, indicating all losses are covered by gains, the transaction is deemed profitable (L12 and L13).

Algorithm 2 handles token conversion by enumerating all possible swap paths between tokens. At L1, the algorithm traverses a directed graph using DFS to identify all acyclic paths from *inToken* to *outToken*, where each path represents a swap sequence. The nested loop (L3–L8) processes these paths, converting *inToken* based on derived exchange rates. For instance, if $tkn_i$ can be swapped to $tkn_k$ via intermediate tokens $tkn_A$ and $tkn_B$, the resulting $tkn_k$ amounts from both paths are appended to $exOutList$. The algorithm returns the maximum value from $exOutList$ because *exInput* equals $-tknChange.tkn_i$ (see L7 in Algorithm 1). If the maximum obtainable $tkn_k$ amount still yields profit, then regardless of the chosen path, the $tkn_i$ loss is guaranteed to be covered by $tkn_k$ gains.

### C. Identifying Arbitrage & Sandwich Attack

To detect arbitrages and sandwich attacks, we first need to identify if swaps exist in transactions. Thus, we focus on those swap events generated by DEXes[2]. Then, we develop two algorithms to detect arbitrages and sandwich attacks, respectively.

Specifically, to comprehensively identify arbitrages and overcome the second limitation discussed in §IV-A, our method relaxes the heuristic rule for circle detection without requiring sequential orders of swaps and adopts Algorithm 1 to determine the profitability for involved addresses. In other words, if swaps form a loop in any order within the transaction and the transaction is profitable, the algorithm identifies it as an arbitrage. Additionally, as described in §II-B, two types of arbitrages exist. Thus, an arbitrage is classified as front-running if it remains an arbitrage when replayed at the top of the block; otherwise, it is taken as a back-running arbitrage. Such a heuristic was also adopted by [10]. The algorithm is detailed in the external repo: link.

As for sandwich attacks, our method identifies if two attack transactions exist with reverse token swap direction, with victim transactions in the middle. A typical assumption is that only one victim transaction exists. However, we find advanced sandwich attacks where multiple victim transactions are sandwiched between two attack transactions. We refer to this pattern as *multi-layered burger attack* [11]. Additionally, if a sandwich attack involves multiple attack transactions like the one shown in the second case in §IV-A, we term it a *conjoined sandwich attack*. Our method could detect all these three variants. The algorithm is detailed in our repo.

## V. Study Design

With our proposed methods, we raise the following research questions (RQs).

- RQ1 How is the effectiveness of our proposed MEV identification methodologies? What about the full picture of MEV activities in Ethereum?
- RQ2 What are the pros & cons of introducing private transaction architecture?
- RQ3 How does the centralization trend evolve in the MEV ecosystem?

**Dataset Comparison.** After a comprehensive literature review to the best of our efforts, we found 8 studies [5], [8]–[14] are related to detecting arbitrages and sandwich attacks. As for arbitrage MEV activities, among them, only Weintraub [2] has open-sourced the corresponding dataset. As for MEV sandwich attacks, we found all previous studies [3], [12], [15], [16] used the ZeroMEV dataset [17]. Consequently, we take two available MEV datasets for comparison, *i.e.,* Weintraub [2] for arbitrages, and ZeroMEV [17] for sandwich attacks.

**Experimental Setting.** To answer RQ1, we need to apply our MEV identification algorithms on collected transactions and identify if a swap is included (refer to Section IV-C). To achieve this, we review all DEXes listed on DefiLlama [1] and collect their swap patterns. As a result, we have collected 44 swap patterns. We underline *this is the largest swap*

[2]We extend existing works [2], [5] by covering 44 different types of swap patterns, increasing 6.3x and 5.3x, respectively

TABLE II: Overview of identified MEV activities.

| Type | Count |
|---|---|
| Arbitrages | 7,151,723 |
| Sandwich Attacks | 4,989,641 |
| Multi-layered Burger Attacks | 1,125,427 |
| Conjoined Sandwich Attacks | 79,398 |
| Normal Sandwich Attacks | 3,784,816 |
| **Total** | 12,141,364 |

TABLE III: Reasons for exclusively identified arbitrages in our method compared to Weintraub.

| Reasons | Count | % |
|---|---|---|
| Limited swap patterns | 566,076 | 81.6% |
| Strict token amount comparison | 76,506 | 11.0% |
| Strict chronological examination | 26,539 | 3.8% |
| Other | 24,669 | 3.6% |
| **Total** | 693,790 | 100.0% |

*pattern dataset for now*. To answer RQ2, in addition to conducting statistical work before and after the introduction of private transaction architecture, we also analyzed the pros and cons of introducing private transaction architecture using arbitrage MEV activity as an example. Thus, we need to set up a transaction replay environment to distinguish front-running and back-running arbitrages, where we take advantage of Ganache [18]. To answer RQ3, we need to extract the semantics of MEV contracts to filter out the ones with address authorization logic for a more conservative result. Thus, we employed Gigahorse [19], a decompilation and static analysis framework specifically designed for smart contracts, to analyze semantics on the bytecode level.

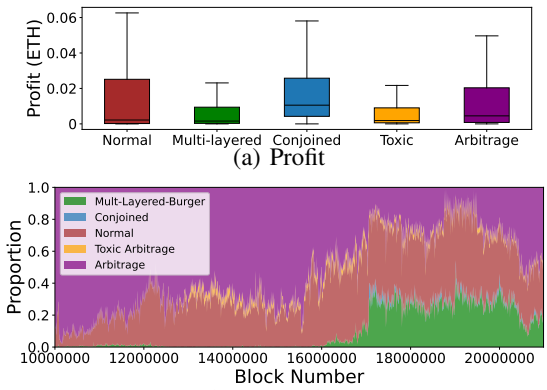
## VI. RQ1: Status of MEV Activities

In this section, we first apply our arbitrage and sandwich attack detection methods in Ethereum. Then, we compare the effectiveness of our methods with existing ones.

### A. Identified MEV Activities

As detailed in Table II, we have successfully identified 12.1 million MEV activities, comprising 7,151,723 arbitrages and 4,989,641 sandwich attacks. There is actually an overlap involving 195,792 transactions. Upon analyzing the behavioral patterns of these overlapping cases, we categorized them as *toxic arbitrages* [20], referring to a type of sandwich attack in which one or both attack transactions simultaneously qualify as profitable arbitrages.

In addition, as we stated in §IV-C, our method can identify sandwich attack variants. In total, there are 1,125,427 (22.6%) multi-layered burger attacks and 79,398 (1.6%) conjoined sandwich attacks (see §IV-C). To depict their characteristics, we first compare the profits among these sandwich attacks and arbitrages, as shown in Fig. 3(a). We found that the median profits of multi-layered burger attacks, toxic arbitrages, and normal sandwich attacks are quite similar, while the profits of conjoined sandwich attacks are almost 5 times more than



(a) Profit
(b) Proportion

Fig. 3: Profit and proportion for MEV activities.

the others. After a comprehensive investigation, we conclude that attackers can use multiple transactions to cause a larger price spread through performing conjoined sandwich attacks, thereby obtaining more profits from the victim transaction. Moreover, we can see that the median profit of arbitrages is around 2x than the one of sandwich attacks. We found that this is because the spread of a sandwich attack is the price difference caused by a single victim transaction swap, while arbitrage can perform multiple swaps between multiple DEXes in one transaction to obtain a larger spread.

We further characterize the proportion of arbitrages and four types of sandwich attacks over time, as shown in Fig. 3(b). We can observe that since #16,000,000, the proportion of multi-layered burger attacks rises sharply, mostly due to an MEV searcher `jaredfromsubway`[3]. This powerful searcher has initiated 64.5% of all sandwich attacks since #16,733,875, of which 58.0% are multi-layered burger and conjoined sandwich attacks. Interestingly, since #20,000,000 (June 2024), the proportion of sandwich attacks started to decrease. We believe this phenomenon is caused by the ETH sharp price decline since then. On the one hand, it makes it easier for price differences to appear between DEXes, providing more arbitrage opportunities. On the other hand, the transaction frequency decreases in a bear market, and the probability of sandwich attacks finding a suitable target transaction decreases.

***Findings:*** Out of 2.5 billion transactions, we identified ∼12.1 million MEV activities. We can observe that complicated MEV activities are significantly fewer in number but can generate higher profits, raising challenges and providing opportunities to MEV searchers.

### B. Comparison with Existing Methods

To evaluate the effectiveness of our proposed methods, we compare our identification results with two datasets as mentioned in §V[4].

*1) Arbitrages Comparison:* Our method identifies 3,910,444 arbitrages compared to Weintraub's 3,358,371 (excluding toxic arbitrages). While 3,216,654 transactions are

[3]Address: Link, which will be studied in §VIII.

[4]We only compare the results on the same block range for fairness.

detected by both methods, 693,790 are exclusively identified by our approach versus 141,717 by Weintraub's.

Analysis of 500 sampled transactions from our exclusively detected arbitrages reveals three primary causes for the discrepancy (Table III). Limited swap patterns account for 81.6% of cases, where Weintraub's scope of recognizable DEX swap patterns is constrained. Strict token amount comparison (11.0%) and strict chronological examination (3.8%) correspond to the limitations discussed in our case study (§IV-A). Despite limited swap patterns dominating, the seemingly intuitive rules still cause Weintraub to mislabel approximately 15% of transactions as false negatives.

For the 141K transactions missed by our method, two main causes emerge. First, our method excludes transactions with final token losses (38,729 transactions, 27.3%). Second, Weintraub indiscriminately labels transactions with swap loops within single DEXes as arbitrages (11,008 transactions, 7.8%). Among the remaining 91,980 transactions, sampling reveals some do generate profits but transfer them to other entities, including addresses labeled as "MEV Bot" by Etherscan. Given blockchain anonymity, our approach conservatively excludes such potentially collusive cases.

Considering the 24,669 unidentified cases as false positives (upper bound) and the 91,980 undetected collusive cases as false negatives (upper bound), our arbitrage detection achieves 0.6% FPR and 2.4% FNR.

*2) Sandwich Attack Comparison:* Our method detects 1,265,929 sandwich attacks versus ZeroMEV's 1,251,072, with 84,944 and 70,088 exclusively identified by each.

Sampling 100 transactions from ZeroMEV's exclusive detections reveals significant issues: 57 are non-sandwich attacks involving various EOAs interacting with public contracts like Uniswap V2 Router without sandwich evidence. Additionally, 19 appear to be sandwich attacks where the beneficiary differs from the initiator; conservatively, we exclude these due to uncertain connections. The remaining 24 transactions result in token losses for initiators, which our method appropriately excludes since MEV activities aim for profit generation.

Manual verification of 100 sampled transactions exclusive to our method confirms all are genuine sandwich attacks.

For performance metrics, the 43% true positive rate from ZeroMEV's exclusive detections yields an upper bound FNR of 2.4% (calculated as $\frac{70{,}088 \times 43\%}{1{,}265{,}929}$). Since our exclusive detections show no false positives in sampling, our method demonstrates negligible FPR compared to ZeroMEV.

***Findings:*** Compared to existing SOTA methods, our method outperforms both of them in identifying arbitrage and sandwich attacks. Under the upper-bound estimation, the FPR and FNR for arbitrage is 0.6% and 2.4%, while these two numbers for sandwich attacks are $\sim$0% and 2.4%.

**Answer to RQ1:** Our identification methods against MEV activities outperform the state-of-the-art ones in both terms of precision and recall. Currently, arbitrage opportunities caused by cryptocurrency price fluctuations and carefully constructed complex MEV strategies are the two main means for MEV searchers to make profits.

## VII. RQ2: Private Pool vs. Mempool

In this section, we aim to depict the pros and cons of introducing private transaction architecture. We first conduct a statistical analysis on various metrics before and after the introduction (§VII-A). Then, we take arbitrage MEV activity as an example to quantitatively illustrate the impact of introducing private transaction architecture (§VII-B and §VII-C)

### A. Overview of MEV Activities across Stages

The red line in Fig. 4 illustrates the number of MEV activities over time. In Stage I, we observe an obvious growth, spanning from May 2020 to October 2020, very likely due to the outburst of the Ethereum DeFi ecosystem. In total, there are 2,698,350 MEV activities in Stage I. Since Stage II, we can distinguish MEV activities conducted in mempool (green line) and private pool (blue line). We can observe a sharp decline in the number of mempool MEV activities, which is wiggling at a low level. Instead, private MEV activities immediately dominated the MEV market. We speculate the reasons for such a significant shift are twofold: (1) private MEV activities with higher transaction fees are prioritized, taking MEV opportunities from mempool; and (2) mempool MEV activities have a lower success rate, forcing MEV searchers to use private transactions (refer to §VII-B). Although the number of mempool MEV activities slightly increased, particularly around the boundary between Stage II and III, the number of mempool MEV activities still decreased after the upgrade was completed. Our statistics show that in Stage II, 36.4% of MEV activities originated from the mempool, while the number went to 2.0% in Stage III. Since 2024 (after block number #18,916,000), we find that only 8,162 (0.2%) MEV activities are from the mempool. This indicates that MEV searchers no longer prefer using the mempool. We believe this is due to the increasing proportion of PBS blocks in Stage III, where private transactions are prioritized and packaged into blocks over mempool transactions in PBS blocks.

***Findings:*** Once the private transaction architecture has emerged, including the centralized Flashbots Relay and the decentralized Relays, due to its processing priority over mempool MEV activities, MEV activities via private transaction pools started to dominate the whole ecosystem.

### B. Case#1: Front-running Arbitrage

In Stage II and III, MEV searchers begin to weigh the options between utilizing the mempool and private transaction architectures. An existing work [2] states that approximately 1.5% of arbitrages conducted through private transaction architecture result in negative returns, thus such an architecture *does not protect these low-resource MEV searchers*. We believe this is a problematic conclusion. In this part, we characterize the actual success rate and expected profit when conducting front-running arbitrages through mempool and private pool.

**Success Rate.** As Flashbots Relay in Stage II and Builders in Stage III verify the status of private transactions in MEV activities, reverted ones will not be included in blocks, allowing MEV searchers to avoid financial losses from paying

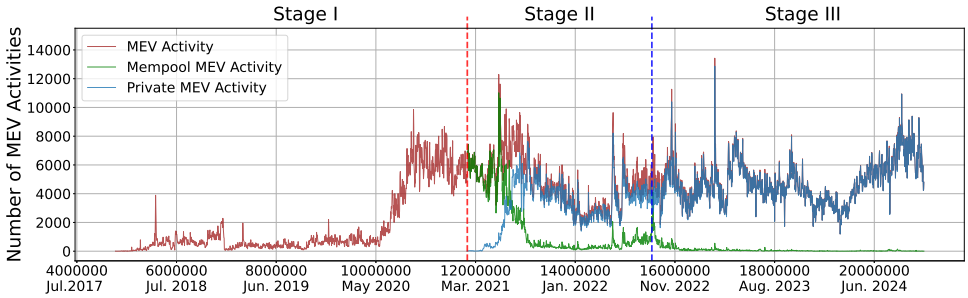
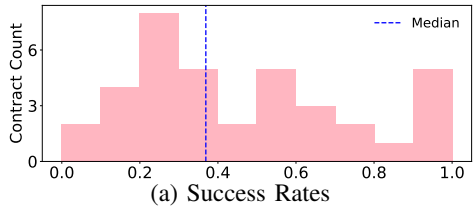
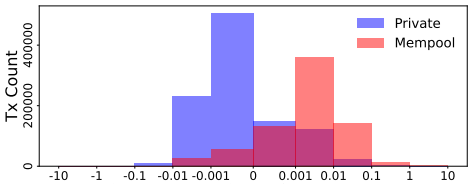


Fig. 4: The number of MEV activities in different stages.



(a) Success Rates



(b) Expected Profit (ETH)

Fig. 5: Distribution of MEV searchers.

the transaction fee [21]. Therefore, the success rate of private MEV transactions is fixed at 100%. For ordinary mempool transactions, however, transaction initiators have to pay the transaction fee whether the transaction is successful or not. We take the front-running arbitrage as an example to evaluate the success rate, whose reasons are twofold: (1) arbitrage represents the most prevalent form of MEV activities, and (2) failed sandwich attacks cannot be reliably detected, as there are multiple transactions in a single MEV activity. We use the method mentioned in §IV-C to detect successful front-running arbitrages. As for identifying failed ones, if a non-identified transaction becomes an arbitrage after replaying it in the front of the block, *i.e.,* another one front-runs it, it is classified as a failed front-running arbitrage.

We replay 20,496,511 transactions from 107 MEV searchers that conduct more than 10,000 arbitrages. Among them, there are 841,937 successful mempool front-running arbitrage, 1,158,966 successful private ones, and 675,065 failed mempool ones. Fig. 5(a) illustrates the distribution of MEV searchers' contracts[5] in terms of success rates. We can observe that the median success rate of top MEV searchers who perform arbitrage through mempool is below 40%. This suggests a high failure rate for conducting MEV activities through mempool, incentivizing MEV searchers to switch to private transaction architecture, which explains the decline in mempool MEV activities in Stage II/III (see §VII-A).

[5]We select 37 MEV searchers' contracts with over 1,000 mempool transactions for success rate analysis.

**Expected Profit.** We introduce Expected Profit (*EP*) to simulate the profit of front-running arbitrage conducted through private transaction architecture if they are broadcasted in mempool. $EP$ is calculated as: $EP = profit \times sr - gp \times fg \times (1 - sr)$ , where $profit$, $sr$, $gp$, and $fg$ refer to the profit, the success rate, the gas price, and the consumed gas, respectively. For 47 MEV searchers that have initiated failed mempool arbitrages, $sr$ is calculated on a contract basis, and $fg$ is the average gas consumption of failed mempool arbitrages. For the remaining 60 MEV searchers' contracts, $fg$ and $sr$ are calculated as the average of the ones of the 47 contracts. *If the expected profit is negative, the MEV searchers tend not to initiate it in mempool*. As shown in Fig. 5(b), we observe that: out of 1,043,009 profitable[6] successful private front-running arbitrages, 744,710 (71.4%) have a negative *EP*. In contrast, out of 734,098 profitable successful mempool arbitrages, only 84,778 (11.5%) have a negative *EP*, underlining the necessity of introducing private transaction architecture. Because around 70% private front-running arbitrages would lead to financial losses to MEV searchers if they are broadcast in the mempool. That is, without private transaction architecture, low-yield MEV opportunities may not be extracted by MEV searchers.

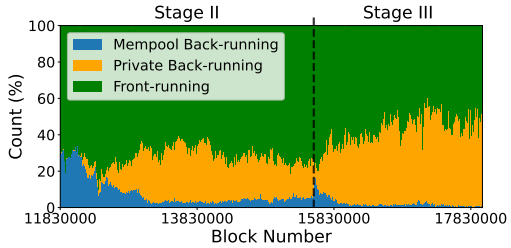
***Findings:*** Contrary to existing conclusions, according to our quantitative experimental results, introducing the private transaction architecture can protect those low-yield MEV searchers as most arbitrage opportunities would yield negative profits if they are executed through the public mempool.

### C. Case#2: Back-running Arbitrage

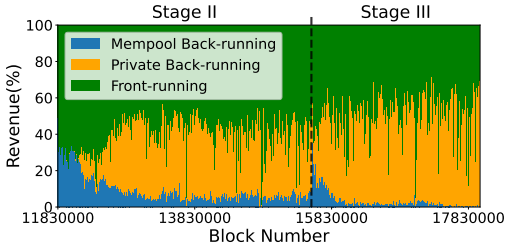
According to McLaughlin [10], back-running arbitrage, due to its ability to extract profit at the same block with the target transaction, has gradually become the primary mechanism for arbitrages due to intensive competition among MEV searchers. Therefore, in this part, we will perform an overview investigation for back-running arbitrages and delve deeper into an emerging phenomenon, *i.e.,* builder-searcher integration.

**Overview of Back-running Arbitrages.** We utilize the method in §IV-C to identify back-running arbitrages. In total, we have identified 1,193,720 back-running arbitrages (866,896 private ones and 326,824 mempool ones) in Stage II and III. Fig. 6 illustrates the ratio of mempool/private back-running arbitrage, as well as the front-running arbitrage identified in

[6]The revenue of the transaction exceeds the gas fee.




(a) Count




(b) Revenue

Fig. 6: The distribution of different types of arbitrages in Stage II and III.

TABLE IV: Labels of the interacted contracts of the target transactions of back-running arbitrages.

| Labels | # Transaction | Proportion(%) |
|---|---|---|
| **Uniswap** | 501,075 | 42.0 |
| **MEV-Bot** | 161,065 | 13.5 |
| **SushiSwap** | 87,701 | 7.3 |
| **1inch** | 80,018 | 6.7 |
| **0xProtocol** | 33,817 | 2.8 |
| **Other** | 330,044 | 27.6 |

§VII-B, in terms of number and revenue. As we can see, back-running arbitrage is gradually becoming the majority. Since Stage III, it has experienced another great boom. We conclude two possible reasons. On the one hand, as private transaction architecture is becoming prevalent, especially in Stage III, which provides transaction ordering services, *e.g.,* Flashbots bundles, lowering the technical barrier to implement back-running arbitrages. On the other hand, the increasing number of MEV searchers leads to intensive competition. The MEV opportunities with high revenue are captured immediately in the same block, *i.e.,* in the form of back-running arbitrage, rather than being left in the next block, which will be captured by front-running arbitrages.

As for the target transactions involved in these back-running arbitrages, we employ the identification method outlined in §IV-C. We find that for 79.8% cases, *no transactions are located between the target transaction and the back-running arbitrage one*. Table IV further shows interacted contracts of all these target transactions with the help of Etherscan. As we can see, 42.0% of target transactions are interacted with Uniswap. Out of them, 59.0% are Uniswap V2 Router [22]. This suggests that such public trading contracts provide a large number of back-running opportunities, which are targeted and captured by MEV searchers.

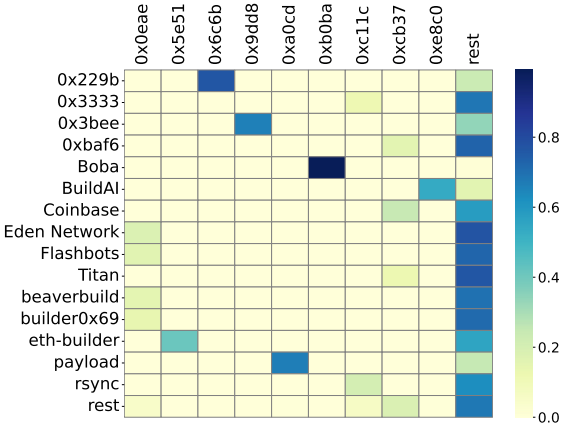



Fig. 7: Proportion of private back-running arbitrages found by MEV searchers (x-axis) included in blocks constructed by builders (y-axis).

**Builder-Searcher Integration.** In Stage III, builders can back-run received transactions to capture MEV profits without competition since opportunities remain invisible to other builders. While applications like MEV-Share [23] and MEV-Blocker [24] return portions of profits to users, some builders retain all profits themselves, referred to as searcher-builder integration [15]. This deviates from PBS's vision that MEV rewards should go to validators.

We analyze 54,273 back-running arbitrages (Sep. 2022 - Aug. 2023), excluding MEV-Share [25] and MEV-Blocker [26] transactions, leaving 33,747 where builders retain all profits. Fig. 7 shows proportion of private back-running arbitrages by searchers (x-axis) in blocks by builders (y-axis). Higher proportions indicate potential collusion. Using 50% threshold, we identify five builder-searcher pairs (Table V) extracting MEV from 2K+ transactions worth 349.4 ETH.

These pairs show median profit margins of 0.3%, 0.2%, 46.1%, 80.6%, and 44.7% respectively, significantly deviating from the typical 6.3% margin. Investigation reveals: (1) the last three pairs avoid competition through builder protection; (2) the first two pairs transfer profits to builders (median block revenue 1.36 ETH). Both of these situations indicate that *builder-searcher integration enhances builders' competitiveness and profitability in block building*. Interestingly, regarding the forth pair, its builder is named Boba Builder, and the MEV searcher contract is also named Boba. According to its website [27], we have not seen any statement indicating that it engages in back-running arbitrages for private transactions.

***Findings:*** Back-running dominates due to intense searcher competition. Builders can stealthily perform MEV activities via private transaction architecture, challenging current PBS mechanisms.

**Answer to RQ2:** Private transaction architecture irreversibly becomes the majority. On the positive side, it could provide protection for low-yield MEV searchers when performing arbitrage, enhancing the decentralization. On the negative side, it breeds builder-searcher collusion, violating the original

TABLE V: Participation of builders in back-running arbitrages in Stage III.

| Builder Address | MEV Searcher Address | #Transactions | Revenue (ETH) |
|---|---|---|---|
| 0xbd3afb0bb76683ecb4225f9dbc91f998713c3b01 | 0xe8c060f8052e07423f71d445277c61ac5138a2e5 | 1069 | 38.3 |
| 0x3bee5122e2a2fbe11287aafb0cb918e22abb5436 | 0x9dd864d39fbfdf7648402746263e451cd4f36af0 | 590 | 134.0 |
| 0xce0babc8398144aa98d9210d595e3a9714910748 | 0xa0cdf33c150b936d0091969c694da3cfeae18446 | 381 | 45.8 |
| 0x3b64216ad1a58f61538b4fa1b27327675ab7ed67 | 0xb0bababe78a9be0810fadf99dd2ed31ed12568be | 377 | 46.1 |
| 0x229b8325bb9ac04602898b7e8989998710235d5f | 0x6c6b87d44d239b3750bf9badce26a9a0a3d2364e | 141 | 85.3 |

position of PBS.

## VIII. RQ3: Centralization Trend

Although decentralization is the starting point of blockchain technology, many existing studies (such as block constructing and smart contract creating) have demonstrated the trend of centralization [28]–[30]. In this section, we conduct a chronological analysis to quantitatively investigate the centralization trend in terms of performing MEV activities.

### A. Clustering Method

To investigate relationships among MEV participants, we perform address clustering focusing on private MEV contracts with authentication mechanisms. Unlike public contracts (*e.g.,* Uniswap router) where caller relationships cannot be determined, private contracts restrict access to authorized addresses with established relationships.

From 12,041 identified MEV contracts, we filter out public ones using Etherscan labels, then apply Gigahorse to detect address authorization logic following Sun *et al.* [31]. This yields 7,019 private MEV contracts for analysis. We apply five heuristic clustering rules to group related EOAs and contracts:

1) Private MEV contracts deployed by an EOA are related.
2) MEV searchers conducting activities via a MEV contract are related.
3) When an EOA initiates transactions through a private MEV contract to transfer Ether or ERC-20 tokens to another address, these three addresses are related[7].
4) Hard-encoded addresses within private MEV contracts are related to that contract.
5) Addresses initiating a sandwich attack are related.

These conservative heuristic rules prioritize avoiding false positives over false negatives, meaning our clustering results represent a *lower bound* of real-world relationships.

### B. Clustering Results

We analyzed clustering results from December 2017 to October 2024, visualized in Fig. 8 and Fig. 9 with color-coded clusters. The data reveals dramatic growth in MEV participation: clusters increased from 34 in 2018 to 640 in 2024, reflecting intensified competition incentives.

To quantify centralization trends, we calculated the Herfindahl-Hirschman Index (HHI): $HHI = 10,000 \times \sum_{i=1}^{N} s_i^2$, where $s_i$ represents cluster $i$'s MEV activity share and $N$ is the total cluster count. Higher HHI values indicate greater centralization. The HHI values show a notable pattern: starting at 6299.6 in 2018, centralization decreased through 2021 (reaching 272.6) as more participants entered the ecosystem, but then resurged significantly from 2023 onwards, reaching 3855.5 in 2024. This recent centralization trend is particularly evident in Fig. 9, where a single cluster (blue) accounts for 43.1% of all MEV activities.

This dominant cluster contains 9 MEV searchers and 29 private MEV contracts, with 28 contracts executing 60,797 arbitrages and 4 contracts performing 1,565,087 sandwich attacks. Notably, one address is labeled `jaredfromsubway`, a well-known sandwich attack bot. Analysis reveals that 33.9% of this cluster's activities involve MEME tokens (*e.g.,* DOGE [32]), with 52.2% of sandwich attack initiations swapping MEME tokens. This focus on MEME tokens leverages their high volatility and speculator popularity, while the cluster's large token holdings provide liquidity advantages that eliminate competition in MEME-related MEV opportunities.

**Answer to RQ3:** By cooperating with other MEV searchers through private MEV contracts, clusters can become oligopolies by holding a large number of certain tokens and then make profits through performing MEV activities, which definitely signifies a tendency towards centralization.

## IX. Limitations

We acknowledge certain limitations of our method. First, in terms of data collection, although we have compiled the most extensive dataset of MEV transactions for arbitrage and sandwich attacks, we still rely on previous methods (see §III) to detect private transactions, which may lead to false positives. Second, regarding evaluating our algorithms for identifying arbitrages and sandwich attacks, our benchmark for manually judging whether a transaction is one of these types is as follows. For MEV activities, they must include addresses that show a profitable balance change in tokens, and the addresses should not be irrelevant ones (such as the Uniswap Router, which is not controlled by any traders). The transactions must also exhibit the characteristics of each type of MEV: sandwich attacks, where the attacking transactions are positioned before and after the victim's transaction, and arbitrage, which involves a cyclic swap pattern. Although this manual inspection process is applicable to most MEV transactions, we still cannot guarantee the precision due to the lack of ground truth.

## X. Related Work

**MEV identification.** Most studies [2], [5], [8], [9], [13] use heuristic methods limited to popular DEXes and specific transaction types. Mclaughlin et al. [10] broadens DEX coverage using ERC-20 transfer events but still relies on heuristics. Li et al. [11] detects MEV via transaction aggregation on Flashbot bundles. Our work differs by broadening DEX coverage,

[7]We exclude irrelevant addresses like coinbase and blackhole.

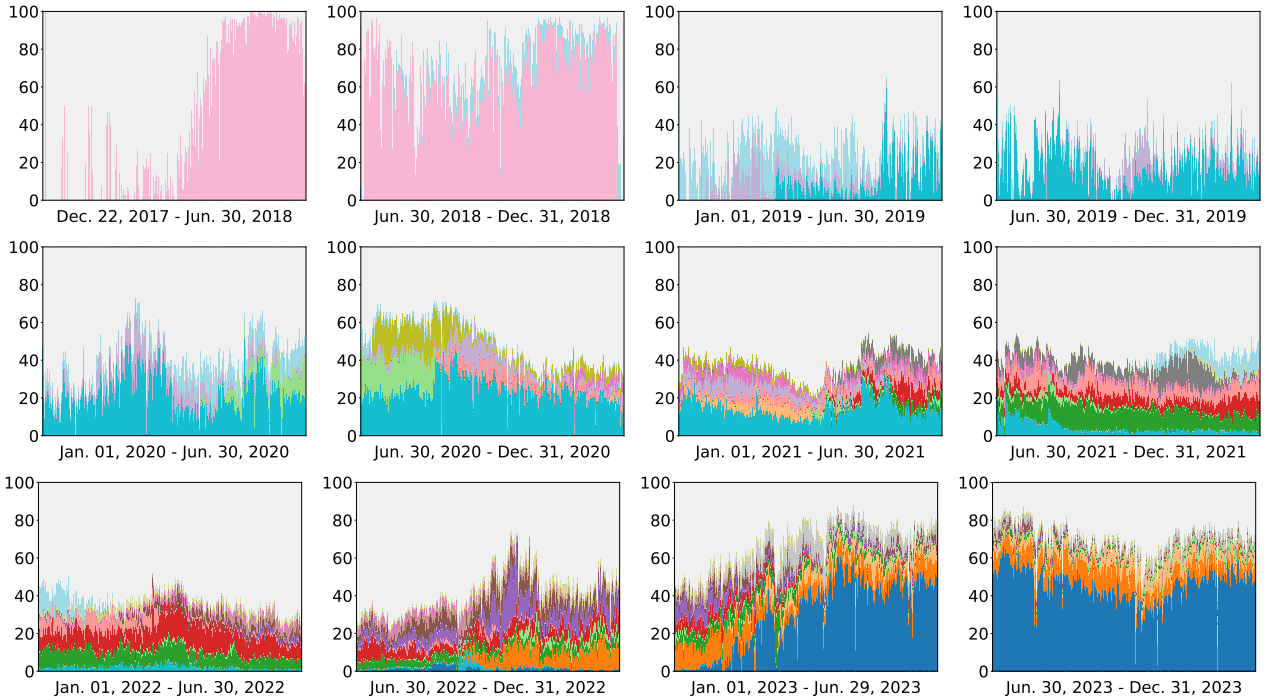



Fig. 8: The MEV searcher clusters from December 2017 to December 2023 (Top 20 colored).

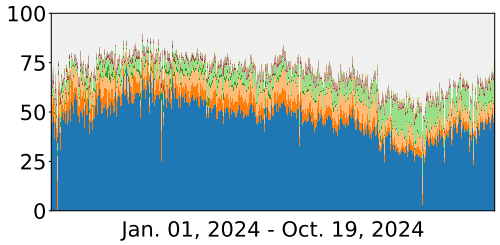



Fig. 9: The proportion of different clusters of MEV activities from January 2024 to October 2024.

refining identification rules for complex transaction subtypes, and addressing violations of conventional heuristics.

**Private transaction architecture.** Pre-PBS studies [2], [13], [33], [34] examined private transaction architecture's relationship with MEV. Post-PBS research [3], [35] explores PBS characteristics and MEV impact, with Wahrstätter et al. [35] analyzing builder submission timing strategies. Unlike their focus on architectural impacts, our research examines how private transaction architectures influence MEV searcher profits and reveals builder-searcher integration patterns.

**MEV applications.** Various applications mitigate MEV-induced losses through front-running prevention via DEX design [36]–[40] or transaction reordering [41]–[43]. Our study reveals that builders use back-running to capture MEV from private transactions without user knowledge, despite services like MEV-Share offering profit redistribution. This highlights ongoing challenges in fair MEV profit distribution.

## XI. Conclusions

This work presents an automated MEV detection framework addressing critical operational challenges for blockchain enterprises. Our graph-based profitability identification algorithm replaces inflexible heuristic methods, achieving 0.6% false positive and 2.4% false negative rates, significant improvements over existing industrial approaches. Validated on 21 million Ethereum blocks containing 2.5 billion transactions, our system identifies 12.1 million MEV activities, including 1.2 million previously undetectable advanced variants. Key findings provide actionable enterprise insights: private transaction architectures protect 71.4% of low-yield MEV opportunities rather than harming participants, contradicting previous assumptions. However, we identify builder-searcher collusion involving 2,000+ transactions worth 350 ETH, highlighting compliance risks. Centralization analysis reveals a single oligopoly controlling 43.1% of MEV activities. Our production-ready framework provides blockchain enterprises with essential tools for MEV monitoring, risk assessment, and compliance management across major Ethereum evolution phases. The automated approach enables enterprises to adapt detection capabilities to emerging MEV strategies while maintaining operational effectiveness in rapidly evolving DeFi environments.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (grant No.62572209 and No.62502414), the Hubei Provincial Key Research and Development Program (grant No.2025BAB057), and HKPolyU Grant (P0054144). Haoyu Wang is the corresponding author.

## REFERENCES


[1] "DefiLlama," https://defillama.com/, 2020.

[2] B. Weintraub, C. F. Torres, C. Nita-Rotaru, and R. State, "A flash (bot) in the pan: measuring maximal extractable value in private pools," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 458–471.

[3] L. Heimbach, L. Kiffer, C. Ferreira Torres, and R. Wattenhofer, "Ethereum's proposer-builder separation: Promises and realities," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, 2023, pp. 406–420.

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

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

[6] "Blocknative Mempool Data Program," https://docs.blocknative.com/mempool-data-program, 2023.

[7] "Flashbots API," https://blocks.flashbots.net/, 2021.

[8] Y. Wang, Y. Chen, H. Wu, L. Zhou, S. Deng, and R. Wattenhofer, "Cyclic arbitrage in decentralized exchanges," in *Companion Proceedings of the Web Conference 2022*, 2022, pp. 12–19.

[9] C. F. Torres, R. Camino *et al.*, "Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1343–1359.

[10] K. C. McLaughlin Robert and G. Vigna, "A large scale study of the ethereum arbitrage ecosystem," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 3295–3312.

[11] Z. Li, J. Li, Z. He, X. Luo, T. Wang, X. Ni, W. Yang, X. Chen, and T. Chen, "Demystifying defi mev activities in flashbots bundle," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 165–179.

[12] S. Park, W. Jeong, Y. Lee, B. Son, H. Jang, and J. Lee, "Unraveling the mev enigma: Abi-free detection model using graph neural networks," *Future Generation Computer Systems*, vol. 153, pp. 70–83, 2024.

[13] J. Piet, J. Fairoze, and N. Weaver, "Extracting godl [sic] from the salt mines: Ethereum miners extracting value," *arXiv preprint arXiv:2203.15930*, 2022.

[14] S. Wu, D. Wang, J. He, Y. Zhou, L. Wu, X. Yuan, Q. He, and K. Ren, "Defiranger: Detecting price manipulation attacks on defi applications," *arXiv preprint arXiv:2104.15068*, 2021.

[15] L. Heimbach, V. Pahari, and E. Schertenleib, "Non-atomic arbitrage in decentralized finance," *arXiv preprint arXiv:2401.01622*, 2024.

[16] S. Yang, F. Zhang, K. Huang, X. Chen, Y. Yang, and F. Zhu, "Sok: Mev countermeasures: Theory and practice," *arXiv preprint arXiv:2212.05111*, 2022.

[17] "Zero mev datasets," https://zeromev.org/, 2021.

[18] "Ganache: A tool for creating a local blockchain for fast ethereum development," https://github.com/trufflesuite/ganache, 2022.

[19] N. Grech, L. Brent, B. Scholz, and Y. Smaragdakis, "Gigahorse: thorough, declarative decompilation of smart contracts," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019, pp. 1176–1186.

[20] Zero MEV, "Toxic arbitrage," https://info.zeromev.org/terms.html#toxic-arbitrage, 2022.

[21] "Flashbots key considerations," https://docs.flashbots.net/flashbots-protect/quick-start#key-considerations, 2023.

[22] "Uniswap v2 router," https://docs.uniswap.org/contracts/v2/reference/smart-contracts/router-02, 2021.

[23] "MEV-Share Docs," https://docs.flashbots.net/flashbots-protect/mev-share, 2023.

[24] "MEV-Blocker," https://mevblocker.io/, 2023.

[25] Flashbots, "Mev-share data," https://flashbots-data.s3.us-east-2.amazonaws.com/index.html, 2024.

[26] "Mev-blocker data," https://dune.com/cowprotocol/mev-blocker, 2024.

[27] "Boba builder docs," http://boba-builder.com/, 2023.

[28] M. Bahrani, P. Garimidi, and T. Roughgarden, "Centralization in block-building and proposer-builder separation," in *International Conference on Financial Cryptography and Data Security*. Springer, 2024, pp. 331–349.

[29] A. Capponi, R. Jia, and S. Olafsson, "Proposer-builder separation, payment for order flows, and centralization in blockchain," *Payment for Order Flows, and Centralization in Blockchain (February 12, 2024)*, 2024.

[30] Z. Lin, J. Chen, J. Wu, W. Zhang, and Z. Zheng, "Definition and detection of centralization defects in smart contracts," *arXiv preprint arXiv:2411.10169*, 2024.

[31] T. Sun, N. He, J. Xiao, Y. Yue, X. Luo, and H. Wang, "All your tokens are belong to us: Demystifying address verification vulnerabilities in solidity smart contracts," in *The 33rd USENIX Security Symposium*, 2024.

[32] "Doge token," https://dogecoin.com/, 2018.

[33] A. Capponi, R. Jia, and Y. Wang, "The evolution of blockchain: from lit to dark," *arXiv preprint arXiv:2202.05779*, 2022.

[34] X. Lyu, M. Zhang, X. Zhang, J. Niu, Y. Zhang, and Z. Lin, "An empirical study on ethereum private transactions and the security implications," *arXiv preprint arXiv:2208.02858*, 2022.

[35] A. Wahrstätter, L. Zhou, K. Qin, D. Svetinovic, and A. Gervais, "Time to bribe: Measuring block construction market," *arXiv preprint arXiv:2305.16468*, 2023.

[36] L. Zhou, K. Qin, and A. Gervais, "A2mm: Mitigating frontrunning, transaction reordering and consensus instability in decentralized exchanges," *arXiv preprint arXiv:2106.07371*, 2021.

[37] M. Ciampi, M. Ishaq, M. Magdon-Ismail, R. Ostrovsky, and V. Zikas, "Fairmm: A fast and frontrunning-resistant crypto market-maker," in *International Symposium on Cyber Security, Cryptology, and Machine Learning*. Springer, 2022, pp. 428–446.

[38] C. McMenamin, V. Daza, M. Fitzi, and P. O'Donoghue, "Fairtradex: A decentralised exchange preventing value extraction," in *Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security*, 2022, pp. 39–46.

[39] C. Baum, B. David, and T. K. Frederiksen, "P2dex: privacy-preserving decentralized cryptocurrency exchange," in *International Conference on Applied Cryptography and Network Security*. Springer, 2021, pp. 163–194.

[40] "Cow protocol overview," https://docs.cow.fi/, 2021.

[41] R. Khalil, A. Gervais, and G. Felley, "Tex-a securely scalable trustless exchange," *Cryptology ePrint Archive*, 2019.

[42] I. Bentov, Y. Ji, F. Zhang, L. Breidenbach, P. Daian, and A. Juels, "Tesseract: Real-time cryptocurrency exchange using trusted hardware," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1521–1538.

[43] D. Malkhi and P. Szalachowski, "Maximal extractable value (mev) protection on a dag," *arXiv preprint arXiv:2208.00940*, 2022.