

# TL;DR

- We analyzed the difference in swap price per block position, recently known as reordering slippage, for Ethereum blocks from September to October 2023. Our analysis reveals significant reordering slippage across different MEV actors and transaction sizes.
- We find reordering slippage costs users \$9m in losses on Uniswap. Of that, reordering slippage negatively affects 72% of swaps and 92% of large-size swaps over \$250k.
- We analyze blockspace in 1% increments we define as block “ticks”, and find that swaps placing right after the top 10% of the block at 10-25% block ticks experience the most negative impact from reordering slippage.
- Telegram trading bots in particular suffered \$2m in reordering slippage, revealing susceptibility to transaction ordering effects, suggested by their use of non-atomic bundles and favoring swaps in illiquid pools.
- This study highlights the need to consider reordering slippage in mev measurements and tooling as there can be up to 6x difference in execution price through a block in some cases

## Abstract

Our study examines reordering slippage costs and bidding behaviors in Ethereum blockspace from September to October 2023. We reveal and analyze transaction ordering impacts on Uniswap V2 and V3 pools' swap prices, quantifying intra-block reordering slippage. Our hypothesis suggests a competitive equilibrium for reordering slippage towards the top of a block, and our data shows most reordering slippage occurs beyond the first 10% of block ticks, defined as 1% blockspace intervals. We observe a significant surge in reordering slippage between the 10-25% range of blockspace ticks, highlighting the critical consequences of not securing premium blockspace.

This study underscores a gap in mev data and tooling availability to measure reordering slippage. By shedding light on this metric, we reveal the competitive dynamics among bundlers for blockspace and the varying economic impact of executing swaps at different block positions.

## Reordering Slippage

Reordering slippage refers to the difference in execution price swaps receive depending on their block position and was initially introduced by Uniswap Labs in their paper analyzing total costs for Uniswap swaps

[here](#)

.

The

[Spectra of MEV by Angeris et al](#)

provides a formal definition of the "cost of MEV" which quantifies the excess profits a proposer can extract by manipulating transaction order, compared to a random order. Suppose

$S$

$T$

$S_T$

$S$

$T$

is a symmetric group on the set of all transactions

$t$

$\in$

$T$

$t \in T$

t

∈

T

being considered for the next block inclusion. Then a finalized block

b

∈

S

T

$b \in S_T$

b

∈

S

T

of size

|

n

|

|n|

|

n

|

is defined as

b

n

=

$\sum$

i

=

0

i

=

n

t

i

$$b^n = \sum_{i=0}^n t_i$$

b

$n$

$=$

$\sum$

$i$

$=$

$0$

$i$

$=$

$n$

$t$

$i$

where

$t$

$i$

$t_i$

$t$

$i$

refers to the transaction index in

$b$

$n$

$b^n$

$b$

$n$

. Please refer to the

[Spectra of MEV paper](#)

for a more in-depth mathematical discussion of the benefits of a definition based on the symmetric group.

## Calculating Reordering Slippage via Bisection Index

While the Uniswap research assumed a random ordering method to calculate reordering slippage for a given price

$p$

$p$

$p$

, we use a bisection method outlined below to quantify a tighter bound for reordering slippage.

We bisect the block continuously at

$j$

j

j

where

0

<

j

<

n

$0 < j < n$

0

<

j

<

n

. Earlier indexes represent the top blockspace and the later indexes represent the bottom blockspace.

For a given liquidity pool

I

$\in$

L

$I \setminus \text{in } L$

I

$\in$

L

with

m

m

m

swaps in a block

b

n

$b^n$

b

n

, let

P

I

=

$$\begin{aligned}
 & \sum_{i=0}^m p_i \\
 & P_I = \sum_{i=0}^m p_i \\
 & P_I \\
 & = \sum_{i=0}^m p_i
 \end{aligned}$$

be the subset of all pool swap execution prices where

$$\begin{aligned}
 & p_i \\
 & p_i \\
 & p_i
 \end{aligned}$$

is the execution price at block index

$$\begin{aligned}
 & i \\
 & i \\
 & i
 \end{aligned}$$

. For an execution price

p  
b  
o  
t  
∈  
P  
l  
(  
j  
,  
m  
]  
 $p_{\text{bot}} \in P_{l_{\text{bot}}(j,m)}$

p  
b  
o  
t  
  
∈  
P  
l  
(  
j  
,  
m  
]  
]

in the bottom block bisection, we define reordering slippage for  
p  
b  
o  
t  
r  
s  
 $p_{\text{bot}_{rs}}$   
p

b  
o  
t  
rs

as:  
Note that

p  
b  
o  
t  
 $p_{\text{bot}}$   
p  
b  
o  
t

is not weighted by volume. A future research direction would be to consider using a volume weighted expectation price.

**Standardized Block Positioning**

The natural question is what is the best bisection value

j  
j  
j  
to choose? Specifically, we partition blocks uniformly into integers called  
t  
i  
c  
k  
s  
 $\in$   
[  
0  
,  
100  
]  
ticks  $\in [0,100]$

$$t_i \in [0, 100]$$

where tick is calculated as

$$t_i = round(\frac{t_i}{t_n})$$

$$t_i = ro_u n$$



d

(

t

n

t

i

)

for a block

b

n

$b^n$

b

n

where

n

n

n

is the number of transactions in the block and

t

i

c

k

tick

t

i

c

k

is rounded to the nearest integer. For example, without a tick, there is no way to compare a transaction

t

5

$t_5$

t

5

between blocks

$b$

50

$b^{50}$

$b$

50

and

$b$

150

$b^{150}$

$b$

150

. However, with ticks, we would know that

$t$

5

$t_5$

$t$

5

would be in ticks 10 and 3 respectively.

In the empirical analysis, we choose the bisection index as the relative transaction index at the 10% tick value. This serves as a good heuristic. A large proportion of Uniswap swaps occur in the first 10% of blockspace ticks, creating a lot of price discovery in this part of the block.

When there are no swaps on both sides of the bisection, reordering slippage can't be calculated using a 10% tick bisection index. Making the bisection index dynamic to calculate reordering slippage differently is left as a future research direction.

## Bidding Mechanics Overview

There are two main ways to bid to block builders for blockspace - by EIP-1559

`max_priority_fee_per_gas`

and submitting bids directly to builders via

`[eth_sendBundle`

]([https://docs.flashbots.net/flashbots-auction/advanced/rpc-endpoint#eth\\_sendbundle](https://docs.flashbots.net/flashbots-auction/advanced/rpc-endpoint#eth_sendbundle))

. The majority of transactions bid for block inclusion using

`max_priority_fee_per_gas`

. There are two components that determine the cost of an EIP-1559 transaction as outlined

[here by Blocknative](#)

. First there is the

`base_fee_per_gas`

, which is the network fee that gets burned. The second is

$\text{max\_priority\_fee\_per\_gas}$

, which is the fee that the block builder collects and is added to the

$\text{base\_fee\_per\_gas}$

to make a blockspace bid more competitive. Thus the

$\text{gas\_price}$

component is the sum of the block

$\text{base\_fee\_per\_gas}$

and priority fee and is calculated as

$\text{gas\_used}$

• (

$\text{base\_fee\_per\_gas}$

+

$\text{max\_priority\_fee\_per\_gas}$

).

Certain users gain direct bidding access to builders by using bots to submit transactions - referred to as bundlers. While normal users bid for block inclusion, bundlers bid for a specific inclusion order for a set of transactions. Submitting a single bid for a bundle of transactions is also more efficient than placing a separate bid for each transaction.

## Searchers

We focus on analyzing the bidding power for specific on-chain searcher strategies - arbitrages, sandwiches, and liquidations. We utilize the libMEV database that classifies searcher bundles. A more in-depth methodology for libMEV searcher labeling

[can be found here](#)

. Note we leave the analysis of cex-dex integrated searchers as a future area of research.

## Telegram Bots

Telegram bots bring advanced algorithmic trading capabilities and automated order execution to mainstream traders using Telegram's platform as the primary interface. These bots specialize in liquidity pool sniping and act as a frontend for users to submit bids directly to builders similar to searchers. However since Telegram users submit transactions independently, this makes Telegram bundles non-atomic at the transaction level.

The below charts visualize the non-atomic bundle nature for a single snipe event on the Uniswap v2 FWS/WETH pool at block 18256601. In the leftmost chart, there is a 6x difference in execution price between the first and last transactions. The gap between the two dotted lines indicates two separate bundles - one first one submitted was submitted by Banana Gun and the second one was submitted by Maestro.

The middle chart shows the correlation between bid amount and transaction ordering. There is not a direct causal link between higher bid and lower transaction placement - the higher bid does not guarantee a better transaction index placement. In particular, the first transaction in the Maestro bundle bid .08 ETH but ended up in around transaction index 39. However, if that same transaction was submitted in the Banana Gun bundle, it would have ended up around transaction index 20.

Finally, the right-hand chart shows the bid amount compared to the swap amount they received, denominated in WETH. There are many high values on the y-axis that indicate users bid .12 WETH to receive .03 WETH worth of FWS. In fact, the majority of users bid more than what they ultimately received. This behavior among Telegram bot users indicates a willingness to incur transaction costs that are multiple times higher than the received amount, suggesting they perceive these costs as justifiable.

Telegram bots have emerged as major blockspace buyers on Ethereum. Router addresses like Banana Gun and Maestro V2 rank among the top gas consumers as of November 2023.

We use the following router addresses:

Note that Alfred router contract was not included as it was created in the first week of November (Alfred Router -

0x8e30dead12d19228cf9cdc984f237f0ad00df195

). Similarly a new Banana Gun Router V2 contract was deployed in December (

0x3328F7f4A1D1C57c35df56bBf0c9dCAFCA309C49

).

## Calculating Bundle Bids

We use the cryo

balance\_diffs

dataset to check the ETH transfers between addresses to the coinbase address. Searcher and telegram bot addresses that send ETH to the coinbase address are then attributed as bundle bids. To calculate the searcher bid cost for every transaction index in the bundle, we average the bundle bid cost across all of the bundle transactions and add the cost to the EIP-1559 cost. We calculate Telegram bots' total bidding cost by adding the bid cost to the EIP-1559 cost since each bid amount is already separated by transaction.

## Data Methodology

We used

[cryo](#)

to retrieve 433.605k blocks and 61.599m transactions for September and October 2023 from block 18039828 to 18473546. Transactions were identified as either public or private using the

[flashbots mempool archive data](#)

using

[mempool dumpster](#).

54.34m transactions were identified as public and 7.25m were private, making private transactions account for ~11.9% of all transactions. Out of these private transactions, MEV and Telegram bots made up about ~2.4m of these private transactions, about 33%. Below is a table that shows the breakdown of transactions:

Uniswap V3 swap data and volume were collected from the

[streamingfast subgraph](#)

using

[subgrounds](#)

. Uniswap V2 swap data was collected using cryo, using a modified version of this

[example script](#)

. In total, there were 12.996m total swaps with 2.225m coming from Uniswap V3 and 10.770m from Uniswap V2 respectively. We remove cases where there are blocks with swaps in only one section of the bisection index, reducing our count to 1.24m swaps.

Since all of the reordering slippage was calculated using float types, the calculations are susceptible to floating point errors, especially for tokens with very large or small values. We restricted the reordering slippage analysis to +/-25% to heuristically filter out floating point errors, reducing the dataset to 961.703k swaps across 1,102 unique Uniswap V3 and 17,682 unique Uniswap V2 pools.

## Block Positioning Analysis - Bidding and Transaction Placement

We find that Telegram bots are consistently outbidding MEV searchers in the top 10% of blockspace, exhibiting higher per-transaction bids and greater transaction volume between the 10-40% block ticks. Throughout the blocks we examined, Telegram bots bid 10,983 ETH over 1.318m transactions (.0082 ETH per transaction) and MEV searchers from the

[libMEV](#)

dataset bid 10,635 ETH over 1.435m transactions (.0074 ETH per transaction).

We analyze bidding behavior overlaid on block positioning. The chart below shows the competitive nature at different block ticks, and that Telegram bot bids are noticeably higher than searcher bids in the top portion of the block.

There are spikes in bidding in sporadic areas of the block for MEV searcher bids. This is due to the fact that builders will place searcher opportunities anywhere in the block, not always at the top.

For example, in block

[18182541](#)

, the HEGIC deployer contract submitted a public transaction for a 100 ETH swap. A searcher bot picked this transaction up from the public mempool and submitted a backrun bundle with about ~20 subsequent transactions. The builder ultimately decided to place this bundle in the middle of the block, starting at the 40% block tick range.

The next chart shows the number of transactions that get included in different areas of the block. As expected, the number of searchers and telegram bundle transactions cluster towards the top of the block.

More surprising is that Telegram bots consistently submit more transactions than searchers in some parts of the top 20% block ticks, which likely attributes to the Telegram bot group's higher bidding power. This block tick range is also where reordering slippage tends to be the highest as seen later.

## Empirical Quantification of Reordering Slippage

We look at the likelihood of how pervasive reordering slippage is for Uniswap swap data. We find that 700k swaps were affected negatively and 271k swaps were affected positively. 72% of the swaps were negatively affected for \$12.811m. The other 28% were positively affected by \$3.749m. Swaps that were negatively affected had median sizes of \$158.19 with a median slippage of \$4.44. Swaps positively affected had median sizes of \$105 with a median slippage of \$3.80.

We find that some pools are more sensitive to reordering slippage than others, which indicates a relation to current block liquidity conditions and volume demand. For example, ~95% of the swaps in the WETH/USDC Uniswap V3 pool were negatively affected for a total of \$299k. Further analysis of the impacts of reordering slippage across pools is warranted.

## Overall Uniswap Reordering Slippage

This section examines reordering slippage from the point of view of Uniswap swaps. Reordering slippage totaled \$9m, with \$7.4m coming from Uniswap V2 and \$1.6m from Uniswap V3. Average reordering slippage was 1.7% and \$15.12 on Uniswap V3 and 5.9% and \$7.99 on Uniswap V2. The below table shows a breakdown of reordering slippage by percent and USDC value as well as transaction type - public and private. \$5.98m of the \$9m reorder slippage was driven by privately submitted transactions on Uniswap V2.

Large Uniswap V3 pools like WETH/USDC, WETH/WBTC, and WETH/USDT exhibit lower mean reordering percentages at .05%, .09%, and .04% respectively for swaps over \$250k.

The charts below show reordering slippage volatility throughout the bottom 90% of the block. While there is larger reordering slippage USDC loss towards the top block ticks, the reordering slippage remains more persistent throughout the block.

Uniswap V2 has a significantly higher amount of reordering slippage, both on a percent and dollar basis. The right-hand chart visually indicates a ~1% lower bound for Uniswap v3 pools and ~5% for Uniswap v2 pools for the bottom 90% of the block, where the highest reordering slippage % is towards the top of the block. The left-hand chart shows that the dollar value for reordering slippage is highest towards the top 10-20% of ticks in block, and falls off significantly after, indicating the fierce contention in blockspace in those ticks compared to others through the block.

This is likely a result of the fact that Uniswap V2 has a large number of pools with illiquid tokens where telegram bots compete against each other for liquidity pool snipes in the top block ticks. Similar to

[Uniswap Lab's findings](#)

, we find that slippage is the primary cost for large trades.

## Telegram Bot Reordering Slippage on Uniswap

Next, we focus on Telegram bots, which reveal to be major victims of reordering costs on Uniswap as explained below. We find \$2m in reordering slippage attributable to these bots, revealing their high sensitivity to transaction sequencing. The combination of illiquidity and Telegram bot bundle non-atomicity further exacerbates reordering slippage. The other major factor is that the choice of bisection index plays a major role in the resulting calculations. It is likely that the bisection index needs to be an even smaller number to be able to realize a profitable trade. This would mean that the 10% tick is more like a

conservative lower bound for the realized cost of reordering slippage.

In the table below, Unibot has the lowest average reordering slippage at 5.5%, which is still fairly high, and \$11 cost per transaction to users. Banana gun is the highest with 8.7% and \$18.97 average cost per transaction to users.

The below chart shows the volatility of reordering slippage for the bottom 90% of the block. Reordering slippage is highest in USDC terms after the top 10% and before the 20% of the block. Reordering slippage % is fairly high throughout the bottom 90% of the block, even for smaller swaps.

## Block0 Bundle Competition

Telegram bots can leverage their users' bidding power to compete for bundles that land at the top of the block starting with transaction 0 or reverting otherwise. We will refer to this type of bundle as a block0 bundle. Telegram bots compete against each other to land these block0 bundles. This section shows that block0 competition can exacerbate the negative impact of reordering slippage for Telegram bot users.

Within these block0 bundles, users themselves must compete against each other for better positions within the bundle by bidding higher than other users within the block0 bundle. We look at a block0 bundle competition between two bots - Banana Gun and Maestro. These bots handle block0 bundle failures differently. If Banana Gun fails to land the block0 bundle, then the bundle reverts. If the Maestro block0 bundle fails, it will revert. However, Maestro will also submit an independent backup transactions to give the user a second chance to land in the block. Full documentation for Banana Bot

[can be found here](#)

and

[here for Maestro](#)

The total swap volume for MSTR/WETH in block 18437662 was \$33,167 with 8.72 ETH bids sent to the builder. Using the 10% tick bisection index, swap volume in the top bisection accounted for \$1,775 and 2.32 ETH bids. The bottom index accounted for \$31,392 with 6.4 ETH bids. The average reordering slippage was 20.4% with an average cost of \$55.28. The total reordering slippage volume was \$5,418.

Notably, the top 10% block ticks contained 26% of the total ETH bid on only 5% of the swap volume. This indicates that even within the Banana Gun bundle, there is fierce competition to jockey for the top of the bundle to secure the best execution price.

## Discussion

Our analysis depicts the detrimental role reordering slippage plays in transaction execution, and highlights how different MEV actors are affected based on their bidding behavior and other characteristics. The main insights from our research are:

We hope to encourage researchers and builders across the community to reveal more of this emerging area and collaborate with us. In response to these challenges, we're developing mev-commit at Primev, an open-source network designed for coordinating with mev actors to receive commitments against such issues. We plan to develop mev-commit use cases that leverage its real time bidding and private commitment mechanisms to combat reordering slippage, offering users a richer transaction toolset.

We will be releasing the reordering slippage notebook calculations and data pipeline in the near future to replicate the analysis. We invite the community to validate the bisection index method against other methodologies or broader datasets, and consider developing these as further use cases with mev-commit. Further research areas we plan to explore are assessing block builder algorithms impact on reordering slippage, analyzing the competitive dynamics of telegram bots and searchers per protocol in greater detail, and expanding reordering slippage research to look at CeFi-DeFi arbitrage.