Quantifying Price Improvement in Order Flow Auctions

Brad Bachu[1], Xin Wan[1], and Ciamac C. Moallemi[2]

[1] Uniswap Labs
brad.bachu@uniswap.org,xin.wan@uniswap.org
[2] Columbia University
ciamac@gsb.columbia.edu

Abstract. We introduce a methodology for empirically evaluating the outcomes of on-chain order flow auctions (OFAs), using price improvement as the key metric and attributing it to factors such as routing efficiency, gas optimization, and priority fee settings. The framework is agnostic to the underlying OFA mechanisms and applies to a broad range of tokens, including those not frequently traded or listed on centralized exchanges (CEXes), enabling comprehensive comparisons of OFA performance. This approach allows for real-world, on-chain evaluations of auction outcomes, providing users with insights into which OFAs perform best and how these improvements are achieved. As an example, we show how the methodology can be applied to 1Inch and Uniswap, demonstrating significant price improvements of 4-5 basis points above the Uniswap router, attributed to added liquidity in large swaps.

1 Introduction

Blockchain-based Automated Market Makers (AMMs) have achieved significant success, with Uniswap surpassing 2 trillion USD in transactions [13]. However, AMMs face challenges including fragmented liquidity and inefficiencies resulting in losses exceeding 540 million USD [17].

Order Flow Auctions (OFAs), such as 1inch Fusion [15], UniswapX [2], CowSwap [16], and MEV-Share [14] address these issues through batching, auctioning, matching orders and rebates. The general idea among all OFAs is that the profit from aforementioned inefficiencies (sometimes referred to as MEV) can be redistributed to the user by auctioning of the right to execute the order or transaction. However, empirical validation of their benefits remains challenging. Comparing prices to centralized exchanges (CEXes) is insufficient as many tokens trade only on DEXes. Cross-platform comparisons by block face selection bias—our dataset shows only 85 overlapping blocks between UniswapX and 1Inch Fusion out of 12,199 total blocks (Table 1). To enable meaningful comparisons, we propose viewing 'price improvement' relative to a consistent baseline. Our methodology decomposes price improvement into routing efficiency, gas optimization, and priority fee settings.

We demonstrate the use of the methodology on 1Inch and Uniswap on Ethereum mainnet [11] for WETH<>USDC over two months in 2023. [11][3]. We select these two as a starting point, primarily due to the similarity in OFA mechanism (Dutch auctions). In this scenario, we find that 1Inch and Uniswap provide similar improvements in trading experience.

Summary of our contributions. In this paper, we make four main contributions:

- *Framework for Price Improvement.* We introduce a systematic approach to evaluate OFA performance through price improvement metrics, enabling consistent comparisons across different interfaces.
- *Methodology for Gas Cost Internalization.* Gas costs can account for over 90% of effective spread in small AMM trades [1]. Our framework internalizes these costs into trade prices, capturing greater variability than median benchmarks and correcting statistical bias in simulated transactions.
- *Price Improvement Attribution Model.* We attribute price improvements to controllable factors, providing actionable insights for OFA optimization.
- *Empirical Application.* Applied to Uniswap and 1Inch on Ethereum, we find that 4-5 basis points (bps) of improvements above the Uniswap router can be achieved, driven by liquidity and routing optimization.

To our knowledge, this is the first formal definition and framework for assessing price improvements in on-chain OFAs, providing granular insights into their effectiveness. Preliminary findings suggest that OFAs may outperform interfaces that rely solely on onchain data and liquidity sources.

	Uniswap Classic	Uniswap X	1Inch Aggregator	1Inch Fusion
Uniswap Classic	1800	38	9	16
Uniswap X	38	9498	43	85
1Inch Aggregator	9	43	1677	17
1Inch Fusion	16	85	17	2701

Table 1: Overlap of blocks with transactions between WETH<>USDC during November and December of 2023.

2 Literature Review

Execution quality in financial markets has been widely studied, with early work focusing on traditional equity markets. [6] reviews methods for measuring execution cost, including quoted, effective, and realized spreads, while [4] shows that these costs can vary based on the measurement methodology. Similarly, [5] and

[3] While our framework handles batch auctions and rebate mechanisms, their empirical analysis is left for future work.

[19] assess execution quality and costs across different US exchanges, highlighting the impact of market structure and policy changes on execution outcomes.

Additional studies have shown that execution quality is multi-dimensional. [7] emphasizes the importance of both cost and speed, while [9] explores the trade-off between these two factors. Intraday patterns further complicate execution quality, as [12] finds compensatory patterns between cost and speed in Nasdaq trades.

Price improvement, a key aspect of execution quality, is also shaped by factors such as payment for order flow and broker competition. [3] explores instances where orders are executed at prices better than quoted, while [18] introduces a theory in which price improvements vary with customer market power. [10] documents differences in price improvement across asset classes, particularly in markets where payment for order flow plays a role.

In the context of decentralized finance, [1] reports that effective spreads on Uniswap are comparable to those in traditional asset classes. However, this study does not account for cross-platform comparisons or the specific impact of gas costs. [20] and [8] further investigate liquidity strategies and execution costs in decentralized markets, but these studies stop short of offering a comprehensive comparison of OFA systems.

To our knowledge, no existing framework systematically compares execution quality across decentralized platforms, particularly when factoring in gas costs and price improvement. This paper bridges this gap by introducing a formal methodology to evaluate OFA performance across platforms and analyze execution quality, taking into account the unique mechanics of decentralized trading

3 Theoretical Framework

We aim to define 'price' and 'price improvement' consistently across OFA systems, where transaction costs (gas fees) are the primary differentiator. While some OFAs internalize gas fees, others require upfront user payment. Direct comparisons are complicated because gas fees may be denominated differently than input tokens, with unclear exchange rates for conversion. Ignoring gas fees leads to inaccurate price comparisons, as they can determine OFA performance differences.

We aim to define 'price' and 'price improvement' consistently across OFA systems, where transaction costs (gas fees) are the primary differentiator. While some OFAs internalize gas fees, others require upfront user payment. Direct comparisons are complicated because gas fees may be denominated differently than input tokens, with unclear exchange rates for conversion. Ignoring gas fees leads to inaccurate price comparisons, as they can determine OFA performance differences.

Our methodology applies to transactions where either input or output token is the gas token (ETH) or its wrapped version (WETH), covering approximately 90% of Ethereum DEX trades[4]. This standardization allows uniform price def-

[4] https://dune.com/queries/3675220

initions across all OFAs, regardless of their gas fee handling. In section 4, we demonstrate how to apply the formalism to Dutch auction based OFAs, and leave additional emperical studies for future work.

3.1 Price

Consider a user with pre-trade balance (a, b) and post-trade balance (a', b') of tokens A and B. We define the price p as

$$p = \frac{b' - b}{a - a'}\,, \tag{1}$$

where the signs reflect the increase in Token B and decrease in Token A. When a user initiates a trade with fixed input i, the price depends on three controllable variables: output amount o (optimized via routing), gas usage g (varies with transaction complexity), and priority fee f (affects transaction ordering). We express these as vector $\boldsymbol{x} = (o, g, f)$, defining price as $p(\boldsymbol{x})$. For example, trading ETH for USDC with gas cost $g(b + f)$ (where b is the base fee) gives:

$$p = \frac{o}{i + g(b + f)}\,. \tag{2}$$

The price calculation varies by OFA type:

- **Traditional**: Gas costs directly affect price through ETH balance changes
- **Gas-free**: Price reflects only token exchange ratios (via Permit2)
- **MEV-aware**: Price includes potential rebates.

See Appendix A for detailed examples of each case.

3.2 Price Improvement

We define *price improvement* (π) as the relative difference between the realized price p and a baseline counterfactual price p', where p' represents the price that would have been achieved under normal conditions without an Order Flow Auction (OFA). Formally:

$$\pi(p, p') = \frac{p - p'}{p'}\,. \tag{3}$$

The counterfactual baseline p' serves as a neutral reference point, enabling us to evaluate execution quality across systems by comparing realized outcomes with the baseline.

3.3 Flexibility in Generating Counterfactual Prices

The baseline serves as a reference point, much like measuring altitude relative to sea level, ground level, or another benchmark: while the chosen reference point affects absolute measurements, it does not impede the ability to make meaningful relative comparisons across different systems (if chosen correctly).

Since price p is a function of output tokens o, gas fees g, and priority fees f, a counterfactual price p' can also be generated by counterfactual values o', g', and f'. Therefore, the counterfactual price p' can be expressed as $p' = p(o', g', f') = p(\boldsymbol{x}')$, where $\boldsymbol{x}' = (o', g', f')$ represents the primed (counterfactual) values for these variables. For more detailed information on how we generate the counterfactual variables $\boldsymbol{x}'$, refer to Section 4.

While alternative baselines, such as CEX prices, could be considered, they would restrict the analysis by limiting available tokens and reducing attribution to factors like gas costs or liquidity routing (see Section 3.5), as they do not provide $\boldsymbol{x}' = (o', g', f')$.

3.4 Price Improvements Across Time

Our primary definition of price improvement π compares realized prices against counterfactual prices at the settlement time t_0. However, it is also useful to evaluate price improvement across various time offsets Δt from t_0, allowing us to account for execution speed and timing differences among OFAs. This approach is conceptually similar to markout in traditional finance, where the performance of a trade is measured relative to its price at different time intervals post-execution.

Implicit in the realized price p is the settlement time t_0, $p = p(t_0)$, but a counterfactual price $p' = p'(t)$ can be generated at any time. Thus, we extend our definition of price improvement by including the time dimension as follows $\pi(p, t_0; p', t) = \frac{p(t_0) - p'(t)}{p'(t)}$.

Since t_0 is fixed historically, the only key differences arise from offsets $\Delta t = t - t_0$. To explore this, we shift the definition by $-t_0$, treating all times relative to the settlement time. This leads to the more practical definition: $\rho(p; p', \Delta t) = \frac{p - p'(\Delta t)}{p'(\Delta t)}$, where we have dropped t_0 for simplicity. When $\Delta t = 0$, we recover our original defition of price improvement $\rho(p; p', 0) = \pi(p, t_0; p', t_0)$. This process is depected in Figure 1.

Evaluating price improvement across different time offsets Δt provides several benefits:

1. Transaction Speed: It helps account for potential differences in transaction inclusion speed between the actual interface and the counterfactual interface, highlighting robustness issues related to execution speed.
2. Order Filling Mechanisms: Different trading interfaces have mechanisms that affect whether or how orders are filled. By analyzing multiple offsets, we can identify and adjust for any selection biases these mechanisms may introduce.
3. Blockchain Conditions: The blockchain environment can be unpredictable, with gas spikes or storage issues affecting outcomes. Evaluating a range of time offsets mitigates the impact of transient conditions, ensuring more accurate results.
4. Transaction Ordering: On platforms like Ethereum, transaction ordering can be adversarial. Comparing against multiple counterfactual orderings reveals sensitivity to transaction ordering strategies.

5. Interpretation Flexibility: Depending on the context, we can compare counterfactual prices at different points in time, such as at arrival or execution, allowing for flexible interpretations of price improvement.

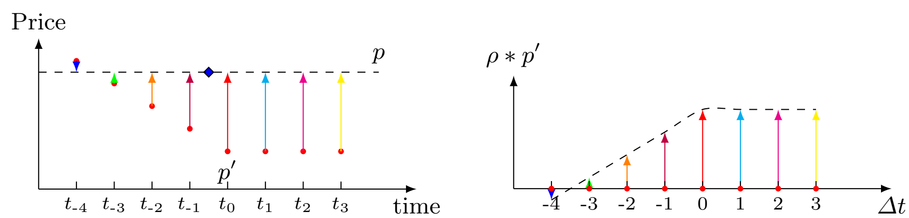

Fig. 1: On the left, we show the real transaction that occurred in the 0th block with price p, and the counterfactual transactions generated at the end of the given blocks with prices $p'(t)$. The arrows represent the differences between p and p' at different times, or the un-normalized price improvement: $\pi(p, t_0; p', t) * p'(t)$. On the right, we show the same un-normalized price improvements, however, with the x-axis now changed to relative time: $\rho(p; p', \Delta t) * p'(t)$.

3.5 PI Attribution

In our framework, price improvement π is conceptualized as an aggregated result of various controllable decisions, such as routing, gas usage, priority fee settings. To provide a more granular insight into how PI was achieved empirically, we decompose π into three economically significant components: routing optimization π^{routing}, gas optimization π^{gas} , and priority fee optimization π^{fee}

$$\pi = \pi^{\text{routing}} + \pi^{\text{gas}} + \pi^{\text{fee}}, \tag{4}$$

where π^{routing} captures the π through optimizing liquidity access, π^{gas} captures the π from reduced gas costs, and π^{fee} captures the π from lower priority fees.

Interface decision making impacts these PI components through several mechanisms:

1. **Route Optimization**: Interfaces optimize routing by selecting liquidity sources, which may include on-chain pools or off-chain sources. While more liquidity is generally better, it comes at the cost of additional gas usage. π^{routing} measures the PI achieved through including more liquidity in the route than the baseline.
2. **Gas Efficiency**: An optimal routing decision minimizes gas usage while maximizing liquidity access. π^{gas} quantifies the PI achieved through using less gas compared to the baseline.
3. **Priority Fee Setting**: Under EIP-1559, any positive priority fee is usually sufficient for inclusion, but some interfaces recommend higher priority fees for faster execution. π^{fee} measures the PI gained by minimizing the priority fee.

To attribute π, we Taylor expand the price function $p(\boldsymbol{x})$ about the baseline variables $\boldsymbol{x}'$,

$$\begin{aligned} p(\boldsymbol{x}) &= p(\boldsymbol{x}') \\ &\quad + \left.\frac{\partial p}{\partial o}\right|_{\boldsymbol{x}'}(o-o') + \left.\frac{\partial p}{\partial g}\right|_{\boldsymbol{x}'}(g-g') + \left.\frac{\partial p}{\partial f}\right|_{\boldsymbol{x}'}(f-f') \\ &\quad + R(\boldsymbol{x},\boldsymbol{x}')\,, \end{aligned} \tag{5}$$

where $R(\boldsymbol{x},\boldsymbol{x}')$ represents the remainder term. Note that, in all the cases we consider, $p(\boldsymbol{x})$ is differentiable in the domain of interest. For example, considering the WETH/ETH out in eq. 8, we have $\frac{\partial p}{\partial o} = \frac{1}{i}$, $\frac{\partial p}{\partial g} = \frac{-(b+f)}{i}$ and $\frac{\partial p}{\partial f} = \frac{-g}{i}$. Rearranging eq. (5) gives us:

$$\begin{aligned} \pi &= \left.\frac{\partial p}{\partial o}\right|_{\boldsymbol{x}'}\frac{(o-o')}{p'} + \left.\frac{\partial p}{\partial g}\right|_{\boldsymbol{x}'}\frac{(g-g')}{p'} + \left.\frac{\partial p}{\partial f}\right|_{\boldsymbol{x}'}\frac{(f-f')}{p'} + \frac{R(\boldsymbol{x},\boldsymbol{x}')}{p'} \\ &= \pi_0^{\text{routing}} + \pi_0^{\text{gas}} + \pi_0^{\text{fee}} + \pi^{\text{rem.}}, \end{aligned} \tag{6}$$

where the 0 subscript indicates leading order contributions. Note that every term here is a function of on-chain values $\boldsymbol{x} = (o, g, f)$ and simulated values $\boldsymbol{x}' = (o', g', f')$, and so is calculable. Of course, it is not guaranteed that remainder term $\pi^{\text{rem.}}$ is small, but we will find that for some cases, it is.

For OFAs where gas is internalized, the amount output token o that the user receives is post-fee. By using onchain data for g, b and f, we can calculate what the pre-fee output would have been, allowing us to attribute π in these cases.

4 Methodology

While our primary contribution is a theoretical framework for evaluating any OFA's price improvements, this section demonstrates its practical application to market leaders that utilize Dutch auctions, Uniswap and 1Inch, highlighting both implementation challenges and potential at scale.

4.1 Sample Interfaces

Uniswap and 1Inch interfaces each offer two execution paths. One provides traditional transactions, which are predetermined routes through public on-chain sources (Uniswap Classic and 1Inch Aggregator), whereas the other (UniswapX and 1Inch Fusion) provides intents and Dutch auction based settlement, where the liquidity can be derived from both private and public on-chain sources. The interfaces dynamically route orders to the path offering better execution. This introduces selection bias: UniswapX and Fusion trades appear to outperform because they are only chosen when advantageous. A comprehensive evaluation must therefore consider both execution paths together to understand true OFA performance in the context of overall routing decisions. These interfaces provide an ideal comparison because they share similar characteristics: Classic and

Aggregator require direct gas payment in ETH, while UniswapX and Fusion internalize gas costs through specialized participants called solvers ot fillers. This parallel structure enables consistent application of our methodology while highlighting key differences in execution approaches.

4.2 Selection of Baseline

Our theory permits any baseline that can generate counterfactual values of output token, gas and priority fees. To illustrate the application of our framework, we select our baseline as a counterfactual simulated swap using the Uniswap Classic routing API at the end of the block in which the actual trade occurred, executed through three steps:

1. **Route Calculation**: Submit identical token pair and amount to the routing API, which finds optimal routes using blockchain state from surrounding blocks (n blocks before/after) for robustness
2. **Route Formation**: API assesses end-block state and pool liquidity (from The Graph), determines optimal route, and formats into calldata with gas estimates
3. **Simulation**: Execute calldata through Tenderly simulator with consistent priority fee (0.1 Gwei), obtaining gas consumption and output details

This choice is beneficial for several reasons:

1. **Assumption Test**: The price improvement (PI) of Uniswap Classic compared to itself should be zero, providing a crucial accuracy check of the baseline simulation at the end of the block.
2. **User Experience**: Provides realistic approximation of typical trading conditions, enabling meaningful comparisons between different OFAs.
3. **Token Coverage**: Uniswap's extensive pool coverage enables comprehensive analysis that can capture long-tail tokens and on-chain dynamics.
4. **Public Access**: Open API enables reproducible analysis by any third party.

The API accesses historical blockchain states, enabling simulations at different block times Section 3.4.

4.3 Data Collection

In this section, we describe the process of collecting the data necessary for our analysis. We focus on historical values for input amounts i, base fees per gas b, output amounts, gas used, and priority fees per gas, represented as $\boldsymbol{x} = (o, g, f)$. Counterfactual baseline values $\boldsymbol{x}' = (o', g', f')$ are generated via the Uniswap Classic routing API.

Our dataset includes all WETH-USDC trades from November and December 2023, which are sourced from Dune Analytics, and validated for Uniswap Labs from provided internal datasets. We collect transactions from Uniswap Classic and UniswapX by filtering interactions via the Uniswap Interface, and for 1Inch

Interface	Path	Size	% Parent	Volume ($)	% Vol
1Inch	Aggregator	1687	36%	37,891,096	22%
	Fusion	2941	64%	134,221,271	78%
Uniswap	Classic	1809	16%	28,573,360	13%
	X	9607	84%	185,599,214	87%

Table 2: Distribution of swaps and volumes across different settlement paths for 1Inch and Uniswap interfaces.

trades, we extract data using the `oneinch` table in Dune Analytics. To focus purely on execution quality, all interface fees are ignored.

To generate a baseline price p', we rely on API calls for historical transactions. For each block time t and input amount i, the API provides counterfactual estimates of the output token o' and gas used g'. We define a baseline function to generate counterfactual prices as $\mathcal{B}$ as $\mathcal{B} : (i, t) \rightarrow (o', g')$, with a consistent baseline priority fee per gas f' of 0.1 Gwei. For more details on counterfactual generation see Appendix A.

4.4 Baseline Gas Corrections

One key challenge in generating accurate counterfactual baselines is estimating gas usage g'. The gas estimated by simulations may not always match the actual gas used g in historical transactions. Figure 2 illustrates the difference between the simulated gas g' and the actual gas g for Uniswap Classic trades. Ideally, g' should closely match g.

Discrepancies arise due to factors like Just-In-Time (JIT) liquidity provision [20], where the actual gas used may be lower than estimated. Additionally, simulations are performed at the end of a block, while trades occur at various times within the block, leading to potential differences.

To correct for this (i.e., to address the systematic error), we apply a correction factor. By comparing g' to the actual gas g, we fit the adjustment factor (β_1) via regression: $g' \approx \beta_1 g$. We then adjust the estimate: $g' \leftarrow g'/\beta_1$. This adjustment (or calibration) reduces the gap between the simulated and actual gas usage, improving the reliability of the price improvement measurement by ensuring our estimates better reflect actual outcomes.

4.5 Uncertainty Analysis

In our analysis of price improvement π and its components $\pi_0^{\text{attribute}}$, we calculate an average value $\bar{\pi}$ weighted by the trade's USD value. This average is influenced by two sources of uncertainty: *statistical* uncertainty (which arises from variations in individual trades) and *systematic* uncertainty (which arises from potential biases in our gas estimates).

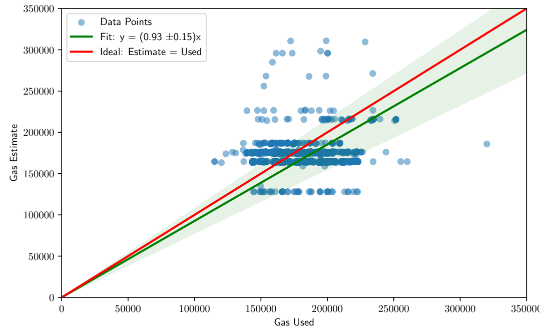

Fig. 2: Comparison between actual gas used g and estimated gas g' for Uniswap Classic transactions. The red line represents perfect gas estimation, and the green dashed line shows the corrected estimate with confidence bounds.

Statistical uncertainty (σ_{stat}) is computed using the weighted standard error. For a trade i with price improvement $z_i \in \{\pi, \pi_0^{\text{attribute}}\}$ and weight w_i, we calculate:

$$\sigma_{\text{stat}}^2 = \frac{\sum_i w_i(\bar{z} - z_i)^2}{n \sum_j w_j}.$$

This gives us the uncertainty due to variability in the sample data.

Systematic uncertainty (σ_{sys}) of the gas estimation is computed by varying β_1 and calculating the bounds: $\sigma_{\text{sys}}^{\text{upper}} = \bar{z}(\beta_1 + \delta\beta_1) - \bar{z}(\beta_1)$ and $\sigma_{\text{sys}}^{\text{lower}} = \bar{z}(\beta_1) - \bar{z}(\beta_1 - \delta\beta_1)$. Note that this is an asymmetric uncertainty. Finally, we combine both sources of uncertainty into the total uncertainty for $\bar{z}$, given by $\bar{z} \pm \sqrt{\sigma_{\text{stat}}^2 + \sigma_{\text{sys}}^2}$.

4.6 Discussion and Additional Limitations

Although various factors introduce uncertainty, we find their impact to be minimal since, on average, since the price improvement of Uniswap Classic compared to itself is zero in Figure 3. This validation suggests that while these factors exist, they do not significantly affect the average value. Several factors introduce uncertainty in both gas usage estimates g' and output token estimates o':

1. **Simulation changes**: Updates to the simulation algorithm may cause discrepancies in gas and output estimates.
2. **Intra-block effects**: Simulating at the end of the block overlooks intra-block dynamics, such as liquidity shifts and ticks crossed, which can influence gas usage and output accuracy.
3. **Self-impact**: Transactions interacting with the same liquidity pool may degrade counterfactual results due to self-induced effects, similar to front-running, impacting output token amounts.
4. **Data-integrity**: Un-validated data for 1Inch may introduce biases from indexing delays or missing transactions.

5 Results

Our analysis reveals that auction-based platforms like UniswapX and 1Inch Fusion can achieve substantial price improvements, in this case, primarily through enhanced liquidity access. Uniswap Classic shows no price improvement relative to itself, confirming the accuracy of our framework, while small correction terms validate the use of first-order approximations for most systems. Larger correction terms for auction-enhanced platforms like UniswapX suggest the need to account for higher-order effects.

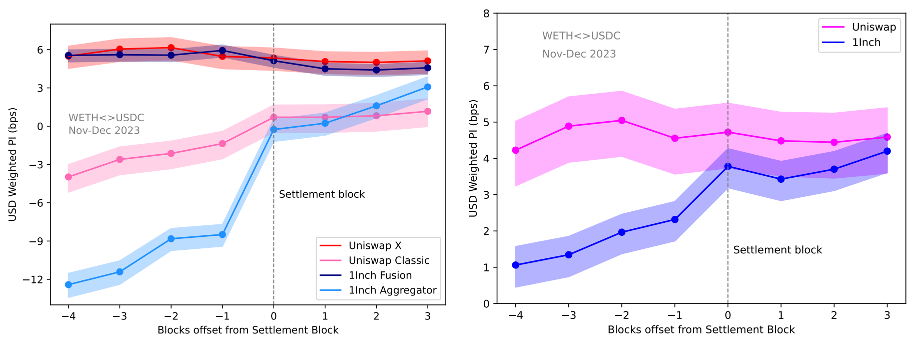

Fig. 3: USD-weighted price improvement trajectory ρ comparing Uniswap Classic, Uniswap X, 1Inch Ag- gregator, and 1Inch Fusion across time (in units of blocks) relative to settlement. The left panel shows settlement-path level decomposition, validating our methodology through Uniswap Classic's zero improvement at settlement, while the rightpanel presents interface-level aggregation revealing systematic differences in execution quality. Shaded areas indicate $\pm 1\sigma$ intervals, accounting for both statistical and systematic uncertainties

Figure 3 shows the USD-weighted price improvements across platforms at different time offsets from the settlement block. As expected, Uniswap Classic exhibits zero price improvement at settlement, confirming our methodology's accuracy in capturing where improvements exist. Both Uniswap-X and 1Inch Fusion, the Dutch auction components, show equivalent price improvements averaging 5-6 basis points. On the right, the combined result for all interface trades (which accounts for the selection bias), shows that the difference in overall performance is a result of the interfaces' traditional routing execution.

Price improvement is further analyzed by trade size in Figure 4. While smaller trades exhibit more volatility, larger trades stabilize with consistent positive price improvement. UniswapX reaches nearly 4 basis points for trades around \$200k, outperforming 1Inch Fusion as trade size increases.

To investigate the differences, Figure 5 and Figure 6 provide a breakdown of price improvement into key attributes: routing efficiency, gas optimization, and priority fee settings. Routing efficiency is the dominant factor, with smaller contributions from gas and priority fees. Gas overheads (0.5-1 basis points) are

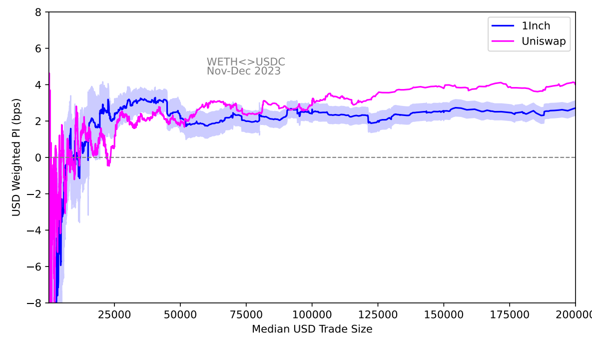

Fig. 4: Price Improvement (PI) analysis across trade sizes for Uniswap and 1Inch interfaces. The rolling USD-weighted PI demonstrates how execution quality varies with trade size, revealing distinct patterns for each interface. Larger trades show more consistent improvements, while smaller trades exhibit higher variability. Shaded areas indicate $\pm 1\sigma$ intervals, combining statistical and systematic uncertainties.

observed in both UniswapX and 1Inch Fusion, though they do not outweigh the routing benefits.

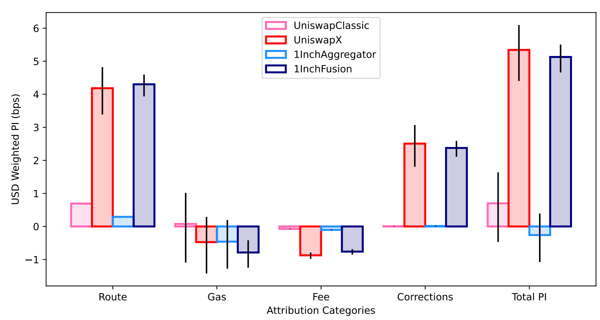

Fig. 5: Decomposition of USD-weighted price improvements across settlement paths, showing relative contributions from routing efficiency, gas optimization, and priority fee settings. The breakdown reveals that routing efficiency dominates price improvements, particularly in auction-based systems (UniswapX and 1Inch Fusion), while gas optimization plays a secondary but significant role.

To validate the approximation, we examine correction terms. For Uniswap Classic and 1Inch Aggregator, these corrections are negligible, confirming the accuracy of the first-order approximation. For UniswapX and 1Inch Fusion, larger correction terms indicate non-linear effects, particularly from gas optimization and priority fees, which can be understood by explicitly extending eq. (6). While first-order terms capture much of the system's behavior, these corrections highlight the complexity of auction-based platforms. Note that, the differences between Uniswap Classic and 1Inch Aggreagator arise primairly from the routes,

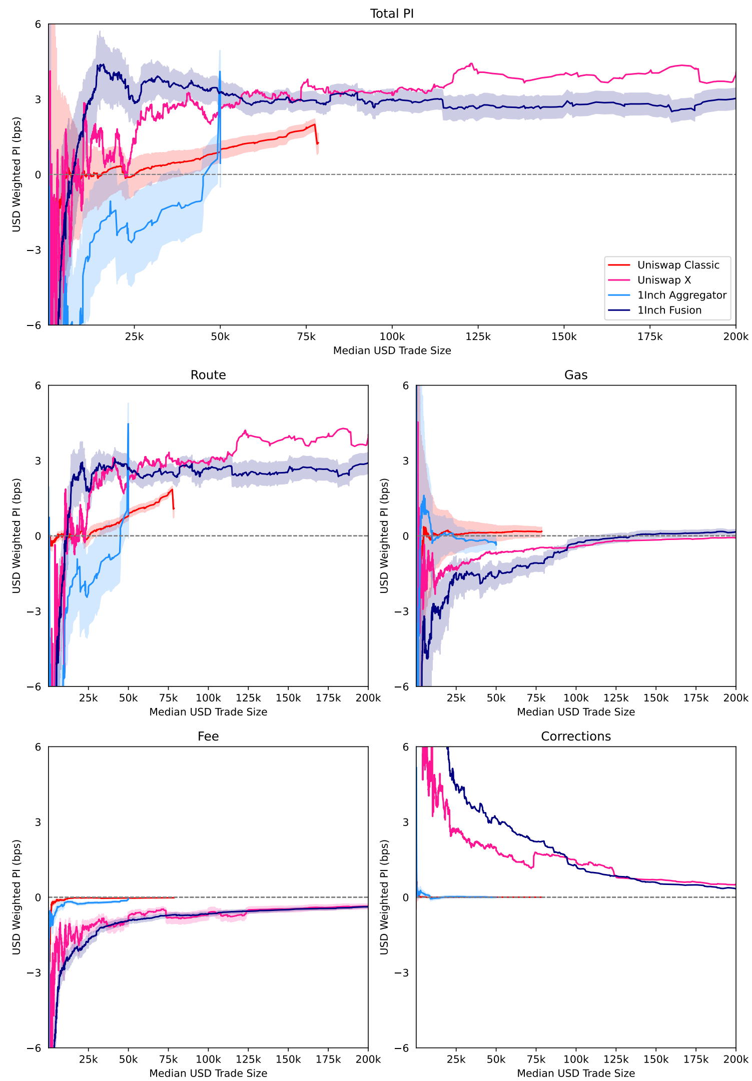

Fig. 6: Component-wise analysis of price improvement contributions across trade sizes for all settlement paths (Uniswap Classic, 1Inch Aggregator, Uniswap-X, and 1Inch Fusion). The breakdown demonstrates how different mechanisms dominate at different trade sizes, with routing efficiency becoming increasingly important for larger trades while gas optimization effects remain relatively constant. Shaded regions represent $\pm 1\sigma$ combined statistical and systematic uncertainties.

whereas the differnces between Uniswap-X and 1Inch Fusion arise primairly due to the filler networks.

6 Outlook

This work provides a framework for analyzing price improvement in OFAs, with extensions to better understand transaction costs. The applications of this exact methodology to batched auctions (COW Swap), and rebate systems (MEV Share, MEV Blocker) is left for future work. Extensions can also include comparisons between fillers that use private and public on-chain liquidity sources, the impact of different benchmarks to validate ordering assumptions, as well as investigating why one OFA type might out perform another in different attributes. Lastly, this works establishes a foundation for comprehensive analysis of on-chain trading, as it can scale to all tokens and answer on-chain specific questions.

Acknowledgements and Disclosures. Ciamac Moallemi was supported by the Briger Family Digital Finance Lab at Columbia Business School and is an advisor to fintech companies.

References

1. Adams, A., Chan, B.Y., Markovich, S., Wan, X.: The costs of swapping on the uniswap protocol. arXiv preprint arXiv:2309.13648 (2023)
2. Adams, H.: Introducing the uniswapx protocol. https://blog.uniswap.org/uniswapx-protocol (2023)
3. Angel, J.: Who gets price improvement on the nyse (1994)
4. Bessembinder, H.: Issues in assessing trade execution costs. Journal of financial markets **6**(3), 233–257 (2003)
5. Bessembinder, H.: Trade execution costs and market quality after decimalization. Journal of Financial and Quantitative Analysis **38**(4), 747–777 (2003)
6. Bessembinder, H., Venkataraman, K.: Bid-ask spreads: Measuring trade execution costs in financial markets. Encyclopedia of quantitative finance pp. 184–190 (2010)
7. Boehmer, E.: Dimensions of execution quality: Recent evidence for us equity markets. Journal of Financial Economics **78**(3), 553–582 (2005)
8. Capponi, A., Jia, R., Zhu, B.: The paradox of just-in-time liquidity in decentralized exchanges: More providers can sometimes mean less liquidity. Available at SSRN (2023)
9. Engle, R.F., Ferstenberg, R., Russell, J.R.: Measuring and modeling execution cost and risk (2006)
10. Ernst, T., Spatt, C.S.: Payment for order flow and asset choice. Tech. rep., National Bureau of Economic Research (2022)
11. Flashbots: Orderflow.art. https://orderflow.art/frontends
12. Garvey, R., Wu, F.: Intraday time and order execution quality dimensions. Journal of Financial Markets **12**(2), 203–228 (2009)
13. Labs, U.: https://x.com/Uniswap/status/1776218671058239621 (2024)

14. Miller, B.: Mev-share: programmably private orderflow to share mev with users. https://collective.flashbots.net/t/mev-share-programmably-private-orderflow-to-share-mev-with-users/1264 (2023)
15. 1inch Network: The 1inch network releases a major upgrade, fusion. https://blog.1inch.io/the-1inch-network-releases-a-major-upgrade-fusion/ (2022)
16. Protocol, C.: General faq. https://swap.cow.fi/#/faq
17. Qin, K., Zhou, L., Gervais, A.: Quantifying blockchain extractable value: How dark is the forest? In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 198–214. IEEE (2022)
18. Rhodes-Kropf, M.: Price improvement in dealership markets. The Journal of Business **78**(4), 1137–1172 (2005)
19. Wah, E., Feldman, S., Chung, F., Bishop, A., Aisen, D., Exchange, I.: A comparison of execution quality across us stock exchanges. Global Algorithmic Capital Markets: High Frequency Trading, Dark Pools, and Regulatory Challenges pp. 91–146 (2017)
20. Wan, X., Adams, A.: Just-in-time liquidity on the uniswap protocol. Available at SSRN 4382303 (2022)

A Price Details

Realized and counterfactual prices depend on if gas is/not internalized and if the input token is/not WETH. **Realized prices:** For traditional transactions where gas is not internalized, we have two cases,

$$p = \begin{cases} \frac{o-g(b+f)}{i}, & \text{when token out address} = \text{WETH/ETH} \\ \frac{o}{i+g(b+f)}, & \text{when token in address} = \text{WETH/ETH} \end{cases} . \tag{7}$$

For modern OFAs using Permit2 signatures[5] enable gas-free user experiences, $p = \frac{o}{i}$. For systems involving MEV rebates, the price is similar to traditional transactions but adjusted for the rebate. **Counterfactual prices:** As mentioned before, the baseline function generates quotes $\mathcal{B}(i) \rightarrow (o', g')$. When the API is given an input i, it can generate a token out amount estimate o', and gas use estimate g'. When the gas is not internalized , we compute p' as

$$p' = \begin{cases} \frac{o'-g'(b+f')}{i}, & \text{when token out address} = \text{WETH/ETH} \\ \frac{o'}{i+g'(b+f')}, & \text{when token in address} = \text{WETH/ETH} \end{cases} . \tag{8}$$

When gas is internalized, we compute a token in amount gas adjusted $i' = i - g'(b + f')$, we define the following $\mathcal{B}(i') = (o'', g'')$, and

$$p' = \begin{cases} \frac{o'-g'(b+f')}{i}, & \text{when token out address} = \text{WETH/ETH} \\ \frac{o''}{i'+g'(b+f')}, & \text{when token in address} = \text{WETH/ETH} \end{cases} . \tag{9}$$

Note that the denominator is just i, but we have written it in a way that allows us to estimate the amount of input token that you have available to route i', and the amount that you must pay in gas $g'(b + f')$. The amount of token that you have available to route is then used to generate the token out amount estimate o''.

[5] https://blog.uniswap.org/permit2-and-universal-router