Strategic Analysis of Just-In-Time Liquidity Provision in Concentrated Liquidity Market Makers

Bruno Llacer Trotti
Federal University of Rio de Janeiro (UFRJ), Brazil

Weizhao Tang
Carnegie Mellon University, Pittsburgh, PA, USA

Rachid El-Azouzi
Avignon University, France

Giulia Fanti
Carnegie Mellon University, Pittsburgh, PA, USA

Daniel Sadoc Menasché
Federal University of Rio de Janeiro (UFRJ), Brazil

Abstract

Liquidity providers (LPs) are essential figures in the operation of automated market makers (AMMs); in exchange for transaction fees, LPs lend the liquidity that allows AMMs to operate. While many prior works have studied the incentive structures of LPs in general, we currently lack a principled understanding of a special class of LPs known as Just-In-Time (JIT) LPs. These are strategic agents who momentarily supply liquidity for a single swap, in an attempt to extract disproportionately high fees relative to the remaining passive LPs. This paper provides the first formal, transaction-level model of JIT liquidity provision for a widespread class of AMMs known as Concentrated Liquidity Market Makers (CLMMs), as seen in Uniswap V3, for instance. We characterize the landscape of price impact and fee allocation in these systems, formulate and analyze a non-linear optimization problem faced by JIT LPs, and prove the existence of an optimal strategy. By fitting our optimal solution for JIT LPs to real-world CLMMs, we observe that in liquidity pools (particularly those with risky assets), there is a significant gap between observed and optimal JIT behavior. Existing JIT LPs often fail to account for price impact; doing so, we estimate they could increase earnings by up to 69% on average over small time windows. We also show that JIT liquidity, when deployed strategically, can improve market efficiency reducing slippage for traders, albeit at the cost of eroding passive LP profits by up to 44% per trade on average.

2012 ACM Subject Classification Information systems → Electronic commerce

Keywords and phrases Concentrated Liquidity Market Makers, Uniswap, Just-in-Time

Digital Object Identifier 10.4230/LIPIcs.AFT.2025.8

Related Version *Full Version*: https://arxiv.org/abs/2509.16157

Supplementary Material
Software (Source Code): https://github.com/brunoCCOS/JITUniswapOptimization

Funding This work was partially funded by CAPES; by FAPERJ under grant E-26/204.268/2024; and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, under grants 403601/2023-1 and 315106/2023-9. It was also supported in part by the National Science Foundation under grant CNS-2325478, as well as by the Initiative for Cryptocurrencies and Contracts (IC3) and the CyLab Secure Blockchain Initiative, together with their respective industry sponsors.

1 Introduction

Recent years have seen increased adoption of Decentralized Exchanges (DEXs) [29] and Automated Market Maker (AMM) protocols [30], which automatically set trade prices and execute trades; prominent markets running AMMs include Uniswap [1], SushiSwap [24], and PancakeSwap [22]. At the time of writing, the annual trading volume on AMM-based exchanges exceeds hundreds of billions of dollars [19].

Liquidity providers (LPs) are users who lend tokens to specific pools, thereby providing liquidity for trades in exchange for trading fees. LPs are central to the effective operation of AMMs; consequently, many prior works have sought to analyze their behavior [7, 9, 12, 21, 25]. For example, several of these works characterize conditions under which LPs are incentivized to contribute liquidity, and strategies for doing so to optimize returns [8, 25].

Despite a growing body of work analyzing the incentives of LPs in AMMs, to our knowledge, few have studied an important class of LPs known as *Just-in-Time Liquidity Providers* (JIT LPs) [11, 26]. These are informed strategic agents who aim to extract profit by providing liquidity in a highly targeted manner. Rather than holding a liquidity position over a period of time, a JIT LP provides liquidity for a single trade. This can be executed by a sandwich attack [10] on the *target* transaction by executing a swap that involves frontrunning the target transaction with an "add liquidity" transaction and backrunning it with a "withdraw liquidity" transaction. By temporarily capturing a higher share of liquidity, the JIT trader can earn a larger portion of the trading fee that is paid to all LPs pro rata to each LP's liquidity share.[1]

Although JIT LPs may appear to operate at the margins of an AMM, they play a crucial role in the ecosystem. They act as a frictional force against price movements, they help pools stay aligned with external market opportunities, and they reduce slippage for traders [26, 28].[2] At the same time, they introduce adversarial costs to passive LPs and arbitrageurs. The prevalence of JIT LPs is expected to grow with the emergence of technologies such as Flashbots [17] and Uniswap V4 [2], which introduce optimized mechanisms for JIT operations (e.g., hooks, bundled transactions). However, we currently lack a theoretical understanding of JIT liquidity provision and its effects on AMMs.

This work fills a current gap in the literature by developing and analyzing a principled model of JIT LP incentives for an important class of AMMs known as *Concentrated Liquidity Market Makers (CLMMs)*. Introduced in Uniswap V3 [1], CLMMs allow LPs to invest liquidity over fine-grained price ranges; the LP's liquidity is only used to support trades when the AMM price lies within the specified range. This helps LPs reduce risk due to large price fluctuations. Today, CLMMs are widely used in Uniswap V3 and the mechanism is generalized in emerging Uniswap V4 markets [2].

Under a model of the CLMM paradigm, we formulate and analyze the JIT resource allocation optimization problem, study its solution properties analytically, and provide an algorithm for computing optimal strategies (§5). Our analysis reveals new insights into how optimal JIT strategies depend on transactional and pool-specific parameters.

We summarize our contributions as follows.

- **Price impact landscape:** We analyze how swap-induced price changes affect LP revenue and formally characterize the conditions under which liquidity provision leads to gains or losses. These effects are often called "impermanent loss"; we instead use the term "price

[1] Note that the "trading fee" (pool fee) is a percentage of the swap value paid by the trader to liquidity providers, whereas the "transaction fee" (gas fee) is paid to Ethereum validators to include the transaction in a block. In this paper, except otherwise noted, fees refer to trading fees.

[2] Slippage is a phenomenon by which traders lose money in an AMM due to price changes over the course of a single trade.

impact" to reflect that they do not always result in losses (Def. 2). This analysis yields a precise description of the parametric region in which LPs benefit from price movements (§2.3). This result holds for both *passive* (regular) LPs and JIT LPs.

- **Formulation and analysis of utility optimization for JIT LPs:** We model the decision-making process of JIT LPs as a non-linear optimization problem and define a transaction-level utility function that captures the trade-off between fee revenue and price impact. This formulation enables a rigorous exploration of the strategy and utility space (§2.2.1 and §4) . We classify optimal JIT behavior across three swap scenarios – overpriced, arbitrage-driven, and overshooting trades – and present an algorithm for identifying optimal liquidity positions (§5 and §6).
- **Empirical insights on real-world JIT performance:** Using on-chain data, we report three key empirical findings (§6). First, JIT traders today appear to allocate trades suboptimally, without accounting for price impact; our experiments suggest that this unawareness reduces JIT profit by up to 41% on average. Second, although JIT LPs currently account for a small share of total fee revenue, most of JIT revenue comes from price impact (93.7%) rather than extracting fees from passive LPs. Third, JIT liquidity provision can become a serious competitor to passive LPs by reducing their fee income by up to 40%, while slightly reducing the cost of slippage for traders.

2 Background and Model

2.1 Introduction to AMMs

Automated Market Makers (AMMs) enable decentralized token exchanges through algorithmic pricing instead of traditional order books. We begin by explaining one of the most common classes of AMMs known as constant product market makers (CPMMs) [6]. A CPMM maintains a liquidity pool of tokenized assets X and Y. After initialization, the CPMM maintains a pool price q, which is the price of token Y in terms of token X. When an LP enters the CPMM, they contribute an amount of liquidity L, which comprises token amounts x of X and y of Y, according to the following rule:

$$x = \frac{L}{\sqrt{q}}, \qquad y = L\sqrt{q}. \tag{1}$$

Consequently, the token amounts x, y of each liquidity position of amount L always follow the constant product rule $xy = L^2$. Additionally, when there exist multiple liquidity positions $\{(x_i, y_i, L_i)\}_{i=1}^{N}$, the corresponding summations also follow the constant product rule

$$x_{\text{total}} \cdot y_{\text{total}} = \left(\sum_{i=1}^{N} x_i\right) \cdot \left(\sum_{i=1}^{N} y_i\right) = \left(\sum_{i=1}^{N} L_i\right)^2 = L_{\text{total}}^2. \tag{2}$$

A CPMM supports three basic operations – mint (add liquidity), burn (remove liquidity), and swap (trade).

- Mint and burn transactions are executed by an LP who specifies liquidity L to add/remove, and will deposit/withdraw token amounts (x, y) following (1). q remains constant.
- Swaps are executed by a trader who specifies an amount of either token to pay. Without loss of generality, assume the trader pays Δx tokens of type X. L remains unchanged for each liquidity position, so the trader will get Δy Y tokens following (2):

$$(x_{\text{total}} + \Delta x)(y_{\text{total}} - \Delta y) = L_{\text{total}}^2.$$

As a result, the pool has a new price $q' = (y_{\text{total}} - \Delta y)/(x_{\text{total}} + \Delta x)$ and the equivalent token amount (x', y') of each liquidity position L will update by plugging q' into (1).

Slippage is an important concept in this domain; it refers to the difference between the execution price of a trade and the price expected at the time of trade initiation. In the context of AMMs, slippage arises from the fact that large trades move the price along the bonding curve, resulting in a marginally worse exchange rate for each successive unit traded. Slippage is an inherent byproduct of price impact in AMMs.

▶ **Example 1** (Slippage). Consider an AMM with 100 ETH and 10,000 USDC, so $q = \frac{\#\mathbf{ETH}}{\#\mathbf{USDC}} = 100$. Suppose a trader wishes to buy 10 ETH. Due to the constant-product formula $x \cdot y = k$, the required USDC input is such that $100 \times 10{,}000 = (100 - 10) \times y_{\text{final}}$, i.e., $y_{\text{final}} \approx 11{,}111.11$. Thus, the trader must pay approximately 1,111.11 USDC to receive 10 ETH, implying an average price of 111.11 USDC per ETH – higher than the original pool price. The slippage is: $(111.11 - 100)/100 = 11.11\%$. This increase reflects the adverse price movement caused by the trader's own order.

2.2 Concentrated Liquidity Market Makers

Since the emergence of CPMMs, more complex AMM designs have been proposed; among the most widespread of these is Concentrated Liquidity Market Makers (CLMMs) [14]. They allow LPs to choose not only the amount of liquidity they wish to add to the pool, but also the price range in which the liquidity is active. When an LP adds liquidity to the CLMM, the LP creates a *liquidity position* by specifying a tuple (L, a, b), where L is the amount of liquidity, and (a, b) represents the price range in which the position is effective. During a trade, if the AMM price exits the specified range (a, b) of an LP's position, that liquidity will not be used to support the trade outside the specified price range. At price q, this liquidity position reserves token amounts

$$x = x^{(a,b)}(q) \triangleq L \cdot \left(\frac{1}{\sqrt{\hat{q}_{a,b}}} - \frac{1}{\sqrt{b}} \right), \qquad y = y^{(a,b)}(q) \triangleq L \cdot \left(\sqrt{\hat{q}_{a,b}} - \sqrt{a} \right), \tag{3}$$

where $\hat{q}_{a,b} = \min\{b, \max\{a, q\}\}$ is the projection of the price q onto the interval (a, b). By setting $(a, b) = (0, \infty)$, this relation reduces to (1), which implies that a CPMM is a special case of a CLMM when every LP chooses the full price range. However, in general cases where $0 < a < b < \infty$, the reserves of the CLMM liquidity position remain constant with $x = 0$ when $q \geq b$ and with $y = 0$ when $q \leq a$. This implies that 1) the position is unused when q moves out of range (a, b), and 2) there exists a maximum reserve of token X and token Y for such positions.

A CLMM also supports mint, burn and swap operations, as long as they are consistent with the total reserves in the pool (e.g. do not try to swap tokens that do not exist). A swap outputs Δy in the following steps:

1. **Calculate the new price q'**: Since, for each position, x is a continuous and monotonic function of q, the total liquidity reserve x_{total} is also continuous and monotonic in q. Therefore, we can determine the post-swap price q' by solving for the value of q' that satisfies $x_{\text{total}}(q') = x_{\text{total}}(q) + \Delta x(q')$, which is well-defined due to the invertibility of $x_{\text{total}}(\cdot)$. Provided that q' remains within the union of all active price ranges, the swap can be successfully executed (refer to [1] for implementation details).
2. **Update the token reserve**: Each position keeps (L, a, b) constant, so we recalculate x' and y' by plugging q' into (3). The change in the second token, Δy, can then be obtained by summing all y' values.

By concentrating liquidity within a narrower price range, CLMMs enable each active position to support a higher volume of trades with the same amount of capital. This increased capital efficiency was designed to reduce price slippage for traders - as larger trades can be absorbed with smaller price movements [1]- and to allow LPs to extract more from their money, since they can have a higher liquidity, therefore a higher share of the fees, using the same budget by picking a narrower range. At the same time, when the market price exits a position's specified range, that liquidity becomes inactive or "frozen", and only the remaining active positions support the trade. As a result, traders of such trades may experience higher slippage compared to a scenario where liquidity is distributed across the entire price spectrum.

In practice, CLMMs discretize the price axis using a system of *ticks*. Let T be the set of possible ticks, $T = \{t_i \mid i = 0, \dots, M\}$, with $M \in \mathbb{Z}_{>0}$.[3] As a result, the set of valid price ranges is given by $\mathcal{R} = \{(a, b) \in T \times T \mid a < b\}$. Liquidity providers are restricted to choosing ranges $(a, b) \in \mathcal{R}$, so liquidity remains constant between consecutive ticks (t_m, t_{m+1}). Within each such interval (or across consecutive intervals with identical liquidity), price changes follow the structure defined in (3), which reveals a tight coupling between price movement and the liquidity available in the pool.

2.2.1 Fees

One of the incentives for liquidity providers to participate in the pool is the collection of fees. Each transaction pays a fee proportional to the input amount, and this fee is distributed pro-rata among all providers with active liquidity in the corresponding price range. The fee amount varies across pools, each characterized by a parameter $\alpha \in (0, 1)$ that determines the fraction of the input amount collected as a trading fee.

In a CLMM, every time a provider mints a position (L, a, b), they increase the liquidity present in every tick $t_m \in (a, b)$. To assess the liquidity available at a specific tick m, we sum the liquidity of all positions that include it. Formally, we define the liquidity provided at tick m by a provider $n \in [N]$ as $P_{n,m}$ and calculate the total liquidity as:

$$P_m = \sum_{n \in [N]} P_{n,m} \tag{4}$$

Let $\delta = (\delta_m)_{m=1}^{M}$ denote the vector of fees (in dollars) collected per tick during a swap. Then, $\delta_m \triangleq \Delta x_m \cdot \alpha \cdot p_x$, where Δx_m is the amount of tokens exchanged in tick m

$$\Delta x_m = P_{n,m} \left(\frac{1}{\sqrt{\hat{q}'_m}} - \frac{1}{\sqrt{a_m}} \right).$$

As discussed above, the total fee from a swap is distributed pro-rata among all providers with active liquidity. The fee earned by provider n is given by:

$$\mathcal{F}_n = \sum_{m=1}^{M} \delta_m \cdot \frac{P_{n,m}}{P_m},$$

where δ_m is the total fee from swap m and $P_{n,m}$ is the liquidity contributed by passive provider n over tick m.

[3] Ticks are defined by exponential spacing: $T = \{t_i \mid i = 0, \dots, M\}$, where $t_i = 1.0001^{\tau(i-\iota)}$ with $\tau, M \in \mathbb{Z}_{>0}$ and $\iota \in \mathbb{Z}$.

2.3 Price Impact (a.k.a. Impermanent Loss)

Prior literature commonly assumes that the price ratio between two tokens is stable and externally set by the market, given by $q_{\text{market}} = \frac{p_x}{p_y}$, where p_x and p_y denote the market prices of tokens X and Y, respectively [3, 4, 25]. However, the actual price within a CLMM pool is determined endogenously – it is only changed by trades in the pool. If it does not coincide with the market price, arbitrage activities become profitable. At equilibrium – when no arbitrage opportunities remain – the two values align, such that $q = q_{\text{market}}$. This alignment is often assumed in theoretical analyses, particularly in the absence of active participants like JIT LPs and arbitrageurs.

The divergence between the pool price q and the market price thus represents both a risk and an opportunity. On one hand, price mismatches expose LPs to impermanent loss – the difference in value between a liquidity position and the equivalent token holdings outside the pool. On the other hand, such mismatches create arbitrage opportunities for sophisticated agents, including JIT LPs. To encompass both favorable and adverse outcomes of price divergence, we adopt a more general term and refer to the impermanent loss effect as *price impact* throughout the remainder of this paper.

Consider an LP n who mints a position when the pool price is q and later withdraws it at pool price q'. Let p_x and p_y denote the external market prices of tokens X and Y at the time of minting, and let p'_x and p'_y be their respective market prices at the time of withdrawal. Following (3), we denote by $x_n(q)$ and $y_n(q)$ the token amounts associated with the position of LP n when the pool price is q (at minting), and by $x_n(q')$ and $y_n(q')$ the corresponding amounts when the price is q' (at withdrawal). The dollar value of the position at the time of minting and at the time of withdrawal is:

$$V_{\text{mint}}(L, p_x, p_y, q) = p_x \cdot x_n(q) + p_y \cdot y_n(q), \quad V_{\text{withdraw}}(L, p'_x, p'_y, q') = p'_x \cdot x_n(q') + p'_y \cdot y_n(q'). \tag{5}$$

▶ **Definition 2** (Absolute price impact). *The* absolute price impact *is the net change in value of the position:*

$$\mathcal{C}_n(\Delta x, L; p_x, p_y, p'_x, p'_y, q, q') \triangleq V_{mint}(L, p_x, p_y, q) - V_{withdraw}(L, p'_x, p'_y, q'). \tag{6}$$

In what follows, to simplify notation we omit the explicit dependencies on parameters of the above quantities, e.g., referring to $\mathcal{C}_n(\Delta x, L; p_x, p_y, p'_x, p'_y, q, q')$ simply as $\mathcal{C}_n$. Note that a *negative* price impact ($\mathcal{C}_n < 0$) corresponds to a *gain* for the liquidity provider relative to holding the assets outside the pool. Conversely, a *positive* price impact implies a relative loss.

▶ **Definition 3** (Relative price impact). *The* relative price impact *is the ratio of this value difference to the initial position value:*

$$\mathbf{PI} \triangleq \mathcal{C}_n / V_{mint}. \tag{7}$$

Throughout the rest of this paper, we omit the LP subscript n when the context is clear, and refer to quantities such as $x_n(q)$, $y_n(q)$ simply as $x(q)$, $y(q)$.

3 When is Price Impact Beneficial to LPs?

As explained previously, an LP n gains when the fees accrued $\mathcal{F}_n$ are larger than the price impact $\mathcal{C}_n$. However, LPs may struggle to predict these tradeoffs for three reasons: (1) Price fluctuations and transaction patterns are inherently unpredictable. (2) Given information about price and transaction fluctuations, we currently lack closed-form expressions showing when LPs will benefit as a function of transaction size and current market conditions. (3) Even if we had knowledge of items (1) and (2), most LPs invest on a slow timescale (days or even weeks [16]), over which it is not possible to take advantage of per-transaction profits.

A key observation is that Just-In-Time (JIT) LPs do not suffer from the 1st and 3rd constraints – they can act quickly enough to move on a per-transaction basis using current price data. Hence, if we can resolve item (2), JIT LPs can determine with some certainty when CLMM transactions will be profitable.

In this section, we resolve challenge (2) by formally characterizing the conditions under which $\mathcal{C}$ is non-positive or positive in CLMMs; we show that the answer depends on whether the pool price moves toward or away from the prevailing market price. Our results in this section are not specific to JIT LPs, but as mentioned previously, passive LPs may not be able to take advantage of these results, as they move their liquidity on much slower time scales.

Assumptions. While deriving the following results, we assume that the market prices of tokens X and Y remain constant throughout the trade, i.e., $p_x = p'_x$ and $p_y = p'_y$. This is because the trade occurs instantaneously when the block including the transaction is finalized on the blockchain. In this case, the price impact simplifies to the difference in the dollar value of the LP's position before and after the trade, evaluated using fixed market prices. According to Definition 2:

$$\mathcal{C}(\Delta x, L; p_x, p_y, q, q') = p_x\,(x(q) - x(q')) + p_y\,(y(q) - y(q')) = -p_x \Delta x(q,q') - p_y \Delta y(q,q'). \quad (8)$$

In what follows, to simplify notation we refer to $\mathcal{C}(\Delta x, L; p_x, p_y, q, q')$ simply as $\mathcal{C}$.

3.1 Conditions for Favorable Price Impact

Our first main result consists of necessary and sufficient conditions under which a price change $q \to q'$ results in non-positive price impact. This threshold condition delineates precisely when an LP's position becomes favorable or unfavorable relative to simply holding the tokens at market prices

▶ **Theorem 4** (Threshold condition for price impact). *Let $\hat{q} \triangleq \max\{a, \min\{b, q\}\}$, and similarly for $\hat{q}'$. The change in pool price relates to price impact as follows:*

Case 1: *$q' < q$: $\mathcal{C} \le 0$ if and only if*

$$\frac{1}{\hat{q}}\left(\frac{p_x}{p_y}\right)^2 \ge \hat{q}'. \quad (9)$$

Case 2: *$q < q'$: $\mathcal{C} \le 0$ if and only if*

$$\frac{1}{\hat{q}}\left(\frac{p_x}{p_y}\right)^2 \le \hat{q}'. \quad (10)$$

Proof sketch. (Full proof in [18]) We establish conditions under which an LP experiences non-positive price impact. Replacing $p_x = p'_x, p_y = p'_y$ into Eq.(7):

$$\mathbf{PI} = 1 - \frac{V_{\text{withdraw}}(L, p_x, p_y, q')}{V_{\text{mint}}(L, p_x, p_y, q)} \le 0 \iff 1 \le \frac{p_x x_n(q') + p_y y_n(q')}{p_x x_n(q) + p_y y_n(q)}. \quad (11)$$

Therefore, the inequality holds iff $V_{\text{withdraw}} \ge V_{\text{mint}}$. The result follows from substituting Eq. (3) into the above expression, followed by algebraic manipulation. ◀

Based on this result, we can establish sufficient conditions for either favorable or adverse price impact. These are summarized in the following corollary.

▶ **Corollary 5** (Gains under diverging prices and losses under converging prices). *When initial and final prices do not cross p_x/p_y, gains and losses are determined by the direction of the price movement:*

- *Gain under diverging prices: $\mathcal{C} \leq 0$ if $q' < q \leq p_x/p_y$ or $q' > q \geq p_x/p_y$, i.e., if a trade moves the AMM price* away *from the external market price, the LP experiences gains.*
- *Loss under converging prices: $\mathcal{C} \geq 0$ if $q < q' < p_x/p_y$ or $q > q' > p_x/p_y$, i.e., if a trade moves the AMM price* towards *the external market price, the LP experiences losses.*

▶ Remark 6 (No-loss if pool price initially aligned with market). Note that if the pool price is initially aligned with the market price, i.e., $q = p_x/p_y$, then the price impact $\mathcal{C}$ experienced by a liquidity provider is *always* non-positive: $\mathcal{C} \leq 0$. This immediately follows from the fact that there is gain under diverging prices (first case considered in the above corollary), and if the market price is initially aligned with pool price, prices will diverge.

Intuitively, when the pool price moves toward the market price (first case in Corollary 5), traders obtain better execution, acquiring tokens from the pool at prices closer to fair market value – resulting in a loss for LPs. Conversely, when the pool price moves away from the market price (second case in Corollary 5), traders face worse execution, effectively overpaying and enriching LPs.

3.2 The Effects of Liquidity

We analyze the limiting behavior of price impact as available liquidity changes. Suppose we have a price range (a, b) which is currently covered by total liquidity L by passive LPs. A JIT LP is considering adding liquidity to this range; intuitively, when liquidity L tends to zero, the position is too small to have any meaningful participation, and the price impact vanishes. Conversely, as $L \to \infty$, the position becomes large enough that the price becomes almost static. Formally, we establish the following result

▶ **Lemma 7** (Limiting behavior of price impact). *Let Δx be the quantity of token X exchanged in a trade. Consider $q \in (t_m, t_{m+1}]$ the current pool price and consider (L, a, b) to be the single position of some provider n such that $q \in (a, b)$, in words, consider L to be all the liquidity minted before the transaction by some provider n. Then:*

$$\lim_{L\to\infty} \mathcal{C} = \Delta x \cdot (p_y \cdot q - p_x), \quad \textit{and} \quad \lim_{L\to 0} \mathcal{C} = 0.$$

Proof sketch. (Full proof in [18]) The proof follows from Eq. (8) and the following facts derived in [1] (see also Eq. (3)): $\frac{\Delta x}{L} \triangleq \Delta \frac{1}{\sqrt{q}} = 1/\sqrt{q'} - 1/\sqrt{q}$ and $\frac{\Delta y}{L} \triangleq \Delta\sqrt{q} = \sqrt{q'} - \sqrt{q}$. Then, $\Delta y = -\Delta x \sqrt{q'}\sqrt{q}$. As $L \to \infty$, we have $q' \to q$. In the limit, $\Delta y \to -q\Delta x$, and $\mathcal{C} \to -p_x \Delta x + p_y q \Delta x$ (see Eq. (8)). As $L \to 0$, the LP's position is infinitesimal, yielding negligible participation in the trade and thus $\mathcal{C} \to 0$. ◀

This result reinforces the intuition that liquidity acts as a dampener of price impact: as liquidity increases, the pool becomes less sensitive to trade-induced price shifts, and the value transfer is increasingly governed by the degree of misalignment between pool and market prices. To illustrate the broader implications of price dynamics and their relationship with price impact, we present the following example.

▶ **Example 8.** Figure 1 provides a geometric interpretation of price impact in the (q, q')-plane, where the market price ratio is fixed at $p_x/p_y = 1$. The space is partitioned into colored regions: green for $\mathcal{C} \leq 0$ (favorable to LPs), and red for $\mathcal{C} \geq 0$ (unfavorable to LPs). Corollary 5 provides sufficient conditions for LPs to experience non-positive price impact – Regions VI and II. Conversely, Corollary 5 also identifies conditions under which price impact is non-negative – Regions VII and III. More intricate behavior arises when the price crosses

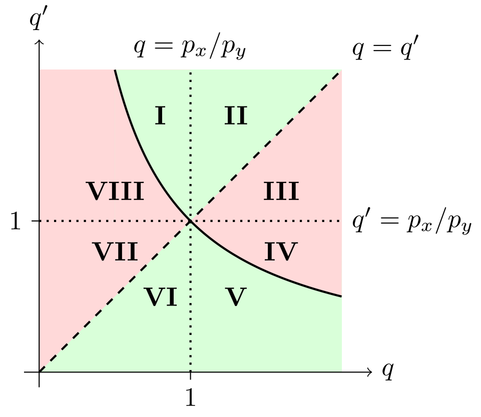


Figure 1 Partition of the (q, q')-space with $p_x/p_y = 1$; black curve corresponds to $q' = 1/q$. Red and green regions correspond to **PI** ≥ 0 and **PI** ≤ 0, respectively.

the market value (Theorem 4). For example, in Region I ($q' > p_x/p_y > q$ and $q' > 1/q$), price impact is favorable; in Region VIII ($q' < 1/q$), it becomes unfavorable. A symmetric situation arises for Regions IV and V depending on whether $q' < 1/q$.

Analogous to our analysis of price impact, we can show that the fee function is continuous with respect to the liquidity variable and converges to a finite limit as liquidity grows large. This result is formalized in the following lemma

▶ **Lemma 9** (Effect of Liquidity). *Let q be the current price in a CLMM, where $q \in (t_m, t_{m+1}]$. Suppose we have a target trade that pays Δx X tokens to the pool and asks for Y tokens, which can be supported by the CLMM. Immediately before the trade, a JIT LP adds liquidity $L \in [0, \infty)$ to price range (a, b) where $a \leq t_m < t_{m+1} \leq b$. Let $q'(< q)$ be the resulting price after the trade as a function of L. Then, q' is continuous, strictly decreasing in L, and $\lim_{L\to\infty} q' = q$.*

Proof sketch. (Full proof in [18]) We model the final price q' after the swap as a function of added liquidity L. The swap demand Δx must be fulfilled across ticks, so we express q' as an implicit function of L. As L increases, the same Δx has less price impact, so $q' \uparrow$. The mapping is strictly decreasing in L and continuous due to the functional form of the AMM and the monotonicity of the inverse square root. As $L \to \infty$, $q' \to q$, meaning the AMM price becomes increasingly resistant to change. ◀

▶ Remark 10. Note that, since $P_m = \sum_{i\in[N]} P_{n,m}$, Lemma 9 is also true for any $P_{n,m}$; that is, $\lim_{P_{n,m}\to\infty} q' = q$ $\forall n$ such that $t_m < q \leq t_{m+1}$. This means that it is sufficient for **any** provider to inject high amount of liquidity in order to force the limit condition.

Lemma 9 highlights one of the central messages of this work: by concentrating liquidity when and where it is needed, JIT LPs can dampen price volatility, controlling slippage and choosing how the price will change. This empowers the JIT LP to optimize their revenue by balancing earned fees and losses due to price changes (See §2.3).

4 Just-In-Time Liquidity Allocation: An Optimization Perspective

We now define the formal structure of the decision process faced by a JIT LP observing a pending transaction. We begin by explaining the relevant properties of JIT LPs (§4.1), which informs our model, including the strategy space available to the JIT LP (§4.2.1), and the utility function that governs decision-making. Finally, we establish conditions under which an optimal strategy exists (§4.2.2).

4.1 Background on Just-In-Time LPs

Unlike ordinary LPs who passively leave their liquidity in the liquidity pool, Just-In-Time (JIT) LPs react to market conditions in real-time to extract profit [26]. We next present two important aspects of JIT LP behavior, and explain how we model them.

(1) Per-transaction optimization. JIT LPs operate by monitoring the public *mempool*[4] [27, 29] for trade opportunities. This allows JIT LPs to selectively decide whether to provide liquidity for each trade before the trade is validated and finalized on the blockchain.

Once a JIT LP identifies a favorable trade, it can submit a *bundle*, which is a set of one or more transactions submitted together for inclusion in a block, typically via a relay to a block builder (e.g., via FlashBots [17]). These bundles are guaranteed atomic execution: the entire bundle either succeeds or fails as a group, ensuring riskless, single-trade provisioning. Exploiting this atomicity, the JIT LP can execute various operations-e.g., *sandwich attacks*, which mint liquidity just before a swap and burn it immediately after [28].

Modeling Implications: JIT LPs exploit this information asymmetry to compute their expected utility for a given trade. Based on this foresight, they can make optimal, *transaction-level* decisions about whether to mint and burn liquidity, factoring in both trading fees and the impact of price movements.

This naturally induces a two-stage optimization framework: first, passive LPs select their liquidity allocation strategy for a longer period [25]; second, for each observed trade, JIT LPs react by solving an online optimization problem to maximize net expected revenue, potentially adjusting for inclusion costs such as gas fees or auction bids. Our model focuses on this second optimization.

(2) Competition among JIT LPs. With multiple JIT LPs potentially competing over the same trade, each constructs and submits a bundle. The bundle includes a tip to the block builder, who selects bundles to maximize the total value extractable from the block by adjusting transaction ordering, inserting additional transactions, or censoring transactions.

This creates a sealed-bid auction: each JIT LP submits a bundle with a bid, without visibility into others' bids, and the builder selects the one that maximizes their expected revenue, i.e., only the highest-bidding bundle is included. The winning JIT LP earns a portion of the fees and pays a tip bounded below by the second-highest bid. This mechanism forms the basis of our transaction-level utility model, where we analyze how JIT LPs optimize their strategy under fixed surplus and uncertain inclusion.

Modeling Implications: Modeling this, recall that for a swap of Δx tokens of X and fee rate α, the total fee generated by the swap is $\delta = \alpha \cdot p_x \cdot \Delta x$, where p_x is the market price of token X in dollars. Since both Δx and p_x are visible in the mempool, δ is known to all JIT LPs in advance. As a result, they compete over a fixed surplus, strategically offering portions of δ as tips to the block builder to secure inclusion.

[4] The mempool is a public set of pending transactions propagated through the blockchain peer-to-peer network.

4.2 Model

Formally, we represent the JIT optimization problem as a tuple $([N], \mathcal{U}, \mathcal{S}, \theta, \rho)$. Here, $[N]$ denotes the set of all passive liquidity providers in the pool. The strategy space $\mathcal{S}$ encapsulates all feasible actions available to the JIT LP, while $\mathcal{U}$ is the utility function mapping each strategy to a real-valued profit – computed as the difference between earned fees, incurred price impact, and bidding costs. The vector θ includes all relevant swap parameters, such as trade size, fee rate, price information, and the liquidity distribution of passive providers. Finally, ρ is a constant that denotes the JIT budget.

Note that the JIT optimization problem assumes that the passive LP positions are established a priori. We denote by $\boldsymbol{s}$ the strategy profile of passive LPs, encoded as a vector of liquidity allocations across ticks: $\boldsymbol{s} = (L_i, a_i, b_i)_{i\in[N]}$. The aggregate effect of the passive LP strategies $\boldsymbol{s}$ induces a liquidity distribution function $P = (P_m)_{m\in M}$, where each P_m denotes the total passive liquidity available at tick m. Formally, P is derived via a deterministic mapping from $\boldsymbol{s}$. In the remainder of this section, we use $\boldsymbol{s}$ to refer to the underlying strategies and P to denote the induced state of the pool.

4.2.1 Strategy Space $\mathcal{S}$

A JIT LP's choice of action consists in selecting one position[5] (L, a, b) to be minted immediately before a swap and burned immediately after. Since the entire state of the pool is observable at decision time, the JIT LP can make an informed choice based on current market and pool conditions. To formalize this, we define the swap as a 6-tuple $\theta \triangleq (\Delta x, q, P, p_x, p_y, \alpha)$, where:

- Δx is the amount of token X being exchanged by the trader in the target trade;
- q is the current pool price;
- $P = (P_m)_{m\in M}$ is the per-tick liquidity distribution from passive LPs;
- p_x and p_y are the external market prices (in dollars) of tokens X and Y, respectively;
- α is the pool's fee rate.

Let q^* denote the price after the swap without the presence of JIT LPs. We define

$$\mathcal{R}^{(q,q^*)} \triangleq \mathcal{R} \cap \left\{(a,b) \middle| \forall m : a \le t_m < t_{m+1} \le b,\ [t_m, t_{m+1}] \cap (\min\{q, q^*\}, \max\{q, q^*\}) \neq \varnothing \right\}.$$

Intuitively, $\mathcal{R}^{(q,q^*)}$ includes all feasible price ranges that do not contain a subrange $[t_m, t_{m+1}]$ that is never "touched" when price moves from q to q^*. We claim that the JIT LP only considers price ranges in $\mathcal{R}^{(q,q^*)}$ for their choice of liquidity position. Otherwise, we can break the position (L, a, b) down to two positions (L, a, c) and (L, c, b) where one of their ranges does not intersect with the price moving range $(\min\{q, q^*\}, \max\{q, q^*\})$. Let it be (a, c) without loss of generality [25]. During the trade, (L, a, c) will incur fee income $\mathcal{F} = 0$ and zero token amount change, which further implies price impact $\mathcal{C} = 0$ since p_x and p_y are both constant. Hence, in comparison with an action that chooses (L, c, b) instead, the (L, a, b) has equal utility impact while wasting budget on (a, c). Accordingly, we define the JIT strategy space as:

$$\mathcal{S} = \left\{(L, a, b) \;\middle|\; (a,b) \in \mathcal{R}^{(q,q^*)},\ L \cdot V_{(a,b)} \le \rho,\ q \in [a,b] \right\} \cup \{\perp\},$$

where $\perp$ denotes non-participation, ρ denotes the LP's budget, and $V_{(a,b)}$ is the dollar cost per unit of liquidity at price q over the interval (a, b).

[5] We do not consider multiple-position actions for two main reasons: 1) it is commonly assumed in the literature for JIT to mint constant liquidity [28], and 2) the data provided in [25] reveals that transactions from 2024 to 2025 by JIT LPs include only a single position.

4.2.2 Utility Function

The objective of the JIT LP is to maximize a utility function $\mathcal{U}$, defined as the net profit obtained from participating in a given swap. This profit captures the trade-off between fees earned and the price impact incurred. Given a strategy $s = (L, a, b) \in \mathcal{S}$ and swap parameters θ, the utility is defined as:

$$\mathcal{U}(s;\theta) = \mathcal{F}(s;\theta) - \mathcal{C}(s;\theta), \tag{13}$$

where $\mathcal{F}(s;\theta)$ is the total fee income and $\mathcal{C}(s;\theta)$ denotes the price impact associated with the strategy. The JIT LP aims to solve:

$$s^* = \arg\max_{s\in\mathcal{S}} \; \mathcal{U}(s;\theta). \tag{14}$$

In practice, a JIT LP needs to additionally pay a cost v for placing bids in auctions (such as Flashbots [17]) to land the sandwich attack on the trade transaction. Ideally, we should incorporate the auction in our model. Two natural approaches are (1) to model the game among multiple JIT LPs, or (2) model the probability of winning the bid as a function of v. Both approaches require empirical study on historical auctions that involves not only auction winners, but more importantly, losers. Such data is not currently accessible, to the best of our knowledge, and it is unrealistic to make ad hoc assumptions. Hence, for simplicity, we assume v to be a constant that does not relate to the choice of liquidity position. As a result, we can easily extend our model by updating the budget constraint as $L \cdot V_{(a,b)} + v \leq \rho$ and the utility function as $\mathcal{U}(s;\theta) = \mathcal{F}(s;\theta) - \mathcal{C}(s;\theta) - v$.

Given a specific strategy $s \in \mathcal{S}$ and the current swap context, we can compute the total utility associated with executing the strategy. From this point forward, we omit explicit dependence on θ and s whenever the context is clear.

The total fee earned by the JIT LP is decomposed tick-wise. Since the JIT LP deploys a single position (L, a, b), let L be the liquidity of the JIT on tick m. The JIT's share of the fees at tick $m \in M$ is:

$$\mathcal{F}_m \triangleq \underbrace{\Delta x_m \alpha p_x}_{\text{Fee } = \delta_m} \cdot \frac{L}{\sum_{i\in[N]} P_{i,m} + L}, \qquad \mathcal{F} \triangleq \sum_{m\in M} \mathcal{F}_m. \tag{15}$$

Let t_m denote the price at tick m. Recall that α, p_x are trade parameters, and that Δx_m depends on q' and denotes the amount of token X that is injected at each tick m (see §2.2.1). As the swap affects only the ticks crossed during execution, the tick-level fee δ_m is nonzero only for ticks m such that $t_m \in (q, q')$. Under these conditions, the fee allocation simplifies to:

$$\mathcal{F}_m = \delta_m \cdot \frac{L}{\sum_{i\in[N]} P_{i,m} + L}, \qquad \mathcal{F} = \sum_{m\in M \,|\, t_m \in (a,b)} \mathcal{F}_m. \tag{16}$$

From it, we can show the following lemma:

▶ **Lemma 11.** *Consider an LP n with a single position (L, a, b), such that $q \in (a, b)$. Let there be a swap that pays $\Delta x > 0$ X tokens. The total fee earned by the provider is given by:*

$$\mathcal{F}(L) = \sum_{\forall m | t_m \in (a,b)} \delta_m \cdot \frac{L}{L + \tilde{P}_{n,m}},$$

where $\tilde{P}_{n,m} = \sum_{i\neq n} P_{i,m}$ is the total liquidity in tick m without $L = P_{n,m}$. In addition, $\mathcal{F}(L)$ is continuous in L and is bounded above by $\alpha \cdot \Delta x$,

$$\lim_{L\to\infty} \mathcal{F}(L) \leq \alpha \cdot \Delta x. \tag{17}$$

Proof sketch. (Full proof in [18].) We express the total fee earned by the LP as a sum over ticks, weighted by the LP's share of liquidity (16). Since each fee component depends continuously on L through the liquidity share, the full fee function is continuous in L. As $L \to \infty$, the LP's share of each tick approaches 1, but the total fee is still bounded above by $\alpha \cdot \Delta x$, since that is the maximum total fee generated by the swap. Hence, the limit exists and is bounded. ◀

The price impact $\mathcal{C}$ is likewise decomposed across ticks as:

$$\mathcal{C}_m \triangleq p_x \cdot (x(\hat{q}_m) - x(\hat{q}'_m)) + p_y \cdot (y(\hat{q}_m) - y(\hat{q}'_m)) \,, \quad \mathcal{C} \triangleq \sum_{m \in M \,|\, t_m \in (a,b)} \mathcal{C}_m, \tag{18}$$

where $\hat{q}_m = \min\{t_{m+1}, \max\{q, t_m\}\}$ is the entry price, $\hat{q}'_m$ the exit price within tick m and $x(q)$, $y(q)$ are given by Eq. (3).

This decomposition allows the utility to be expressed in tick-wise form:

$$\mathcal{U}_m = \mathcal{F}_m - \mathcal{C}_m, \quad \mathcal{U} = \sum_{m \in M \,|\, t_m \in (a,b)} \mathcal{U}_m. \tag{19}$$

Recall that all parameters $\theta \triangleq (\Delta x, q, P, p_x, p_y, \alpha)$ are fixed by the given trade, and since q' depends on (L, a, b), the only free variables are $L, (a, b)$. Using Lemmas 9, 7, and 11, we establish the following theorem:

▶ **Theorem 12.** *The utility function $\mathcal{U}$, as defined in Eq. (19), attains a global maximum over the strategy space $\mathcal{S}$. That is, there exists an optimal strategy $s^* \in \mathcal{S}$ such that $\mathcal{U}(s^*) \geq \mathcal{U}(s)$ for all $s \in \mathcal{S}$.*

Proof sketch. (Full proof in [18].) The utility function $\mathcal{U}$ depends on liquidity L and the chosen price range (a, b). Since tick prices are discrete, there are finitely many such intervals. For a fixed interval, $\mathcal{U}(L)$ is continuous due to the continuity of both fee ($\mathcal{F}$) and cost ($\mathcal{C}$) functions in L. Also, $\mathcal{F}$ is bounded above while $\mathcal{C}$ grows at most linearly in L, constrained by a budget ρ. Therefore, the domain of feasible L is compact, and $\mathcal{U}$ attains a maximum on it. Optimizing over all intervals proves existence of an optimal strategy. ◀

This guarantees that for any swap, there exists a profit-maximizing strategy for the JIT LP. In the following section, we propose an algorithm to compute this strategy and empirically explore how it behaves across different swap conditions.

5 An Algorithmic Solution to the JIT Optimization Problem

We now present an algorithmic approach for computing the optimal strategy $s^* = (L^*, a^*, b^*)$ that maximizes the utility $\mathcal{U}$ of a JIT LP in response to a given target swap. This is made possible by Theorem 12, which guarantees the existence of a global optimum.

The utility function can be challenging to optimize since there are many parameters and the utility function is nonconcave in general. In addition, we lack a closed-form formula for q', which is calculated algorithmically. To address these challenges, we use empirical observations to reduce the search space. First, we note that θ, the fixed trade parameters, are known beforehand by the JIT trader; this turns the utility into a function of three variables: a, b, L.

In theory, the JIT trader could choose any (a, b) interval, of which there are $O(M^2)$. However, due to the monotonicity of q' in the liquidity, the original price q cannot move farther than q^*, where q^* is the anticipated post-swap price in the *absence* of JIT intervention.

Hence, we limit ourselves to tick ranges $(a, b) \in \mathcal{R}^{(q,q^*)}$. While this does not reduce the asymptotic worst-case complexity of our search, in our data we observe that for 95% of transactions, there are at most 2 ticks between q and q^* (see §6). Hence, enumerating the intervals in $\mathcal{R}^{(q,q^*)}$ is practically feasible.

Algorithm 1 provides a computational method to search for the optimal JIT liquidity provisioning strategy in response to an observed swap. The algorithm iterates over all candidate price intervals $(a, b) \in \mathcal{R}^{(q,q^*)}$ (line 3). For each interval (a, b), it solves a one-dimensional, non-linear, non-concave optimization problem to identify the liquidity level L^* that maximizes the JIT LP's utility:

$$L^*(a, b; \theta) = \arg\max_L \left(\mathcal{F}^{(a,b)}(L; \theta) - \mathcal{C}^{(a,b)}(L; \theta)\right) \tag{20}$$

subject to the budget constraint $p_x x^{(a,b)}(q') + p_y y^{(a,b)}(q') \leq \rho$

$$\mathcal{F}^{(a,b)}(L; \theta) = \sum_{m \in M | t_m \in (a,b)} \Delta x_m \cdot \alpha \cdot \frac{L}{L + P_m} \cdot p_x \tag{21}$$

$$\mathcal{C}^{(a,b)}(L; \theta) = \sum_{m \in M | t_m \in (a,b)} \left(-p_x \Delta x_m + p_y \Delta x_m \hat{q}_m \cdot \frac{L + P_m}{\Delta x_m \sqrt{\hat{q}_m} + L + P_m}\right) \cdot \frac{L}{L + P_m}. \tag{22}$$

If the resulting utility exceeds the current maximum, the optimal strategy is updated accordingly (lines 6–8). After examining all admissible ranges, the algorithm returns the best strategy $s^* = (L^*, a^*, b^*)$ (line 11) that maximizes the JIT LP's profit in the given swap scenario. The optimal liquidity pair (a^*, b^*) and the corresponding optimal liquidity L^* are given by $L^* = \arg\max_L \mathcal{F}^{(a^*,b^*)}(L; \theta) - \mathcal{C}^{(a^*,b^*)}(L; \theta)$ such that $\mathcal{U}^{(a^*,b^*)}(L^*; \theta) = \mathcal{U}(s^*; \theta) \geq \mathcal{U}(s; \theta)\ \forall\ s \in \mathcal{S}$, where s^* is given by the triplet $s^* = (L^*, a^*, b^*)$ (see §4.2.1).

As long as we can find the global optimum of a bounded continuous univariate function, Algorithm 1 returns an optimal strategy to maximize JIT utility. In our experiments, we tried both Particle Swarm Optimization (PSO) and binary search to solve for L^*. Since in most cases the optimal solution was to use the entire budget, both algorithms found the best strategy, but in non-trivial cases, PSO performed better at the cost of more computational power. Our implementation of Algorithm 1 can be found on GitHub.[6]

▶ Remark 13 (Token Directionality). All derivations in this section assume $\Delta x > 0$, i.e., token X is being sold. To handle the symmetric case $\Delta y > 0$, where token Y is sold instead, we define a transformation $Y(\theta) = (\Delta y, 1/q, P, p_y, p_x, \alpha)$. This mapping ensures that all formulas and results continue to hold under the appropriate relabeling of tokens.

5.1 Strategic Archetypes in CLMM Swaps

We classify trades into three canonical scenarios – referred to as *strategic archetypes* – that capture the dominant patterns and analytical results of Theorems 4 and 12.

- **Overpriced trade:** The trade moves the pool price away from the market price, resulting in the purchase of the more expensive token. Formally, this occurs when $q' < q \leq p_x/p_y$ or $q' > q \geq p_x/p_y$. According to Corollary 5, such movements are associated with non-positive price impact, meaning the LP benefits (Figure 2(a1)).

[6] `https://github.com/brunoCCOS/JITUniswapOptimization`

Algorithm 1 JIT Optimal Strategy Search.

```
Require: Swap parameters $\theta = (\Delta x, q, P, p_x, p_y, \alpha)$  Budget $\rho$
Ensure: Optimal strategy $s^* = (L^*, a^*, b^*)$
 1: Initialize $\mathcal{U}_{\max} \leftarrow -\infty$
 2: Set $s^* \leftarrow \perp$
 3: for each $(a, b) \in \mathcal{R}^{(q,q^*)}$ do
 4:     Solve for optimal liquidity:  $L^*(a, b; \theta)$ [Optimization (20)]
 5:     Evaluate resulting utility:  $\mathcal{U} \leftarrow \mathcal{F}^{(a,b)}(L^*; \theta) - \mathcal{C}^{(a,b)}(L^*; \theta)$
 6:     if $\mathcal{U} > \mathcal{U}_{\max}$ then
 7:         Update best utility: $\mathcal{U}_{\max} \leftarrow \mathcal{U}$
 8:         Update best strategy: $s^* \leftarrow (L^*, a, b)$
 9:     end if
10: end for
11: return $s^*$
```

- **Arbitrageur trade:** The trade brings the pool price closer to the market price by acquiring the cheaper token. This corresponds to $q < q' \leq p_x/p_y$ or $q > q' \geq p_x/p_y$. As shown in Corollary 5, this condition implies positive price impact, i.e., a loss for LPs. Although this trade is optimal for arbitrageurs, it is typically unprofitable for JIT LPs unless the fees compensate for the loss (Figure 2(a2)).
- **Overshoot trade:** A corrective trade crosses the market price, starting by buying the cheaper token but pushing the pool price past the market value, ultimately overpaying. This configuration – $q < p_x/p_y < q'$ or $q > p_x/p_y > q'$ – lies on the boundary conditions specified in Theorem 4. In this case, whether the price impact is favorable or not depends on whether the post-trade price q' crosses the threshold condition in (9) or (10) (Figure 2(a3)).

Recall that q is the pool price of token Y in terms of token X, so $1/q$ is the pool price of X in Y. When $q > p_x/p_y$, token Y is overpriced in the pool relative to the market, implying that X is relatively cheaper on-chain.

5.2 Optimized Investment Strategies by Archetype

Among all archetypes, participating in *overpriced trades* yields the highest utility. These trades allow JIT LPs to sell tokens above market value across the entire chosen range (a, b).

In *overshoot trades*, JIT LPs can profit by targeting only the favorable segment – i.e., by minting liquidity only in the price range where $\Delta x < 0$ and $q < p_x/p_y < q'$. By concentrating liquidity narrowly in this segment, JIT LPs increase their share of fees while avoiding loss-inducing zones. Figure 2(b) gives an illustrative example; the initial and final price sandwich the market price $q < p_x/p_y < q'$, so the trade is "overshooting" the price. Therefore, in all ticks between $(q, p_x/p_y)$ the price impact will be negative; hence the best strategy is to mint liquidity only on feasible ranges inside $(p_x/p_y, q')$ where the price impact is positive.

In contrast, engaging in *arbitrageur trades* is generally unfavorable unless the fee revenue is high enough to offset the incurred price impact. The following result formally establishes a sufficient condition under which fees are insufficient to cover price impact:

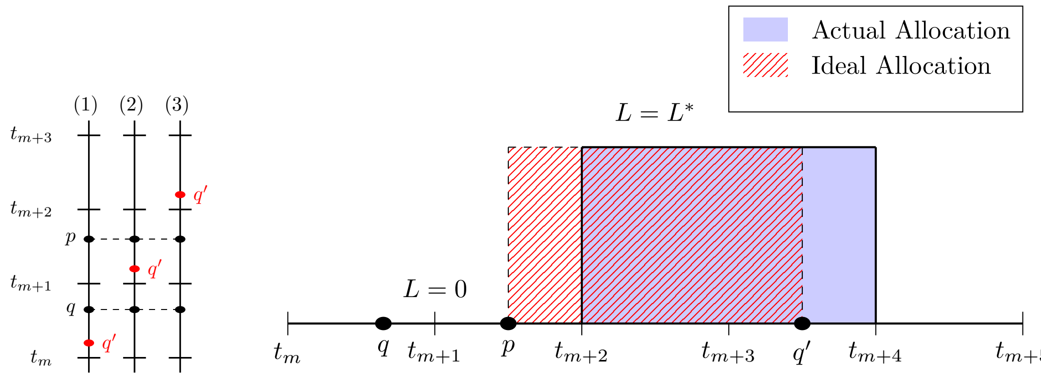


Figure 2 (a) Illustration of the three kind of trades. Each vertical line represents a price axis with different positions for q, p, q' which fully characterizes each type of trade. (1) **Overpriced** trade: q' is further away from p than q. (2) **Arbitrageur** trade: New price q is closer to the market price. (3) **Overshoot** trade. (b) Liquidity distribution across ticks as a function of price. In this overshoot trade example, the optimal strategy is to allocate the entire budget immediately after the pool price crosses the market price. Formally, all liquidity should be placed within the interval (p, q'), where $p = p_x/p_y$, as price impact is non-positive in this region. Since liquidity can only be minted at discrete tick levels, the optimal placement corresponds to the nearest available ticks – in this case, (t_{m+2}, t_{m+4}). This behavior extends to both overpriced and arbitrage trades, depending on tick granularity and pool conditions.

▶ **Proposition 14** (Fees Insufficient to Offset Price Impact). *For a transaction with swap parameters θ (definition in §4), suppose without loss of generality that $\Delta x > 0$. For a price range (a, b), if*

$$1 + \alpha < \frac{\Delta y_m \cdot p_y}{\Delta x_m \cdot p_x} \quad \forall m \in (p(a), p(b)),$$

i.e., if the gained value of Y tokens over the paid value of X tokens is more than the nominal fee rate, then utility is always negative: $\mathcal{U}^{(a,b)} < 0$.

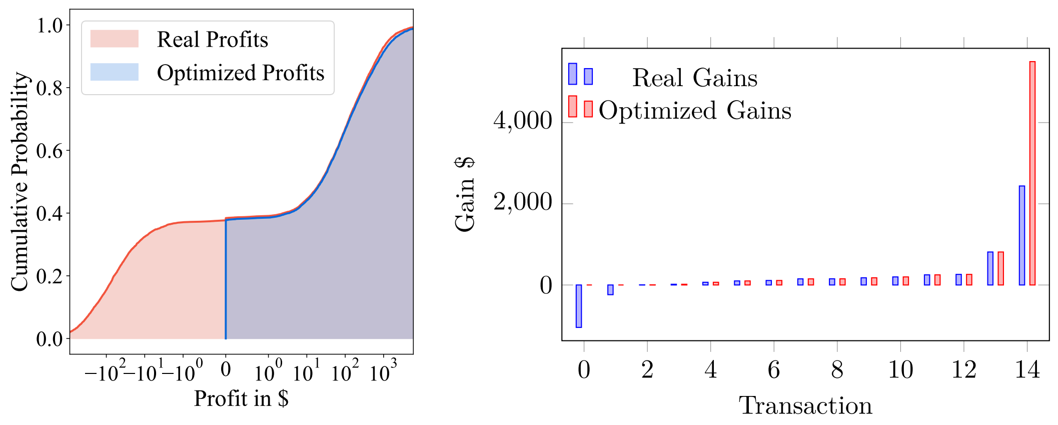
The proof of the above proposition can be found in [18].

6 Empirical Results Using On-Chain Data

In this section, we empirically evaluate the theoretical insights using data from Uniswap V3.

Data Collection. We collected data from a USDC/WETH[7] pool over 6 months: January-June 2024, which comprises 1,013,147 total transactions. The data was collected from the Allium platform [5]. From this data, we extracted all JIT transactions, which were identified by using the same criterion as [11], i.e. matching transactions which were sandwiched by a mint and burn transaction by the same provider. The resulting number of JIT swaps was 6,829. In this section, our results are generated by taking real JIT transactions and simulating our optimized version of JIT investment strategies. We assume throughout that a JIT transaction is realized if and only if it was realized in the source data (i.e., we do not simulate different results of bidding auctions among JIT LPs). To simulate a JIT budget, we

[7] 0x88e6a0c2ddd26feeb64f039a2c41296fcb3f5640 pool hash



(a) CDF of profit for a real JIT LP vs. optimized strategy. Optimized JIT avoids costly trades with high price impact.

(b) Sample of 15 transactions comparing absolute profits. While real JITs occasionally perform optimally, losses arise from engaging in losing trades and sometimes suboptimal allocation.

Figure 3 Comparison between actual and optimized JIT LP behavior. (a) focuses on aggregate performance, while (b) zooms in on individual transactions.

always assume that the JIT original transaction uses the trader's entire budget. Doing this, we ensure that our simulations do not use more money than the real execution, so our best results come exclusively from better resource utilization.

6.1 Results

We observe three main findings from our empirical evaluation:

1. JIT LPs in the pool we studied currently invest suboptimally and could significantly increase their profits. (§6.1.1)
2. JIT LP utility is primarily driven by price impact rather than fees. (§6.1.2)
3. If JIT LPs were to optimize their investment strategies, it would help JIT LPs and traders, at the expense of passive LPs. (§6.1.3)

6.1.1 Suboptimal Investment Strategies

Empirical data reveals that real-world JIT LPs underperform significantly relative to their potential gains, as illustrated in Figure 3. Many JIT agents engage in transactions where the adverse effects of price movements outweigh the fees earned, resulting in negative net utility (Figure 3(a)). In our sample of 6,884 JIT transactions, real JIT LPs could have earned up to 69% more than their current profit if they had adopted Algorithm 1. Our optimized allocation strategy not only helps JIT LPs identify when *not* to invest, but it can also increase returns on already-profitable transactions; for example, Figure 3(b) shows a random sample of 15 JIT transactions, comparing their realized gain in $ and the estimated gain under Algorithm 1. Our optimized strategy never loses money, and it sometimes increases gains relative to the current strategy of JIT traders. These observations highlight the importance of modeling price dynamics and underscore the need for strategy optimization beyond mere fee maximization.

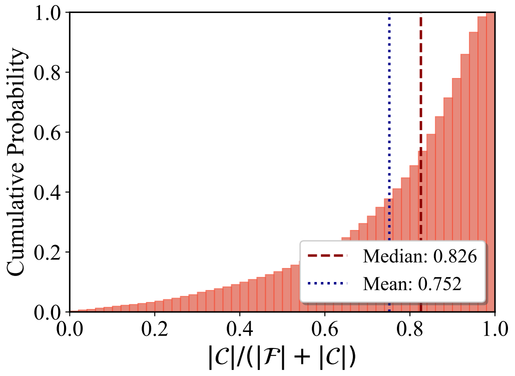


Figure 4 CDF of the proportion of JIT returns resulting from price impact as opposed to fees, i.e., $\frac{|\mathcal{C}|}{|\mathcal{C}|+|\mathcal{F}|}$. In most JIT transactions, price impact has a much greater impact on returns than fees, accounting for 75% of the total returns on average.

6.1.2 JIT Utility Is Driven Primarily by Price Impact

Our analysis shows that the primary driver of JIT profitability is not fee income but price impact (Figures 4). Optimized JIT strategies leverage price dislocations to extract value, often outperforming naive approaches that focus solely on fees. This illustrates the value of our study relative to prior works that focus on optimizing fee share [25].

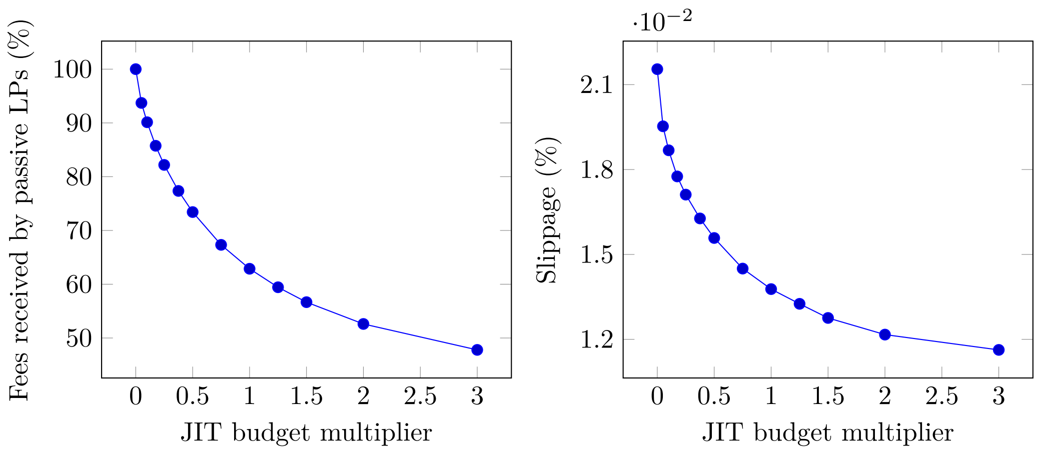
6.1.3 Market-Level Implications of Optimized JIT Behavior

Currently, the participation of JIT LPs in CLMMs is limited, and their share of fee revenue remains negligible when compared to passive LPs; in our dataset, JIT LPs claimed less than 2% of total fees paid in the system. However, simulations suggest that increased adoption of optimized JIT strategies could substantially reshape the CLMM ecosystem. In particular, JIT LPs affect both the distribution of fees among liquidity providers and the execution quality experienced by traders. Figures 5(a) and 5(b) illustrate two key market-level effects. As JIT LPs increase their capital allocation, they capture a growing share of the fee revenue, which reduces the earnings of passive LPs by up to 44% per trade, when the JIT budget is large (Figure 5(a)). At the same time, traders benefit from improved execution due to reduced price impact, as the injected liquidity narrows the effective spread and enables more efficient absorption of large orders (Figure 5(b)).

Averaging these values across transactions reveals two trends: the marginal benefit to passive LPs decreases with JIT participation, while traders benefit from reduced slippage. Together, these results indicate that although optimized JIT activity may diminish returns for passive LPs, it can improve capital efficiency and trading experience in CLMMs. This reinforces the view that well-designed JIT strategies can contribute positively to market structure.

7 Discussion and Implications

Our results highlight the strategic advantage of JIT in CLMMs. Specifically, a JIT LP that enters just before a trade – when the pool price is aligned with the market price – can benefit from the resulting swap-induced price movement. In contrast, passive LPs are exposed to losses when price deviations from the market are corrected via arbitrage.



(a) Fees received by passive LPs. (b) Trader surplus attributable to JIT liquidity.

Figure 5 Both plots compare the effects of higher JIT budget with a baseline where no JIT engage (x=0). (a) Effect of JIT budget on passive LP fees. Passive LP fee gains diminish as JIT liquidity increases. (b) Trader benefits via reduced slippage with higher JIT budgets. The Y-axis shows the slippage suffered by the trader under different budgets (Recall slippage from §1).

Importantly, JIT profitability arises only under specific conditions (Proposition 14), often requiring precise timing and positioning (§5.1). Given the competitive nature of bidding for inclusion (e.g., via Flashbot bundles), it is not surprising that real-world JIT activity remains limited.

JIT LPs complement the actions of arbitrageurs. While arbitrageurs restore price alignment after deviations, JIT providers preemptively mitigate such deviations by injecting liquidity in trades with negative price impact $\mathcal{C}$. In this sense, JIT acts as a frictional force – reducing the magnitude of price dislocations (Figure 5) that would otherwise create larger arbitrage opportunities. This increased liquidity is beneficial to traders, as it reduces slippage and improves execution quality. However, it also comes at the expense of passive LPs, who face reduced fee income.

7.1 Real dynamics of providers, arbitrageurs and traders

A common perception is that arbitrageurs profit primarily at the expense of passive LPs through the price impact. However, this is not always the case. Consider a model where the market price P_t evolves stochastically, and the pool price Q_t follows it with some lag due to noise. Suppose a trade moves the pool price from $Q_t = P_t$ to $Q_{t+1} \neq P_{t+1}$, and an arbitrageur subsequently restores alignment at $Q_{t+2} = P_{t+2}$. If $P_t = P_{t+1} = P_{t+2}$, the passive LP's price impact $\mathcal{C}_n$ is zero, but fees $\mathcal{F}_n > 0$ are still earned from both trades.

In this case, the arbitrageur's profit comes not from the LP, but from the noisy trader who executed the initial swap at a worse price – e.g., buying at $\sqrt{Q_t Q_{t+1}} > Q_t = P_t$. The arbitrageur captures this overpayment at $\sqrt{Q_{t+1} Q_{t+2}}$. JIT LPs benefit even more from such trades. By sandwiching the swap, they can collect both price impact gains and fees. Moreover, their presence dampens the price displacement, thereby reducing the arbitrageur's profit and the passive LP's fee share from the second trade.

To illustrate this, suppose a noisy trade pushes the price to $Q_{t+1} > Q_t$. A JIT LP that enters before the trade adjusts the effective spot price to Q^*_{t+1}, such that $Q_{t+1} > Q^*_{t+1} > Q_t$. The arbitrageur, now facing a smaller price deviation, captures less profit: $\sqrt{Q^*_{t+1} Q_{t+2}} < \sqrt{Q_{t+1} Q_{t+2}}$.

Assuming all arbitrage opportunities are rapidly corrected, the pool price trajectory is driven by market price movements plus correctable noise. This noise does not generate price impact directly; rather, it creates temporary dislocations that can be monetized by arbitrageurs and JIT LPs. In fact, this type of exploitable noise is the basis of Loss-versus-Rebalancing (LVR) [20], a cost borne by passive LPs even when price impact vanishes.

JIT LPs exploit this noise by positioning liquidity around anticipated dislocations. Their interventions reduce price volatility, preempt arbitrage, and extract part of the fees that would otherwise accrue to passive LPs or arbitrageurs. If arbitrage profits stem from noisy trades rather than passive LPs, JIT LPs can be seen as first movers that mitigate price deviations while capturing a portion of this surplus.

8 Related Work

Despite a growing body of work analyzing LP incentives in AMMs, including recent theoretical models of JIT LPs [11, 26], prior efforts either address market-level dynamics under informational asymmetry [11] or empirically characterize JIT activity [26]. To our knowledge, no existing work models the transaction-level, optimization-based behavior of JIT LPs in CLMMs such as Uniswap V3. We fill this gap by formally modeling the per-transaction utility of JIT LPs based on price impact (§2.3) and fee accrual (§2.2.1), analyzing how optimal strategies vary with trade direction and liquidity depth, and comparing predictions with observed on-chain behavior.

Complementing our work, Cartea et al. [12] develop a continuous-time model for strategic liquidity provision in CLMMs. They derive optimal dynamic strategies for LPs adjusting liquidity ranges over time, accounting for fee income, predictable loss, and concentration risk. Their results show how exchange rate drift and rebalancing costs shape behavior. Unlike our discrete, transaction-level model where JIT LPs react to individual swaps, their focus is on long-term LPs managing ongoing exposure. Still, their analysis highlights the need to model fee income and adverse selection in CLMMs.

Tang et al. [25] present a game-theoretic model of liquidity provision in CLMMs, focusing on interactions between passive LPs. Their static model captures how budget constraints and price tick granularity affect equilibrium behavior, identifying a unique Nash equilibrium with a water-filling strategy. By comparing theory to on-chain behavior, they show LPs in risky pools deviate more from optimality than those in stable pools. However, they do not model or consider the effects of JIT LPs.

In this work, we analyze price impact as a key metric. Price impact and LVR coincide for individual trades [3, 4]. While LVR matters for passive LPs over time, JIT LPs primarily care about price impact.

Recent studies explore how informational asymmetries and adversarial timing affect CLMM dynamics [11, 13, 15, 23]. Capponi et al. [11] model adverse selection and value extraction by informed traders. Qin et al. [23] and Daian et al. [13] examine MEV, showing how latency arbitrage and frontrunning harm LPs. While these works address systemic risks, we focus on microstructure-level decisions by JIT LPs under fee competition and price impact. Our formal framework captures their transaction-specific optimization, and shows how misjudging price impact can erode profits – even for strategically-timed liquidity.

9 Conclusion

Emerging tools such as CLMMs, Flashbots, and Uniswap V4 Hooks are reshaping DeFi by enabling JIT liquidity providers to improve capital efficiency, manage risk, and align strategies with real-time on-chain dynamics. Yet, empirical evidence shows that most JIT agents fail to exploit profitable opportunities and often incur losses due to simplistic strategies.

This work demonstrates that significant gains are achievable via optimization-based approaches that explicitly weigh fee accrual against price impact. Our transaction-level framework guides JIT agents toward utility-maximizing strategies that allocate capital more effectively and reduce adverse outcomes such as slippage or loss.

More broadly, the results support a shift toward reactive, transaction-driven market models. By analyzing individual trades, rather than long-term aggregates, we obtain sharper insights into the microstructure of AMMs and the role of JIT behavior.

Future research should explore extensions of AMM models that account for JIT strategies and protocols that safeguard fairness for passive LPs. Closing the loop between protocol design and agent strategy is key to building more adaptive and sustainable DeFi ecosystems.

References

1 Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. `https://uniswap.org/whitepaper-v3.pdf`, 2021. Accessed: 2025-08-07.

2 Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v4 core. `https://uniswap.org/whitepaper-v4.pdf`, 2024. Accessed: 2025-08-07.

3 Abe Alexander and Lars Fritz. Impermanent loss and loss-vs-rebalancing I: some statistical properties, 2024. `arXiv:2410.00854`.

4 Abe Alexander, Guillaume Lambert, and Lars Fritz. Impermanent loss and Loss-vs-Rebalancing II, 2025. `arXiv:2502.04097`.

5 Allium. Allium – Enterprise blockchain data platform. `https://www.allium.so/`. (Accessed on 10/04/2024).

6 Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, AFT '20, pages 80–91, New York, NY, USA, 2020. Association for Computing Machinery. `doi:10.1145/3419614.3423251`.

7 Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of Uniswap markets, 2021. `arXiv:1911.03380`.

8 Alif Aqsha, Philippe Bergault, and Leandro Sánchez-Betancourt. Equilibrium reward for liquidity providers in automated market makers, 2025. `arXiv:2503.22502`.

9 Jan Arvid Berg, Robin Fritsch, Lioba Heimbach, and Roger Wattenhofer. An Empirical Study of Market Inefficiencies in Uniswap and Sushiswap. In *International Conference on Financial Cryptography and Data Security*, pages 238–249. Springer, 2022. `doi:10.1007/978-3-031-32415-4_16`.

10 Andrea Canidio and Robin Fritsch. Arbitrageurs' profits, lvr, and sandwich attacks: batch trading as an amm design response, 2025. `arXiv:2307.02074`.

11 Agostino Capponi, Ruizhe Jia, and Brian Zhu. The Paradox Of Just-in-Time Liquidity in Decentralized Exchanges: More Providers Can Sometimes Mean Less Liquidity. *arXiv preprint arXiv:2311.18164*, 2023. `doi:10.48550/arXiv.2311.18164`.

12 Álvaro Cartea, Fayçal Drissi, and Marcello Monga. Decentralized finance and automated market making: Predictable loss and optimal liquidity provision. *SIAM Journal on Financial Mathematics*, 15(3):931–959, 2024. `doi:10.1137/23M1602103`.

13 Philip Daian, Steven Goldfeder, et al. Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges. In *IEEE S&P*, 2020.

14 Robin Fritsch. Concentrated liquidity in automated market makers. In *Proceedings of the 2021 ACM CCS Workshop on Decentralized Finance and Security*, DeFi '21, pages 15–20, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3464967.3488590.

15 Lioba Heimbach, Eric Schertenleib, and Roger Wattenhofer. Risks and returns of Uniswap V3 liquidity providers. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT)*, pages 89–101, 2022. doi:10.1145/3558535.3559772.

16 Alfred Lehar, Christine Parlour, and Marius Zoican. Fragmentation and optimal liquidity supply on decentralized exchanges, 2024. arXiv:2307.13772.

17 Zihao Li, Jianfeng Li, Zheyuan He, Xiapu Luo, Ting Wang, Xiaoze Ni, Wenwu Yang, Xi Chen, and Ting Chen. Demystifying defi mev activities in flashbots bundle. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, pages 165–179, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3576915.3616590.

18 Bruno Llacer Trotti, Weizhao Tang, Rachid El-Azouzi, Giulia Fanti, and Daniel Menasché. Strategic analysis of just-in-time liquidity provision in concentrated liquidity market makers (extended version). *arXiv e-prints*, 2025. URL: https://arxiv.org/abs/2509.16157.

19 Annika Masrani. Pancakeswap hits record $310b trading volume in 2024. *Nasdaq*, 2024. URL: https://www.nasdaq.com/articles/pancakeswap-hits-record-310b-trading-volume-2024.

20 Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing, 2024. arXiv:2208.06046.

21 Deborah Miori and Mihai Cucuringu. DeFi: Modeling and Forecasting Trading Volume on Uniswap v3 Liquidity Pools. *SSRN Electronic Journal*, May 2023. Available at SSRN: https://ssrn.com/abstract=4445351 or http://dx.doi.org/10.2139/ssrn.4445351. doi:10.2139/ssrn.4445351.

22 PancakeSwap. PancakeSwap: AMM on Binance Smart Chain. https://docs.pancakeswap.finance/, 2021. Accessed: 2025-05-25.

23 Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214. IEEE, 2022. doi:10.1109/SP46214.2022.9833734.

24 SushiSwap. SushiSwap System. https://docs.sushi.com/, 2020. Accessed: 2025-05-25.

25 Weizhao Tang, Rachid El-Azouzi, Cheng Han Lee, Ethan Chan, and Giulia Fanti. Game Theoretic Liquidity Provisioning in Concentrated Liquidity Market Makers. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 9(1):1–45, 2025. doi:10.1145/3711700.

26 Xin Wan and Austin Adams. Just-in-Time Liquidity on the Uniswap Protocol. *SSRN 4382303 Electron. J.*, 2023.

27 Shuzheng Wang, Yue Huang, Wenqin Zhang, Yuming Huang, Xuechao Wang, and Jing Tang. Private order flows and builder bidding dynamics: The road to monopoly in ethereum's block building market, 2024. doi:10.48550/arXiv.2410.12352.

28 Xihan Xiong, Zhipeng Wang, William Knottenbelt, and Michael Huth. Demystifying just-in-time (JIT) liquidity attacks on uniswap v3. Cryptology ePrint Archive, Paper 2023/973, 2023. URL: https://eprint.iacr.org/2023/973.

29 Deniz Yüksel. A Retrospective Analysis of Public and Private Order Flow on the Ethereum Blockchain. Master's thesis in informatics, Technische Universität München, 2024.

30 M. Yuksel et al. Automated market makers and decentralized exchanges: a defi primer. *Financial Innovation*, 7(1):1–27, 2021. URL: https://jfin-swufe.springeropen.com/articles/10.1186/s40854-021-00314-5.