

Just-in-time Liquidity on the Uniswap Protocol *

Xin Wan¹ and Austin Adams²

1. Uniswap Labs, xin@uniswap.org
2. Uniswap Labs, austin@uniswap.org

Abstract. Just-in-time (“JIT”) liquidity is a liquidity provision strategy found in blockchain-based automated market makers like the Uniswap Protocol, where a liquidity provider (“LP”) mints and burns a concentrated position immediately before and after a swap. There have been wide-spread concerns in the community about the potential adverse impact of JIT liquidity on the economic security of the system. In this paper, we study the market micro-structures that allow JIT transactions to take place. We show that JIT liquidity occurs rarely in practice, and when it does, JIT liquidity improves execution quality of the trader to whom liquidity is provided. Finally, we derive and validate a model of the price improvement offered by JIT liquidity, which we show to have an upper bound at twice the fee rate of the pool that JIT occurs in.

Keywords: Automated market maker · Liquidity provision · Uniswap

1 Introduction

There has been some confusion around the concept of JIT liquidity on Uniswap. Paradigm’s Dan Robinson, a co-author of the Uniswap v3 whitepaper, ran a poll on Twitter surveying people’s estimates of the percentage of v3 liquidity supplied by JIT.¹ The most common answer was “over 40%”, with just 16.5% of respondents correctly answering “less than 1%”. In this paper, we investigate historical JIT on Uniswap, and find the aforementioned belief to be false. We document the empirical distribution of JIT, and tie it back to economic intuitions. Finally, we dive deeper into the anatomy of JIT transactions and the micro economic considerations of a JIT LP, deriving some interesting theoretical results. The numbers stand in stark contrast to public perception.

1.1 Liquidity Provisioning

Liquidity provisioning takes many forms. In a central limit order book, liquidity providers submit limit orders that are filled by liquidity takers. In an Over The Counter (OTC) market, liquidity is often provided directly and exclusively by

* We thank Dan Robinson for his tweet that inspired this analysis. We are grateful for comments from Matteo Leibowitz and Aseem Sood.

¹ <https://twitter.com/danrobinson/status/1549514746411552775>

dealers to inquiring clients as quotes, or generated through auctions (bid wanted in competition) sent by institutions to a select set of dealers.

Automated Market Makers (AMMs) like the Uniswap Protocol (“Uniswap”) have created a unique market structure, and with it, a new form of liquidity provisioning. Liquidity providers (“LPs”) deposit tokens into liquidity pools controlled by smart contracts, which then directly quote and trade tokens in the pools with traders. Traders send in one token and get back another token.

In Uniswap v3, LPs can deposit and withdraw liquidity at any time, and the fee they earn will be proportional to the amount of liquidity they supplied against traders’ order flows during the period in which their liquidity was deposited.²

1.2 What is JIT?

Just-in-time (“JIT”) liquidity refers to a specific type of LP strategy whereby an LP:

1. Observes a (usually sizable) pending swap transaction in the public mempool
2. Adds liquidity to the Uniswap pool that the swap will go through immediately before the swap happens, usually concentrated in the one tick that the swap will be in-range for
3. Lets the swap execute, and receives a fee in return for providing liquidity to the swap
4. Removes the liquidity and fees accrued immediately after the swap

Simultaneously, the LP will complete a hedging transaction in a different liquidity venue to offset their inventory risk, earning the difference between the transaction fees associated with the hedge and the LP fee from the Uniswap pool. We will talk more about this hedging process later in this article.

Figure 1 below illustrates a JIT transaction in real life.

Note that in the example, all three transactions happened in the same block (Block 13733864), and they are all sequential. The JIT liquidity provider bundled the three transactions and sent them through Flashbots.

Since the size of the swap and the point-in-time liquidity distribution was known to the JIT LP before their transactions, they could have (and likely had) computed the exact fee earning and position change from the trade.

2 Historical JIT volume and Distribution on Uniswap v3

In this section, we quantify the amount of JIT liquidity that has been supplied on Uniswap v3.

² <https://uniswap.org/blog/uniswap-v3>

JIT Example

Action	Details	Etherscan
Add liquidity	0x769f... adds >11M USDC to USDC/ETH 5bps pool	Note 1
Swap	0x1261... swaps ~1500ETH for ~7M USDC in this pool. Note that the LP's position changed from 11M USDC to 5.5M USDC + 1200 ETH	Note 2
Remove liquidity	0x769f... removes ~5.5M USDC and ~1200 ETH	Note 3

Fig. 1. Note 1³, Note 2⁴, Note 3⁵

2.1 Methodology

We only consider a set of transactions to be JIT liquidity provision when the following conditions are met:

1. An LP position was minted and burnt in the same block
2. The mint transaction and burn transaction are exactly 2 positions apart in the block
3. The transaction between mint and burn is a swap transaction in the same pool

Once we confirm a pair of mint — burn transactions to be JIT, we compute the difference in the amount of assets deposited versus those withdrawn. We take one side of the transaction and mark it to US dollars as of the block time.⁶

2.2 Results

Total From the deployment of Uniswap v3 on May 5, 2021 to July 18, 2022, we identified a total of 8,287 JIT liquidity provision attempts. Fifty-seven of them failed to provide any liquidity (some incorrectly sandwiched non-swap transactions; some added liquidity at the wrong tick). Over 95% of JIT liquidity was supplied by one single account, and less than 20 addresses have ever attempted to supply JIT liquidity.

In aggregate, successful JIT liquidity provided just over \$2bn dollars of liquidity to traders. For reference, Uniswap v3 supported over \$600bn of trading volume in the same period, meaning that JIT liquidity filled 0.3% of all liquidity demand.

⁶ See Appendix for logic for converting transaction volume to USD.

This directly refutes the public confusion around how much of v3's liquidity is supplied by JIT. The most common answer was "over 40%", with just 16.5% of respondents correctly answering "less than 1%".

Distribution through time The historical distribution of successful JIT activities is shown in Figure 2 below.

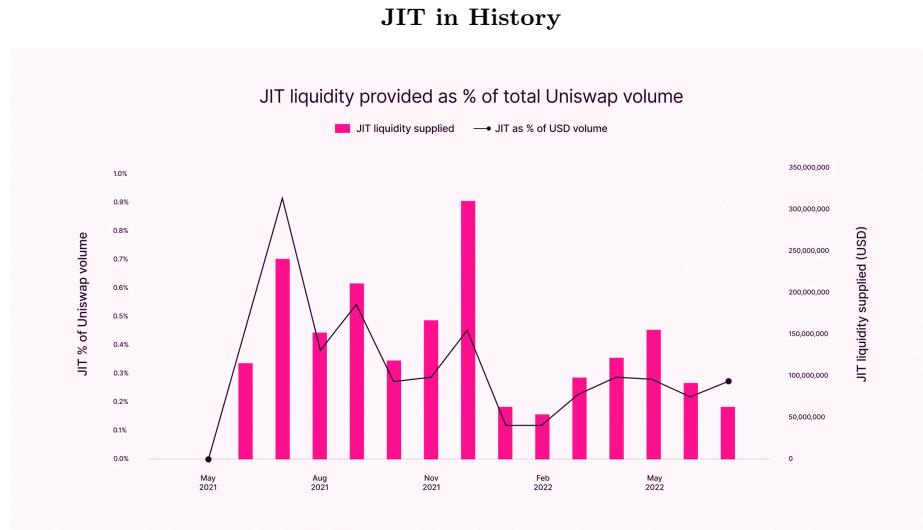


Fig. 2. Source: Uniswap Labs data, Dune Analytics

Absolute USD volume of JIT liquidity has not grown significantly since June 2021. In most months, total trading volume against JIT liquidity was under \$200mm, with recent months mostly under \$100mm.

As a percentage of total USD trading volume on Uniswap, JIT has not been significant or growing, either. Over the past 6 months, Uniswap supported an average of \$50bn of trading volume per month on Ethereum mainnet. With the exception of July and September 2021, the amount of trading volume executed against JIT liquidity has never risen above 0.5% on a monthly basis.

Distribution across pools JIT liquidity has historically been concentrated across the top pools on Uniswap by trading volume. See Figure 3 for details.

Comparing blue bars (percentage of total historical JIT liquidity supplied in each pool) to pink bars (percentage of total trading volume / liquidity demand in each pool), we can see clearly that JIT liquidity provisioning is much more concentrated in the top pools. The most active pool (USDC-WETH 5bps pool) alone accounts for over half of all JIT liquidity ever supplied. The top ten pools

Distribution of JIT across Liquidity Pools

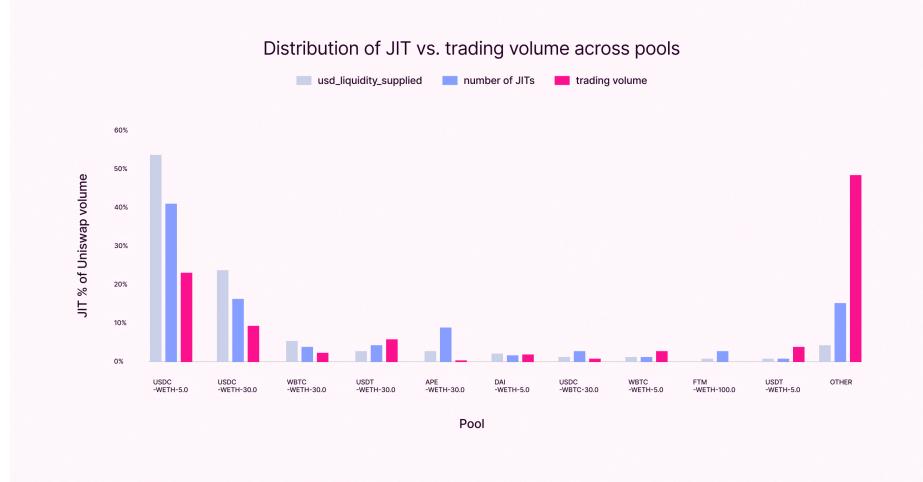


Fig. 3. Source: Uniswap Labs data, Dune Analytics

account for 95%. On the demand side, the top 10 pools account for just 55% of total trading volume.

Intuitively, this pattern makes sense: JIT transactions are only profitable when the swap in the middle is large enough, and large trades are much more concentrated in the top pools. Another relevant factor is that pairs in large pools have greater liquidity on centralized exchange venues, making the hedging transaction much easier and cheaper to complete.

Distribution across fee tiers JIT transactions, both by count and by volume, concentrate heavily in the middle two fee tiers. Compared to the distributions of trade count and volume in the Uniswap Protocol, 5bp pools see relatively more count of JIT, while 30bp pools see more volume.⁷

A few different factors are at play here.

First, given fixed costs, JIT only makes sense when the transaction size reaches certain thresholds. Since larger-sized transactions are more likely to be routed through lower-fee pools, this factor shifts concentration of JIT towards those pools.

On the other hand, there is a hedging cost to JIT transactions. We will discuss this in more detail in the next section, but it is reasonable to assume at least 1-2bps of hedging cost to a JIT transaction. This makes JIT in 1bp pools

⁷ In the table, the first and second data columns show the distribution of transaction count and volume of JIT across different fee tiers. The third and last data columns show the distribution of trade count and volume of swaps on the Uniswap Protocol, as a reference for the distribution of JIT.

Distribution of JIT across Fee Tiers

Fee (Bps)	JIT Txns	JIT Volume	Uniswap Txns	Uniswap Volume
1	0%	0%	2%	6%
5	44%	58%	30%	59%
30	55%	39%	42%	30%
100	11%	3%	26%	5%

Fig. 4. Source: Uniswap Labs data, Dune Analytics

money-losing in most cases, and in general shifts concentration of JIT towards higher fee-tiers.

Distribution of JIT liquidity provided per transaction Each time an LP provides JIT liquidity, they pay miners an amount roughly equal to two gas costs, plus some share of Maximum Extractable Value (“MEV”) if there is competition for either the same transaction or space at the front of that particular block. Therefore, JIT transactions are usually much larger than the typical swap sizes on Uniswap, so that the profit generated for the LP can justify the gas costs for minting and burning the JIT LP position.

Distribution of USD Value of JIT Liquidity

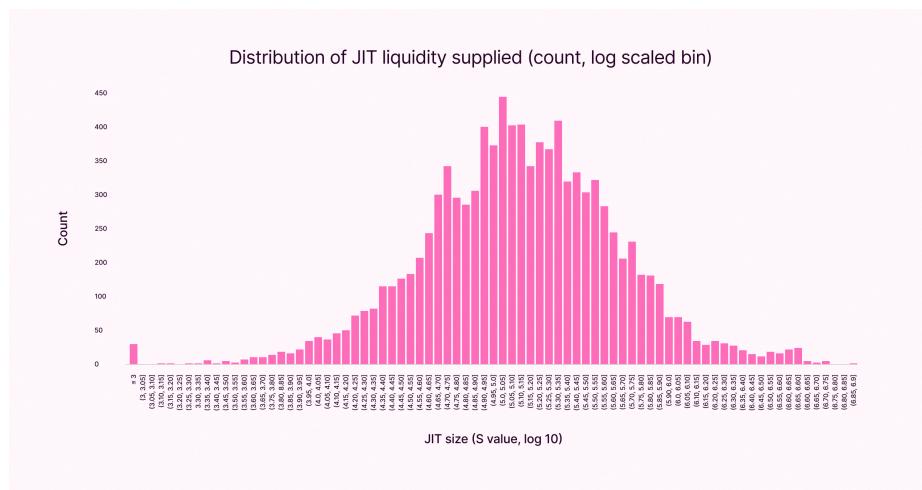


Fig. 5. Source: Uniswap Labs data

Over half of JIT transactions supplied more than \$100,000 of liquidity individually. Only 245 out of more than 8,000 transactions supplied less than \$10,000 of liquidity.

On the other hand, if we plot the same data with log-scaled x-axis, it is notable that the distribution of JIT sizes past the one million dollar mark is generally smooth.

3 Economic Considerations of a JIT

So far we have given some summary statistics of the empirical distribution of JIT liquidity. However, can we model the decision making process of a JIT LP for each JIT transaction?

3.1 Model

Two necessary conditions need to be met for an LP to supply JIT liquidity⁸:

1. The expected profit is positive.
2. The LP can outbid other MEV opportunities in Flashbots auctions.

The two constraints combined result in a limited set of scenarios where it is optimal to provide JIT liquidity.

Expected Profit A JIT LPs' profit comes from the difference between LP fee earned and the cost of hedging. In the simplest form, the JIT LP supplies liquidity on Uniswap, and simultaneously executes an offsetting trade on a different venue (e.g. Coinbase) with a lower fee rate to hedge.

The revenue of the trade is:⁹

$$\begin{aligned} \text{Revenue} = & \text{Liquidity supplied} \times \text{Uniswap pool fee rate} \\ & + \text{Price impact on Uniswap pool} \end{aligned} \tag{1}$$

The cost of the trade is:¹⁰

⁸ Inventory rebalancing needs of LPs could technically be another reason for JIT. However, since they are unobservable and introduce much more complexity, we leave them out of this analysis.

⁹ While JIT usually pushes price impact close to zero, technically there is still a small price impact from the swap remaining; so assuming "true market price" does not move, the JIT LP still acquired the position a tiny bit cheaper than the global market price.

¹⁰ Note that the last three items are all included in the miner payment, but economically it is helpful to break them out and treat them separately, since they are driven by dynamics in different markets.

$$\begin{aligned}
\text{Cost} = & \text{Liquidity supplied} \times \text{Other liquidity venue fee rate} \\
& + \text{Price impact / slippage on other liquidity venues} \\
& + \text{Gas fee of adding liquidity} \\
& + \text{Gas fee of removing liquidity} \\
& + \text{Profit share with miner ("miner share")}
\end{aligned} \tag{2}$$

It is reasonable to assume the trade starts with no arbitrage condition satisfied, and therefore pricing on Uniswap is the same as on the other liquidity venue. Furthermore, since a JIT LP wants to maximize fee earnings, they will almost always supply a liquidity amount large enough for the entire swap to trade against, and concentrate all the liquidity in one tick space. Therefore, we can assume the price impact on the Uniswap pool is 0 in the tick space where JIT liquidity is minted. The second item (Price impact on Uniswap pool) from the Revenue formula is then either 0 if the JIT LP decides to mint in the current tick, or a small amount d if they decide to mint in the next tick. For simplicity, we assume $d = 0$.¹¹

While fee rates on other exchanges vary, it is reasonable to assume that they are non-negative. Therefore, the first cost item

$$\text{Liquidity supplied} \times \text{Other liquidity venue fee rate}$$

has a lower bound of 0. It is also reasonable to assume that the JIT LP will need to take liquidity while hedging, causing a non-negative price impact / slippage. Hence, the second cost item

$$\text{Price impact/slippage on other liquidity venues}$$

can be reasonably assumed to have a lower bound of 0 as well.

Plugging in the aforementioned lower bounds for cost items, and expressing the JIT profit as the difference between revenue and cost, we obtain an upper bound of the expected profit from a JIT transaction:

$$\text{Profit} \leq \text{Liquidity supplied} \times \text{Uniswap fee rate} - \text{Total gas} - \text{Miner share}$$

Since JIT LPs require expected profit to be positive, the inequality above implies the maximum miner share the JIT LP will be willing to pay is:

$$\text{Miner share} \leq \text{Liquidity supplied} \times \text{Uniswap fee rate} - \text{Total gas}$$

This constraint will play a large role in determining how often JIT transactions can occur, as we will show below.

¹¹ Empirically, $d = 0$ in $\frac{2}{3}$ of the cases. See Appendix for in-depth discussions around what happens when $d \neq 0$.

Opportunity Cost / Other MEV opportunities Having a positive expected profit is not a sufficient condition for a JIT transaction to happen. Since JIT liquidity is almost always supplied through a Flashbots bundle, the LP needs to outbid other searchers who also want to use the swap in their bundle. With perfect information and rational players, this should only happen when a JIT transaction is more profitable than all other possible MEV bundles using the same swap (such as sandwich attacks, back-running arbitrage, etc). We will see that this is largely true, but not always.

Consider a simple alternative to the JIT strategy: an MEV searcher sees a large swap in the mempool and instead of providing JIT liquidity, they submit a bundle that starts with the swap going through the pool and creating a large price impact and ends with their own swap in the opposite direction, essentially taking on the opposite side of the swap at a cheaper price.

Similar to the calculation above, we can compute the MEV searcher's expected profit:

$$\text{Profit} = \text{Size of swap} \times (\text{price impact of swap} - \text{Uniswap fee rate}) - \text{Total gas} - \text{Miner share}$$

To ensure positive expected profit, we have:

$$\text{Miner share} \leq \text{Size of swap} \times (\text{price impact of swap} - \text{Uniswap fee rate}) - \text{Total gas}$$

For a JIT to happen, we need the expected profit from JIT to be greater than or equal to the expected profit from a simple backrunning swap:

$$\begin{aligned} & \text{JIT liquidity supplied} \times \text{Uniswap fee rate} - \text{JIT gas fee} \\ & \geq \\ & \text{Size of swap} \times (\text{price impact of swap} - \text{Uniswap fee rate}) - \text{swap gas fee} \end{aligned}$$

Since liquidity supplied in a JIT transaction is at most 100% of the swap, and Uniswap fee rate is positive, we have:

$$\begin{aligned} & \text{JIT liquidity supplied} \times \text{Uniswap fee rate} - \text{JIT gas fee} \\ & \geq \\ & \text{JIT liquidity supplied} \times (\text{price impact of swap} - \text{Uniswap fee rate}) - \text{swap gas fee} \end{aligned}$$

Since JIT involves a mint and a burn while backrunning only has one swap, the gas fee for JIT should always be higher, everything else equal. Therefore:

$$\begin{aligned} & \text{JIT liquidity supplied} \times \text{Uniswap fee rate} \\ & \geq \\ & \text{JIT liquidity supplied} \times (\text{price impact of swap} - \text{Uniswap fee rate}) \end{aligned}$$

Finally we can cancel out the JIT liquidity supplied term, and be left with:

$$2 \times \text{Uniswap fee rate} \geq \text{price impact of swap}$$

Since the price impact of the swap with JIT is close to 0, the right-hand side is basically the price improvement that the trader gets from JIT liquidity. With some simplifying assumptions, we have derived an upper bound of price improvement: two times the fee rate in the Uniswap pool that the swap is going through.

It is worth mentioning that the result derived above is only an upper bound - it is not guaranteed to be, and highly likely not, the actual tight upper bound. For example, instead of simply backrunning the large swap, an MEV searcher could potentially generate more profit from a sandwich attack. However, the bound we derived should still hold true.

3.2 Empirical Tests

We can test the result from the last section with empirical data.¹²

For each swap that traded against JIT liquidity, we simulate the same swap without JIT. We then compute the price improvement this swap has received compared to the counterfactual. If the price improvement is greater than two times the pool fee, we call the JIT “suboptimal”, since the JIT LP could have profited more by simply submitting a backrunning trade. The table in Figure 6 below summarizes our findings.

Empirical Optimality in JIT Execution			
Fee	JIT Optimal	Count	Percentage
5 bps	True	3,368	94.9%
	False	181	5.1%
30 bps	True	3,223	99.9%
	False	3	0.1%
100 bps	True	710	99.7%
	False	2	0.3%

Fig. 6. Source: Uniswap Labs data

The vast majority of JIT trades satisfy our derived bound. However, in some cases, we do see violations.

¹² Out of all JIT transactions we identified, over 90% of them happened in the top 14 pools, with each pool containing more than 100 JIT transactions. Our empirical tests are based on this subset.

These violations are not necessarily indicative of either flaws in our derivation or JIT LP's implementation. Rather, it could come from assumptions we made when calculating the counterfactual output of the swap.

For example, one assumption we made was that the swapper would always swap in the exact amount of token. In reality, the amount of input / output could be dynamic based on the state of the pool, which changes as JIT liquidity is injected and removed.

In any case, swappers benefited from the participation of JIT LPs. The distribution of price improvements from JIT liquidity can be found in Figure 7 below. Consistent with our results, most of the swaps fell to the left of 0.3, which is our upper bound for price improvements in a 100-bp pool.¹³

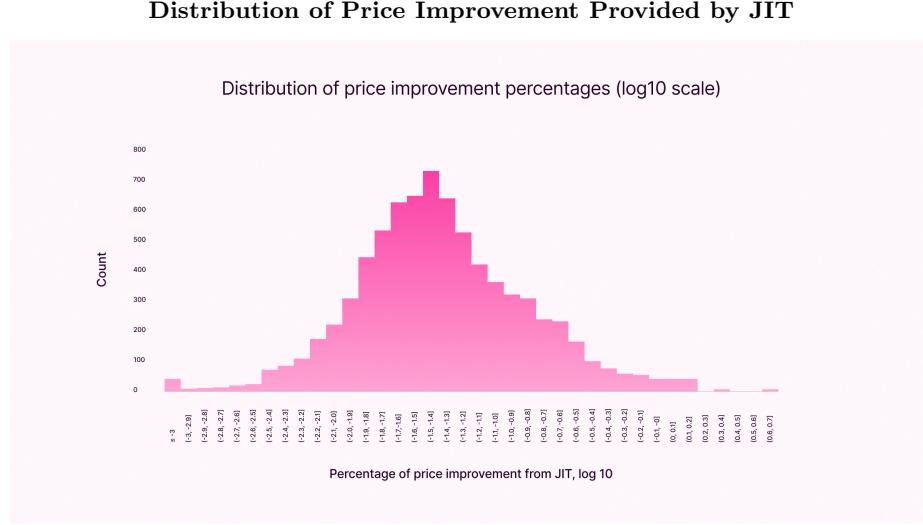


Fig. 7. Source: Uniswap Labs data

4 Conclusion

In this analysis, we gave a comprehensive overview of JIT liquidity, from real-world examples to economic considerations. We showed that JIT liquidity has historically played a minimal role within the Uniswap Protocol: it is typically heavily concentrated in top pools and mostly provided by a single LP address.

We also showed that JIT liquidity provided traders with improved execution. We explicitly modeled the micro-economic consideration of a JIT liquidity

¹³ Our upper bound is 2 times fee rate, which is 2 percent for a 100bp pool. Since the histogram has a log-scaled x-axis, and $\log(2) = 0.3$ with a base of 10, our upper bound implies swaps should fall to the left of 0.3 on the x axis.

provider, and showed that an optimally implemented JIT should provide price improvement no more than twice the fee rate of the pool that JIT liquidity was provided in.

A few areas remain unexplored in this analysis.

First, the derived upper bound on price improvement is likely not tight, because we only considered the most naive form of competition in the searcher market. A more sophisticated searcher could generate more MEV with more complex bundles such as sandwich attacks, leaving the JIT LP less room to work with.

Second, it is worth exploring new mechanism designs that allow LPs to more easily compete with price improvements for traders. Currently, whether an LP's JIT transaction gets included in a block has nothing to do with the degree of price improvement they quote to traders. In an ideal scenario, the LP that provides the most price improvements (quotes the best price) should outcompete other LPs as well as all sandwich attackers. Such a design would not only incentivize more LP participation but also bring higher trader welfare, leading to an overall improvement in market dynamics.

5 Appendix

5.1 Logic of marking trades to USD

Most trades have either ETH or USD on one side. We convert the trade volume to USD in the following order:

1. If ETH is on one side, use ETH price at that minute.
2. If USD stablecoin is on one side, use that amount (assuming stablecoin is 1:1 to USD).
3. Only 3 transactions remained; we manually marked them.

5.2 Consideration of a JIT LP when choosing between current or next tick space

Consider the case when the JIT LP observes an incoming swap. They have two options: 1) provide JIT liquidity in current tick space; 2) provide JIT liquidity in next tick space.

The tradeoffs are the following: by choosing option 2) over option 1), the LP is forgoing part of the fee (on the volume required to cross the tick) in exchange for a lower cost basis for the position they are acquiring. The LP would only choose to mint one-sided liquidity in the next tick if the gain from lowered cost basis is larger than the forgone fee income.

Denote:

1. The size of the incoming swap to be S
2. The distance between starting marginal price in the pool and the next tick to be d

3. The fee rate of the pool to be r
4. The volume required to cross the tick (or the amount of LP capital deposited between starting marginal price and next tick) to be L

The forgone fee when the LP chooses to mint one-sided in the next tick space is:

$$L \times r$$

The gain from acquiring the remaining position at a lower cost basis is:

$$(S - L) \times d$$

The LP would choose to do next tick if the gain is larger than the forgone fee:

$$(S - L) \times d \geq L \times r$$

Rearranging, we have:

$$d \geq \frac{L \times r}{S - L}$$

Note that this result is consistent with our intuition:

1. The deeper the liquidity in the current tick (L), the less likely the LP would choose to go for the next tick space, since they would lose a larger percentage of the fee income;
2. The larger the swap size (S), the more likely they would go for the next tick space, since the gains from lowered cost basis would be multiplied by a larger amount;
3. Given the same amount of volume required to cross the next tick, the larger the price impact, the higher the gain from price impact for the LP, and the more likely they will choose the next tick.