

Liquidity Provisioning in Uniswap V3

Sheldon Benard sheldon.benard@berkeley.edu	Pradeepta Das pradeepta@berkeley.edu
Srajan Garg srajangarg@berkeley.edu	Sarthak Sagar sarthak_sagar@berkeley.edu
Philip Spalding pspalding@berkeley.edu	

Team 04
Advisor: Dr. Christine Parlour

March, 2022

Abstract

Uniswap v3 allows liquidity providers to concentrate their capital within a range, effectively leveraging their investment and increasing the capital efficiency of the Uniswap markets. This feature allows liquidity providers to execute a variety of provisioning strategies, but also introduces competition between liquidity providers. We explore the dynamics of liquidity provider returns in Uniswap v3 markets and evaluate empirically whether the largest markets are in equilibrium. We observe that, currently, these pools have too much liquidity compared to the level of trading, such that the fee revenue generated does not sufficiently compensate liquidity providers for the risk of adverse selection that they are exposed to, making it difficult for liquidity providers to earn positive risk-adjusted returns.

Acknowledgements

We'd like to express our deepest thanks to our advisor and mentor Dr. Christine Parlour. Her expertise and many insights significantly guided us in our exploration of decentralized exchanges. We'd also like to thank Dr. Eric Reiner, for his feedback and guidance.

Contents

1	Introduction	1
1.1	Background	2
1.2	Related Work	5
2	Uniswap	6
2.1	Uniswap v2	7
2.2	Uniswap v3	13
2.2.1	Mechanics of Uniswap v3	14
2.2.2	Valuing a Uniswap v3 Position	20
2.2.3	Capital Efficiency	23
3	Arbitrage in Uniswap v3	25
4	Liquidity Provisioning	30
4.1	Liquidity Curves	31
4.2	Liquidity Provider Returns	34
5	Empirical Findings	45
6	Conclusion	50
	Appendix A Splitting Trades is Equivalent to a Single Trade	54
	Appendix B Price Bound Simulation Results	55
	Appendix C Example Liquidity Curves	55
	Appendix D Expected Daily Returns to Liquidity Providers	58

Glossary

AMM Automated Market Maker. A computer program that automates the process of providing liquidity and facilitates trading between currencies.

Blockchain A distributed ledger that is shared and verified by the nodes of a computer network.

CEX Centralized Exchange. A traditional exchange, particularly one that utilizes a central limit order book.

CFMM Constant Function Market Maker. Like CLOBs in CEXs, DEXs use CFMMs to facilitate trades. CFMMs execute trades based on a function that depends on the current reserves in the pool.

CLOB Central Limit Order Book. Used in traditional exchanges to aggregate quotes and to match and execute trades.

Concentrated liquidity A new feature of Uniswap v3 that allows LPs to provide liquidity in a price range of their choice aimed at improving capital efficiency.

CPMM Constant Product Market Maker. Type of CFMM that requires that any trade must change the reserves, denoted x, y , such that $xy = k$. In other words, the product of the reserves must remain unchanged after each trade.

dApp Decentralized Application. Applications that leverage a blockchain’s decentralized infrastructure. Because these applications use the Ethereum blockchain (or any other Turing-complete blockchain), they are: robust, as they don’t require centralized servers; transparent, since the code and data is stored on the public blockchain; and censorship resistant.

DeFi Decentralized Finance. An emerging financial ecosystem based on secure distributed ledgers similar to those used by cryptocurrencies.

DEX Decentralized Exchange. A dApp/smart contract that allows for secure, direct peer-to-peer cryptocurrency transactions without the need for an intermediary.

ETH Ether cryptocurrency; the token for the Ethereum blockchain.

Flash swap Allows users to borrow assets from a Uniswap pool and execute arbitrary logic as long as the assets are either returned along with a small fee or are paid for like a normal swap by the end of the transaction (Adams et al., 2020). Whereas in traditional markets arbitrageurs must have sufficient capital to take advantage of arbitrage opportunities, flash swaps effectively allow anyone to perform capital free arbitrage.

Gas fee Every transaction on the Ethereum and Polygon blockchains incurs a gas fee that is paid to miners to include the transaction in a block. Gas fees on the Ethereum blockchain are quoted in units of gwei, where 1 gwei is 10^9 ETH.

LP Liquidity Provider. Refers to individuals or organizations providing liquidity to a Uniswap pool, thereby functioning as a kind of market maker.

Miner Miners validate transactions on a blockchain, which can require significant computational power. Miners are compensated for such work with the blockchain’s cryptocurrency.

Pool A Uniswap smart contract that enables users to trade one token (e.g. WETH) for another (e.g. USDC). Because the reserves provided by LPs are pooled together in a smart contract, we commonly refer to these contracts as pools.

Price innovation A price movement in the underlying asset; in our case, a cryptocurrency.

Smart contract Programs that reside and run on a blockchain. The Ethereum blockchain originally introduced smart contracts in 2013. Smart contracts are “not controlled by a user, instead they are deployed to the network and run as programmed. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts can define rules, like a regular contract, and automatically enforce them via the code. Smart contracts cannot be deleted by default, and interactions with them are irreversible” (Eth, 2022). These programmatic contracts have applications in areas which require trusted transactions between parties without the need for a central authority.

Token Tokens represent fungible and tradable assets or utilities that reside on a blockchain. In context of this report, a token always refers to an ERC20 token. ERC20 is a standard used for creating and issuing smart contracts on the Ethereum blockchain that was implemented in 2015. ERC stands for “Ethereum request for comment.”

TVL Total Value Locked. The total value of assets deposited in a DeFi protocol. It is considered one of the key metrics for gauging interest in a protocol in the decentralized community.

USDC a cryptocurrency stablecoin, pegged to the United States dollar (USD).

USDT a cryptocurrency stablecoin, pegged to the United States dollar (USD).

WBTC Wrapped Bitcoin. WBTC follows the ERC-20 standard, while BTC does not. WBTC was created because BTC cannot be used for various DeFi applications. Thus, wrapping the BTC token enables it to be used across a wide spectrum of dApps. WBTC’s value is pegged to Bitcoin (BTC).

WETH Wrapped Ether. WETH follows the ERC-20 standard, while ETH does not. WETH was created because ETH cannot be used for various DeFi applications. Thus, wrapping the ETH token enables it to be used across a wide spectrum of dApps. WETH’s value is pegged to ETH.

Select Notation

x	Actual reserves, in the context of Uniswap v2, or “virtual” reserves, in the context of Uniswap v3, for <code>token0</code> .
y	Actual reserves, in the context of Uniswap v2, or “virtual” reserves, in the context of Uniswap v3, for <code>token1</code> .
p	The price of the Uniswap pool defined in terms of its reserves: $p = \frac{y}{x}$.
m	The market price of the asset, which may deviate from the pool price, p . This can generally be thought of as the mid-quote from a traditional centralized exchange.
γ	The transaction fee for the Uniswap pool.
$L, L(p)$	The liquidity available in a Uniswap v3 pool at a give price, p . We drop the dependence on p when it is clear that L is constant within the range being discussed, which is consistent with the notation in the Uniswap v3 white paper.
q	The square root of the price, $q = \sqrt{p}$, used to formulate our analytical approximations for swapping in Uniswap v3.

1. Introduction

More than a decade ago, Nakamoto (2008) proposed the Bitcoin electronic cash system: a peer-to-peer network that allows for direct online payments without the need of a financial intermediary. The underlying blockchain technology has fueled the growth of a new and innovative financial ecosystem that has been coined Decentralized Finance (DeFi). Leveraging the blockchain infrastructure, Decentralized Applications (dApps) are actively being developed that are: robust, as they don't require centralized servers; transparent, since the code and data is stored on the public blockchain; and censorship resistant, allowing any party to transaction on the network as long as they follow the protocol's rules.

Decentralized Exchanges (DEXs) are one such dApp that allow for secure, direct peer-to-peer cryptocurrency transactions without the need for an intermediary. Motivated by the novelty of these exchanges and their growing popularity in the cryptocurrency space, our report examines Uniswap v3, launched in May 2021, which is one of the largest and fastest-growing DEXs, as can be seen in Figure 1. Uniswap v3 employs an Automated Market Maker (AMM) and features several enhancements to previous versions that improve capital efficiency and introduce competition into the Uniswap marketplace. In this paper, we aim to explain the return dynamics of Uniswap market makers, called Liquidity Providers (LPs), explore whether the largest Uniswap v3 markets are in equilibrium, and understand empirically whether providing liquidity to such markets is profitable for LPs.

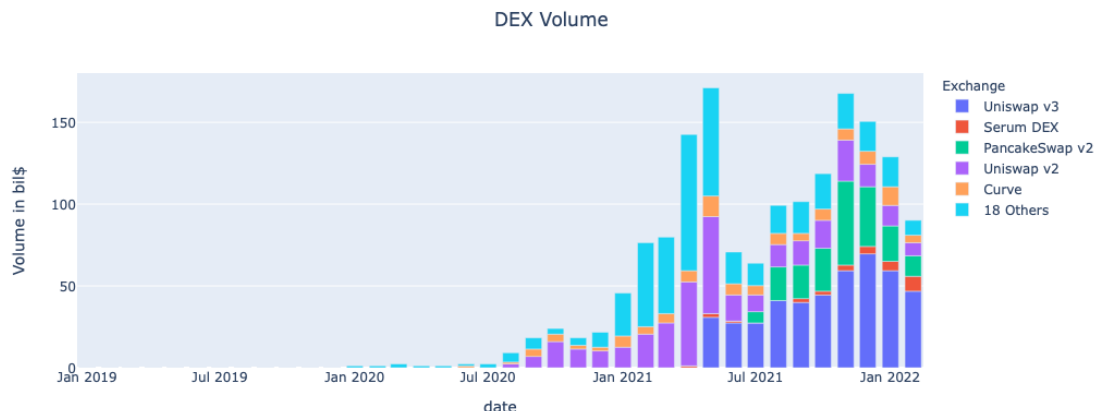


Figure 1: DEX trading volume over time. (Source: theblockcrypto.com)

Our report is organized as follows: Section 1 introduces DEXs and reviews related work on Uniswap and other AMM protocols. Section 2 details Uniswap, outlining its implementation, key concepts, and mathematical underpinnings. Section 3 discusses arbitrage in Uniswap v3 and the importance of arbitrageurs in ensuring that the Uniswap market prices approximately track the true market price. Section 4 explores liquidity provisioning in a Uniswap v3 market, and we present a series of simulations to illustrate the return dynamics for such an investment strategy. Section 5 empirically explores liquidity provisioning for some of the largest Uniswap v3 markets and evaluates whether such markets are in equilibrium. Section 6 concludes with a summary of our findings and a brief discussion on future work.

1.1 Background

In traditional markets, there are two main types of participants: liquidity takers and liquidity providers, commonly referred to as market makers. In markets organized around a Central Limit Order Book (CLOB), professional market makers provide two-way prices, bid and ask, for an asset, supplying immediate liquidity. Liquidity takers execute against these prices and are provided with the ability to trade on demand. The cost of this liquidity is the bid-ask spread. Liquidity takers must cross the spread in order to execute an order, generating revenue for the market makers. Beau (2014) succinctly describes the importance of market makers in traditional markets: “market-makers serve a crucial role in financial markets by providing liquidity to facilitate market efficiency and functioning.”

The increased demand for cryptocurrencies has led to the proliferation of traditional Centralized Exchanges (CEXs)¹, organized around CLOBs, for such assets. However, the existence of traditional market makers - a role that is generally dominated by professional investors - and financial intermediaries introduces superfluous agents, defeating the purpose of decentralization. As such, DEXs have also exploded in popularity, especially amongst those in the DeFi community. As of December 2021, DEXs held over \$72B worth of crypto assets and had a monthly trading volume of \$152B, making them a major player in the cryptocurrency space. Compared to CEXs, DEXs currently account for approximately 13% of total cryptocurrency trading volume and are

¹Popular CEXs include Binance, Coinbase, and Kraken.

continuing to grow, as shown in Figure 2.



Figure 2: DEX vs. CEX trading volume over time. (Source: theblockcrypto.com)

DEXs are “smart contracts” - programs that reside and run on a blockchain. Most DEXs run on the Ethereum blockchain, which originally introduced smart contracts in 2013, or the Polygon blockchain, which is compatible with Ethereum smart contracts. Smart contracts are “not controlled by a user, instead they are deployed to the network and run as programmed. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts can define rules, like a regular contract, and automatically enforce them via the code. Smart contracts cannot be deleted by default, and interactions with them are irreversible” (Eth, 2022). Because users retain custody of their tokens when trading on a DEX, DEXs remove the custody risk that traders are exposed to when trading on CEXs and provide additional utility to users who may benefit from having access to their tokens on the blockchain. Technically, CLOBs could be organized as smart contracts. However, representing the state of the order book and supporting the complicated matching logic CLOBs use would be extremely costly as users must pay for space and compute power in the smart contract environment (Angeris, Kao, Chiang, Noyes, and Chitra, 2021).

Instead, DEXs utilize an AMM, which uses a relatively simple predefined set of rules to determine how trades are executed. In order to facilitate transactions, though, a DEX must have reserves, tokens that LPs deposit in exchange for a portion of the transaction fees generated by the exchange. Most DEXs utilize a separate smart contract for each token pair that users want to

trade; for example, Uniswap has a smart contract that enables users to trade WETH for USDC and another smart contract that allows users to trade WBTC for USDC. Because the reserves provided by liquidity providers are pooled together in a smart contracts, we commonly refer to these contracts as pools. Once a pool has reserves, traders can trade with the contract based on the rules defined by the AMM.

AMMs have been studied extensively in algorithmic game theory, starting with Hanson’s logarithmic market scoring rule used in prediction markets Hanson (2002). Many of the most popular DEXs utilize a Constant Function Market Maker (CFMM), a family of AMMs that share certain mathematical and theoretical properties. A Constant Product Market Maker (CPMM) requires that any trade must change the reserves, denoted x, y , such that $xy = k$. In other words, the product of the reserves must remain unchanged after each trade. One notable drawback of this model is that the convexity of the function results in price impact when trading.

Constant sum market makers are another class of CFMMs, characterized by the equation $x + y = k$. With this scheme, the sum of the reserves must remain constant after each trade. Mohan (2022) notes that, unlike CPMMs, trading with a constant sum market maker results in no price impact. However, if the market price is variable, arbitrageurs “may drain the [constant sum market maker] of one token for the other.” This is problematic since the liquidity in one of the tokens would be destroyed and the LPs could experience significant losses due to adverse selection.

The constant mean market maker generalizes the CPMM and, for reserves $(x_1, x_2, x_3, \dots, x_n)$, is characterized by the equation $\prod_{i=1}^n x_i^{w_i} = k$, where $\sum_{i=1}^n w_i = 1$. Balancer, another popular DEX, utilizes a constant mean market maker. This generalization allows for AMMs to handle $n \geq 2$ tokens. However, this scheme results in price impact and increased complexity. Hybrid function market makers have also been introduced that aim to incorporate the benefits of different AMM models, such as reducing price impact (as in constant sum market makers) while simultaneously preventing reserves being drained during trading (as in constant product market makers). Figure 3 illustrates these three AMM schemes.

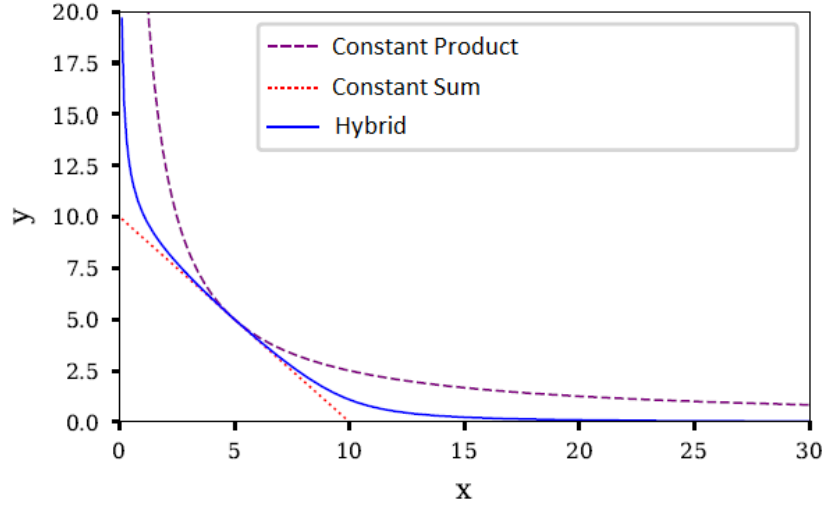


Figure 3: Different AMM schemes (Source: Curve Whitepaper, curve.fi)

The Uniswap v1 protocol, released in November 2018, was the first decentralized platform to utilize a CPMM. While quite successful, Uniswap v1 required that all trades occur through ETH, which resulted in serious transfer limitations and additional costs. Uniswap v2 was released in May 2020, allowing for common token pairs to be exchanged without using ETH as an intermediary. Uniswap v3, launched in May 2021, addresses issues related to capital efficiency in Uniswap v2 pools. Most notably, Uniswap v3 allows LPs to concentrate their liquidity by “bounding” it within a desired price range. This feature allows LPs to execute various provisioning strategies, which not only may improve capital efficiency but also introduce competition between LPs.

1.2 Related Work

While Uniswap v3 is still quite new, there has been notable research on earlier versions of Uniswap and other AMMs that provides a foundation for understanding the dynamics of such markets and the risks and returns of providing liquidity to an AMM.

Angeris et al. (2021) provides a comprehensive analysis of CPMMs, such as Uniswap v2, including deriving theoretical price bounds for a pool’s price compared to the true market price due to arbitrage. Guillaume Lambert has a series of Medium posts, including Lambert (2021a), Lambert (2021b), and Lambert (2021c), that derive several important properties of Uniswap v3, including

examining the value of an LP’s investment through the lens of option pricing and providing several suggestions on how to effectively provide capital to a Uniswap v3 pool. Clark (2021) also looks at Uniswap v3 through an option pricing framework and derives the replicating portfolio of a constant product market with bounded liquidity.

Lehar and Parlour (2021) compare Uniswap v2 pools to traditional CLOB markets and characterize the equilibrium liquidity for Uniswap v2 pools, providing evidence that existing Uniswap v2 pools are stable in practice. Additionally, consistent with traditional CLOB markets, the authors find that pools are larger (i.e., have more liquidity) the lower the volatility of the token price and the higher the level of liquidity trades compared to informed trades. Aoyagi (2020) develops a model for the equilibrium liquidity of a Uniswap v2 market, showing that the “equilibrium size of market liquidity tends to be stable and does not diverge or evaporate even if a (small) temporary shock hits the market.”

Two of the most relevant papers related to providing liquidity in Uniswap v3 actually arrive at quite different conclusions. Neuder et al. (2021) explore a class of “reset” strategies for providing liquidity in Uniswap v3, whereby a position is rebalanced once the pool’s price moves too far from the center of the LP’s position. The authors find that simple LP strategies for Uniswap v3 can earn over 200x more than providing liquidity in Uniswap v2, at least in their stylized environment. On the other hand, Loesch et al. (2021) explores impermanent loss for Uniswap v3 and find that the majority of LPs do not outperform a simple long only strategy, holding the assets that would have otherwise been invested in the pool, and that approximately 50% of LPs experience negative returns on their capital.

2. Uniswap

Uniswap v3 builds on Uniswap v2, introducing several new features, such as concentrated liquidity, that result in a more complicated protocol. As such, we first introduce Uniswap v2 and its mechanics before diving into Uniswap v3 in detail.

2.1 Uniswap v2

Each Uniswap smart contract is setup to allow traders to swap a single pair of tokens (e.g., WETH/USDC, WBTC/USDT, etc.). Utilizing the notation of the Uniswap v3 white paper for consistency across sections, we denote **token0** as the risky asset token and **token1** as the numeraire token. In order for traders to swap tokens in the pool, LPs must provide the pool with reserves of **token0**, x , and reserves of **token1**, y . The price of **token0** in terms of **token1** for the pool is then $p = \frac{y}{x}$. When a pool is created, the initial reserves deposited into the pool define the pool's price. However, if an LP wants to contribute new reserves to the pool or remove their existing reserves from the pool, they must do so at the prevailing price ratio (see equations (3) and (4) below). In return for providing liquidity to a pool, Uniswap LPs receive UNI tokens that entitle them to a portion of the transaction fees generated when traders swap tokens.

It is important to note that every transaction between a user and the pool, including both swapping and adding/removing liquidity incurs a gas fee. Gas fees are costs associated with executing transactions on the blockchain. Each transaction requires a known number of computational steps, each of which requires a known amount of gas, which is independent of market conditions (Lehar and Parlour, 2021). While the amount of gas to execute a transaction may be fixed, users offer additional amounts of ETH as incentives to miners so that their transactions will be included in a block (Lehar and Parlour, 2021). Users that offer more ETH will be prioritized by miners and will have their transactions executed before other users that offer less ETH, and such incentive fees can vary with market conditions (Lehar and Parlour, 2021). While Ethereum has made several improvements to reduce gas costs, high gas costs on the Ethereum blockchain are a known issue and can result in high costs for both traders and LPs in a Uniswap pool. Uniswap v3 transactions generally incur gas fees of approximately \$100 on the Ethereum blockchain, but these costs fluctuate and we have seen fees as high as \$400.

Once a pool has reserves, traders can then use the liquidity in the pool to swap tokens. If a trader wants to swap $\frac{\Delta y}{1-\gamma}$ of **token1**, where γ is the percentage transaction fee, for $-\Delta x^2$ of **token0**

²In our formulation, Δy and Δx have different signs, so swapping $\frac{\Delta y}{1-\gamma} > 0$ of **token1** results in $\Delta x < 0$, so the trader receives $-\Delta x$.

(i.e., buy the risky asset token), Uniswap requires that the following must hold (Zhang et al., 2018):

$$(x + \Delta x)(y + \Delta y) = k = xy \quad (1)$$

$$\Delta x = \frac{k}{(y + \Delta y)} - x \quad (2)$$

In other words, any trade must change the reserves so that, in the absence of fees, the product of reserves remains equal to a constant, k . Note that if a trader swaps $\frac{\Delta y}{1-\gamma}$ of **token1**, only $\frac{\Delta y}{1-\gamma}(1-\gamma) = \Delta y$ is used in the CPMM formula above; the remainder $\frac{\Delta y}{1-\gamma}\gamma$ is the transaction fee for the swap, which is added to the pool's reserves, increasing k . Immediately after a swap, the pool's state changes as follows:

$$\begin{aligned} y' &= y + \frac{\Delta y}{1-\gamma} \\ x' &= x + \Delta x \\ k' &= \left(y + \frac{\Delta y}{1-\gamma}\right)(x + \Delta x) = k + \frac{\gamma\Delta y}{1-\gamma}(x + \Delta x) > k \\ p' &= \frac{y + \frac{\Delta y}{1-\gamma}}{x + \Delta x} \end{aligned}$$

Additionally, it is important to note that fees are collected in the tokens swapped into the pool: if a trader swaps **token1** for **token0** (buys the risky asset token), fees are collected in **token1** whereas, if a trader swaps **token0** for **token1** (sells the risky asset token), fees are collected in **token0**, resulting a slight change to the formulas above.

The $xy = k$ CPMM formulation results in a bonding curve that determines the price impact, Δp , from a swap. Figure 4 below illustrates different bonding curves for different values of k .

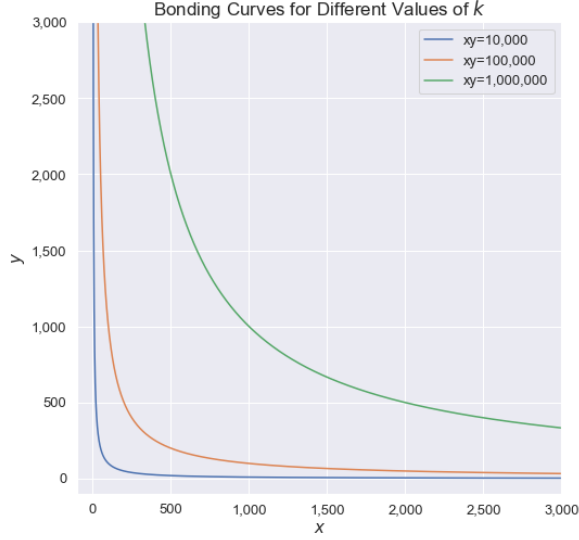


Figure 4: Bonding curves for various values of k .

The steepness of the bonding curve corresponds to the price impact from swapping. As k increases, the bonding curve becomes less steep and the price impact for any swap is reduced. We can use equation (1) and the fact that $p = \frac{y}{x} = \frac{y^2}{k}$ to determine the price impact from swapping in the absence of fees ($\gamma = 0$):

$$\begin{aligned}
 p + \Delta p &= \frac{y + \Delta y}{x + \Delta x} \\
 \Delta p &= \frac{(y + \Delta y)^2}{k} - p \\
 \Delta p &= \frac{2\Delta y + \Delta y^2}{k}
 \end{aligned}$$

Using this relationship we can see that as $k \rightarrow \infty$, $\Delta p \rightarrow 0$. k can therefore be thought of as a measure of liquidity. Consistent with traditional markets, pools with more liquidity result in less price impact for traders.

To make the mechanics of Uniswap v2 more concrete, we proceed with a simple example. Let's assume that we have a pool with a 1% transaction fee³ and with existing reserves of 800 of `token1` and 8 `token0`. The pool's current price is $p = \frac{y}{x} = \frac{800}{8} = 100$ and $k = xy = 6,400$. If a new

³All Uniswap v2 pools have a fee of 0.3%. The highest fee for a Uniswap v3 pool is 1%, but most pools have lower fees of either 0.3% or 0.05%, depending on the volatility and popularity of the tokens. 1% is chosen for simplicity

LP arrives, they must contribute tokens at a ratio of 100 of **token1** for every 1 of **token0**. If the LP contributes 200 of **token1** and 2 of **token0** - a position worth $200 + 2p = 400$ of **token1**, the numeraire asset in our example - the new reserves are now 1,000 and 10 respectively. The LP has contributed 20% of the total reserves and will, therefore, be entitled to 20% of the reserves, including fees generated, when they remove their position. After this transaction, $k = 10,000$, but the price remains unchanged. Adding liquidity to a pool has no effect on the pool's price. Now, a trader arrives and wants to buy **token0**; they swap in $\frac{100}{1-0.01} = 101.01$ of **token1**. $101.01(0.01) = 1.01$ of their swap is collected as transaction fees and the remaining 100 is used to determine how much of **token0** they receive:

$$\Delta x = \frac{k}{(y + \Delta y)} - x$$

$$\Delta x = \frac{10,000}{(1,000 + 100)} - 10 \approx -0.909$$

The trader receives approximately 0.909 of **token0**. The reserves of the pool - and, therefore, the pool's price - have changed, though. Now, the pool has 1,101.01 of **token1** and 9.09 of **token0**, so the pool's new price is $p' = 121.12$. If the LP now decides to withdraw his position, he can extract 20% of the pools reserves at the current ratio, and he receives 220.20 of **token1** and 1.818 of **token0**. The total value of his position, including fees collected, denominated in **token1** is $220.20 + 1.818p' = 440.40$.

To create (remove) an LP position with $\Delta L = \sqrt{k}$ liquidity⁴, an LP must contribute (receive) $\Delta X, \Delta Y$ ⁵ of **token0** and **token1**, respectively:

$$\Delta Y = \Delta L \sqrt{p} \tag{3}$$

$$\Delta X = \Delta L \frac{1}{\sqrt{p}} \tag{4}$$

Let us assume that an LP created their position when the pool's price was p , contributing $\Delta X, \Delta Y$ to the pool. When the LP liquidates its position at the new price, p' , the LP receives

⁴The reason for using the ΔL notation will become clear when we discuss Uniswap v3.

⁵Note that we use capital $\Delta X, \Delta Y$ when describing tokens added or removed from the pool due to liquidity transactions and lower case $\Delta x, \Delta y$ when describing tokens added or removed from the pool due to swaps.

$\Delta X', \Delta Y'$. When $p' > p$, $\Delta Y' > \Delta Y$ and $\Delta X' < \Delta X$, so an LP receives more of the numeraire asset and less of the risky asset, which has appreciated in value. Conversely, when $p' < p$, $\Delta Y' < \Delta Y$ and $\Delta X' > \Delta X$, so an LP receives less of the numeraire asset and more of the risky asset, which has depreciated in value. In other words, when an informed arbitrageur trades, moving the pool's price, they effectively rebalance the LP's portfolio at disadvantageous terms, which constitutes a form of adverse selection (Lehar and Parlour, 2021). As compensation for such risk, though, LPs earn fees from transactions, including any trades from informed arbitrageurs.

Using equations (3) and (4) above, we can easily compute the value of an LP position (again in the absence of fees), assuming that the pool's price, p , equals the market price of the risky asset:

$$V(p) = \Delta Y + p\Delta X = 2\Delta L\sqrt{p} = 2\sqrt{kp}$$

Further, the gross return on such a position is given by:

$$\frac{V(p')}{V(p)} = \frac{2\sqrt{kp'}}{2\sqrt{kp}} = \sqrt{\frac{p'}{p}}$$

As Uniswap and other AMMs became popular, investors began to compare LP returns to the returns of simply holding the assets. The relative difference between an LP's ending value and the value of a long position in the initial assets invested was dubbed “impermanent loss” (IL) by Pintail (2019). If an investor was to simply hold the assets invested that would have been used to provide liquidity in the pool - the “HODL” strategy - the value of the portfolio at the new price, p' , would be:

$$\begin{aligned} \text{HODL}(p') &= \Delta Y + p'\Delta X \\ &= \sqrt{kp} + p \frac{p'}{p} \frac{\sqrt{k}}{\sqrt{p}} \\ &= \sqrt{kp} \left(1 + \frac{p'}{p} \right) \end{aligned}$$

We can now defined IL as:

$$\begin{aligned}
\text{IL}\left(\frac{p'}{p}\right) &= \frac{V(p')}{\text{HODL}(p')} - 1 \\
&= \frac{2\sqrt{kp\frac{p'}{p}}}{\sqrt{kp}\left(1 + \frac{p'}{p}\right)} - 1 \\
&= \frac{2\sqrt{\frac{p'}{p}}}{1 + \frac{p'}{p}} - 1
\end{aligned} \tag{5}$$

Figure 5 plots equation (5) as a function of $\frac{p'}{p}$. The loss is minimized at $\text{IL}\left(\frac{p'}{p} = 1\right) = 0$, so, in the absence of fees, providing liquidity always under-performs the long portfolio, which is a direct result of adverse selection - i.e., the LP's portfolio being rebalanced against them. IL is purely a return concept and does not account for the fact that the long portfolio and the LP position have notably different risk profiles, especially when including fees, which we explore more during our discussion of Uniswap v3. IL, however, does make it clear that fees earned are a critical part of an LP's return and, in the absence of such fees, an LP is unlikely to earn substantial risk-adjusted returns.

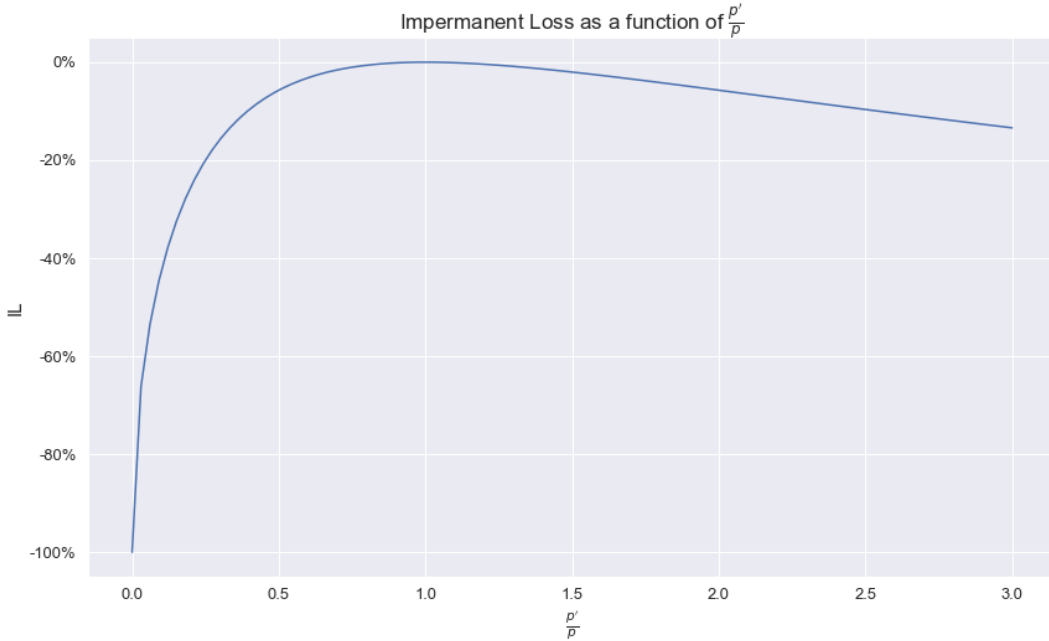


Figure 5: Impermanent loss as a function of $\frac{p'}{p}$.

Despite its popularity, Uniswap v2 suffers from low capital efficiency. When an LP supplies liquidity, the liquidity is distributed across the whole space of prices (i.e. $p \in (0, \infty)$). As such, a significant portion of the assets provided to the pool is not involved in facilitating swaps, especially if the token pair has a relatively stable price ratio, such as stablecoin pairs. As such, LPs collect fewer fees (relative to their contribution) and the pool experiences higher price impact. Uniswap v3, which we discuss next, introduced concentrated liquidity to address these issues.

2.2 Uniswap v3

In Uniswap v2, LPs simply decide how much liquidity they want to contribute to the pool and the amounts of `token0` and `token1` they provide are wholly determined by the pool’s price. Such liquidity provided to the pool was effectively distributed uniformly across the entire price range $(0, \infty)$, resulting in a single bonding curve $xy = k$. Uniswap v3 uses the same CPMM curve as Uniswap v2 but introduced the concept of concentrated liquidity. In the Uniswap v3 pools, LPs can now bound their liquidity within an arbitrary price range, improving “capital efficiency and allow[ing] LPs to approximate their preferred reserves curve, while still being efficiently aggregated with the rest of the pool” (Adams et al., 2021).

Consistent with the Uniswap v3 white paper, we refer to liquidity within a given price range as a position. Each position needs enough reserves to support trading within its range: a position needs enough of `token0` to cover price movements to its upper bound as a trader swapping in `token1` and receiving `token0` pushes the price up, and enough of `token1` to cover price movements to its lower bound as trading swapping in `token0` and receiving `token1` pushes the price down (Adams et al., 2021). Within this range, the pool acts like a CPMM, similar to Uniswap v2, with larger reserves, known as “virtual” reserves. Once the price moves outside of the range of a position, the position’s liquidity is no longer active, so no fees can be earned, and the position is composed entirely of a single token; if the price, p , is above the position’s upper range, the position is composed of `token1` whereas, if the price is below the position’s lower range, the position is composed of `token0` (Adams et al., 2021). LPs can create as many positions as they want, allowing LPs to distribute liquidity along the space of prices as they see fit. However, because adding or removing liquidity

to a pool imposes a gas fee on the LP, creating many very small positions would be very costly in practice. Multiple LP positions can be aggregated to form a “liquidity curve”, $L(p)$, which tell us how much liquidity is available in the pool at any given price. A simplified Uniswap v3 liquidity curve is presented in Figure 6.

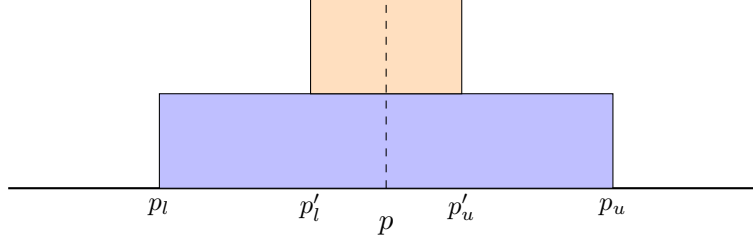


Figure 6: Sample liquidity curve for Uniswap v3. LP_1 has a position over $[p_l, p_u]$ (represented by the blue box), and LP_2 has a position over $[p'_l, p'_u]$ (represented by the orange box). For $p \in [p'_l, p'_u]$, both LPs collect fees proportional to their contribution in that range. Also, since both positions are active, the higher liquidity in this range reduces price impact. When p leaves this range, LP_1 will collect 100% of the fees, provided $p_l < p < p_u$. Otherwise, there is no liquidity and swaps can't be facilitated.

The implementation of concentrated liquidity required several changes from Uniswap v2. First, transaction fees are no longer added to a pool's reserves but are set aside in a separate account. Second, the range of possible prices was divided into discrete ticks, which define the edges of an LP's position and determine how the tokens received from any swap are calculated. Next, we discuss the mechanics of Uniswap v3, focusing on the implementation changes related to concentrated liquidity.

2.2.1 Mechanics of Uniswap v3

One of the largest changes in Uniswap v3 was the introduction of ticks. Ticks are identified by their index, i , and a range between two ticks can be defined in terms of i_l , the lower tick index, and i_u , the upper tick index. A tick exists at every price that is an integer power of 1.0001:

$$p(i) = 1.0001^i$$

so that each tick is a 0.01% (1 basis point) price movement away from its neighboring ticks (Adams et al., 2021). Each tick also identifies a range of prices. All prices within $[p(i), p(i + 1))$ represent

the tick i , and we can calculate the tick index for a given price as:

$$i = \lfloor \log_{1.0001}(p) \rfloor$$

Additionally, each Uniswap v3 pool specifies a `tickSpacing` variable; only ticks with indexes that are divisible by `tickSpacing` can be used to define an LP's liquidity position (Adams et al., 2021). The smaller the `tickSpacing`, the more precisely LPs can define their liquidity ranges. However, this also results in higher gas costs for traders as more calculations must be done to determine how much a trader receives from a swap (Adams et al., 2021).

Within each tick, liquidity is constant as no position can begin or end within a tick. As such, if a swap is small enough so that the price stays within the current tick, i_c , the contract acts like a CPMM, $xy = k$. Unlike Uniswap v2, though, x, y used in the CPMM formulation are now virtual reserves. In fact, a Uniswap v3 smart contract actually tracks L and \sqrt{p} and uses these values to calculate the virtual reserves within the current tick, i_c , using the following formulas:

$$L = \sqrt{xy} \tag{6}$$

$$\sqrt{p} = \sqrt{\frac{y}{x}} \tag{7}$$

(Adams et al., 2021). Note that Uniswap v3 uses L^2 instead of k to denote the constant product, but this is simply a change in notation from Uniswap v2: $L^2 = k$ in areas where L is constant. Using equations (1), (6), and (7), we can also write L as

$$L = \frac{\Delta y}{\Delta \sqrt{p}} \tag{8}$$

In other words, liquidity, L , can be thought of as the amount that `token1` reserves change for a given change in \sqrt{p} (Adams et al., 2021). Let us assume that a trader wants to swap $\frac{\Delta y}{1-\gamma}$ of `token1` for some amount of `token0`. $\gamma \frac{\Delta y}{1-\gamma}$ is set aside as transaction fees and $(1-\gamma) \frac{\Delta y}{1-\gamma} = \Delta y$ is added to the pool. The pool then uses the following equations, which can be derived from equations (2)

and (8), to determine how much of **token0** the trader receives:

$$\Delta\sqrt{p} = \frac{\Delta y}{L} \quad (9)$$

$$\Delta x = \Delta \frac{1}{\sqrt{p}} L \quad (10)$$

Similarly, if Δx is added to the pool, the following equations can be used to determine Δy :

$$\Delta \frac{1}{\sqrt{p}} = \frac{\Delta x}{L} \quad (11)$$

$$\Delta y = \Delta\sqrt{p}L \quad (12)$$

(Adams et al., 2021). Because fees from a swap are not added to the pool’s reserves, swapping only changes p (and \sqrt{p}), but not L ; L only changes when an LP adds or removes liquidity from the pool.

If the calculated $\Delta\sqrt{p}$ is large enough that the new price, p' , would move outside of the current tick, i_c , the swap is only partially executed within the current tick. I.e., $\Delta\sqrt{p}$ is limited to the difference between the current price, \sqrt{p} and the price at the tick edge - either $\sqrt{p(i)}$ if swapping in **token0** or $\sqrt{p(i+1)}$ if swapping in **token1** - and partial $\Delta x, \Delta y$ are calculated for the current tick using this adjusted $\Delta\sqrt{p}$. After the swap is partially executed within the current tick, the contract must “cross the tick”, at which point liquidity, L , may change, depending on the liquidity at the new tick, and the swap continues to execute in this new tick in the same manner (Adams et al., 2021). This process continues until the amount of **token1** or **token0** swapped in is exhausted. If a swap pushes the price to the minimum or maximum tick that defines a liquidity position, after which there is no more liquidity, then a swap may only be partially executed. It is also important to note that fees are earned within each tick in proportion to the amount of the swap that is executed within the tick. For example, if $\Delta y = 100$ but only 50 of **token1** is needed to push the price to the next tick, after which the remaining 50 of **token1** can be swapped, then 50% of the fees are allocated to the first tick and 50% of the fees are allocated to the second tick.

Because swapping in Uniswap v3 requires moving tick-to-tick, adjusting liquidity as the swap progresses, it is difficult to arrive at simple, analytical formulas (such as equation (2) for Uniswap

v2 above) for how much a trader receives from swapping one token for another. We can, however, use the Uniswap v3 equations above to come up with analytical approximations under certain assumptions.

Recall that within a tick, where L is constant, the following holds:

$$L = \frac{\Delta y}{\Delta \sqrt{p}} \implies \Delta y = L \Delta \sqrt{p}$$

For notational convenience, let $\sqrt{p} = q$. We start by trying to find the amount, Δy , such that swapping in Δy of `token1` moves the the square root of the price from q_0 to $q_1 > q_0$. Let us assume that ticks can be arbitrarily spaced, so that $q_1 = q_0 + n\Delta q$ where $\Delta q = \frac{q_1 - q_0}{n}$. Additionally, we assume that L is constant within each interval, regardless of the n chosen, which we denote L_i . We can then calculate Δy as

$$\Delta y = \sum_{i=1}^n L_i \Delta q$$

This takes a form similar to a left Riemann sum, and as we let $n \rightarrow \infty$ the above converges to the integral:

$$\Delta y = \int_{q_0}^{q_1} L(q) dq \tag{13}$$

and L becomes a continuous function of p or q . In practice, $L(p)$ is a complex step function, but we allow $L(p)$ to be a smooth, continuous function for our approximations for simplicity. Note that, if $q_1 < q_0$, $\Delta y < 0$ and we would replace L_i with L_{i-1} above as we are swapping the other direction. In this case, the equation is a right Riemann sum but still converges to the same integral.

Now, if we assume that a pool's liquidity can be approximated, at least locally, by some continuous function, $L(q)$, and given the pool's current square root price, q_0 , and a fixed amount to swap, Δy , we can find the resulting square root of the price, q_1 . We start by assuming that $L(q)$ is linear in q - i.e., L is linear in the square root of the price - or that we can approximate L as a

linear function in q around the current square root price, q_0 . Writing L as:

$$L(q) = \alpha + \beta q$$

we can solve for q_1 :

$$\begin{aligned}\Delta y &= \int_{q_0}^{q_1} (\alpha + \beta q) dq \\ \Delta y &= \alpha q + \frac{\beta}{2} q^2 \Big|_{q_0}^{q_1} \\ \Delta y &= \alpha q_1 + \frac{\beta}{2} q_1^2 - \alpha q_0 - \frac{\beta}{2} q_0^2 \\ \frac{\beta}{2} q_1^2 + \alpha q_1 - [\alpha q_0 + \frac{\beta}{2} q_0^2 + \Delta y] &= 0\end{aligned}\tag{14}$$

Note that when $\beta = 0$, the above reduces equation (2) for a Uniswap v2 swap. We can now use the quadratic formula to solve for q_1 :

$$\begin{aligned}q_1 &= \frac{-\alpha \pm \sqrt{\alpha^2 - 4 \frac{\beta}{2} [\alpha q_0 + \frac{\beta}{2} q_0^2 + \Delta y]}}{2 \frac{\beta}{2}} \\ q_1 &= \frac{-\alpha + \sqrt{(\alpha + \beta q_0)^2 + 2\beta \Delta y}}{\beta}\end{aligned}\tag{15}$$

We discard the negative solution as $q > 0$. Now, since we know what the final price is for a swap of size Δy , we can also calculate the amount of `token0` for the swap:

$$\begin{aligned}\Delta x &= \Delta \frac{1}{q} \\ \Delta x &= \left(\frac{1}{q_1} - \frac{1}{q_0} \right) \frac{\Delta y}{q_1 - q_0} \\ \Delta x &= \frac{-\Delta y}{q_1 q_0}\end{aligned}\tag{16}$$

A natural linear approximation for $L(q)$ at q_0 is the tangent line, so we can use this approximation

for swaps where liquidity is changing linearly around the current square root price:

$$\begin{aligned}\hat{L}(q) &= L(q_0) + L'(q_0)(q - q_0) \\ \hat{L}(q) &= (L(q_0) - L'(q_0)q_0) + L'(q_0)q \\ \implies \alpha(q_0) &= L(q_0) - L'(q_0)q_0 \text{ and } \beta(q_0) = L'(q_0)\end{aligned}$$

With these formulas we can calculate the amount of **token0** bought or sold, Δx , for Δy of **token1** and the new price of the pool immediately after the swap, q_1 , in terms of $q_0, L(q), \Delta y$. While this is only an approximation, we use this formulation for swaps when trying to solve for the optimal arbitrage trade in the presence of non-constant liquidity in Uniswap v3 pools in Section 3.

Concentrated liquidity also has an effect on the tokens an LP contributes to or receives from the pool. Unlike Uniswap v2, in which LPs provide and receive tokens at the ratio defining the pool's price, the amount of **token0** and **token1** an LP contributes or receives depends on the position's range and current price: "the amount of **token0** (ΔX) or **token1** (ΔY) that needs to be deposited can be thought of as the amount that would be sold from the position if the price were to move from the current price (p) to the upper tick or lower tick (for **token0** or **token1**, respectively)" (Adams et al., 2021). The following equations are provided in the Uniswap v3 white paper, but can also be derived from the equations above:

$$\Delta Y = \begin{cases} 0 & i_c < i_l \\ \Delta L(\sqrt{p} - \sqrt{p(i_l)}) & i_l \leq i_c < i_u \\ \Delta L(\sqrt{p(i_u)} - \sqrt{p(i_l)}) & i_c \geq i_u \end{cases} \quad (17)$$

$$\Delta X = \begin{cases} \Delta L\left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}}\right) & i_c < i_l \\ \Delta L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p(i_u)}}\right) & i_l \leq i_c < i_u \\ 0 & i_c \geq i_u \end{cases} \quad (18)$$

where ΔL is the amount of liquidity added to the pool, i_l, i_u are the ticks defining the position's lower and upper bounds, respectively, and i_c is the tick for the pool's current price, p . As discussed

above, when the pool's price is outside a position's range, the amount of tokens added to or removed from the pool is entirely composed on a single token. Only when the price is within the position's range does an LP add or receive both `token0` and `token1`.

2.2.2 Valuing a Uniswap v3 Position

Each LP position is defined by three variables, ΔL , $p(i_l)$, $p(i_u)$, but the tokens that such a position is entitled to is determined by (17) and (18) above. As such, we can use these equations to determine the value of an LP position, excluding accrued fees, for any given value of p . We use a similar derivation for the value of the position as (Lambert, 2021a), but use the notation of the Uniswap v3 white paper for consistency. The value of any position, denominated in the numeraire token, is simply:

$$V(p) = \Delta Y + p\Delta X$$

Using the formula (17) and (18) above for $\Delta X, \Delta Y$, we can write the value of the position as:

$$V(p) = \begin{cases} \Delta L p \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right) & i_c < i_l \\ \Delta L \left(2\sqrt{p} - \frac{p}{\sqrt{p(i_u)}} - \sqrt{p(i_l)} \right) & i_l \leq i_c < i_u \\ \Delta L (\sqrt{p(i_u)} - \sqrt{p(i_l)}) & i_c \geq i_u \end{cases} \quad (19)$$

In all cases, the value of the position scales linearly with ΔL , so we can think of this term as a notional value that determines the size of the initial position. The value of the position scales differently with p depending on the range of the position selected. We plot the value of a position with $\Delta L = 1$ in Figure 7 below.



Figure 7: Value of an LP's position as a function of p for $\Delta L = 1$.

As the range of the position gets smaller, the value curve becomes “sharper” and as $p(i_l) \rightarrow p(i_u)$, the value converges to the payoff diagram of a covered call or a cash covered short put, again, excluding any fees earned (Lambert, 2021a). Equation (19) provides us with the value of the position for any given value of p , but what is the expected value of some investment with a fixed time horizon, $T - t$, at the current price, p_t ? While finding the discounted, expected value of the position is difficult analytically, we can use well-known results from option pricing to find a portfolio that statically replicates the payoff of the position at “maturity.” Similar to Clark (2021), we can use the results of Carr et al. (1998) to decompose the value of the position into a portfolio of bonds, forwards, and puts and calls on the risky asset, `token0`. Carr et al. (1998) show that, in the absence of arbitrage, the value of a twice differential payoff at time t with time-to-maturity $T - t$ can be written as:

$$\begin{aligned}
 V_t(S_t) &= e^{-r(T-t)} V_T(S_T = K^*) + V'_T(S_T = K^*)(S_t - K^* e^{-r(T-t)}) \\
 &\quad + \int_0^{K^*} V''_T(K) P_t(S_t, K) dK + \int_{K^*}^{\infty} V''_T(K) C_t(S_t, K) dK
 \end{aligned}$$

where S_t is the value of the underlying at time t , r is the risk free rate, $P_t(S_t, K), C_t(S_t, K)$ are the values of a put and call, respectively, both with strike K and time-to-maturity $T - t$, and K^* is an arbitrary “starting” strike. To create such a portfolio, we must first calculate the first and second derivatives of the payoff:

$$\frac{dV}{dp} = \begin{cases} \Delta L \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right) & i_c < i_l \\ \Delta L \left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p(i_u)}} \right) & i_l \leq i_c < i_u \\ 0 & i_c \geq i_u \end{cases}$$

$$\frac{d^2V}{dp^2} = \begin{cases} 0 & i_c < i_l \\ \frac{-\Delta L}{2p^{\frac{3}{2}}} & i_l \leq i_c < i_u \\ 0 & i_c \geq i_u \end{cases}$$

If we select $K^* = p(i_u)$, and note that $S_t = p_t$ (where we assume that the pool’s price equals the market price), then we can write the current value of the position as:

$$V_t(p_t) = e^{-r(T-t)} \Delta L (\sqrt{p(i_u)} - \sqrt{p(i_l)}) + \int_{p(i_l)}^{p(i_u)} \frac{-\Delta L}{2K^{\frac{3}{2}}} P_t(p_t, K) dK \quad (20)$$

This formulation illustrates that the value of the LP position is the value of a bond plus a portfolio of short put options on **token0** with strikes between $p(i_l)$ and $p(i_u)$. Like a covered call or a cash covered short put, in which the investor collects the premium from selling an option, the LP gives up some of the upside in the risky asset in order to collect transaction fees. While impermanent loss is widely discussed in the crypto community, comparing an LP’s position to a long position in the risky asset is not a valid comparison; we can clearly see that an LP’s position would have a very different volatility than a long position in **token0**. While option pricing provides an interesting lens through which to understand the value of an LP’s position, we leave this as an area for future work as devising a model for the fees earned by an LP is much more difficult than deriving equation (20) given the complex and competitive dynamics of a Uniswap v3 pool.

2.2.3 Capital Efficiency

In Uniswap v2, LPs contribute capital to the entire space of prices, $p \in (0, \infty)$. However, most of this capital is largely idle as swaps generally occur within a relatively tight range of prices. To illustrate this ineffective deployment of capital, consider a hypothetical Uniswap v2 stablecoin pair USDC/DAI pool, represented by Figure 8. If an LP wants to contribute $\Delta L = \sqrt{k} = 10$ liquidity to the pool, they must contribute 10 DAI and 10 USDC, (represented by the red area in Figure 8). This represents a 10% stake in the pool. Given that both tokens are stablecoins (i.e. pegged to an external value; in this case, pegged to USD), it is reasonable to assume that the price, p , will remain in $[0.99, 1.01]$. If a trader arrives and swaps 0.5 DAI, after fees, for $\Delta x = \frac{k}{(y+\Delta y)} - x = 0.4975$ USDC, the price moves from $p = 1.0 \rightarrow p' \approx 1.01$. Only $\Delta x \times 10\% \approx 0.05$ USDC of the 10 USDC tokens that the LP contributed are actually involved in the swap. As such, a significant amount of the capital provided is never utilized to facilitate swaps.

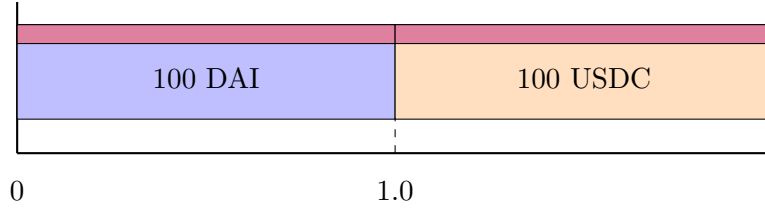


Figure 8: Continuous price range of USDC/DAI, determined by the ratio of the assets in the pool: $\frac{100\text{DAI}}{100\text{USDC}} = 1.0$. The red region is the contribution from an LP that represents 10% of the pool.

Concentrated liquidity in Uniswap v3 increases capital efficiency as far fewer tokens are required to create a comparable amount of liquidity around a bounded range. To add ΔL liquidity to a pool in Uniswap v2, an LP contributes (see equations (3) and (4) above):

$$\Delta Y = \Delta L \sqrt{p}, \quad \Delta X = \frac{\Delta L}{\sqrt{p}}$$

to the pool. However, in Uniswap v3, an LP only needs to contribute (see equations (17) and (18))

above):

$$\begin{aligned}\Delta Y &= \Delta L(\sqrt{p} - \sqrt{p(i_l)}) < \Delta L\sqrt{p} \\ \Delta X &= \Delta L\left(\frac{1}{\sqrt{p}} - \frac{1}{\sqrt{p(i_u)}}\right) < \frac{\Delta L}{\sqrt{p}}\end{aligned}$$

Note that we assumed that $p(i_l) < p < p(i_u)$ for simplicity and comparability. As the position's range becomes larger, $p(i_l) \rightarrow 0, p(i_u) \rightarrow \infty$, the amount of tokens a Uniswap v3 LP must contribute to the pool converges to the Uniswap v2 amounts. Conversely, as the position's range becomes smaller, $p(i_l) \rightarrow p, p(i_u) \rightarrow p$, the amount of tokens required becomes very small. As such, an LP can contribute a relatively large amount of liquidity to the pool with a relatively small investment of `token0` and `token1`.

Going back to our example above, a Uniswap v3 LP could contribute $\Delta L = 10$ liquidity to the USDC/DAI pool, with $p = 1.0$ and a range of $p(i_l) = 0.99, p(i_u) = 1.01$ for:

$$\begin{aligned}\Delta Y &= 10(\sqrt{1} - \sqrt{0.99}) \approx 0.5 \\ \Delta X &= 10\left(\frac{1}{\sqrt{1}} - \frac{1}{\sqrt{1.01}}\right) \approx 0.5\end{aligned}$$

Because we expect the price to stay within such a tight range, the LP contributes 95% fewer tokens to supply the same level of liquidity to the pool. In fact, if the same trade discussed above were to occur, all of the LP's tokens would be utilized to facilitate the trade. If the LP still wanted to contribute 10 USDC and 10 DAI, he could create a position with $\Delta L = 2,000$, significantly increasing liquidity and reducing the price impact from swaps.

If the price stays within the range, a Uniswap v3 LP earns the same amount of fees as a Uniswap v2 LP, but with a smaller required capital outlay, increasing the returns of the LP. Unfortunately, such amplified gains come at a cost. Lambert (2021c) shows that impermanent loss becomes increasingly worse as the position's range tightens⁶, which, again, is a direct consequence of negative portfolio rebalancing due to adverse selection. Given that a Uniswap v3 position amplifies both the gains and losses of a comparable Uniswap v2 position by reducing the capital required for

⁶In fact, impermanent loss in Uniswap v3 is always worse than in Uniswap v2 where a position's range is always $(0, \infty)$.

the position, we can think of a Uniswap v3 position as a leveraged Uniswap v2 position limited to a range, with the amount of leverage increasing as the position’s range tightens and fewer tokens are required to fund the position.

3. Arbitrage in Uniswap v3

Uniswap pools rely on the existence of arbitrageurs to ensure that the pool’s price is close to the true market price. In fact, Uniswap v2 introduced “flash swaps”, which allow users to effectively borrow assets from the pool and execute arbitrary logic as long as the assets are either returned along with a small fee or are paid for like a normal swap by the end of the transaction (Adams et al., 2020). Whereas in traditional markets arbitrageurs must have sufficient capital to take advantage of arbitrage opportunities, flash swaps effectively allow anyone to perform capital free arbitrage.

In the presence of transaction fees, gas fees, and other market frictions, trading from arbitrageurs will not move the pool’s price all the way to true market price. Angeris et al. (2021) show that in Uniswap v2 pools, in the presence of transaction fees, γ , but no other market frictions, the pool’s price, p , bounds the true market price, m :

$$p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$$

Unfortunately, because a Uniswap v3 pool’s liquidity can take on arbitrary shapes, it is difficult to analytically arrive at such theoretical bounds for a pool’s price. We can, however, use our swap approximations, equations (14) and (15), to numerically solve for optimal arbitrage when liquidity changes with p .

When $m > p$, an arbitrageur can profit by borrowing `token1`, swapping it into the Uniswap pool to receive `token0`, and converting `token0` at the market price, m , back into `token1` to repay the loan, profiting from the mispricing of the Uniswap pool. This process adds `token1` and removes `token0` from the pool’s reserves pushing the pool’s price towards the market price. Since the arbitrageur is swapping in `token1`, fees are also collected in `token1`. The arbitrageur, therefore, solves the following constrained optimization problem to determine how much of `token1` to borrow

and swap:

$$\max_{\Delta x, \Delta y} (-\Delta x)m - \frac{\Delta y}{1 - \gamma}$$

This objective function is constrained as we require $-\Delta x > 0, \Delta y > 0$ ⁷ and that $\Delta x, \Delta y$ follow the CPMM formula at each tick within where the swap is executed. Using our analytical approximations, with a linear approximation of $L(q)$, we can simplify the problem, removing some of the constraints:

$$\begin{aligned} & \max_{\Delta y} -\frac{-\Delta y}{q_1 q_0} m - \frac{\Delta y}{1 - \gamma} \\ &= \max_{\Delta y} -\frac{-\Delta y}{q_0 \frac{-\alpha + \sqrt{(\alpha + \beta q_0)^2 + 2\beta \Delta y}}{\beta}} m - \frac{\Delta y}{1 - \gamma} \\ &= \max_{\Delta y} \frac{\Delta y \beta m}{q_0 (-\alpha + \sqrt{(\alpha + \beta q_0)^2 + 2\beta \Delta y})} - \frac{\Delta y}{1 - \gamma} \end{aligned} \tag{21}$$

subject to $\Delta y > 0$

This equation can be easily solved numerically to approximate the optimal arbitrage in a Uniswap v3 pool with non-constant liquidity.

In order to better understand how a Uniswap v3 pool's price bounds the market price in the presence of arbitrageurs and non-constant liquidity, we can simulate a Uniswap v3 market and compare the simulated price bounds to the theoretical price bounds for a Uniswap v2 pool. We use our Python Uniswap v3 simulator⁸ for all simulations in this paper and implement the following simulation procedure:

1. We assume that the market price, m , follows a geometric Brownian motion process with drift μ and volatility σ . We generate 30 days of trading with 2 price innovations per day for a total of 60 prices. We assume that there are no market frictions when trading at the market price outside of the Uniswap pool.

⁷Remember, if $\Delta y >$, then $\Delta x < 0$ in our formulation.

⁸Our simulator is based on the Uniswap v3 white paper and the actual Uniswap v3 project on GitHub: <https://github.com/Uniswap/v3-core>. Our implementation is a simplified version that includes only the core functionality and uses floating-point precision rather than integer arithmetic.

2. After each price innovation, a single liquidity trader arrives swapping enough to move the pool’s price approximately 10% above the current market price. The 10% price movement was chosen so that the price moves outside of the theoretical Uniswap v3 price bounds for any reasonable fee rate, but is relatively arbitrary. A smaller value could be used, especially for pools with lower fee rates, without any impact to the analysis. Immediately after this liquidity trade, an arbitrageur arrives and solves the above optimization problem, equation (21), trading enough to maximize their profit and moving the pool’s price closer to the current market price. Next, another liquidity trader arrives, moving the pool’s price approximately 10% below the current market price. Once again, an arbitrageur arrives to take advantage of the new price deviation. We record the pool’s price immediately after each arbitrage trade, which should provide us with the upper and lower bounds for the current price.
3. After these series of trades happen, we move to the next price innovation, repeating the above until we arrive at the end of the simulation.

Because L cannot be continuous nor is L usually linear in \sqrt{p} , our arbitrage optimization problem is only an approximation. The closer L is to a step function, the more discretization error is introduced. The more curvature L exhibits, the worse the linear approximation becomes. To alleviate some of these errors, rather than having the arbitrageur only execute a single swap, we allow the arbitrageur to swap multiple times, incrementally moving the pool’s price towards the market price until arbitrage is no longer profitable. In the presence of proportional transaction costs, γ , and no gas fees, executing a single, large swap is equivalent to executing several smaller swaps totalling the same amount of tokens swapped in; refer to Appendix A for a simple proof. This is unique to Uniswap v3. Angeris et al. (2021) show that, in Uniswap v2, splitting trades is always more expensive than trading an equivalent amount in a single trade. However, this is due to the fact that fees from each transaction are added to the pool’s reserves in Uniswap v2, while, in Uniswap v3, fees are set aside in a separate account.

In our implementation, an arbitrageur takes several small steps towards the current market price, using the secant line between the pool’s current price and a target price as a linear (in the \sqrt{p}) approximation for L . After each step, the step size is reduced so that the linear approximation

becomes more accurate as we approach the market price. For example, if $m = 100, p = 110$, an arbitrageur first trades to the target price of 105, using the secant line between 110 and 105 as an approximation for L . Once this swap is executed, the next target price becomes 102.5 and the arbitrageur swaps again, and finally, the target price becomes 100 and the arbitrageur swaps again, assuming it is still profitable to do so.

We performed the above simulation for multiple different fees, between 0.05%, the lowest transaction fee for any Uniswap v3 pool, to 3%, which is actually higher than the highest transaction fee, 1%, for any pool. We explored the effect of non-constant liquidity by looking at different liquidity curves. Each curve is linear in price with varying slopes, but the liquidity at the initial price, $p = 100$, is the same. Figure 9 presents 6 different liquidity curves used, all of which are centered at $p = 100, L(100) = 5,000$ (though we note that the results are the same regardless of the overall level of liquidity).

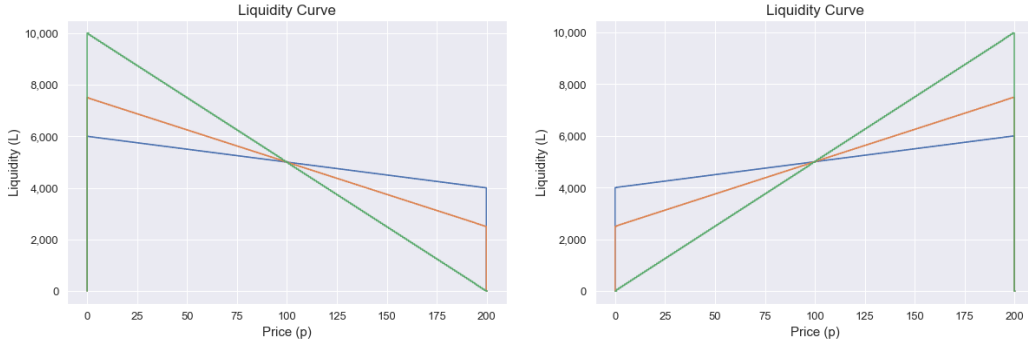


Figure 9: Linear liquidity curves for our price bound simulations.

In all cases, the simulated price bounds converged to the theoretical Uniswap v2 price bounds. Figure 10 presents the results of one such simulation for $L(p) = 50p + 0$ with $\gamma = 0.03$. We present the 3% fee simulation in the report as the larger fee results in a larger price bound, which is easier to see. However, we note that the results are the same for the smaller fees. We present the results for the different liquidity curves presented in Figure 9 for 3 different price simulations in Appendix B.

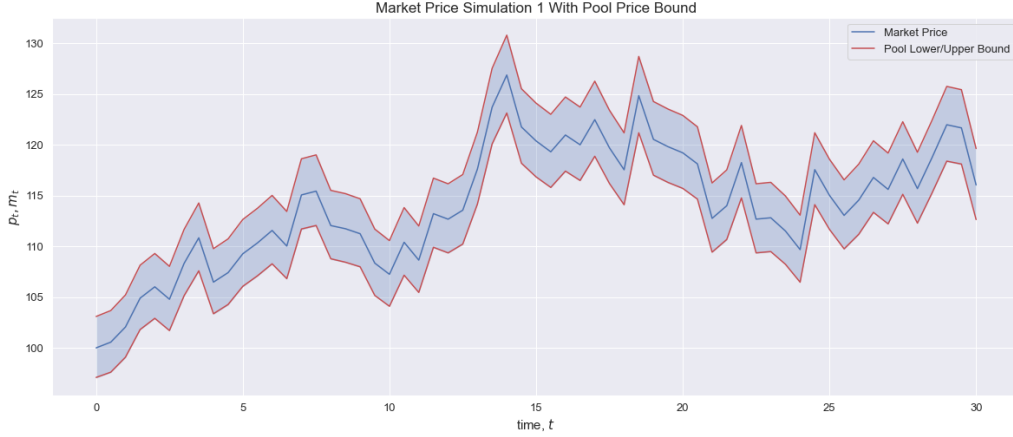


Figure 10: Simulated price bounds for a $L(p) = 50p + 0$ liquidity curve with $\gamma = 0.03$. The blue bands represent the theoretical Uniswap v2 price bounds: $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$. The red lines are the simulated Uniswap v3 price bounds.

These results suggest that the shape of the liquidity curve for Uniswap v3 pools does not affect the theoretical price bounds in the presence of sufficient arbitrageurs and no market frictions, including gas fees. While gas fees, like transaction fees, clearly increase the price bounds, the bound’s invariance to the shape of L indicates that Uniswap v3 pools are quite robust to multiple shapes and levels of liquidity curves that LPs may, as a whole, create.

While the simulated price bounds for Uniswap v3 may converge to the theoretical price bounds for Uniswap v2, the real question is whether there is sufficient arbitrage in practice that such price bounds are enforced. To determine whether the pool prices appropriately follow market prices, we obtained minute level price data from the Gemini exchange and use the average of the open-close, high-low prices for each minute as a proxy for the true market price, m , during each time interval. We then examined how often the pool’s price deviated from the theoretical price bound, $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$, and how quickly such deviations were corrected. Figure 11 shows the estimated market price (i.e., the reference value), the price bounds, and the second level pool prices for four Uniswap v3 pools on January 27, 2022. The pools’ prices clearly stay within the price bound for the larger 0.3% fee pools. There are notable spikes in the pool’s price away from the reference value for the smaller 0.05% fee pools; however, the price quickly moves back into the price bound (in many cases, the next swap corrects the price) and we do not see persistent deviations from the theoretical price bounds. Lehar and Parlour (2021) note that attackers often deliberately push

markets out of equilibrium, so it is not immediately clear whether these spikes are due to large liquidity traders, informed traders, or attackers trying to manipulate the pool’s price. The longest deviation we identified for the pools we examined was approximately two hours on May 19, 2021, a day with relatively high volatility in the crypto markets and when the Uniswap v3 pools were still quite new.

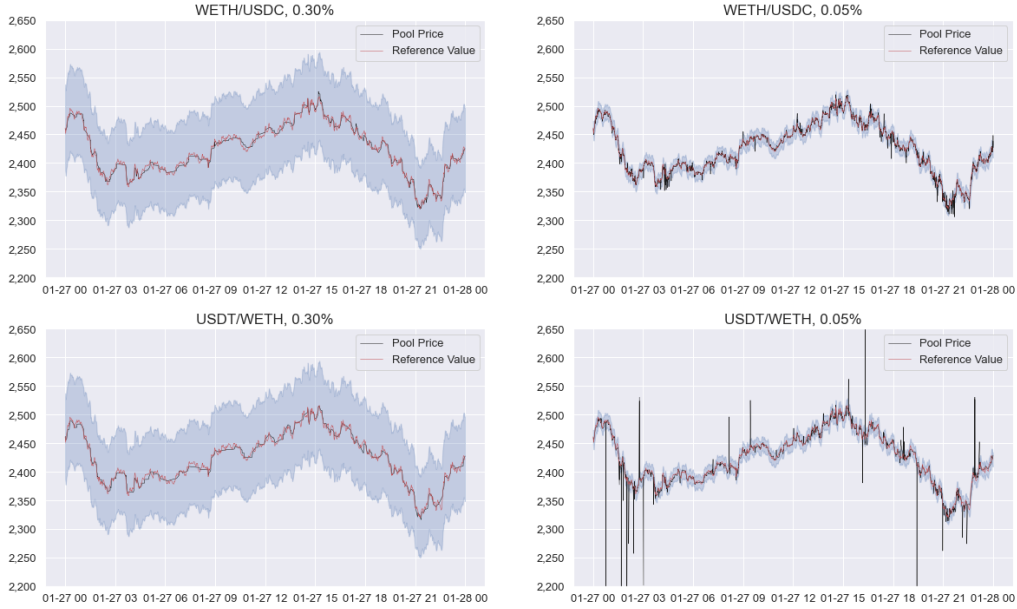


Figure 11: Actual price bounds for four different WETH/stablecoin pools on January 27, 2022. The theoretical price bounds, $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$, are identified by the blue bands. The x and y axes have been aligned across all images. In some cases, the deviations from the price bounds for the USDT/WETH pool actually extend off the plots. Note also that all prices are presented with the stablecoin as the numeraire asset, which may differ from the pool’s implementation.

4. Liquidity Provisioning

With the introduction of concentrated liquidity in Uniswap v3, LPs can now decide within what price range they want to provide liquidity, allowing LPs to execute various provisioning strategies. In addition to potentially improving capital efficiency, such decisions also result in competition between LPs, unlike Uniswap v2 where liquidity provisioning is not rivalrous (Lehar and Parlour, 2021). In Uniswap v2, LPs must decide whether their investment will generate enough fees to cover the risk of adverse selection, but no LP has a competitive advantage over others. This has changed with concentrated liquidity. In the absence of gas fees - i.e., when there are no costs associated

with setting up or rebalancing a liquidity position - each Uniswap v3 LP could potentially center their liquidity around the current price, rebalancing their position as the price changes, similarly to how market makers in traditional CLOB markets cancel and resubmit quotes as prices fluctuate. High gas fees on the Ethereum blockchain, however, are a known problem, and these costs may actually give large LPs a potential advantage over smaller LPs in Uniswap v3 markets. A large LP, for which gas fees may constitute a relatively small fraction of their investment, will be able to more frequently rebalance their position, ensuring that their liquidity is always generating fees. For smaller LPs, though, frequent rebalancing may be prohibitively expensive, and they may need to provide liquidity within a larger range to ensure that their positions continue to generate fees as prices fluctuate. However, by spreading their capital over a larger range of prices, their proportion of fees earned at any one price will be smaller.

In this section, we explore the returns to LPs in Uniswap v3. We begin by discussing how concentrated liquidity has changed the level and shape of liquidity across the space of prices, which affects the amount of price impact that traders experience. Then, using simulations, we explore how LP returns vary by the number of liquidity traders in the market, the size of such liquidity trades, the dynamics of the true market price, and the level and shape of liquidity provided, as a whole, into the pool. In section 5, we use data from the seven largest Uniswap v3 pools⁹ as of February 2022 to understand whether providing liquidity in Uniswap v3 pools is profitable and whether such pools are in equilibrium.

4.1 Liquidity Curves

In Uniswap v2, all LPs provide liquidity across the whole space of prices, $p \in (0, \infty)$, so the liquidity curve is always flat. The liquidity curves that we observe in Uniswap v3, however, take on various shapes as these curves reflect LPs' combined decisions on how much liquidity to provide for any given price range. Figure 12 presents two such liquidity curves for the WETH/USDC and USDT/WETH pools on January 27, 2022¹⁰. There is a large amount of liquidity centered

⁹We exclude pools that are stablecoin pairs as there is very little price movement and liquidity is tightly centered around the pegged price.

¹⁰We do not show the scale of the y axis for the liquidity curves herein as liquidity, L , varies in scale based on different decimal conventions used in the implementation of each pool, which depends on which tokens are included

around the current price for the WETH/USDC pool above, and the curve roughly resembles a bell curve. Liquidity for the USDT/WETH pool, on the other hand, is skewed to the right of the price, suggesting either that LPs are expecting prices to increase and positioning their liquidity to take advantage of where prices will be in the future or that LPs are not frequently rebalancing their positions.

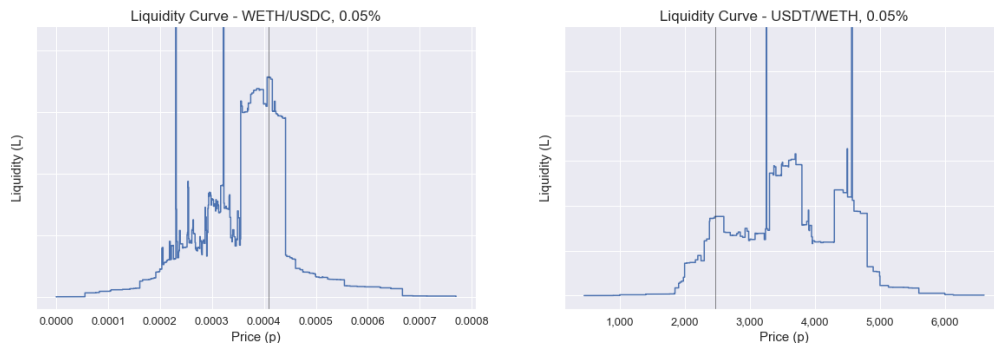


Figure 12: Liquidity curves for the WETH/USDC and USDT/WETH pools at the beginning of January 27, 2022. Black horizontal lines denote each pool's current price.

Both curves also have notable spikes in liquidity. These spikes represent range orders - orders that mimic limit orders in traditional markets by taking advantage of the fact that a position's token composition changes as the pool's price moves from one side of the position to the other. If an LP provides liquidity within a very narrow range to one side of the current price, the position will be entirely composed of a single token. If and when the price crosses the position so that the price is on the other side of the range, the position will now be composed of the other token, and the LP can remove his position having effectively executed a swap around the price at which he set his position. Because an LP is providing liquidity within such a narrow range, even a small investment may significantly increase L within the price range, resulting in the spikes we see in the liquidity curves. There are, however, several differences between range orders and traditional limit orders. First, there is a limit to how narrow a position's range can be as each tick has a width and each pool has a minimum spacing between which ticks can be used to define a position's range; as such, if the price is within that range, the order will only be partially executed (Adams et al., 2021). Second, once the position has been crossed, the LP must withdraw their investment. If they

in the pool and which token is set as `token0` and `token1`. For figures where we show multiple liquidity curves for the same pool, we set the scale of the y axis so that the level of each pool can be observed over time (e.g., Figure 13).

do not, and the price crosses back to the other side of the position, the swap will effectively be reversed (Adams et al., 2021).

Not only do we see variation in liquidity curves across pools, but we also see notable changes in liquidity across time for each pool. Figure 13 below presents the liquidity curves for the WETH/USDC pool across three different days. On July 18, 2021 (left) ETH prices were at one of the lowest points since the introduction of Uniswap v3 in May 2021. We can see that liquidity is skewed to the left of the pool’s current price, suggesting that LPs may be expecting the ETH price to increase (`token1` for the WETH/USDC pool is WETH, so the pool’s price generally tracks the inverse of the ETH market price, $1/\text{ETH}$). On November 6, 2021 (middle), ETH prices were at their highest point since Uniswap v3’s introduction. Liquidity is much more concentrated around the current price, but now liquidity is skewed to the right of the price, suggesting that LPs may be expecting the price of ETH to decline. Indeed, in both cases presented here, prices move in the direction where LPs have skewed their liquidity. While these results are purely anecdotal, we would expect that, like CLOBs, liquidity curves provide information about the distribution of future prices. While we do not explore this more in our paper, it is an interesting area for future work. We present liquidity curves for the other six pools examined on the same three days in Appendix C.

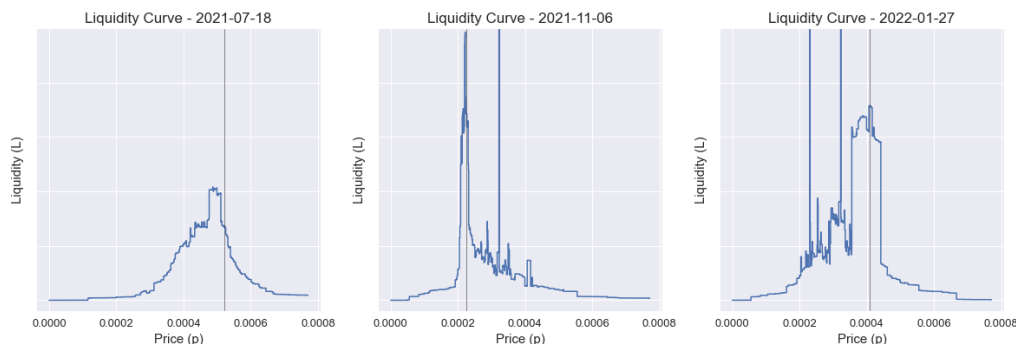


Figure 13: Liquidity curves for the WETH/USDC pool on three different days. Black horizontal lines denote each pool’s current price.

4.2 Liquidity Provider Returns

Finding an optimal LP strategy is difficult in Uniswap v3 markets because of the convexity of the bonding curve, the arbitrary shapes and levels of liquidity that other LPs can create, the complex dynamics of a pool’s price process, and high gas fees. To understand how different variables affect returns for different positions that LPs can create, we perform a number of simulations aimed at illustrating the kinds of factors that contribute to returns and that LPs should consider when devising a provisioning strategy. We structure our simulations as follows:

1. We assume that the market price, m , follows a geometric Brownian motion process with drift, μ , and volatility, σ . At the beginning of each simulation, the pool’s price equals the beginning market price which we set at $p = m = 100$. We continue to use `token1` as our numeraire asset and `token0` as the risky asset in defining the price.
2. Each simulation is a single period with a single price innovation in the market price, m . Once the price innovation occurs, arbitrageurs trade until there is no more profit, bringing the pool’s price close to the market price: $p(1 - \gamma) \leq m < \frac{p}{1-\gamma}$ (see our discussion of arbitrage above for details). For all simulations, we use $\gamma = 0.01$ for simplicity.
3. As noted by Lehar and Parlour (2021), if “price innovations arrive with an exogenous intensity, then a higher number of trades in a given interval must correspond to more liquidity trading.” As such, before each price innovation occurs, $n \sim \text{Poisson}(\alpha)$ liquidity traders arrive, buying or selling a stochastic amount $s \sim \text{Exponential}(\frac{1}{\beta})$ ¹¹ of `token1`. Immediately after each liquidity trade, an arbitrageur arrives and trades to take advantage of any price deviations, if possible, similar to our simulations in section 3.
4. Before any trading occurs, LPs provide liquidity creating one of two liquidity curves, shown in Figure 14. The blue liquidity curve is simply constant at some value, $L(p) = l$, within the range $0 < p \leq 200$. This curve is analogous to a Uniswap v2 liquidity curve as, given the parameters we use in our simulations, it is highly unlikely that the pool’s price will exceed 200. The orange curve is a simple bell curve given by, $L(p) = c \text{sech}^2\left(\frac{p-a}{b}\right)$. The constant c

¹¹We parameterize the exponential distribution this way as it has the property that the mean trade size is β .

controls at what level of liquidity the curve peaks at, which we set so that both curves have the same amount of liquidity at the starting price, $p = 100$. In our initial simulations, we set $l = c = 10,000$.

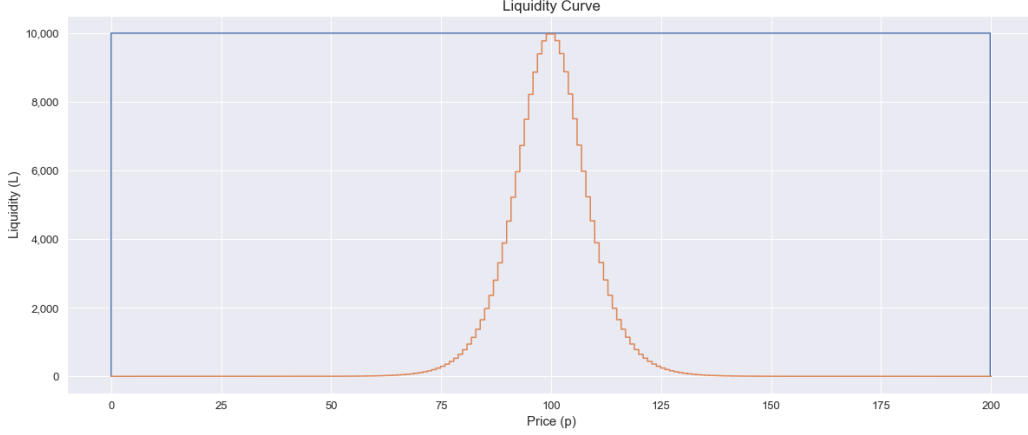


Figure 14: Two liquidity curves used for our return simulations. Both curves are centered at 100, which is the pool’s starting price for all simulations.

The liquidity curves above are constructed as 200, approximately one price unit width, bins; e.g., the first position spans $0 < p < 1$, the second position spans $1 \leq p < 2$, and so on. Note that because of the positioning of Uniswap v3 ticks, no position is exactly one price unit wide; however, since each tick is a 1 basis point movement from its neighbors, we can create positions that are very close to this width within the range $0 < p \leq 200$. We record the initial value of each position in terms of the numeraire token: $V(p) = p\Delta X + \Delta Y$, where $\Delta X, \Delta Y$ are the amounts of `token0`, `token1`, respectively added to the pool to create the position. With the exception of the position that spans $p = 100$, each position is composed entirely of a single token: `token1` for all positions to the left of $p = 100$ and `token0` for all positions to the right of $p = 100$.

5. Once all liquidity traders and arbitrageurs have traded, including arbitrage from the single price innovation, the simulation ends. We calculate the total return to each position, including fees earned and any change in value due to disadvantageous rebalancing of the position from adverse selection, and the new market price. Because the ending pool price, p' , may deviate from the ending market price, m' , due to the arbitrage induced price bounds, we use m' to

calculate the total value of the position and its returns. We then average returns over 10,000 simulations to estimate the expected returns for each position.

We start with an initial simulation with $l = c = 10,000, \mu = 0, \sigma = 0.1, \alpha = 50, \beta = 1,000$. The mean returns for each position are plotted in Figure 15 below. This graph allows us to see how returns vary depending on both the level of liquidity provided and on where an LP provides liquidity relative to the beginning price. Since our simulation only has a single price innovation, our simulations are akin to relatively short term returns. Figure 15 has several notable characteristics. First, returns are highest around the pool's starting price. This is unsurprising as most trades occur around this price and each trade generates fees for the positions. Even though the liquidity trades move the pool's price away from the initial price, arbitrageurs subsequently move the price back, generating additional fees for some positions. The dashed black lines represent the theoretical pool price bounds due to arbitrage, $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$, discussed in section 3.

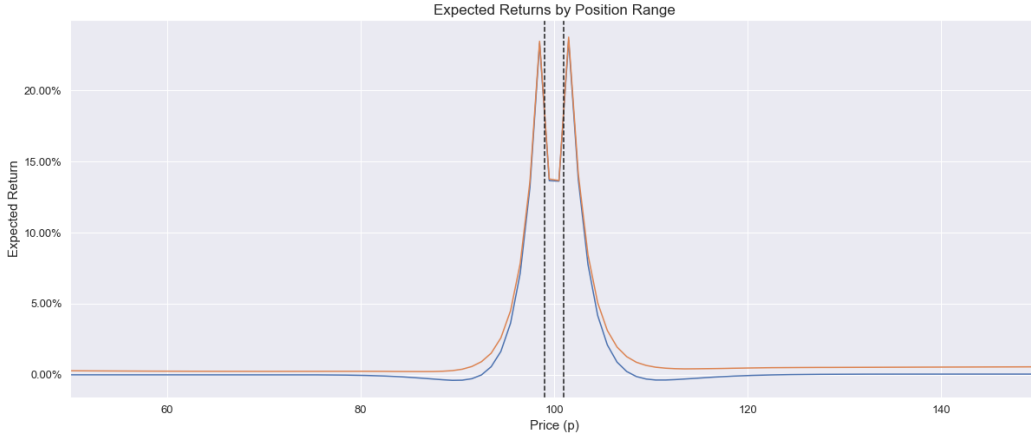


Figure 15: Expected returns for the base simulation. The orange line presents the returns to the bell shaped liquidity curve; the blue line presents the returns to the constant liquidity curve. The dashed black lines represent the theoretical pool price bounds due to arbitrage, $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$.

We can see that fees are actually lower in this range as arbitrageurs do not move the price all the way to $p = 100$, but to the edge of the theoretical bound. Returns from fees decrease exponentially as we move away from the pool's initial price. Because the size of each liquidity trade, s , is exponentially distributed, liquidity trades move the price by varying magnitudes. Most liquidity trades are small with relatively little price impact, so fewer positions participate in a swap and earn fees; however,

there are also larger trades that have a significant price impact, in which many positions along the space of prices earn fees.

Unlike the constant liquidity curve, liquidity for the bell-shaped curve decreases as we move away from the initial price, resulting in a larger price impact for each swap. The orange curve in Figure 15, which correspond to the orange, bell-shaped liquidity curve in Figure 14, has consistently higher returns than the blue curve, which corresponds to the constant liquidity curve. The reason for this is twofold. First, the larger price impact results in more positions participating in the swaps, spreading out fees further than if there was constant liquidity. Second, positions away from the initial price require less capital to establish compared to their constant liquidity counterparts, so, given a fixed amount of fees for two positions spanning the same price range, the returns to the bell curve position will be larger than the returns to the constant liquidity position.

As we move away from the space of prices that m is likely to end at, positions are less likely to earn fees or experience negative rebalancing. As such, positions far to the left of the initial price act simply like long positions in **token1**, which as the numeraire have returns of 0%, and positions far to the right of the initial price act like long positions in **token0**, which are exposed changes in the market price. This is apparent from Figure 7 above and is easy to show using equation (19). As long as $p < p(i_l)$, the value of the position changes linearly with p . However, once $p \geq p(i_u)$, the position's value is constant, and there is no upside as prices increase further. For a position whose range is consistently below the pool's price, $p \geq p(i_u) \forall p$, so that the position is composed entirely of **token1** and does not earn fees, the return on the position is:

$$\frac{V(p')}{V(p)} - 1 = \frac{\Delta L(\sqrt{p(i_u)} - \sqrt{p(i_l)})}{\Delta L(\sqrt{p(i_u)} - \sqrt{p(i_l)})} - 1 = 0 \quad (22)$$

Conversely, for a position whose range is consistently above the pool's price, $p < p(i_l) \forall p$, so that the position is composed entirely of **token0** and does not earn fees, the return on the position is:

$$\frac{V(p')}{V(p)} - 1 = \frac{\Delta L p' \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right)}{\Delta L p \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right)} - 1 = \frac{p'}{p} - 1 \quad (23)$$

Note that, for simplicity, we assume that $m' = p'$ above. Because prices follow a geometric Brownian motion process, we know that the ending price $m' \sim \text{Lognormal}((\mu - \frac{\sigma^2}{2})t, \sigma^2 t)$; with $\mu = 0, \sigma = 0.1$, the expected return is 0, which is consistent with the results in Figure 15.

While our initial simulation is interesting in its own right, understanding how returns vary as different variables vary provides much richer information on the dynamics of Uniswap v3 liquidity positions. Next, we perform the same simulation, except we vary α , the expected number of liquidity traders, and β , the expected size of the liquidity trade. Note that we use the same random seed for all simulations discussed in this section in order to isolate the effects of changing certain variables.

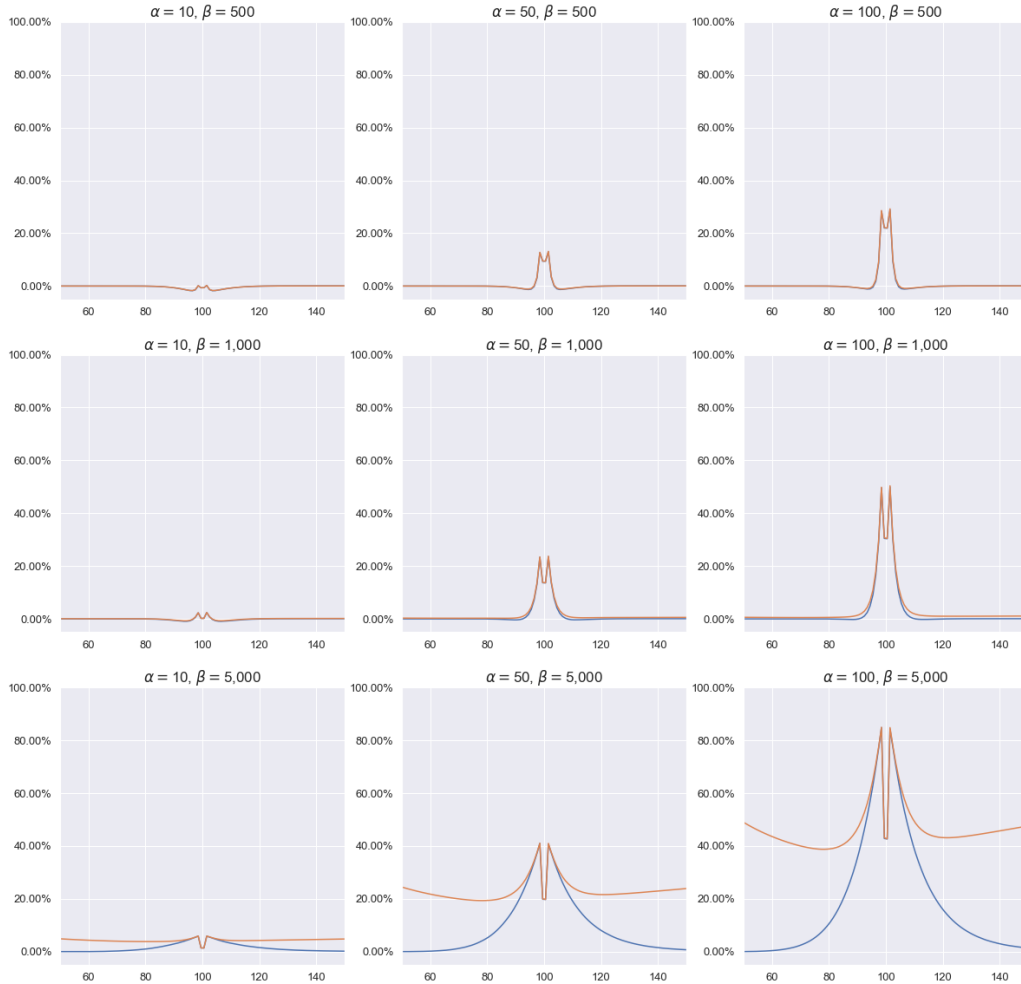


Figure 16: Expected returns for varying expected number of liquidity trades, α , and expected liquidity trade size, β . The orange line presents the returns to the bell shaped liquidity curve; the blue line presents the returns to the constant liquidity curve.

Figure 16 presents the simulation results for combinations of $\alpha \in \{10, 50, 100\}$ (left/right) and $\beta \in \{500, 1000, 5000\}$ (up/down). For a given β , increasing the expected number of liquidity trades, α , has the effect of uniformly increasing the level of returns. While more liquidity trades will increase the total fees earned, if the size of the trades is fixed, the distribution of fees remains unchanged. Alternatively, for a given α , increasing the expected size of each liquidity trade, β , both increases total fees, resulting in a level increase in returns, and results in more positions participating in each swap, expanding the number of positions that earn fees and, in turn, experience positive returns. These effects are also clearly additive, as can be seen from the increase in expected returns as we move from the upper left simulation ($\alpha = 10, \beta = 100$) to the lower right simulation ($\alpha = 100, \beta = 5000$).

For the next set of simulations, we reset α, β to their original values and vary the drift and volatility parameters of the price process: μ, σ . Figure 17 presents the simulation results for combinations of $\mu \in \{-0.05, 0.0, 0.05\}$ (up/down) and $\beta \in \{0.05, 0.1, 0.2\}$ (left/right). The effect of changes in the drift of m is the most apparent. Positions far to the right of the initial price either experience notably negative, flat, or positive returns depending on whether there is negative, zero, or positive drift, respectively; as discussed above, these positions are akin to long positions in `token0`. The effects of volatility are more nuanced, but still consequential. As σ increases, we can see the positions around the center (around $p = 90, p = 110$) begin to become concave and even negative. Volatility has a direct effect on the adverse selection experienced. Since each position for our simulation is only one price unit wide, almost every position that experiences disadvantageous rebalancing from adverse selection is rebalanced so that the token composition shifts entirely from one token to the other. As such, we can use Figure 7 and equation (19) above to understand how such adverse selection impacts returns in the absence of fees.

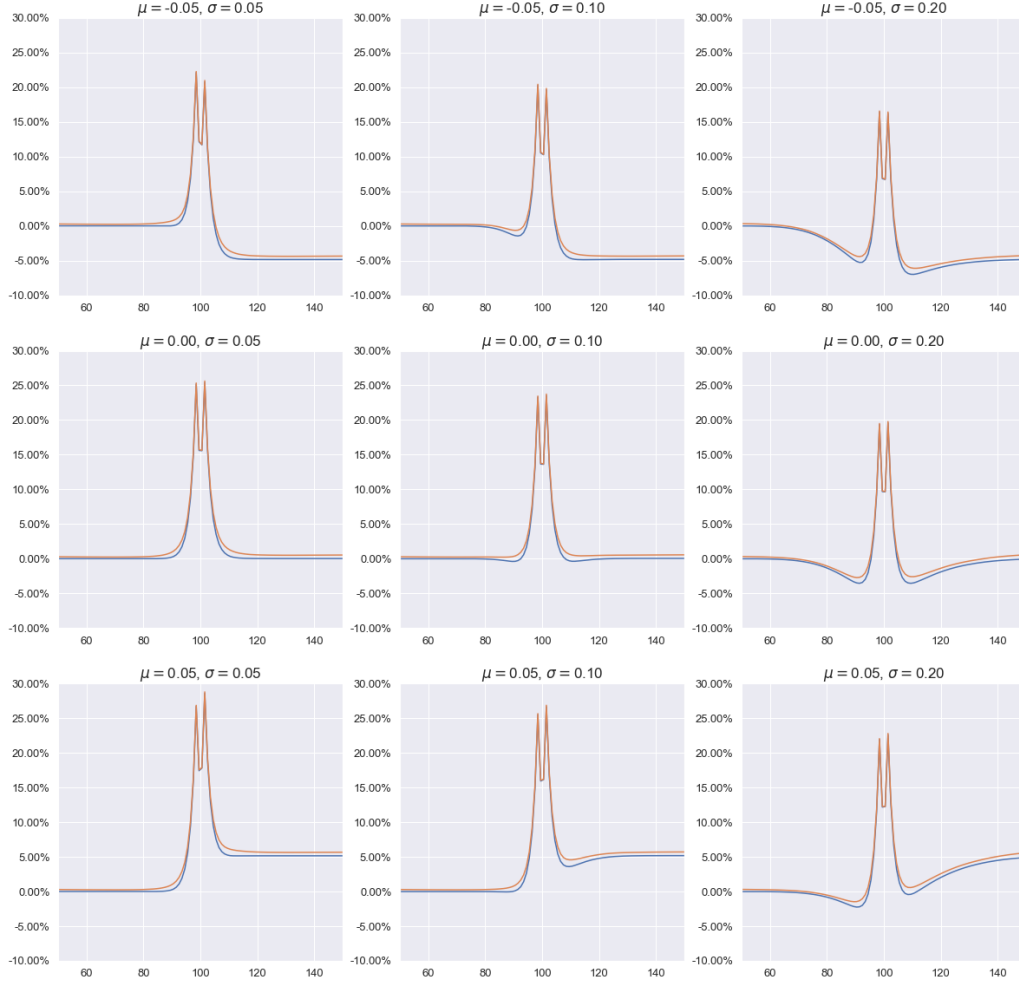


Figure 17: Expected returns for varying drifts, μ , and volatilities, σ , for the market price process. The orange line presents the returns to the bell shaped liquidity curve; the blue line presents the returns to the constant liquidity curve.

Let us first consider a position that spans a set of prices above the current price, $p < p(i_l)$. If prices decrease, so that $p' < p(i_l)$, the position's composition remains the same and the value of the position changes linearly with p , as described in equation (23). If the price increases from $p < p(i_l)$ to $p' \geq p(i_u)$, the position is rebalanced from 100% `token0` to 100% `token1`. We can see in Figure 7 that this price movement pushes the value from the range where $V(p)$ changes linearly with p ($p < p(i_l)$) to where $V(p)$ is constant ($p' \geq p(i_u)$). As such the return from this price movement is:

$$\frac{V(p')}{V(p)} - 1 = \frac{\Delta L(\sqrt{p(i_u)} - \sqrt{p(i_l)})}{\Delta L p \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right)} - 1 = \frac{\sqrt{p(i_u)p(i_l)}}{p} - 1 \quad (24)$$

In our case, where $p(i_l) \approx p(i_u)$, we can see that the upside is limited to approximately $\frac{p(i_u)}{p} - 1$. This limited upside is a direct cause of the negative returns for positions around $p = 110$. As volatility increases and large price declines become more likely, these positions experience larger losses. However, since they have limited upside, they do not participate in the larger potential gains from higher volatility. Therefore, in expectation, their returns are negative.

Next, let us consider a position that spans a set of prices below the current price, $p \geq p(i_u)$. If prices increase, $p' < p(i_l)$, the position's composition remains the same and the value of the position is constant, as described in equation (22). On the other hand, if the price decreases from $p \geq p(i_u)$ to $p' < p(i_l)$, the position is rebalanced from 100% `token1` to 100% `token0`. We can see in Figure 7 that this price movement pushes the value from the range where $V(p)$ is constant and at it's highest point ($p \geq p(i_u)$) to where $V(p)$ changes linearly with p ($p' < p(i_l)$). As such the return from this price movement is:

$$\frac{V(p')}{V(p)} - 1 = \frac{\Delta L p' \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right)}{\Delta L (\sqrt{p(i_u)} - \sqrt{p(i_l)})} - 1 = \frac{p'}{\sqrt{p(i_u)p(i_l)}} - 1 \quad (25)$$

Because we only see returns of this form when prices decline and the position is negatively rebalanced, this return is bounded between $(-1, 0)$. The limited upside coupled with notable downside risk is the direct cause of the negative expected returns for positions around $p = 90$. As σ increases and large price declines become more likely, these positions experience larger losses. However, these positions begin at their maximum value as $p \geq p(i_u)$, so there is no upside whatsoever. This, again, results in negative returns in expectation. While positions between $90 < p < 110$ are also exposed to adverse selection risk, the fees generated by these positions more than offset the downside from negative rebalancing and disadvantageous price movements, resulting in positive returns in expectation.

While LPs can estimate the number and size of liquidity trades and the dynamics of the market price process, they do not directly control these variables¹². LPs do, however, determine the level of liquidity in a pool. If an LP thinks a pool is too crowded, they can decide not to allocate

¹²Theoretically, more liquidity in a pool may attract more liquidity traders as traders will experience less price impact.

their capital to the pool. In our next set of simulations, we revert back to our baseline values, $\mu = 0, \sigma = 0.1, \alpha = 50, \beta = 1,000$, but now vary the amount of liquidity in the pool by changing l, c . Figure 18 shows the return plots for $l, c \in \{10000, 25000, 50000\}$. Increasing liquidity has a profound impact on positions' returns. First, with more liquidity, there is less price impact, so fewer positions participate in a swap and earn fees, tightening the range of prices for which returns are positive. In fact, as illustrated in the right graph, if liquidity is high enough, a swap will not even move the price outside of the theoretical price bounds, which removes the dip in returns at $p = 100$ that we see in other plots. Second, for a fixed number of liquidity trades, increasing liquidity results in positions earning the same amount of fees on a larger amount of capital invested, which simply decreases fee returns. Because the ending market price, m' , is unaffected by the level of liquidity, increasing liquidity only reduces fee returns; it does not reduce a position's adverse selection risk. As such, too much liquidity in a pool can result in very few LPs earning sufficient fees to cover the costs of disadvantageous rebalancing from adverse selection.

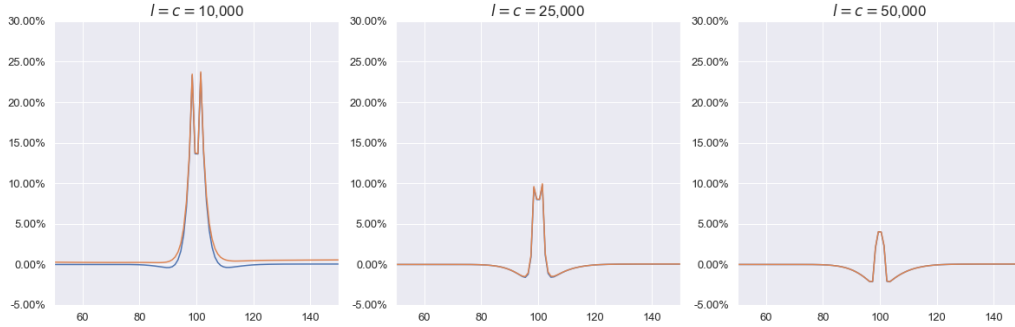


Figure 18: Expected returns for varying levels of liquidity, l, c . The orange line presents the returns to the bell shaped liquidity curve; the blue line presents the returns to the constant liquidity curve.

Even for $l, c = 50,000$, there are some positions that earn positive returns due to the fees generated around the initial price. What happens if we increase liquidity even further? Figure 19 plots the results of the same simulations with $l, c = 100,000$ and $l, c = 200,000$. At $l, c = 100,000$, only positions within the theoretical price bounds earn fees as there is so much liquidity that the price impact of liquidity trades does not move the price outside of the bounds; i.e., liquidity trades are not large enough to push the price outside this range. Interestingly, because the price does not move outside of the theoretical bounds, arbitrageurs do not trade immediately after each liquidity trade, which actually reduces the total number of swaps that occur, further reducing the

fees generated. With the exception of positions far from the initial price, which again earn either zero returns or simply the return on m , all other positions experience negative returns as adverse selection easily outweighs the fees generated. At $l, c = 200,000$, there is now so much liquidity that even positions at the initial price do not earn sufficient fees to outweigh adverse selection.

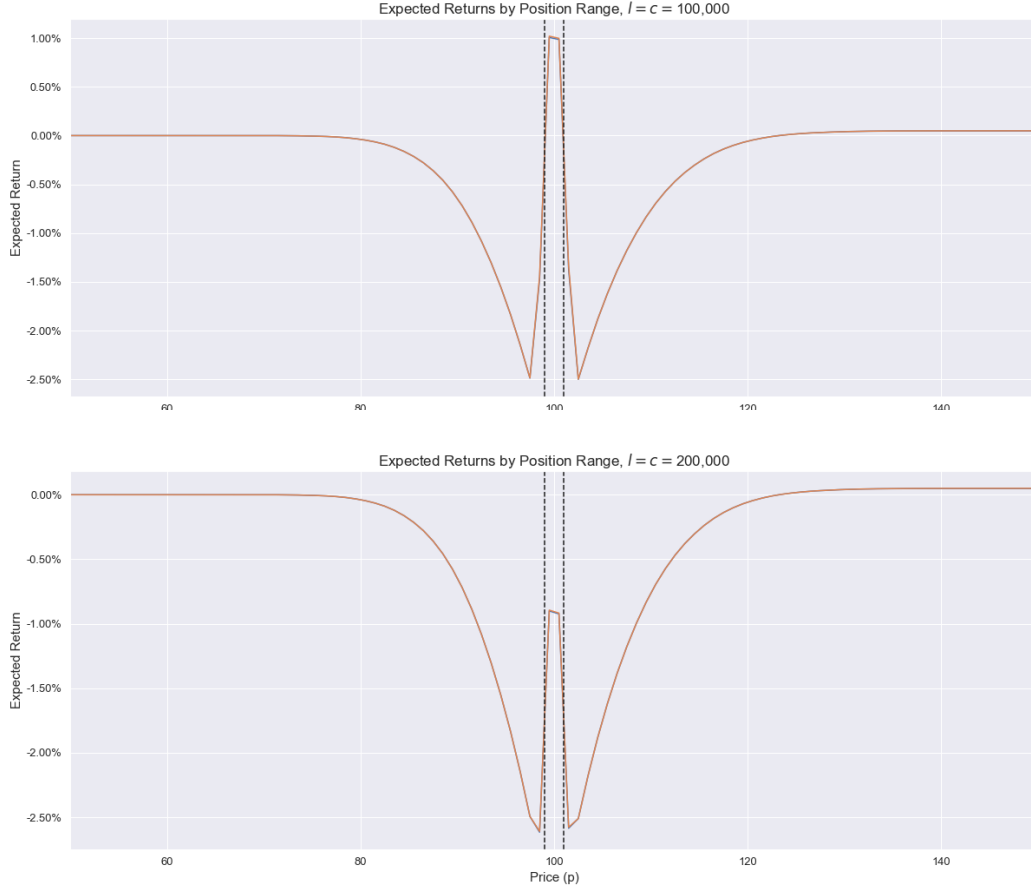


Figure 19: Expected returns for varying levels of liquidity, l, c . The orange line presents the returns to the bell shaped liquidity curve; the blue line presents the returns to the constant liquidity curve. Dashed black lines represent the theoretical pool price bounds due to arbitrage, $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$.

In all of our simulations so far, fee income peaks around the initial price as there is only one price innovation. However, in reality, the market price continually fluctuates - we have many small price innovations instead of large, infrequent price innovations. As such, we perform the same simulations except now we allow for five smaller price innovations before the simulation ends. To make these next simulations comparable to the previous simulations, we keep the length of the simulation constant but adjust our parameters. We divide α and μ by five so that our simulation has the

same total number of liquidity trades and the same total drift. We divide σ by $\sqrt{5}$ (as volatility scales with the square root of time) so that m' takes on the same distribution as the original simulations. Figure 20 shows our results of these simulations for $l, c = 10,000$ and $l, c = 200,000$. For $l, c = 10,000$, the low liquidity case, fees earned are now more spread out, which then spreads out returns. We still see the same effect on returns within the arbitrage price bounds as there are at least a few liquidity trades that occur around the initial price, but the effect is far less pronounced.

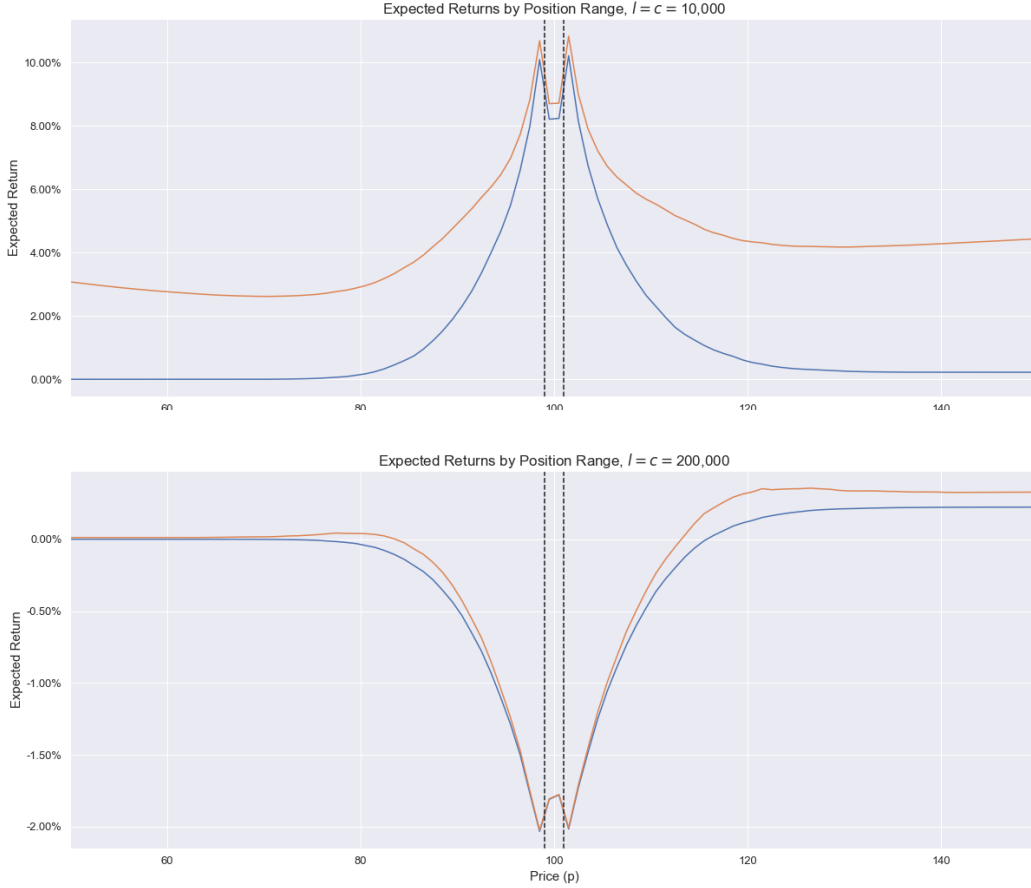


Figure 20: Expected returns for varying levels of liquidity, l, c , for a simulations with multiple price innovations. The orange line presents the returns to the bell shaped liquidity curve; the blue line presents the returns to the constant liquidity curve. Dashed black lines represent the theoretical pool price bounds due to arbitrage, $p(1 - \gamma) \leq m < \frac{p}{1 - \gamma}$.

We see similar results for $l, c = 200,000$, but returns are notably negative. As expected, if the market price continually moves, centering liquidity at the current price provides fewer benefits as the price may quickly move away from the position. Again, this suggests that rebalancing is extremely important for capturing sufficient fee income to overcome the risk of adverse selection.

Unfortunately, with high gas, such rebalancing can also be very expensive and result in the need to generate even more fee income to produce positive returns.

5. Empirical Findings

Armed with a better understanding of the dynamics of LP returns, we now examine returns for actual Uniswap v3 pools. Our goal is to understand whether these pools result in positive returns for LPs and whether these pools are in equilibrium. In equilibrium, the aggregate amount and position of liquidity (i.e., the level and shape of the liquidity curve) is such that liquidity suppliers are indifferent between committing more capital within any tick in the pool or to their outside option (Lehar and Parlour, 2021). As such, if a pool is in equilibrium, we would expect that returns to each tick, in expectation, would be approximately equal (and which we assume will be zero). Such a zero-profit condition assumes that there are very low costs to enter or exit a position. While Ethereum gas fees may be high, they do not present a notable barrier to entry or exit, so characterizing equilibrium in this way is appropriate. Lehar and Parlour (2021) show that the equilibrium size of a Uniswap v2 pool balances fee revenue against the negative rebalancing of adverse selection. This holds for Uniswap v3 as well, but now we expect it to hold within each tick. If any particular tick experiences positive (negative) returns, LPs could add (remove) liquidity to (from) that range until the return moves to zero. Consistent with Aoyagi (2020), our simulations above suggest that the equilibrium amount of liquidity at any tick is negatively related to the volatility of the risky token, which affects adverse selection, and positively related to the number and size of liquidity trades, which affects fee revenue.

We examine pools with more than \$50 million in weekly volume as of January 2022 that are not stablecoin pairs, for which liquidity is tightly bound around 1.0. This results in 7 pools, 6 of which are WETH/stablecoin pools with fees of either 0.3% or 0.05%, the last of which is a WBTC/stablecoin pool with a fee of 0.3%. We collected data, using the Covalent Unified API ¹³, on all liquidity transactions and swaps from each pool’s inception through the end of January 2022. Summary statistics for each of the seven pools is presented in table 1 below. While there is some variation among pools, we can see that, given a fee rate, pools with more trading result in more

¹³<https://www.covalenthq.com/>

fees and generally have higher Total Value Locked (TVL), which measures the amount of liquidity locked into the pool. For each pool, total fees to TVL varies between approximately 0.3% to 1.25%, and it is reasonable to expect that LPs in pools with higher fee/TVL ratios may experience higher returns as fees are earned on a relatively smaller base of capital.

Pair (token1/token0)	Fee Tier	# of Trades	Volume (USD)	Fees	TVL (USD)	Fees/TVL
WETH/USDC	0.05%	41,651	5.07b	2.53m	203.64m	1.24%
WETH/USDC	0.30%	2,835	466.43m	1.40m	238.43m	0.58%
USDT/WETH	0.05%	17,956	771.06m	385.53k	37.99m	1.01%
USDT/WETH	0.30%	2,117	208.84m	626.51k	115.52m	0.54%
WETH/DAI	0.05%	5,339	142.34m	71.17k	14.87m	0.47%
WETH/DAI	0.30%	1,334	76.64m	229.93k	60.38m	0.38%
USDC/WBTC	0.30%	1,716	246.45m	739.34k	94.17m	0.78%

Table 1: Summary statistics for the 7 pools examined for 1st week of March 2022.
(Source: <https://dune.xyz/gammastrategies/Uniswap-v3-Volume-and-Fees-Collected>)

With this data and our Uniswap v3 simulator, we are able to approximately recreate each pool at any point in time. There are two limitations that do not allow us to exactly recreate the pool. The first and more significant limitation is that time stamps for transactions are only precise to the second. If multiple transactions occur in the same second, it is not possible¹⁴ to determine which transaction occurred first. However, order matters in Uniswap pools: each time a swap occurs the price changes, which determines how much of each token an LP must provide for a given level and range of liquidity; each time an LP changes its position, liquidity changes, which changes swap dynamics and the price impact of each swap. Even though we are not able to exactly replicate the pools, we found that the pools’ prices all tracked the relevant reference prices obtained from other data sources, so we do not believe such approximation error has a significant impact on our results. Our Uniswap v3 simulator uses floating-point precision for all calculations while the actual Uniswap smart contracts all use integer arithmetic. The use of floating-point precision introduces another source of error in our approximations; however, we found that this error was immaterial.

If the liquidity provided to a pool is in equilibrium, then, by the zero profit condition discussed above, we would expect that returns to each tick range would be approximately equal to zero in expectation. To test whether we see this in the pools in practice, we estimate the daily expected

¹⁴Theoretically, we could determine the correct order of transactions through trial and error by validating that our pool’s state matched the pool’s state recorded on the Ethereum blockchain. This would require additional data on the pool’s state after each transaction, which we were not able to obtain.

return to providing liquidity in different areas around the pool's initial price. For each day, we created 22 bins around the pool's initial price, p_0 . 20 of the bins are each 1σ wide, where σ is the approximate daily volatility for each asset token, WETH and WBTC, both of which have daily volatility of $\sigma \approx 0.04$, which is what we use all of our experiments. For example, the first few bins around the initial price would be $[p_0(1 - 2\sigma), p_0(1 - \sigma))$, $[p_0(1 - \sigma), p_0)$, $[p_0, p_0(1 + \sigma))$, $[p_0(1 + \sigma), p_0(1 + 2\sigma))$. The last two bins span the areas outside $\pm 10\sigma$: $(0, p_0(1 - 10\sigma))$ and $(p_0(1 + 10\sigma), \infty)$. Figure 21 illustrates the bins used for the WETH/USDC liquidity curve for a single trading day.

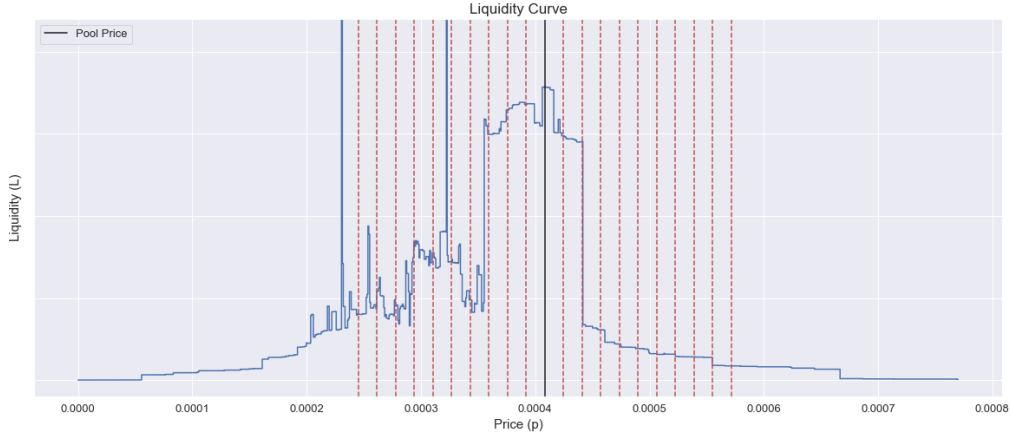


Figure 21: Liquidity bins for the WETH/USDC pool on January 27, 2022.

We calculated the daily return to each bin as the money-weighted rate of return (MWRR) between the beginning of the day and the end of the day. The MWRR takes into account cash flows that occur during the day, which is important as LPs may rebalance their positions during the day. The MWRR is effectively the internal rate of return, which is the return, r , that results in the net present value (NPV) of a series of cash flows equaling zero:

$$\text{NPV} = \sum_{t=0}^n \frac{\text{CF}_t}{(1+r)^t} = 0$$

To calculate the MWRR, we assume that the liquidity curve is constructed at the beginning of the day, which results in the initial negative cash flow at time $t = 0$, and completely divested at the end of the day, including collecting fees earned, which results in the last positive cash flow at $t = 1$. The only cash flows intraday are due to LPs adding or removing liquidity. The daily returns to the

22 bins are then averaged over the life of the pool to arrive at the estimated expected values.

We present the expected returns, along with their 95% confidence intervals, for the USDT/WETH pool in Figure 22 below and a condensed table of results for all pools in table 2; similar graphs for the other pools are included in appendix D, but all curves have a similar shape and similar properties.

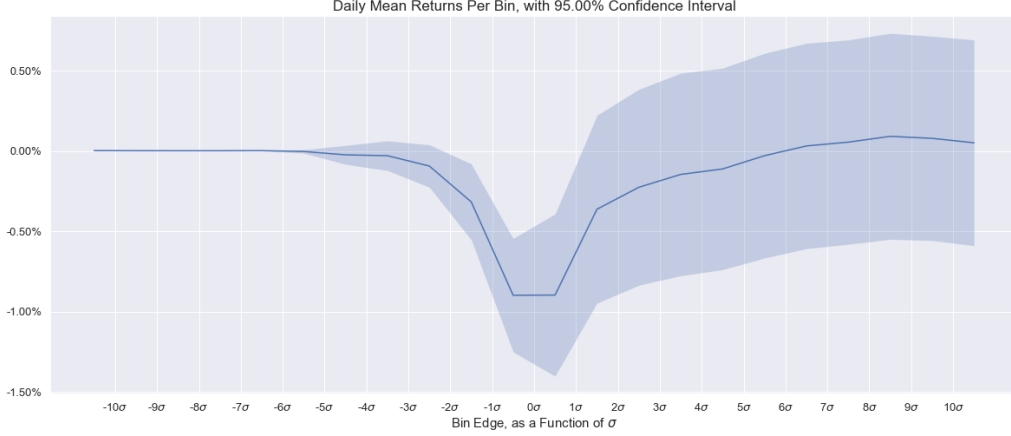


Figure 22: Mean daily returns for the USDT/WETH pool along with the 95% confidence interval.

	$[-4\sigma, -3\sigma)$	$[-3\sigma, -4\sigma)$	$[-2\sigma, -4\sigma)$	$[-1\sigma, 0\sigma)$	$[0\sigma, 1\sigma)$	$[1\sigma, 2\sigma)$	$[2\sigma, 3\sigma)$	$[3\sigma, 4\sigma)$
WETH/DAI, 0.3%	-0.04%	-0.08%	-0.19%	-0.53%	-0.53%	-0.22%	-0.04%	-0.04%
WETH/DAI, 0.05%	0.25%	-0.17%	-0.50%	-1.19%	-1.11%	-0.74%	-0.53%	-0.47%
WETH/USDC, 0.3%	-0.04%	-0.09%	-0.31%	-0.84%	-0.73%	-0.17%	0.03%	0.03%
WETH/USDC, 0.05%	-0.15%	-0.29%	-0.52%	-0.95%	-1.02%	-0.34%	0.18%	-0.09%
USDT/WETH, 0.3%	-0.07%	-0.11%	-0.24%	-0.75%	-0.81%	-0.31%	-0.19%	-0.15%
USDT/WETH, 0.05%	-0.03%	-0.09%	-0.32%	-0.90%	-0.90%	-0.36%	-0.23%	-0.15%
USDC/WBTC, 0.3%	0.02%	0.03%	-0.09%	-0.44%	-0.52%	-0.17%	-0.10%	-0.10%

Table 2: Daily expected returns for different bins of liquidity relative to the beginning price for the day for each pool.

Returns to LPs that provide liquidity around the current price, $[p_0(1 - \sigma), p_0(1 + \sigma))$, are on average negative; we see this in all 7 pools examined. In fact, the shape of the empirical mean daily returns is remarkably similar to the $l, c = 200,000$ simulated expected returns in Figure 20, suggesting that the reason for such negative expected returns is too much liquidity around the current price. The majority of transaction fees are generated around the pool's current price, so LPs will center their positions in this range in order to maximize their fee revenue. However, as more LPs add liquidity to this range, each LP's share of the fees generated becomes smaller. If too many LPs add liquidity to the range, the fees earned will not be enough to offset potential

depreciation in their positions’ values due to negative rebalancing from adverse selection. While such negative returns are somewhat surprising, our results are consistent with the results of Loesch et al. (2021). Loesch et al. (2021) find that, for the 17 pools they examine, roughly 40% to 60% of LPs incur negative returns, depending on the pool, and that in aggregate 49.5% of all LPs incur negative returns. Further, the authors find that LPs that contribute more capital are more likely to experience negative returns, suggesting that an LP’s size may not result in any competitive advantage as originally hypothesized.

For all pools, the standard error of returns either grows from zero or shrinks to zero as we move from left to right of the current price. Whether the volatility in returns increases or decreases as we move left to right is determined by which token, `token0` or `token1`, for the pool is the stablecoin and our numeraire token. While mean daily returns for positions entirely denominated in the risky asset (WETH or WBTC in the pools we examine) may be close to zero, these positions are exposed to notable volatility. In practice, it is unlikely that an LP would create or leave intact a tight position that is far away from the current price. However, this suggests that many LPs are creating positions that span a relatively large range of prices. Unlike a CLOB market where market makers can freely submit and cancel orders, Uniswap LPs incur gas fees whenever they update their positions. Lehar and Parlour (2021) suggest that, in Uniswap v2, Ethereum gas fees act as a “commitment device” for LPs to remain in a pool. Similarly, gas fees may act like such a commitment device for LPs to stay within any given position. Because rebalancing is costly and volatility for WBTC and WETH is relatively high, it is understandable that LPs may create positions that they do not have to rebalance as often to earn fees. Lambert (2021b) suggests that positions with tight ranges kept in for short amounts of time outperform wider positions that are left for longer periods. To actually execute such a strategy, an LP must actively rebalance its position as prices change, which, again, may be prohibitively expensive for smaller investments. Recently, Uniswap has also created pools on the Polygon blockchain, which is compatible with all Ethereum smart contracts but has notably lower gas fees. While these pools are still in their infancy, it will be interesting to see whether liquidity becomes less distributed and is rebalanced more often as gas fees come down.

Loesch et al. (2021) find that 25% of positions last less than a day whereas 50% of positions last for a week or longer. As such, we performed the same procedure for hourly returns and

weekly returns as well as for various values of σ (i.e., different sized bins). The results for all such experiments were effectively the same as the results for daily returns presented herein: negative returns around the current price with positions further from the center acting like long positions in `token0` or `token1`. Loesch et al. (2021) also find that about 5% of LP positions only exist for a single block. These “flash LPs” are actually the only group of LPs that the authors find outperform impermanent loss. This is, in a sense, taking the advice of Lambert (2021b) to an extreme. If an LP can reasonably predict when large trades will occur¹⁵, an LP can add liquidity exactly where the swap will generate fees, collecting the maximum amount of revenue given their capital. By immediately removing this position after the transaction (or after an arbitrageur moves the price back to the true market price), the LP also minimizes the negative rebalancing that occurs, reducing any depreciation in their position’s value.

Surprisingly, we did not find any instances of positive expected returns for positions around the current price, suggesting that these pools simply have too much liquidity relative to the level of liquidity trades and the amount of volatility that these tokens exhibit. We expect that, if LPs experience prolonged negative returns, LPs will exit the pools, decreasing liquidity until such liquidity remaining no longer earns negative returns. Too much liquidity, though, is undoubtedly a good thing for traders, whose trades will experience less price impact. Therefore, we may also expect that this excess liquidity may attract more traders, increasing the fee revenue earned by LPs until the existing LP base begins to experience non-negative returns. Like all markets, if there is too much supply, an increase in demand (e.g., an increase in traders using the Uniswap v3 pools) or a decrease in supply (e.g., existing LPs exiting the Uniswap v3 pools) will help shift the market back towards equilibrium. Given the additional complexity introduced by concentrated liquidity in Uniswap v3, arriving at such an equilibrium may take longer than in the simpler Uniswap v2 pools as researchers are still actively exploring the dynamics of these new AMMs.

6. Conclusion

With the introduction of concentrated liquidity, Uniswap v3 added a layer of complexity to providing liquidity and introduced competition into Uniswap markets. While sophisticated investors

¹⁵Theoretically, an LP could front-run a swap as well.

may be able to take advantage of the additional functionality now available to LPs, earning positive, risk-adjusted returns is nontrivial and an optimal liquidity provisioning strategy is far from clear. As researchers continue to explore Uniswap v3 pools and as LPs better understand the dynamics of their investments, we do expect clearer LP strategies to emerge though, similar to the “flash LPs” identified by Loesch et al. (2021). Further, as Ethereum continues to improve the gas fees associated with transactions and as Uniswap pools are deployed on the Polygon blockchain, we also expect that more sophisticated liquidity provisioning strategies will emerge as the cost of rebalancing a portfolio declines. While such strategies may increase competition among LPs, they may also improve price efficiency to the extent that LPs are able to more effectively deploy their capital in the range that traders demand.

Our results, however, indicate that, currently, the largest non-stable coin pair pools have too much liquidity given the level of liquidity trades and the volatility that the risky asset tokens exhibit, making it very difficult to earn positive returns. As LPs readjust their strategies, potentially even leaving the Uniswap pools, and as DEXs continue to grow in popularity, attracting more traders, we expect that the level and positioning of liquidity will move into equilibrium over time.

Finally, we have discussed multiple avenues that would be interesting areas for future work. First, option pricing theory provides a robust framework for valuing LP positions, and we may be able to use such a framework to model Uniswap dynamics and devise optimal provisioning strategies. Second, Uniswap liquidity curves, like CLOBs, may provide valuable information on future prices that could be used to develop investment strategies. Lastly, Uniswap v3’s deployment on the Polygon blockchain may have already changed how LPs strategically allocate capital, and comparing our results herein to LPs’ returns on the Polygon blockchain could be very informative.

References

Introduction to smart contracts, 2022. URL

<https://ethereum.org/en/developers/docs/smart-contracts/>.

Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 core, 2020. URL

<https://uniswap.org/whitepaper.pdf>.

Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core, 2021. URL <https://uniswap.org/whitepaper-v3.pdf>.

Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets, 2021.

Jun Aoyagi. Liquidity provision by automated market makers. *ERN: Other IO: Theory (Topic)*, 2020.

Denis Beau. Market-making and proprietary trading: industry trends, drivers and policy implications. 2014. URL <https://www.bis.org/publ/cgfs52.pdf>.

Peter Carr, Morgan Stanley, and Dilip Madan. Towards a theory of volatility trading. In *Reprinted in Option Pricing, Interest Rates, and Risk Management, Musiella, Jovini, Cvitanic*, pages 417–427. University Press, 1998.

Joseph Clark. The replicating portfolio of a constant product market with bounded liquidity, August 3 2021. URL <https://ssrn.com/abstract=3898384>.

Robin Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. 2002. URL <https://mason.gmu.edu/~rhanson/mktscore.pdf>.

Guillaume Lambert. Understanding the value of uniswap v3 liquidity positions, July 14 2021a. URL <https://lambert-guillaume.medium.com/cdaaee127fe7>.

Guillaume Lambert. A guide for choosing optimal uniswap v3 lp positions, part 2, September 15 2021b. URL <https://lambert-guillaume.medium.com/4a94b0a12886>.

- Guillaume Lambert. Calculating the expected value of the impermanent loss in uniswap, October 1 2021c. URL <https://lambert-guillaume.medium.com/bfbfebbefed2>.
- Alfred Lehar and Christine A. Parlour. Decentralized exchanges. *Investments eJournal*, 2021.
- Stefan Loesch, Nate Hindman, Mark B Richardson, and Nicholas Welch. Impermanent loss in uniswap v3. *arXiv preprint arXiv:2111.09192*, 2021.
- Vijay Mohan. Automated market makers and decentralized exchanges: a defi primer. 2022. URL <https://jfin-swufe.springeropen.com/track/pdf/10.1186/s40854-021-00314-5.pdf>.
- Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
- Michael Neuder, Rithvik Rao, Daniel J. Moroz, and David C. Parkes. Strategic liquidity provision in uniswap v3, 2021.
- Pintail. Uniswap: A good deal for liquidity providers?, July 11 2019. URL <https://pintail.medium.com/104c0b6816f2>.
- Y. Zhang, X. Chen, and D. Park. Formal specification of constant product ($xy=k$) market maker model and implementation. 2018.

Appendix A Splitting Trades is Equivalent to a Single Trade

Angeris et al. (2021) show that splitting a trade is always more expensive in the presence of proportional transactions fees and zero gas fees in Uniswap v2 pools. This is not the case in Uniswap v3 pools because transaction fees are not added to the pool's reserves after a swap. We prove this below, following a similar procedure as Angeris et al. (2021), for the simple case where the swap is small enough so that L is constant for the entire range of the swap.

We show that two sequential swaps, $\Delta x, \frac{\Delta y}{1-\gamma}$ and $\Delta x', \frac{\Delta y'}{1-\gamma}$, are equivalent to a single swap, $\Delta x^{\text{total}}, \frac{\Delta y + \Delta y'}{1-\gamma}$, in the absence of gas fees and constant liquidity, L . Since we are swapping in **token1**, the amount of **token1** added to the pool after fees are set aside is $\frac{\Delta y}{1-\gamma}(1-\gamma) = \Delta y$, $\frac{\Delta y'}{1-\gamma}(1-\gamma) = \Delta y'$. The first swap satisfies:

$$(x + \Delta x)(y + \Delta y) = L^2$$

This swap updates the pool's price from p to $p' = \frac{y + \Delta y}{x + \Delta x}$, so the next swap then satisfies:

$$\begin{aligned} (x + \Delta x + \Delta x')(y + \Delta y + \Delta y') &= L^2 \\ \implies \Delta x + \Delta x' &= \frac{L^2}{y + \Delta y + \Delta y'} - x \end{aligned}$$

If a trader swaps in $\frac{\Delta y + \Delta y'}{1-\gamma}$ in a single trade, then $\frac{\Delta y + \Delta y'}{1-\gamma}(1-\gamma) = \Delta y + \Delta y'$ is added to the pool, and the swap must satisfy:

$$\begin{aligned} (x + \Delta x^{\text{total}})(y + \Delta y + \Delta y') &= L^2 \\ \implies \Delta x^{\text{total}} &= \frac{L^2}{y + \Delta y + \Delta y'} - x = \Delta x + \Delta x' \end{aligned}$$

This result also holds in the case of non-constant liquidity. When a swap is executed over a price range where liquidity changes, the swap is split into multiple parts and executed over each range where liquidity is constant and then summed together. As long as the pool's reserves do not change between two sequential swaps - which only happens when LPs add or remove liquidity as fees are not added to a pool's reserves - sequentially swapping is the same as a single swap in Uniswap v3.

Appendix B Price Bound Simulation Results

Below we presented the simulated price bounds for three different price series (left/right) and three different increasing liquidity curves (up/down), $L(p) = 10p + 4,000$, $L(p) = 25p + 2,500$, and $L(p) = 50p + 0$, with $\gamma = 0.03$, as discussed in section 3. While we also ran these simulations for three different decreasing liquidity curves, the resulting graphs are identical, illustrating that the price bounds are the same whether liquidity is increasing or decreasing around the current price.

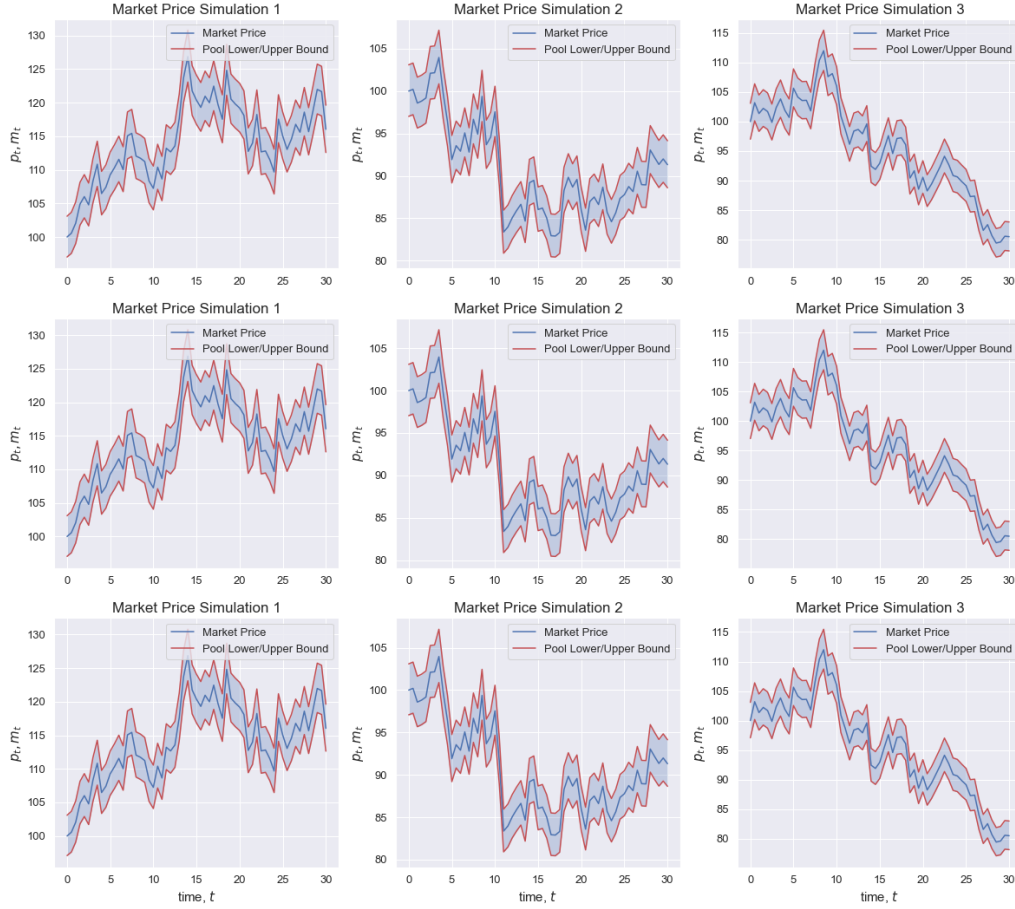
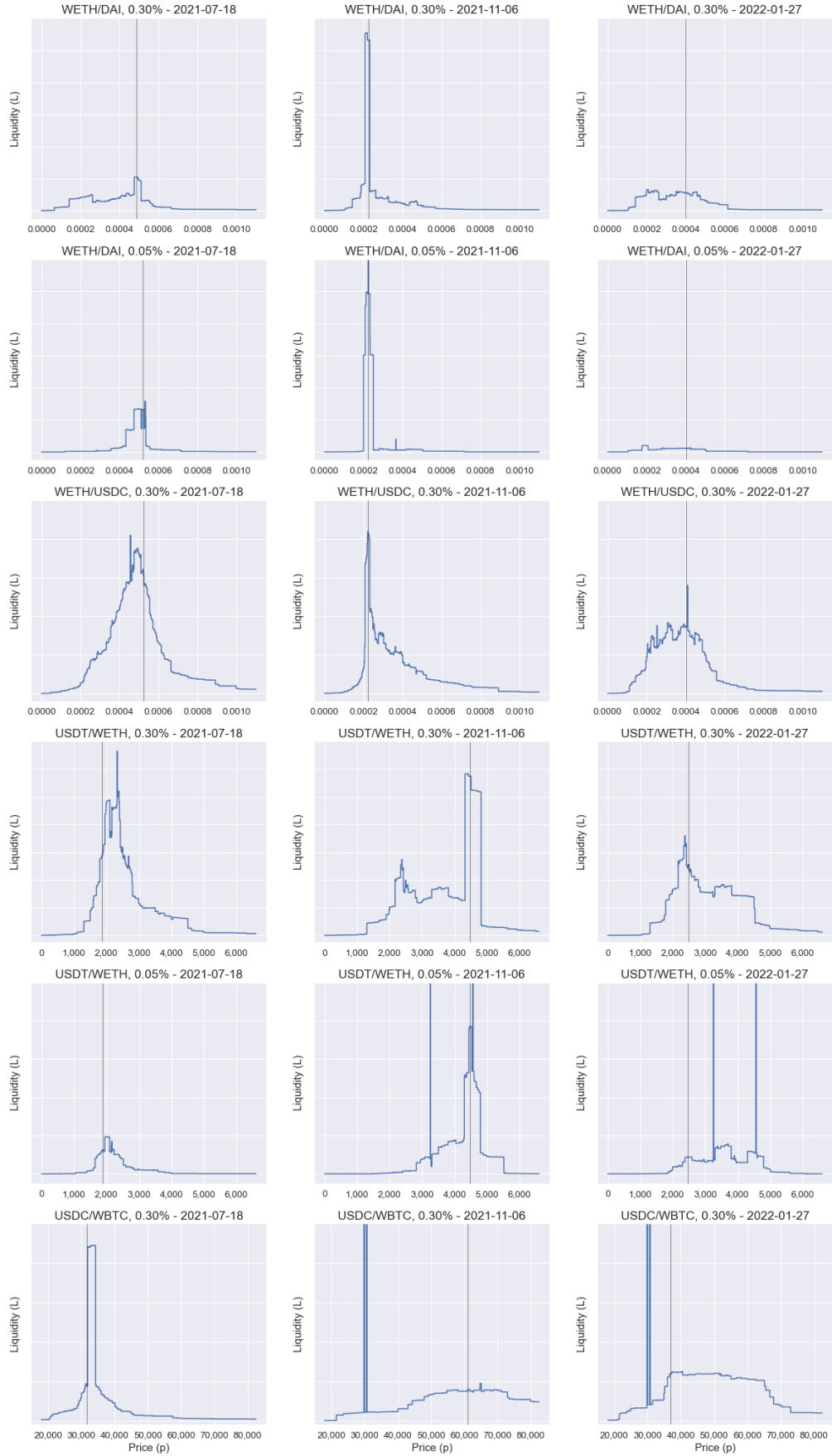


Figure 23: Simulated price bounds for three different price series (left/right) and three different liquidity curves (up/down), $L(p) = 10p + 4,000$, $L(p) = 25p + 2,500$, and $L(p) = 50p + 0$, with $\gamma = 0.03$. Results are the same for the three decreasing liquidity curves discussed above.

Appendix C Example Liquidity Curves

Liquidity curves for six of the seven pools (up/down) examined (note that the liquidity curves for the WETH/USDC, 0.05% fee pool are included in section 4.1 above). The pools presented are

as of July 18, 2021 (left), November 6, 2021 (middle), and January 27, 2022 (right). We do not show the scale of the y axis for the liquidity curves as liquidity varies in scale based on different decimal conventions used in the implementation of each pool. However, we set the scale of the y axis so that the relative level of liquidity for each pool can be observed over time.



Appendix D Expected Daily Returns to Liquidity Providers

Estimated daily mean returns, along with their 95% confidence intervals, for six of the seven pools examined (note that the results for the USDT/WETH, 0.05% fee pool are included in section 5 above). For all pools, the standard error of returns either grows from zero or shrinks to zero as we move from left to right of the current price. Whether the volatility in returns increases or decreases as we move left to right is determined by which token, `token0` or `token1`, for the pool is the stable coin and our numeraire token.

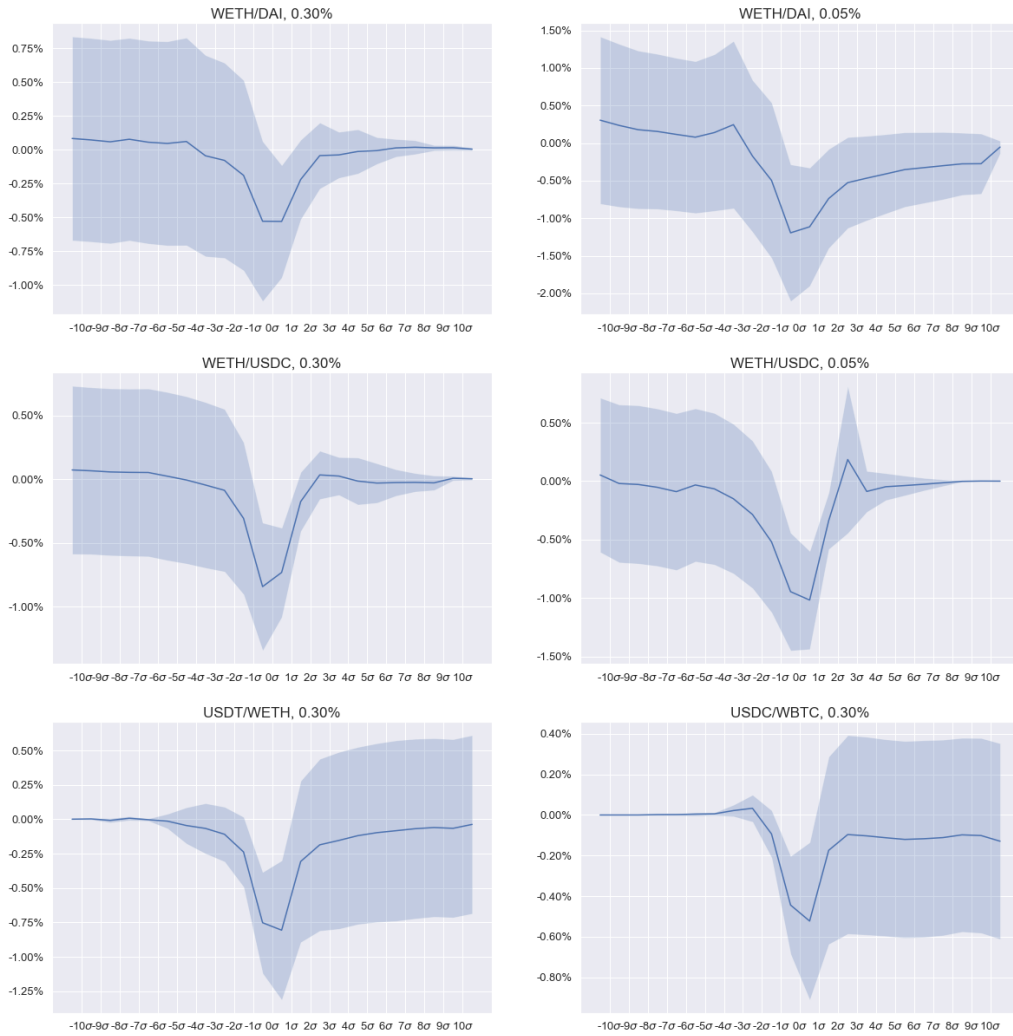


Figure 24: Estimated daily mean returns, along with their 95% confidence intervals, for six of the seven pools examined.