Differential Liquidity Provision in Uniswap v3 and Implications for Contract Design

Zhou Fan†
zfan@g.harvard.edu
Harvard University
Cambridge, MA, USA

Francisco Marmolejo-Cossío†
fjmarmol@seas.harvard.edu
Harvard University
Cambridge, MA, USA

Ben Altschuler
baltschuler@college.harvard.edu
Harvard University
Cambridge, MA, USA

He Sun
he_sun@g.harvard.edu
Harvard University
Cambridge, MA, USA

Xintong Wang
xintongw@g.harvard.edu
Harvard University
Cambridge, MA, USA

David C. Parkes‡
parkes@eecs.harvard.edu
Harvard University
Cambridge, MA, USA

ABSTRACT

Decentralized exchanges (DEXs) provide a means for users to trade pairs of assets on-chain without the need of a trusted third party to effectuate a trade. Amongst these, constant function market maker (CFMM) DEXs such as Uniswap handle the most volume of trades between ERC-20 tokens. With the introduction of Uniswap v3, liquidity providers are given the option to differentially allocate liquidity to be used for trades that occur within specific price intervals. In this paper, we formalize the profit and loss that liquidity providers can earn when providing specific liquidity positions to a contract. With this in hand, we are able to compute optimal liquidity allocations for liquidity providers who hold beliefs over how prices evolve over time. Ultimately, we use this tool to shed light on the design question regarding how v3 contracts should partition price space for permissible liquidity allocations. Our results show that a richer space of potential partitions can simultaneously benefit both liquidity providers and traders.

CCS CONCEPTS

• Theory of computation → Algorithmic game theory and mechanism design.

KEYWORDS

blockchain, decentralized finance, Uniswap v3, liquidity provision

ACM Reference Format:
Zhou Fan, Francisco Marmolejo-Cossío†, Ben Altschuler, He Sun, Xintong Wang, and David C. Parkes. 2022. Differential Liquidity Provision in Uniswap v3 and Implications for Contract Design. In . ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3533271.3561775

†Full Online Version of Paper: https://arxiv.org/abs/2204.00464
‡Both authors contributed equally to this research.
§also DeepMind.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICAIF '22, November 02–04, 2022, New York, NY
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9376-8/22/11 $15.00
https://doi.org/10.1145/3533271.3561775

1 INTRODUCTION

A key application in decentralized finance (DeFi) is that of decentralized exchanges (DEXs). DEXs offer smart contracts that allow users to trade tokens without the need of a trusted-third party. Such an implementation reduces hacking risks typically suffered by centralized off-chain exchanges. Amongst DEXs, there are two prevailing algorithmic paradigms: order book DEXs and automated market maker (AMM) DEXs. Order book DEXs maintain a list of buy and sell orders at distinct prices for a given pair of assets to be traded. These orders are received, matched and executed. An AMM contract provides buy and sell prices for trades, where said prices are typically computed as a function of the contract’s assets.

Currently, AMMs are the most common form of DEXs, amongst which Uniswap contracts handle a substantial proportion of trading volume. Uniswap contracts serve as constant function market makers (CFMM) that compute the price of buying and selling between two assets by preserving a functional invariant of its existing liquidity reserves. We briefly describe the operation of a CFMM.

Let x and y denote the liquidity reserves that the contract has for token A and token B respectively. The trading invariant can be expressed as $F(x, y) = C$, for a given function $F$ and a constant $C$ provided by the contract. A trader who sells $\Delta x > 0$ of token A sends $\Delta x$ units of token A to the contract and receives $\Delta y > 0$ of token B, such that the functional invariant is maintained, i.e., $F(x + \Delta x, y - \Delta y) = C$. Thus, $\Delta y/\Delta x$ represents the per-unit price of token A for the trade (in terms of token B). As $\Delta x \to 0$, this ratio gives the instantaneous price of token A in terms of token B.

Liquidity providers (LPs) are agents who provide assets to the contract and enable trades. An LP lends the contract a bundle of both A and B tokens, which is traded against each other as the relative price of token A (or token B) changes. Liquidity provision is rewarded by transaction fees on trades. In 2021, Uniswap v3 was introduced [2], and v3 contracts enable LPs to differentially allocate liquidity to be used for trades in a specified price interval. Transaction fees for trades are shared proportionally amongst LPs who provide liquidity on intervals that contain a price change. If the price exits the interval of an LP’s liquidity position, their liquidity no longer earns fees on trades.

With Uniswap v3, LPs can use the same capital to obtain more aggressive liquidity positions around tight price intervals of interest. This may lead to more transaction fees, albeit at the risk of losing out
on fees all together if prices exceed the specified interval. Another important consideration for LPs is the change in the composition of its capital as external prices of token $A$ and token $B$ evolve and trades occur to (approximately) match the external price. For example, if prices return to where they were when liquidity was lent, the LP can withdraw its liquidity in the same quantities as initially lent, whereas if prices have changed, the bundle of tokens to which an LP has claim has a lower value under the new prices. This is referred as the *impermanent loss* of a liquidity position, a crucial component for LP which we consider throughout this work.

Given the rapid increase of DEX usage\(^1\), it is important to understand the strategic considerations facing LPs. In this paper, we build off existing work to provide a new empirical and theoretical understanding of LP behavior, and further use these techniques to provide design recommendations for Uniswap v3 contracts.

**Our Contributions.** We begin by providing an overview of Uniswap dynamics in Section 2. In Section 3 we show how risk-neutral and risk-averse LPs can optimize their profit and loss over a finite time horizon as they hold stochastic beliefs of how prices will evolve. In Section 4, we apply these tools to the concrete design question of how v3 contracts should partition price space into potential liquidity positions. Through experiments that are calibrated to real price data, we provide empirical evidence that a greater diversity space of price partitions can benefit both LPs and traders. Furthermore, we also study how different degrees of risk-aversion impact optimal liquidity allocations for LPs. We find that as LPs become more risk-averse, they spread their liquidity across larger price ranges to the variance of their profit and loss. Moreover, we also show in the full version of the paper that increased volatility in price sequences also causes risk-averse LPs to spread their liquidity allocations farther over a larger price range.

**Related Work.** Our work extends a growing body of literature around liquidity provision incentives stemming from on-chain implementations of CFMMs. Most related to ours is the work by Neuder et al. [15], which studies strategic liquidity provision in Uniswap v3. They assume LPs hold beliefs that contract prices will evolve according to a Markov chain, and provide a method for LPs to maximize fees earned in the steady state. The main difference in our work is that we model the potential loss an LP can suffer as the asset price change with trades, i.e., *impermanent loss*. By modeling both fees earned and impermanent loss suffered, we aim to understand decision-theoretic implications faced by LPs who allocate liquidity optimally with respect to beliefs they may hold on how prices evolve. Moreover, we use this methodology to glean insight into the design of how v3 contracts should discretize price space for potential liquidity positions, a problem which to our knowledge has not been studied before in the CFMM literature.

Other prior literature studies Uniswap v2. Several work shows how LPs can replicate the payoffs of financial derivatives [4, 5, 13], Angeris et al. model arbitrage opportunities under trading fees and formulate potential LP earnings under simple price changes [7]. They show that under reasonable conditions v2 exchanges closely track reference markets. Similar guarantees are provided for a more general class of CFMMs, and techniques are extended to CFMMs supporting multi-asset trades [3]. Angeris et al. study the implications of curvature in reserve curves for traders and LPs, providing tradeoffs for when high and low curvature regimes favor each of these two classes of agents [6]. Capponi and Jia study the adoption of Uniswap v2 by considering a sequential game between LPs and traders [11]. We adopt a similar stochastic model for price updates in computational experiments. Our work here contributes to the line of literature by considering a richer space of liquidity allocations enabled by v3 and studying design questions in v3 contracts.

Liquidity provision in Uniswap shares similarities with market making in traditional limit-order book markets. Avellaneda M. models the market making problem of posting bid and ask quotes in a maximized exponential utility framework [10]. Our study of LP revenue as a function of fidelity in setting liquidity positions is also related to the *tick size design* problem in traditional limit order markets, which draws great attention from both an economics and regulatory perspective. European exchanges, for example, compete directly on the minimum pricing increment in the limit order book to capture market shares of quoted and executed volumes [14]. In 2016, the US Securities and Exchange Commission (SEC) launched the *Tick Size Pilot Program* to assess the impact of increasing tick sizes on the market quality for illiquid stocks [12].

### 2 OVERVIEW OF UNISWAP

As mentioned in the introduction, Uniswap contracts are a family of CFMM DEXs, designed to facilitate trades between ERC-20 token pairs. We provide an overview of Uniswap v3 contracts. The following exposition is necessarily brief due to page constraints, however everything that we explore can be derived from the algebra governing Uniswap contracts, which is outlined in the relevant Uniswap white papers [1, 2]. In addition, an in-depth description of Uniswap v2 and v3 contracts, along with relevant examples, can be found in the full version of our paper.

Going forward we consider a Uniswap contract which trades between token $A$ and token $B$, where we typically denote units of token $A$ by $x$ and units of token $B$ by $y$. Furthermore, when we refer to price going forward, we refer to the price of a unit of token $A$ in terms of token $B$. A v3 contract can be associated with fee rate, $y \in (0, 1)$ and a set of $(m + n + 1)$ buckets $\mu = \{B_{-m}, \ldots, B_n\}$. Buckets partition the space of all possible prices, $(0, \infty)$ into intervals. Furthermore, for each bucket, a liquidity provider (LP) can send bundles of $A$ tokens and $B$ tokens to the contract to be used as liquidity for trades that occur at the prices pertaining to that bucket. Similarly, if an LP owns liquidity in a certain bucket, they can extract it at any moment of time to obtain a commensurate bundle of $A$ and $B$ tokens (quantities being defined by Uniswap algebra). The process of providing and removing liquidity from a contract can thus be viewed as an LP “purchasing” and “selling” liquidity units for a given bucket $B_i \in \mu$, and hence LP's more generally interact with a v3 protocol by owning (and potentially shifting) liquidity allocations $\ell = (\ell_{-m}, \ldots, \ell_0, \ldots, \ell_n)$, where $\ell_i$ is the amount of liquidity units the LP owns for bucket $B_i$.

The overall liquidity allocation gives rise to a functional invariant maintained by the reserves of $A$ and $B$ tokens in the contract (i.e.

\(^1\)As of this writing, the daily average trade volume in Uniswap v2 and v3 contracts is approximately 160 Million USD and 1.25 Billion USD respectively. Total liquidity locked by users to facilitate trades in v2 and v3 contracts is 2.8 Billion USD and 4.4 Billion USD. See https://v2.info.uniswap.org/home and https://v3.info.uniswap.org/home.
for all \((x, y), F(x, y) = C\) for some function \(F\) and constant \(C\). Thus at any given moment, the contract maintains an internal state consisting of: the liquidity allocation of all LPs and the reserves of the contract, \((x, y)\). Trading (at a fee rate of \(\gamma = 0\)) corresponds to maintaining the functional invariant, so that if a trader sends \(\Delta y > 0\) units of token \(B\), they receive \(\Delta x\) units of token \(B\) such that \(F(x - \Delta x, y + \Delta y) = C\), resulting in a price of \(\Delta x / \Delta y\). The limit as \(\Delta y \to 0\) gives the contract price \(P\), and this is used to specify which bucket’s liquidity is used for trading. If \(\gamma > 0\), when sending \(\Delta y\) to the contract, it is first the case that \(\gamma \Delta y\) is levied as fees to be shared proportionally amongst LPs who have provide liquidity to the price interval of the trade. The remaining \((1 - \gamma) \Delta y\) is used traded as per the above process (respecting the functional invariant).

From the above it follows that the contract’s state implicitly contains contract price \(P \in (0, \infty)\). The final important detail for \(v3\) contracts is the fact that the bundle of tokens required to purchase liquidity in a bucket \(B_i = [a, b]\) is a function of \(P\). Changes in market value and subsequent trades change \(P\), which in turn changes the token bundles that an LP can cash their liquidity for if they exit their position. The discrepancy the equivalent token bundle value of a liquidity position at different market prices is called an *impermanent loss*. We consider LPs who borrow capital to open a static liquidity position over a finite time horizon, with the aim of earning fees and eventually settling their debt. Thus an LP’s profit and loss (PnL) is precisely their fees earned minus the impermanent loss incurred by their liquidity position.

Most importantly, we can compute the profit and loss of an LP’s position \(t = \{t_i\}_{i=-m}^m\) over a sequence of changes to contract price and market price. In general, we let \(P = (P_c, P_m)\) be a contract/market price pair and consider sequences of contract/market pairs, \(P = (P_1, \ldots, P_T)\), such that \(P_i = (P_{c,i}, P_{m,i})\). We denote said profit and loss by \(PnL(t, P)\), and note that this quantity is in fact linear in \(t\) for any fixed sequence \(P\). This fact can be readily extrapolated from the mechanics of Uniswap contracts, but due to space constraints we forego providing algebraic details. For full details, we once again point the reader to the full version of our paper.

### 3 OPTIMAL LIQUIDITY PROVISION

As mentioned in the previous section, for a given \(v3\) bucket structure \(\mu\) and contract/price sequence \(P = (P_0, \ldots, P_T)\), one can unpack the mechanics of Uniswap contracts in order to express \(PnL(t, P)\) as a linear function of \(t\). In this section, we assume that an LP believes contract/market prices sequences are generated stochastically and provide techniques for computing an optimal liquidity provision strategy of an LP over a finite time horizon. Going forward, we use the notation \(\mathcal{P}\) to denote a distribution over contract/market price sequences and call an instance of \(\mathcal{P}\) the belief profile of an LP prior to opening a \(v3\) liquidity position.

#### 3.1 Liquidity-independent LP Belief Profiles

In general, for a \(v3\) contract with bucket structure \(\mu\), liquidity provision is an intricate game where the set of agents consist of all LPs and traders, and an important manifestation of the interdependence of agents in the game is via the distribution \(\mathcal{P}\) over contract/market prices over a given time horizon.

We make the simplifying assumption that \(\mathcal{P}\) is independent of the aggregate liquidity of all LPs in the pool (a reasonable assumption for smaller LPs) and denote such belief profiles as *liquidity-independent*. Once \(\mathcal{P}\) is fixed, the question of how to optimally allocate liquidity becomes decision-theoretic, for an LP’s profit and loss only depends on the liquidity allocation they provide to the contract. Though the model fails to capture game-theoretic aspects of liquidity provision, this is a first step in understanding key design considerations in Uniswap \(v3\) contracts.

### 3.2 Optimization for Risk-averse LPs

In accordance with the definition of \(PnL(t, P)\) we assume that an LP borrows the capital required to create a liquidity position over buckets given by \(\mu\) for a fixed time horizon. Once the time horizon ends, the LP removes their liquidity from the contract, and uses this capital alongside accrued fees to pay the amount owed for creating the position. For a risk-neutral LP, the relevant quantity to optimize is the expected profit and loss over a distribution of price sequences generated by the liquidity-independent belief profile \(\mathcal{P}\):

\[
PnL_\mathcal{P}(t) = \mathbb{E}_{P \sim \mathcal{P}}(PnL(t, P)).
\]

Since \(PnL(t, P)\) is linear for any choice of \(P\), it follows that \(PnL_\mathcal{P}(t)\) is also linear in \(t\). In addition, we introduce the notion of risk-aversion and use the exponential (constant absolute risk aversion) utility function [8, 16]. The exponential utility captures comprehensive sources of risks by modeling higher moments of PnL.

**Definition 3.1.** Let \(u_a(x)\) denote the exponential utility function:

\[
u_a(x) = \begin{cases} 
(1 - e^{-ax}) / a & a \neq 0 \\
\infty & a = 0.
\end{cases}
\]

If \(a < 0\) the LP is risk-seeking, if \(a > 0\) the provider is risk-averse, and if \(a = 0\) the LP is risk-neutral.

We can now express the utility of an LP obtains for a given price sequence as follows.

**Definition 3.2 (Risk-averse PnL).** Suppose that an LP has a liquidity position given by \(t\) over a bucket instance \(\mu\). For a given contract/market price sequence, \(P = (P_0, \ldots, P_T)\), we denote the risk-averse profit and loss of the LP by \(PnL^a(t, P)\) for risk-profile \(a \geq 0\), given by:

\[
PnL^a(t, P) = u_a(PnL(t, P)).
\]

If \(P\) is in addition generated according to an LP belief profile \(\mathcal{P}\), then we let \(PnL^a_{\mathcal{P}}\) denote the expected risk-averse profit and loss of the LP at risk-profile \(a \geq 0\):

\[
PnL^a_{\mathcal{P}}(t) = \mathbb{E}_{P \sim \mathcal{P}}(PnL^a(t, P))
\]

**Lemma 3.3.** \(PnL^a_{\mathcal{P}}(t)\) is concave in \(t\) for any choice of \(a\) and \(\mathcal{P}\).

**Proof.** \(u_a(x)\) is smooth and concave for any choice of \(a\), as \(e^{-ax}\) within the expression is convex. We have previously seen that \(PnL(t, P)\) is linear in \(t\) for any choice of \(P\), hence \(PnL^a(t, P)\) is itself concave in \(t\) for any choice of \(P\). Finally, concavity is preserved over the expectation in the definition, hence the claim holds. \(\square\)
We consider an LP that has an initial normalized budget of \( W = 1 \) unit of token \( B \) and wishes to create an optimal liquidity position. For each bucket \( B_i \in \mu \), we let \( w_i \) denote the token \( B \) worth of 1 unit of liquidity at the initial parity contract/market price \( P_0 = (P_{c,0}, P_{m,0}) = (1, 1) \). Summing over buckets, the budget constraint is \( \sum_i \ell_i w_i \leq W \). Given this, the optimization problem for optimal risk-averse liquidity allocations is as follows:

\[
\begin{aligned}
\min_{\ell} & \quad - PnL_a^a(\ell) \\
\text{s.t.} & \quad \sum_i \ell_i w_i \leq W \\
& \quad \ell_i \geq 0, \text{ for all } i
\end{aligned}
\]  

(5)

We let \( \text{OPT}_a(\mathcal{P}, \mu) \) denote the value of the optimal solution, and we let \( \ell^a \) denote this liquidity position. Notice that as a consequence of Lemma 3.3, the optimization problem is convex. In subsequent sections we exploit convexity to provide efficient methods for approximating \( \text{OPT}_a(\mathcal{P}, \mu) \). In addition, we also note that when \( a = 0 \) (risk-neutral LP) the optimization problem becomes linear, and optimal liquidity allocations are forcibly concentrated on a single bucket. Such allocations do not occur in practice, providing further credence to studying risk-averse LPs.

### 3.2.1 Computational Methods

The main difficulty in solving the optimization problem corresponding to Equation 5 lies in the fact that the objective function is an expectation over a large set of price paths that can arise from \( \mathcal{P} \) over a given time horizon. In this regard, we approximate \( PnL_a^a(\ell) \) as the average \( PnL \) over a sample of sequences from the stochastic process governing contract/market price movements.

More specifically, we begin by taking \( N \) i.i.d. sample paths, \( P_1, \ldots, P_N \sim \mathcal{P} \), and define the following objective function

\[
PnL_a^a(\ell | P_1, \ldots, P_N) = \frac{1}{N} \sum_{i=1}^{N} PnL_a^a(\ell | P_i)
\]

(6)

Notice that if we take expectations over the \( P_i \) we get:

\[
\mathbb{E}_{P_1, \ldots, P_N}(PnL_a^a(\ell | P_1, \ldots, P_N)) = PnL_a^a(\ell)
\]

(7)

We can define a corresponding optimization problem for a risk-averse LP seeking to approximately optimally create a liquidity allocation, subject to budget-constraints:

\[
\begin{aligned}
\min_{\ell} & \quad - PnL_a^a(\ell | P_1, \ldots, P_N) \\
\text{s.t.} & \quad \sum_i \ell_i w_i \leq W \\
& \quad \ell_i \geq 0, \text{ for all } i
\end{aligned}
\]  

(8)

This is a convex optimization problem, and we can evaluate gradients for the objective function via standard methods. As before, we focus on the case where \( W = 1 \), and use projected gradient descent as a solution method.

### 3.3 Optimal Risk-averse PnL as a function of Bucket Characteristics

\( \text{OPT}_a(\mathcal{P}, \mu) \) is intricately tied to the characteristics of the buckets available in a v3 contract. In practice, the Uniswap v3 contract uses buckets \( \mu = \{B_{-m}, \ldots, B_0, \ldots, B_n\} \) with endpoints that correspond to multiplicative increases and decreases of the parity price \( P = 1 \). The contract maintains a fixed set of price ticks \( \{t(-q), \ldots, t(0), \ldots, t(q)\} \), where \( t(1) = 1.0001^1 \) and \( q = 2^{12} \). In addition, each contract has a positive integer variable called tickspacking, which we denote by \( \Delta \) and dictates which of all possible tick values can be used as bucket endpoints. A tick can only be a bucket endpoint if \( i \equiv 0 \mod \Delta \). This means that we can express this bucket structure by letting the \( i \)-th bucket, \( B_i \), represent an interval \([a_i, b_i]\) such that \( a_i = 1.0001^{\Delta i} \) and \( b_i = 1.0001^{\Delta(i+1)} \).

**Proposition 3.4.** Suppose that \( \mu(\Delta) \) is the bucket list that results from setting tickspacking to \( \Delta \geq 1 \). Furthermore, let \( \Delta' = k\Delta \) for any integer \( k \geq 2 \). For any choice of LP belief profile \( \mathcal{P} \), and risk-aversion parameter \( a \), we have:

\[
\text{OPT}_a(\mathcal{P}, \mu(\Delta')) \leq \text{OPT}_a(\mathcal{P}, \mu(\Delta)).
\]  

(9)

Proof. These bucket designs are nested, with \( \mu(\Delta') \) a coarsened version of \( \mu(\Delta) \), which means that any liquidity position over \( \mu(\Delta') \) can also be represented by a position over \( \mu(\Delta) \). Suppose a position over \( \mu(\Delta') \) consists of allocating \( \ell_i \) units of liquidity to \( B_i \). If an LP creates a position with \( \ell_i \) units of liquidity in each of the \( k \) buckets in \( \mu(\Delta) \) that correspond to this bucket, then the overall bundle value of the position is identical to \( \ell_i \) units of liquidity in \( B_i \) and results in the same fees for the LP.

### 3.4 Trader Gas Fees

Finer bucket lists allow LPs to create more complicated liquidity positions that potentially make use of more active endpoints. As seen in Proposition 3.4 this generally leads to larger expected PnL for LPs in v3 contracts. However, said refinement also results in increased gas fees for traders, for a desired trade is to push a contract price outside of the current active interval of the v3 contract, a new active interval must be computed, alongside the set of active and active liquidity. This computational overhead is passed on to traders as they must cover gas fees for the trade to be processed.

For each bucket, \( B_i \in \mu \), we can compute the expected number of crossings of its left endpoint, \( a_i \), over the course of the the stochastic process dictated by LP belief profile \( \mathcal{P} \). We denote this quantity \( c_i(\mathcal{P}, \mu) \). What matters for gas fees, though, is the number of active bucket endpoints that are crossed. For there to be computational overhead, there needs to be an LP with a position that uses the endpoint. To proceed, we assume that each bucket endpoint is active in this sense. We call this the \textit{bucket coverage assumption}. This is justified empirically; we find that the liquidity placed in Uniswap v3 contracts by the population of LPs almost invariably covers the spread of buckets around the market price of an asset (which in turn closely matches the contract price otherwise an arbitrage opportunity arises). In particular, for buckets close to the contract price, we find empirically that almost invariably there exists at least one LP who has a position ending in that bucket’s endpoint.
**Definition 3.5 (Trader Gas Cost).** Let us consider a liquidity provision instance dictated by LP belief profile $\mathcal{P}$ and buckets given by $\mu$. We let $\text{GAS}(\mathcal{P}, \mu)$ denote the expected gas fees incurred by all traders over the time horizon $T$. Under the bucket coverage assumption, this is given by:

$$\text{GAS}(\mathcal{P}, \mu) = \sum_{i=-m}^{n} c_i(\mathcal{P}, \mu)$$  \hspace{1cm} (10)$$

In our definition of GAS, we have normalized the gas fees per crossing of an active endpoint to be 1. Indeed, the per-crossing gas fees can fluctuate, but our expression is correct up to a calibration constant. We will ultimately only be interested in the relative gas cost between distinct bucketing schemes, hence linearly scaling GAS does not change our results. Finally, we highlight that as in the computation of $\text{OPT}_\mu(\mathcal{P}, \mu)$, we can take a sampling based approach to approximate $\text{GAS}(\mathcal{P}, \mu)$, whereby we sample $k$ sequences of contract/prices pairs from $\mathcal{P}$ and compute the average number of bucket endpoint crossings as an approximation to $\text{GAS}(\mathcal{P}, \mu)$.

### 3.5 The Uniswap v3 Contract Design Problem

We are interested in how choices for bucket design $\mu$ impact both $\text{OPT}_\mu$ and GAS, for fixed LP belief profiles $\mathcal{P}$ and risk-aversion values $a \geq 0$. Both objectives are important to a contract, for higher PnL may attract liquidity providers to reduce slippage in trades. On the other hand, increased gas costs can dissuade traders from participating in the contract, thereby reducing rewards for LPs in fees. There is a rich space of potential partitions of price space into buckets, but we typically focus on parametrized families of bucket sets, such as the following:

**Definition 3.6 (Exponential Bucket Scheme).** Suppose that $\theta > 1$ is a real number and $\Delta, m, n \geq 1$ are integers. We call $\mu(\theta, \Delta, m, n)$ a $(\theta, \Delta, m, n)$-exponential bucketing scheme. Buckets in $\mu(\theta, \Delta, m, n)$ are parametrized as follows:

- $B_{-m} = (0, b_{-m}]$ with $b_{-m} = \frac{\theta^m}{\theta^m - 1}$
- $B_n = [a_n, \infty)$ with $a_n = \frac{\theta^n}{\theta^n - 1}$
- For $-m < i < n$, $B_i = [a_i, b_i] = [\frac{\theta^i}{\theta^i - 1}, \frac{\theta^{i+1}}{\theta^{i+1} - 1}]$

For convenience, we also let $\mu(\theta, \lambda)$ denote an exponential bucket scheme where we let $m, n \to \infty$, such that $B_i$ is a finite bucket for all $i \in \mathbb{Z}$. Exponential bucket schemes are a natural extension of the existing bucketing schemes used by v3 contracts in Uniswap. In particular, a v3 contract with tick spacing $= \Delta$ is equivalent to a $(1,0001, \Delta)$-exponential bucketing scheme. With parametrized bucket schemes, we can conveniently express the Pareto frontier of such bucketings with respect to the objectives $\text{OPT}$ and GAS.

**Definition 3.7 (The Uniswap v3 OPT-GAS Pareto Frontier).** Suppose that we fix an LP optimization profile given by $\mathcal{P}$, risk-aversion parameter given by $a \geq 0$ and a parametrized family of bucket sets $\mu(\lambda)$. Furthermore, let $\lambda' \neq \lambda$ be such that:

$$\text{GAS}(\mathcal{P}, \mu(\lambda')) \leq \text{GAS}(\mathcal{P}, \mu(\lambda)),$$

$$\text{OPT}_\mu(\mathcal{P}, \mu(\lambda')) \geq \text{OPT}_\mu(\mathcal{P}, \mu(\lambda)),$$

where one of the inequalities is strict. We say that bucketing scheme $\mu(\lambda')$ Pareto dominates $\mu(\lambda)$ for the LP belief profile $\mathcal{P}$ and risk-aversion parameter $a \geq 0$. We denote this relationship over the parameter space of $\mu$ by $\lambda' >_P \lambda$. In addition, we let $\text{Pareto}(\mathcal{P}, \mu)$ denote the set of all parameters, $\lambda$ which are not Pareto dominated.

### 4 COMPUTATIONAL RESULTS

In this section, we take a computational approach to investigate the v3 contract design problem. We utilize a specific family of LP belief profiles that can be seen as an extension of the model from Capponi and Jia [11], repeated over a longer time horizon.

#### 4.1 Modeling Liquidity-independent Contract-market Price Changes

Our family of liquidity-independent belief profiles generates contract-market prices via a repeated, two-stage process over $T$ rounds. We model a stochastic market price, with this inducing a stochastic contract price. In each round, in a first stage we sample the new market price according to an exogenous stochastic process, and in a second stage we model non-arbitrage traders who either increase or decrease contract price in a stochastic manner. At any point in either stage, if there is a large enough discrepancy between the contract and market price (as a function of the transaction fee rate, $\gamma$), then arbitrageurs trade with the contract to bring the contract price close to market price. Figure 1 illustrates the way that we model changes in contract price in response to market price changes, non-arbitrage trades, and arbitrage trades.

**Figure 1: Market price changes triggering contract price changes, illustrated for three rounds, where each round contains four non-arbitrage and four arbitrage trades following a market price change.**
integral bandwidth parameter of $W \geq 1$. Let $Y' \sim \text{Binom}(2W, p)$, such that $Y = Y' - W$ encodes the maximal change in index from the current price $Z_t$. This means that the price transitions as follows:

$$z_{t+1} = \begin{cases} Z_{r} & \text{if } i + Y < -r, \\
Z_{s} & \text{if } i + Y > s, \\
Z_{i} & \text{otherwise} 
\end{cases} \quad (11)$$

By imposing a constant price ratio of $\omega$ in $Z$, and having price indices transition according to a binomial distribution, the stochastic price process approximates a geometric binomial random walk (GBRW). An exception is when the price is close to the boundaries values $\omega$. Coarse discretization of price space over large time horizons runs $t$ is for two reasons. First, we consider optimal LP strategies with a maximum likelihood estimate (see Appendix of full version of GBRW. We fit these two parameters using historical price data via geometric binomial random walk.

$$\hat{\omega}, \hat{W}$$

are highly correlated. We use prices between Ethereum (ETH) and Uniswap contract with contract price $P$ be the no-arbitrage interval around market price $P_{m}$. As shown in Angeris et al. [7], if a contract price $P_{c}$ is such that $P_{c} \in I_{p}(P_{m})$, then even if $P_{c} \neq P_{m}$, arbitrage is not profitable due to transaction fees. More specifically, non-arbitrage conditions for trading with a Uniswap contract with contract price $P_{c} \in (0, \infty)$ precisely amount to having $P_{c} \in I_{q}(P_{m})$. Consequently, if at $P_{j}$, the contract price is outside the arbitrage interval of the market price, we will assume that arbitrageurs trade in such a way that the contract price reaches the closest point in the no-arbitrage interval.

**Proposition 4.1 (informal).** Suppose that $m = \log(\omega)$. If we let $p = \frac{m^{2} - \sqrt{m^{2} + 4}}{2m}$, then for large bandwidth values, it follows that $E(\frac{Z_{t+1}}{Z_{t}}) = 1$. Consequently, the price ratio stochastic process is approximately on-average stable.

**Empirically Informing the GBRW.** We require five parameters, $(r, s, \omega, W, T)$, to define how the market price evolves. The parameters $(r, s, \omega)$ give rise to the price space, $Z'(r, s, \omega)$. The parameters $\omega > 1$ and $W \geq 1$ define the approximate GBRW that dictates random transitions over $Z'(r, s, \omega)$ and $T \geq 1$ controls the number of times market price can change. Proposition 4.1 implies that the design choice mostly depends on the multiplicative ratio $\omega$, and is not dependent on the total number of price tick $r + s + 1$. This provides a way to obtain a price sequence that approximates an on-average stable geometric random walk, where $\omega$ and $W$ govern the overall volatility, as larger values of $\omega$ imply that price index changes result in larger multiplicative price changes, and larger bandwidth values, $W$, imply that the random walk can make larger jumps in a given time step.

The parameters $\omega$ and $W$ directly impact the volatility of the GBRW. We fit these two parameters using historical price data via a maximum likelihood estimate (see Appendix of full version of paper). We use prices between Ethereum (ETH) and Bitcoin (BTC) for the low volatility regime, as the prices of these two tokens are highly correlated. We use prices between Ethereum (ETH) and USD for the high volatility regime. For ETH/BTC prices, we estimate values $\omega = 1.0005$ and $W = 3$, and for ETH/USD, we estimate values $\omega = 1.0005$ and $W = 7$. Given this, we adopt $\omega = 1.0005$ in all LP belief profiles and consider $W \in \{3, 5, 7\}$ to represent low/medium/high market price volatility regimes.

We set $r = s = 150$, so that the GBRW takes prices in the interval $[0.0728, 1.0778]$, and consider time horizon $T = 100$. Given the fact that our empirical results are informed from per-minute price data, this time-scale corresponds to a roughly two hour window. This choice of a smaller price range and time horizon is for two reasons. First, we consider optimal LP strategies with respect to introducing liquidity at time $t = 0$ and removing it at time $t = T$. In practice we expect such simple strategies to be more prevalent at smaller time scales. Moreover, using a GBRW with a coarse discretization of price space over large time horizons runs the risk of inadequately modeling smaller order price oscillations, in turn impacting LP profit and loss (in terms of fees mostly); working with a smaller time horizon and price space mitigates this risk.

**4.1.2 Contract Price Updates.** As mentioned before, we model contract-price updates via rounds that undergo a two-stage process. The first stage in a round samples the market price, as per the approximate GBRW process from before. The second stage modifies the contract price according to both non-arbitrage and arbitrage trading. These modifications require 3 relevant parameters: $(k, \lambda, \gamma)$. The first parameter $k \in \mathbb{N}$ specifies the number of non-arbitrage trades that occur in a given round. The second parameter $\lambda \in (0, 1)$ specifies the multiplicative price impact a single non-arbitrage trade has in a given round. The final parameter $\gamma \in [0, 1]$ is the trading fee present in the v3 contract at hand.

In more detail, the $t$-th round consists of a sequence of $2(k + 1)$ contract-market price pairs given by $P' = (P^{t}_{1}, \ldots, P^{t}_{2k+1})$. The $j$-th pair is given by $P^{t}_{j} = (P^{t}_{c,j}, P^{t}_{m,j})$, and for all pairs it holds that $P^{t}_{c,j} = z_{j}$; i.e., the market price remains unchanged within this second stage of the round. For all odd $j$, the transition from $P^{t}_{j}$ to $P^{t}_{j+1}$ consists of a single non-arbitrage trade. With probability $1/2$, the trader buys token $A$ from the contract such that contract price increases to $P^{t}_{c,j+1} = (1 - \lambda)P^{t}_{c,j}$, and with probability $1/2$, the seller sells token $A$ to the contract such that contract price decreases to $P^{t}_{c,j+1} = (1 + \lambda)P^{t}_{c,j}$. This form of price-based non-arbitrage trade is similar to that used in Capponi and Jia [11].

For all even $j$, the transition from $P^{t}_{j}$ to $P^{t}_{j+1}$ consists of a (potentially empty) arbitrage trade. Given the fee rate, $\gamma$, and any market price $P_{m} \in (0, \infty)$, we let $I_{q}(P_{m}) = [(1 - \gamma)P_{m}, (1 + \gamma)^{-1}P_{m}]$ be the no-arbitrage interval around market price $P_{m}$. As shown in Angeris et al. [7], if a contract price $P_{c}$ is such that $P_{c} \in I_{q}(P_{m})$, then even if $P_{c} \neq P_{m}$, arbitrage is not profitable due to transaction fees. More specifically, non-arbitrage conditions for trading with a Uniswap contract with contract price $P_{c} \in (0, \infty)$ precisely amount to having $P_{c} \in I_{q}(P_{m})$. Consequently, if at $P^{t}_{j}$, the contract price is outside the arbitrage interval of the market price, we will assume that arbitrageurs trade in such a way that the contract price reaches the closest point in the no-arbitrage interval.

**Relevant Parameter Regimes.** Going forward we fix the empirically-informed values of $r, s = 150, \omega = 1.0005$ and $T = 100$ for precisely the reasons specified at the end of 4.1.1: 1) static LP strategies are more realistic at smaller time scales 2) finer discretizations of price space (and correspondingly smaller price ranges) mitigate the risk of model error arising from smaller order price oscillations. Consequently, we focus on modulating four parameters, $(W, k, \lambda, \gamma)$ to change the way in which market prices are updated and also how non-arbitrage trades and arbitrage trades induce contract price updates. Parameters $k$ and $\lambda$ modulate the number of non-arbitrage trades in a round, and the per-trade price impact of said trades, respectively. Parameters $W$ and $\gamma$ modulate how market prices evolve and when arbitrage trading kicks in.

In Sections 4.2, 4.3, and 4.4 we fix $(W, k, \lambda, \gamma) = (5, 10, 0.00025, 0.01)$. $k$ and $\lambda$ are chosen such that two consecutive non-arbitrage trades, if uninterrupted by arbitrage trades, result in a multiplicative change of $1.0005$ which equals the value of the parameter $\omega$ from the GBRW.
The choice of $γ = 0.01$ is the highest fee tier available in Uniswap contracts, which can have fee tiers $γ \in \{0.0005, 0.003, 0.01\}$. Finally, for these parameters, modulating $W \in \{3, 5, 7\}$ has a small impact on results, hence we fix $W = 5$ to represent intermediate market price volatility. In Section 4.5, we vary $k$, $λ$ and $γ$ to see how $\text{OPT}_a$ and $\text{GAS}$ depend on different scales of non-arbitrage and arbitrage trade while maintaining intermediate market price volatility fixed. In the full version of the paper we explore further parameter regimes that exhibit higher contract-market price volatility and study how this impacts our results.

### 4.2 The Effect of $Δ$ on PnL and Gas Cost

The first observation we can make is that the LP’s optimal PnL increases as bucket sizes decrease, in line with our result from Proposition 3.4. In a similar vein, we also see that as buckets become more fine-grained, the expected gas cost to traders accordingly increases, as finer partitions result in more crossings of active price ticks. We visualize this in Figure 2 for $W = 5$, and $a = 20$ where we simultaneously plot an LP’s optimal expected utility and Gas cost when $θ$ is fixed to 1.002 and we let $Δ \in \{i\}_{i=1}^{40}$. We also map the expected utility of v2 contracts (v2 Gas cost is 0 in our model as no crossings of bucket endpoints ever occur).

![Figure 2: LP’s optimal expected utility and trader Gas Cost, for $(W, k, λ, γ) = (5, 10, 0.00025, 0.01)$ and $a = 20$, and a $(θ, Δ)$-exponential bucketing scheme with multiplicative factor $θ = 1.002$ and bucket spacing $Δ \in \{1, \ldots, 40\}$. The plot also shows the corresponding v2 expected utility.](image)

### 4.3 Uniswap v3 OPT-GAS Pareto Frontier

In Figure 3 we focus on market volatility $W = 5$ and simultaneously plot the performance of multiple $(θ, Δ)$-exponential bucketing schemes for a risk-neutral LP and a risk-averse LP with $a = 20$. Each point in the plot represents the performance of a given $(θ, Δ)$-exponential bucketing scheme, where each $(θ, Δ)$ value is plotted at coordinates given by $(\text{OPT}(P, μ(θ, Δ)), \text{GAS}(P, μ(θ, Δ)))$. In each plot, points of the same color have the same $θ$ value in the bucketing scheme but different $Δ$ values from the set $\{1, \ldots, 20\}$. We also provide plots that highlight which of these points lie on the OPT-GAS Pareto frontier. The most salient observation from these results is the fact that multiple parameter settings for $(θ, Δ)$-exponential bucketing schemes lie on the Pareto frontier, and this is irrespective of the price volatility regime. This tells us that a concrete design improvement for Uniswap v3 contracts lies in allowing more expressive bucketing schemes than the status quo, which is equivalent to $(θ, Δ)$-exponential bucketing schemes with $θ = 1.0001$ and $Δ$ corresponding to tick-spacing in the contract. The top right point of each Pareto frontier plot (in blue) of Figure 3 corresponds to a $(1.002, 1)$-exponential bucketing scheme. This $θ$ value is closest to the v3 status quo of $θ = 1.0001$ and results in relatively high utility for LPs but higher gas fees for traders. Indeed it could be the case that both LPs and traders wish to strike different tradeoffs along the Pareto curve, and to do so, richer partitions of price space for buckets are needed.

### 4.4 The Effect of Risk-aversion

We see that as an LPs risk-aversion increases, optimal liquidity allocations remain centered around the initial unit price, but become more spread out, as visualized in Figure 4. This is to be expected as an LP spreads their position to mitigate the risk of impermanent loss and losing out on fees. In addition, from Figure 3 we can see that with increased LP risk-aversion, the same set of exponential bucketing schemes gives rise to a steeper Pareto curve composed of less points, indicating that different $(θ, Δ)$-exponential bucketing schemes with similar Gas costs give rise to a much wider spread of expected utility in the risk-averse setting. Finally, the risk-averse Frontier itself is still composed of multiple $(θ, Δ)$ values, which indicates that a richer partition of price space can still simultaneously benefit LPs and traders at a wide spread of risk-aversion values.

### 4.5 Modulating Non-arbitrage Trade and Fee Rates

We recall that $k$ and $λ$ modulate the quantity of non-arbitrage trades and their impact on contract price respectively. The most clear observation we make is that both LP PnL and Gas costs are monotonically increasing in both $k$ and $λ$ as expected, which can be visualized in for $k$ in Figure 5. For PnL, as pointed out in [11], LPs only make a profit for non-arbitrage trades, hence this relationship is expected. As for gas costs, it is clear that more contract price movements trigger crossing more active ticks.

The transaction fee rate $γ$ on the other hand affects LP PnL and Gas costs in multiple ways in our model. On one hand, changing $γ$ directly affects LP profits, as price movements consequently bear more fees to the LP. On the other hand, $γ$ affects the no-arbitrage interval around a given market price, which can ultimately change the stochastic process governing contract/market price evolution as per LP belief $P$. In Figure 6 we see that a risk-neutral LP exhibits all fee rates studied on the Pareto frontier, but this rapidly changes as risk-aversion is increased. For $a = 10, 20$, it becomes clear that the highest fee rate, $γ = 0.2$ dominates the rest. Equivalently speaking, to earn the same expected utility as $γ = 0.2$ with other lower fee rates, the contract must make the sacrifice of having larger gas costs. If we only look at this plot, it may seem that a higher fee rate is always better for contract design. In reality, our contract lacks the fact that a higher fee rate decreases desirability of a pool to traders. Ultimately we need to model competition between pools and the delicate interplay between LPs and pool desirability to traders in order to obtain a more precise statement.
In this paper, we provide a detailed decision-theoretic framework for the tradeoffs LPs are faced with regards to how to optimally allocate liquidity in Uniswap v3 contracts. We do so by studying expressions for LP profit and loss that incorporate profits from fees accrued from traders as well as potential impermanent loss from deviations in contract price. Finally, and most importantly, we explore optimal liquidity provision strategies when LPs are endowed with stochastic beliefs over how prices will evolve over a finite time horizon as well as potential aversion to risk in earnings. In this setting, we provide algorithms for computing optimal expected PnL (in both the risk-neutral and risk-averse setting), as well as a measure of the Gas fees incurred by traders for a given bucketing scheme in a v3 contract. We apply these techniques to allocate liquidity in Uniswap v3 contracts.

5 CONCLUSION AND FUTURE WORK

In this paper, we provide a detailed decision-theoretic framework for the tradeoffs LPs are faced with regards to how to optimally allocate liquidity in Uniswap v3 contracts. We do so by studying expressions for LP profit and loss that incorporate profits from fees accrued from traders as well as potential impermanent loss from deviations in contract price. Finally, and most importantly, we explore optimal liquidity provision strategies when LPs are endowed with stochastic beliefs over how prices will evolve over a finite time horizon as well as potential aversion to risk in earnings. In this setting, we provide algorithms for computing optimal expected PnL (in both the risk-neutral and risk-averse setting), as well as a measure of the Gas fees incurred by traders for a given bucketing scheme in a v3 contract. We apply these techniques to allocate liquidity in Uniswap v3 contracts.
study optimal LP profit and loss for empirically-informed belief profiles of differing price volatility. At a high level, we see that LP PnL is maximized for smaller bucket sizes, but this comes at the cost of higher gas fees for traders. At the same time, if we treat the choice of how to partition prices into buckets as a multi-objective optimization problem where we aim to simultaneously optimize for LP PnL and trader Gas Cost, we see that a rich space of bucketing schemes maximally exemplify tradeoffs between these two objectives, strongly indicating that v3 contracts should permit users the flexibility in having richer bucketing schemes beyond that which is currently available. Despite having limitations in capturing more realistic effects of different transaction fee rates, our results also clearly demonstrate that transaction fee rate is a significant parameter in v3 contract design. Finally, we explore how risk-aversion can affect LP behavior, and we see that our evidence for providing richer bucketing schemes persists, and in addition we see optimal LP behavior which involves spreading liquidity over more buckets.

Several interesting future directions arise from our work. First, a natural follow-up may consider more complicated liquidity provision strategies, e.g., enabling LPs to actively re-allocate and adjust liquidity as prices evolve over time. Besides impermanent loss, this requires the modeling of potential gas fees that LPs would need to pay to reallocate liquidity. Second, how can we interpret the liquidity profile of LPs who have locked assets into a given v3 contract? Our work here offers an initial step in this direction, for we can interpret the optimal liquidity allocation (within the simple liquidity strategies we consider) as an on-chain signal of an LP’s belief profile about price movements. More broadly, can we infer the collective price beliefs of LPs from their liquidity provision and their interactions with a contract? Another direction for future work is to build a more sophisticated model to capture more realistic effects of different transaction fee rates for the v3 contract. This requires taking into consideration the competition between pools with different fee rates which affects the trade volume. Finally, an interesting extension is to study strategic interactions among the LPs, taking into account how one agent’s liquidity allocation may affect the price movement and thus the action of other LPs. This requires relaxing the exogenous price assumption and conducting game-theoretic analysis among LPs.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for helping improve earlier versions of this work, as well as insightful conversations with research teams at Wisdomem, Maverick, and Uniswap. This work is supported in part by two generous gifts to the Center for Research on Computation and Society at Harvard University to support research on applied cryptography and society. Our work also benefited from Microsoft Azure credits provided by the Harvard Data Science Initiative.

REFERENCES

[1] Hayden Adams, Noah Zinmsmeister, and Dan Robinson. 2020. Uniswap v2 core. URL: https://uniswap.org/whitepaper.pdf (2020).
[2] Hayden Adams, Noah Zimmermann, Moody Salem, River Keefer, and Dan Robinson. 2021. Uniswap v3 core. Technical Report. Tech. rep., Uniswap.
[3] Guillermo Angeris and Tarun Chitra. 2020. Improved price oracles. Constant function market makers. In Proceedings of the 2nd ACM Conference on Advances in Financial Technologies. 80–91.
[4] Guillermo Angeris, Alex Evans, and Tarun Chitra. 2021. Replicating market makers. arXiv preprint arXiv:2103.14769 (2021).
[5] Guillermo Angeris, Alex Evans, and Tarun Chitra. 2021. Replicating Monotonic Payoffs Without Oracles. arXiv preprint arXiv:2111.13740 (2021).
[6] Guillermo Angeris, Alex Evans, and Tarun Chitra. 2022. When Does The Tail Wag The Dog? Curvature and Market Making. Cryptoeconomic Systems, 2(1) (2022).
[7] Guillermo Angeris, Huien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. 2021. An Analysis of Uniswap markets. Cryptoeconomic Systems, 1(1) (2021).
[8] Kenneth Joseph Arrow. 1965. Aspects of the theory of risk-bearing. Helsinki.
[9] Stoikov S. Avellaneda M. 1992. Stock price volatility: Tests based on the geometric random walk. The American Economic Review (1992).
[10] Stoikov S. Avellaneda M. 2008. High-frequency trading in a limit order book. Quantitative Finance (2008).
[11] Agostino Capponi and Ruizhe Jia. 2021. The adoption of blockchain-based decentralized exchanges. arXiv preprint arXiv:2103.08842 (2021).
[12] Rösch D. Chung K., Lee A. 2019. Tick size, liquidity for small and large orders, and price informativeness: Evidence from the Tick Size Pilot Program. Journal of Financial Economics (2019).
[13] Alex Evans. 2021. Liquidity Provider Returns in Geometric Mean Markets. Cryptoeconomic Systems, 1(2) (2021).
[14] Sean Foley, Tom Meling, and Bernt Arne Ødegaard. 2021. Tick Size Wars: The Market Quality Effects of Pricing Grid Competition. Available at SSRN 2866943 (2021).
[15] Michael Neuder, Rithvik Rao, Daniel J. Moroz, and David C. Parkes. 2021. Strategic Liquidity Provision in Uniswap v3. CoRR abs/2106.12033 (2021). arXiv:2106.12033 https://arxiv.org/abs/2106.12033
[16] John W Pratt. 1978. Risk aversion in the small and in the large. In Uncertainty in economics. Elsevier, 59–79.