# Differential Privacy in Constant Function Market Makers

Tarun Chitra
tarun@gauntlet.network

Guillermo Angeris
angeris@stanford.edu

Alex Evans
ahe4nc@gmail.com

August 2021

**Abstract**

Constant function market makers (CFMMs) are the most popular mechanism for facilitating decentralized trading. While these mechanisms have facilitated hundreds of billions of dollars of trades, they provide users with little to no privacy. Recent work illustrates that privacy cannot be achieved in CFMMs without forcing worse pricing and/or latency on end users. This paper more precisely quantifies the trade-off between pricing and privacy in CFMMs. We analyze a simple privacy-enhancing mechanism called *Uniform Random Execution* and prove that it provides $(\epsilon, \delta)$-differential privacy. The privacy parameter $\epsilon$ depends on the curvature of the CFMM trading function and the number of trades executed. This mechanism can be implemented in any blockchain system that allows smart contracts to access a verifiable random function. We also investigate the worst case complexity over all private CFMM mechanisms using recent results from private PAC learning. These results suggest that one cannot do much better than Uniform Random Execution in CFMMs with non-zero curvature. Our results provide an optimistic outlook on providing partial privacy in CFMMs.

## 1 Introduction

Constant function market makers (CFMMs) have become the most widely used decentralized crypto product. In 2021, these market makers were facilitating over a billion dollars of daily volume, almost as much as centralized exchanges such as Binance, Coinbase, or FTX. These market makers allow those looking for passive yield on a portfolio of assets to be automatically matched with traders looking to execute a swap against their assets. CFMMs work by ensuring that an invariant known as the trading function is kept constant before and after a trade is executed. The trading function, which is a function of the liquidity provided by those seeking passive yield, controls the price displayed by the CFMM that traders can execute a trade at. In order to ensure that liquidity providers (LPs) do not always lose money, as they are effectively buying the currency whose value is going down in exchange for one that is going up, a trading fee is applied to each transaction. Prior work

[AC20, AAE$^+$21, AEC20] has investigated necessary and sufficient conditions for the trading function and choice of fee to lead to profitable outcomes for LPs.

One major problem with CFMMs is their lack of privacy. At a high-level, privacy in CFMMs boils down to preventing an adversary from discerning trade sizes as a function of public prices. The simplest CFMM, the constant sum market maker [AAE$^+$21, §2.4], is the prototypical example of a private CFMM. Constant sum market makers only quote a single price at all times, meaning that an adversary cannot determine if there was a single trade of size 100 or 100 trades of size 1. On the other hand, constant sum market makers have adverse selection for liquidity providers, as they effectively purchase large quantities of an asset whose price is decreasing elsewhere [AEC20, §3]. Quantifying this trade-off between the idealized 'perfect' privacy of a constant sum market maker and the adverse selection faced by LPs is crucial to designing better CFMMs. Furthermore, the dramatic increase in miner extractable value and front-running on Ethereum makes transaction-level privacy increasingly important [QZLG20, QZG21, ZQT$^+$20, AEC21a].

Prior work [AEC21b] has shown that given an initial amount of liquidity, public access to an oracle that furnishes price as a function of trade size, and prices of executed trades, one can uniquely identify the sizes of executed trades. This is a natural (although somewhat indirect) consequence of the concavity of the trading function [AC20, AAE$^+$21]. This work implies that even with modern cryptography such as zero-knowledge proofs (ZKPs), one will need to modify the CFMM mechanism to blind user's trade sizes. For instance, this implies that hiding balances via ZKPs of reserves (which has been proposed and implemented in multiple protocols [CXZ20, Pow21]) is not sufficient for transaction-level privacy.

The two main options presented in [AEC21b] for recovering privacy involve either modifying prices (e.g. adding noise to quoted prices) or batching transactions.[1] Both of these changes will worsen user experience in CFMMs — both options force traders to bear worse price impact while the latter option also means that users face higher latencies for trade confirmation. Assuming that these are the only options available, a natural question to ask is how well can we control the trade-off between worsened price and latency and improved transaction privacy. More specifically, we want to answer two questions:

- What is the minimum number of samples $n(\delta)$ such that an adversary is unable to infer the true trade sizes beyond a precision $\delta$?
- How much worse is the worst price offered to a user via such a mechanism?

The answer the former question is an analogue of sample complexity from learning theory whereas the latter question is measures 'the cost of privacy'.

One framework for answering questions of this form is differential privacy [DR$^+$14]. Differentially private algorithms aim to hide individual user data (e.g. trades) while simultaneously preserving aggregate statistics (e.g. prices or averages). Many differentially private mechanisms work by adding targeted randomness to each individual users' data. As a simple illustration, suppose that we have a sequence of values $x_1, \dots, x_n$ and we want to report

[1] There are two live batching CFMMs in production, CowSwap on Ethereum [Mar21] and ZSwap which relies on a specialized ZKP chain [dV21].

the mean $\mu(S)$ of any sufficiently large subset $S \subset [n]$ without revealing each $x_i$. One can preserve the mean $\mu$ by adding i.i.d., mean zero noise to each value and then then allow users to query for means of subsets. Differentially private algorithms induce a natural trade-off between the privacy and accuracy of a query, much like the trade-off between price impact and privacy in CFMMs. Differentially private algorithms have been used at large-scale and in production at the US Census [Dwo19], Google [ACG$^+$16], and Apple [CJK$^+$18].

We show that this comparison between differential privacy and CFMMs is more than superficial. We construct a black-box algorithm called *Uniform Random Execution* (URE) for making CFMM trade execution differentially private. This algorithm can be viewed as the inverse of batching as it breaks up and splits large trades before subsequently randomly permuting the trade ordering. Randomness is used for both splitting up large trades and for permuting the split up trades. Blockchains with smart contract capabilities that include CFMM ordering as part of consensus rules (e.g. Celo [KOR19], Terra [MSS20], Penumbra [dV21], Osmosis [AO21]) can execute the URE. In particular, any blockchain with a verifiable random function (VRF) [MRV99] that provides public randomness and consensus rules for executing trades in a particular order suffices for URE.

Our analysis of the differential privacy of URE utilizes a novel representation of a sequence of trades as a binary tree. The tree is constructed such that the height of the tree provides a lower bound on the worst case price impact. On the other hand, number of leaves of the tree controls how easy it is for one to invert the precise trades executed. Representing continuous objects (sequences of real-valued trades) as a random discrete data structure allows us to utilize traditional tools from differential privacy. We show that the trade tree controls the maximum price impact of a sequence of trades by utilizing curvature of a CFMM [AEC20, §2]. Curvature represents bounds on market impact cost and liquidity and is crucial for relating the trade tree to worst-case price impact. Subsequently, we analyze the impact of splitting up and randomly permuting trades on the trade tree and then compute bounds on the price impact associated to these actions.

In order to achieve differential privacy, we first prove that splitting up trades can be executed in a differentially private manner (Claim 2). The proof utilizes a recent result for differentially private sampling of Dirichlet distributions [GWH$^+$21]. To split a trade, we sample a random distribution $\pi$ and the split up a single trade according to $\pi$. After splitting up the trades, we then show that randomly permuting the trades leads to an $(\epsilon, \delta)$-differentially private algorithm. We use composition laws [DR$^+$14, KOV15] to combine these two results and show that the URE is differentially private. We note that $\epsilon$ and $\delta$ depend on the CFMM's curvature and logarithmically on the number of trades executed.

One other natural framework for analyzing the privacy vs. price impact trade-off is that of (differentially) private probably approximately correct (PAC) learning [KLN$^+$11]. This framework allows for worst case bounds on adversarial learning. Broadly speaking, PAC learning aims answer when an algorithm can learn a hypothesis (e.g. binary classification) with error less than $\epsilon$ with high probability given at least $n$ samples. Private PAC learning asks the same question but with the constraint that the learning algorithm is differentially private. This means that the learning algorithm can still learn the correct classification without being able to identify (with probability) if a small number of the input data points were changed.

Recent results have begun to classify the types of algorithms that can be made differentially private. One particular line of work has shown that private PAC learnable and online learnable function classes can be related via a combinatorial quantity known as the Littlestone dimension [ALMM19, BLM20, Bun20, JKT20]. The Littlestone dimension of a function class $\mathcal{F}$ is an analogue of VC dimension that measures how well $\mathcal{F}$ shatters ordered sequences of data points. It is defined by constructing a tree of possible sequences of data points and then defining an appropriate analogue of shattering for trees.

Our random binary tree construction allows us to place bounds on the Littlestone dimension of classes of threshold and decision tree learners [RST10]. These learners aim to learn the optimal placement of a front-run or back-run transactions [QZLG20, QZG21] and our bounds effectively show that with enough noise and curvature in the trading function, the Littlestone dimension of this task is $\Omega(\log n)$. The equivalence of differentially private algorithms and online learning algorithms for binary classification [ALMM19, BLM20] therefore implies lower bounds on how well online learning algorithms can deanonymize trade sizes. Combined, these results suggest that if a CFMM has non-zero curvature one cannot do substantially better than the proposed permute and split algorithm.

# 2 Preliminaries

We will cover preliminaries on CFMMs and differential privacy. For more details, please refer to review articles on CFMMs [AAE+21] and differential privacy [DR+14].

## 2.1 Constant function market makers

In this section we describe how CFMMs work. We consider a decentralized exchange (DEX) with $n > 1$ assets, labeled $1, \ldots, n$, that implements a CFMM. Asset $n$ is our numeraire, the asset we use to value and assign prices to the others.

### 2.1.1 CFMM state

**Reserve or pool.** The DEX has some *reserves* of available assets, given by the vector $R \in \mathbf{R}^n_+$, where $R_i$ is the quantity of asset $i$ in the reserves.

**Proposed trade** A *proposed trade* (or *proposed exchange*) is initiated by an agent or trader, who proposes to trade or exchange one basket of assets for another. A proposed trade specifies the *tender basket*, with quantities given by $\Delta \in \mathbf{R}^n_+$, which is the basket of assets the trader proposes to give (or tender) to the DEX, and the *received basket*, the basket of assets the trader proposes to receive from the DEX in return, with quantities given by $\Lambda \in \mathbf{R}^n_+$. Here $\Delta_i$ ($\Lambda_i$) denotes the amount of asset $i$ that the trader proposes to tender to the DEX (receive from the DEX).

The proposed trade can either be rejected by the DEX, in which case its state does not change, or accepted, in which case the basket $\Delta$ is transfered from the trader to the DEX, and the basket $\Lambda$ is transfered from the DEX to the trader. The DEX reserves are updated as

$$R^+ = R + \Delta - \Lambda, \tag{1}$$

where $R^+$ denotes the new reserves. A proposed trade is accepted or rejected based on a simple condition which always ensures that $R^+ \geq 0$.

**Trading function** Trade acceptance depends on both the proposed trade and the current reserves. A proposed trade $(\Delta, \Lambda)$ is accepted only if

$$\psi(R + \gamma\Delta - \Lambda) = \psi(R), \tag{2}$$

where $\psi : \mathbf{R}^n_+ \to \mathbf{R}$ is the *trading function* associated with the CFMM, and the parameter $\gamma \in (0, 1]$ introduces a *trading fee* (when $\gamma < 1$). The 'constant function' in the name CFMM refers to the acceptance condition (2).

We can interpret the trade acceptance condition as follows. If $\gamma = 1$, a proposed trade is accepted only if the quantity $\phi(R)$ does not change, *i.e.*, $\phi(R^+) = \phi(R)$. When $\gamma < 1$ (with typical values being very close to one), the proposed trade is accepted based on the devalued tendered basket $\gamma\Delta$. The reserves, however, are updated based on the full tendered basket $\Delta$ as in (1).

**Assumptions.** We will, in general, assume that the trading function $\psi$ is a strictly concave, increasing function, which holds for essentially all CFMMs barring some special cases such as constant sum market makers. This is true for any CFMM whose reachable set [AC20, §2.3] is a strictly convex set, for all reserves $R$. For example, in the case of Uniswap, or constant product markets, $\psi(R) = R_1R_2$, which is neither concave nor convex, but it can be equivalently written as $\psi(R) = \sqrt{R_1R_2}$, which is strictly concave, increasing whenever $R > 0$. (The notion of equivalence used here is that of [AC20, §2.1], which we will not discuss further in this paper.)

**Curvature.** We briefly summarize the main definitions and results of [AEC20] here; please refer to that paper for further details. Suppose that we have a two-asset CFMM with reserves $R, R'$ and that the trading function $\psi$ is differentiable. The marginal price for a trade of size $\Delta$ is

$$g(\Delta) = -\frac{\partial_1\psi(R, R', \Delta, \Delta')}{\partial_2\psi(R, R', \Delta, \Delta')}$$

where the trade size $\Delta'$ is specified by the implicit condition $\psi(R, R', \Delta, \Delta') = \psi(R, R')$. This is known as the *price impact* function as it represents the output price for a positive sized trade. When there are fees, one can show that $g^{fee}(\Delta) = \gamma g(\gamma\Delta)$. We say that a CFMM is *$\mu$-stable* if it satisfies

$$g(0) - g(-\Delta) \leq \mu\Delta$$

for all $\Delta \in [0, M]$ for a positive $M$. This is a linear upper bound on the maximum price impact that a bounded trade (bounded by $M$) can have. Similarly, we say that a CFMM is $\kappa$-liquid if it satisfies

$$g(0) - g(-\Delta) \geq \kappa\Delta$$

for all $\Delta \in [0, K]$ for a positive $K$. We can define similar upper and lower bounds for $g(\Delta) - g(0)$, $\mu', \kappa'$ which hold when trades $\Delta$ are in intervals $[-M', 0], [-K', 0]$, respectively For the remainder of this paper, we will refer to $\mu$-stability as the upper bound for both $g(0) - g(-\Delta)$ and $g(\Delta) - g(0)$ (and likely for $\kappa$-liquidity). More specificially, given $\mu, \mu'$, we say that a CFMM is symmetrically $\mu''$-stable if $\mu'' = \min(\mu, \mu')$ and symmetrically $\kappa''$-liquid if $\kappa'' = \max(\kappa, \kappa')$.

Angeris, et. al [AEC20] show that the curvature ratio $\frac{\mu}{\kappa}$ controls no-arbitrage pricing when a CFMM is the primary market (e.g. other markets tend to be arbitraged against the CFMM). They also demonstrate that classical economics results, such as the requirement of noise traders for LP profitability, occur and are controlled by curvature parameters $\mu$ and $\kappa$. CFMM curvature parameters can be considered analogous to Kyle's $\lambda$ liquidity parameter [Kyl85] for order books. This analogy is also used to estimate optimal (under no-arbitrage) subsidies needed to compensate LPs for so-called impermanent loss. The higher the CFMM liquidity parameter $\kappa$, the higher the price impact observed. Intuitively, this should control how easy it is to invert whether a trade of size $> \Delta^*$ occurred. In the remainder of this paper, we will focus on using CFMM curvature parameters to bound the impact cost realized, which in turn controls how easily an adversary can invert a trade size from prices.

### 2.2 Differential Privacy

Differential privacy is a framework for classifying how well a randomized algorithm $\mathcal{A}$ anonymizing individual data points.

**Definition 1.** *A randomized algorithm $\mathcal{A}$ is $(\epsilon, \delta)$-*differentially private *if for all $S, S' \in$ $\mathbf{Dom}\,\mathcal{A}$ with $d(S, S') \leq 1$ we have for all measurable $B \subset \mathbf{Range}\,\mathcal{A}$*

$$\mathbf{Prob}[\mathcal{A}(S) \in B] \leq e^{\epsilon}\,\mathbf{Prob}[\mathcal{A}(S') \in B] + \delta$$

In this definition, $\epsilon$ can be thought of as a uniform upper bound on the Kullback-Leibler divergence over the distribution induced by any pair of neighboring data sets. Traditionally, $S, S'$ are thought of as discrete and the metric $d$ corresponded to the Hamming metric. In this paper, we will assume $d$ is the $L^1$ norm, which can still provide differential privacy [DR$^+$14, NRS07]. We provide further details on differential privacy in Appendix A.

## 3 Problem Construction

Angeris, et. al [AEC21b, §3] provide two mitigations for the loss of privacy in CFMMs:

1. *Randomizing price*: One can randomly affect the price quoted by the CFMM in manner resistant to adversaries (while also not destroying liquidity provider returns)

2. *Batching orders*: Picking a size $n \in \mathbf{N}$ of orders to batch prior to execution

Neither of these solutions are perfect however and in [AEC21b], there is no adversarial model for assessing these solutions. The goal of this section is to do this by first formulating a simple adversarial threat model for these solutions and then introducing SURE and URE. We prove that the URE achieves differential privacy with parameters dependent on the number of trades and the curvature. Our results also codify the intuition that the constant sum market maker (which is excluded from the results of [AEC21b]) has "perfect" privacy unlike curved market makers. For the remainder of the paper, we will assume that there are two assets (in order to utilize curvature bounds) and $n$ will refer to a number of trades.

## 3.1 Threat Model

**Adversary definition and attack.** We assume a simple model of an adversary that matches that of [AEC21b]. The adversary, who we will call Eve, attempts to discover the quantities traded by a set of agents referred to as Traders. Eve is unable to see the exact quantities the Traders use to trade with the CFMM, but knows when the Traders transactions $\Delta_1, \ldots, \Delta_n$ are executed as a block. In particular, Eve does not know the order in which the trades are executed and her goal is to estimate the order and sizes of the trades. Eve's only ability is to interact with the CFMM in a state before Alice's transactions and after their transactions are executed.

**Action space.** We assume that Eve has access to two queries:

- $\mathsf{marginalPrice}()$: Computes the marginal (spot) price of the CFMM at its current reserves
- $\mathsf{isValid}(\Delta)$: Takes a trade $\Delta \in \mathbf{R}$ and returns $\mathsf{True}$ if the trade is valid and $\mathsf{False}$ otherwise

We will denote the set of valid trades at reserves $R \in \mathbf{R}^n_+$ as $\mathcal{A}_\varphi(R)$ and note that it can effectively be thought of as the epigraph of the trading function $\varphi$ [AC20].

## 3.2 Simple Uniform Random Execution

One of the simplest ways to introduce entropy into a CFMM is to randomly permute the set of trades to be executed. We will first describe the *simple uniform random execution* (SURE) mechanism that simply permutes observed trades. In practice, this can be executed on-chain provided access to a verifiable random function [MRV99], which exists on chains such as Polkadot [BCC$^+$20] and Cosmos [Buc16].

Formally, suppose that we are given a vector of valid trades

$$\Delta_1 \in \mathcal{A}_\varphi(R), \Delta_i \in \mathcal{A}_\varphi\left(R + \sum_{j=1}^{i-1} \Delta_i\right)$$

For brevity, we will refer to above condition as $\mathcal{A}_\varphi(\mathbf{\Delta})$ for a trade vector $\mathbf{\Delta}$. The SURE mechanism draws a random permutation $\pi \sim_{\mathsf{Unif}} S_n$ and constructs a sequence of trades $\Delta_i^\pi = \Delta_{\pi(i)}$, which arise from permuting the order in which the trades are executed. Consider the marginal prices of the original trades $p_1, \dots, p_n$ and and the permuted prices $p_1^\pi, \dots, p_n^\pi$. Note $p_n = p_{\pi(n)}$ if and only if the CFMM is path-independent (e.g. feeless). Our goal is two-fold: first, we aim to bound the maximum deviation between the true price $p$ and the permuted prices $p^\pi$. That is, we want to compute

$$\mathcal{E}_{SURE} = \mathop{\mathbf{E}}_{\pi \sim S_n}\left[\max_{i \in [n]} |p^\pi(i) - p(i)|\right]$$

This deviation effectively corresponds to a bound on the worst quoted price that a trader can receive (relative to their original order price). Secondly, we want to capture a notion of how difficult it is for an adversary to learn the values of $\pi$ chosen given only the prices $p_n^\pi$.

Before analyzing the SURE mechanism for some classes of trades, let's look at some simple examples. If all of the trades $\Delta_i$ are unique — e.g. $\nexists i, j \in [n]$ such that $\Delta_i = \Delta_j$ — then computing $\Delta_i$ given $p^\pi$ in some sense be difficult to invert to a precision higher than $\kappa \min_{i,j} |\Delta_i - \Delta_j|$. This is because if $\pi$ is a single adjacent transposition $(i\ i+1)$, then $g(\sum_{j=1}^i \Delta_i) - g(\sum_{j=1}^{i-1} \Delta_j) \geq \kappa \min(\Delta_i, \Delta_{i-1}) \geq \kappa \min_{i,j} |\Delta_i - \Delta_j|$. Moreover, we should expect that SURE should work better when $\sum_{i=1}^n \mathsf{sgn}(\Delta_i) \approx 0$. This is because the probability of having a long run of trades in the same direction is very low. For instance, if $\Delta_i$ is a Rademacher random variable (e.g. uniformly drawn from $\{-1, 1\}$) then the expected maximum length of a run is $\Theta(\log n)$ [ER75, Theorem 1].

On the other hand, if there is a set $S \subset [n]$ with $|S| = \Omega(n)$ such that for all $i, j \in S, \Delta_i = \Delta_j$, then it will be much easier to invert the set of trades. There is a loss of entropy in the output trade sequences as there will be many permutations $\pi, \pi', \pi \neq \pi'$ such that $\Delta^\pi = \Delta^{\pi'}$. Let's consider an explicit numerical example. Let $\Delta_1 = 100$ and $\Delta_i = 1$ for all $i \in \{2, \dots, n\}$. Even though we are sampling from $n!$ permutations, there are only $n$ output sequences that SURE output: $\Delta_j^\pi = \Delta_{\pi(1)} = 100$ in the $j$th position for $j \in [n]$. Suppose we consider a permutation $\pi$ with $\pi(1) = j$. For any trade in position $i$ with $\pi(i) < j$, the trade gets significantly better execution than they did initially. This is because their trade is executed before the trade of size 100 is executed, giving them significantly less impact. Therefore, SURE requires the trade distribution to have sufficient entropy and the distribution of trade sizes to not be too concentrated in order to work.

We will first analyze SURE on a subset of the total set of allowable input trades. This subset will be defined via simple constraints on $\min_{i,j} |\Delta_i - \Delta_j|$. We will later relax these by splitting up large trades in a manner that ensure that the trade size distribution satisfies these constraints with high probability. To analyze SURE, we will start by obtaining upper and lower bounds on the worst case expected price discrepancy, $\mathbf{E}[\max_i |p^\pi(i) - p(i)|]$. This analysis will provide insight into what subset of admissible trades provide provable bounds on price discrepancy and identifiability.

**Maximum of the Price Process and Random Binary Trees.** Suppose the price impact function $g$ is $\kappa$-liquid and $\mu$-stable on an interval $[-M, M]$. By definition this implies

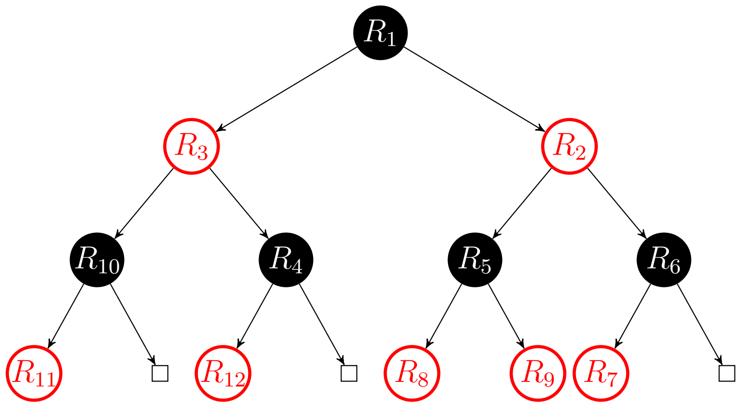


**Figure 1:** Depiction of the tree $T(\mathbf{R}(\boldsymbol{\Delta}, \pi))$ where $R_i = R_i(\boldsymbol{\Delta}, \pi)$ and $R_{11} < R_{10} < R_3 < R_{12} < R_4 < R_1 < R_8 < R_5 < R_9 < R_2 < R_7 < R_6$

that for all $i \in [n]$

$$\sum_{j=1}^{i} \kappa\Delta_{\pi(j)} - \mu\Delta_j \leq p^\pi(i) - p(i) \leq \sum_{j=1}^{i} \mu\Delta_{\pi(j)} - \kappa\Delta_j$$

This means that we have

$$\kappa\, \mathbf{E}\left[\max_i \left|\sum_{j=1}^{i} \Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j\right|\right] \leq \mathbf{E}[\max_i |p^\pi(i) - p(i)|] \quad \leq \mu\, \mathbf{E}\left[\max_i \left|\sum_{j=1}^{i} \Delta_{\pi(j)} - \frac{\kappa}{\mu}\Delta_j\right|\right] \quad (3)$$

Therefore, bounds on partial sums of permuted trades will allow us to bound the worst case price impact of SURE. Define the partial sum $R_i(\boldsymbol{\Delta}, \pi) = \sum_{j=1}^{i} \Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j$. Now consider the binary search tree $T(\mathbf{R}(\boldsymbol{\Delta}, \pi))$ whose root is $R_1(\boldsymbol{\Delta}, \pi)$. Each element $R_j(\boldsymbol{\Delta}, \pi)$ is inserted sequentially to construct the tree (see Figure 1 for an example).

This representation of the partial sums as a tree provides a natural geometric description of the maximum price deviation. In particular, $\max_i R_i(\boldsymbol{\Delta}, \pi)$ is necessarily a leaf node in this tree. This provides the following bounds using (3)

$$\max_i |p^\pi(i) - p(i)| \leq \mu \left(|R_1(\boldsymbol{\Delta}, \pi)| + \max_j \left|\Delta_{\pi(j)} - \frac{\kappa}{\mu}\Delta_j\right| \cdot \mathsf{height}(T(\mathbf{R}(\boldsymbol{\Delta}, \pi)))\right) \quad (4)$$

$$\max_i |p^\pi(i) - p(i)| \geq \kappa \left|R_1(\boldsymbol{\Delta}, \pi) + \min_j \left(\Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j\right) \cdot \mathsf{height}(T(\mathbf{R}(\boldsymbol{\Delta}, \pi)))\right| + O(1) \quad (5)$$

Note that the second bound comes from bounded support of curvature (§2.1.1):

$$\begin{aligned} \max_j |R_j(\boldsymbol{\Delta}, \pi)| \geq |R_j(\boldsymbol{\Delta}, \pi)| &= \left|R_1(\boldsymbol{\Delta}, \pi) + \sum_{i=1}^{j} \left(\Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j\right)\right| \\ &\geq \left|R_1(\boldsymbol{\Delta}, \pi) + \min_j \left(\Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j\right) \cdot \mathsf{height}(T(\mathbf{R}(\boldsymbol{\Delta}, \pi)))\right| + O(1) \end{aligned}$$

Moreover, the number of leaves in the tree represent the number of left-to-right local maxima of $R_j$. Note, furthermore, that by using curvature and the tree structure, we have reduced the maximum price deviation problem (a continuous problem) into a combinatorial one regarding a random tree. If the tree is roughly balanced (e.g. height is $O(\log n)$) and there are $\Omega(n)$ leave nodes then it is unlikely that a small change to the permutation $\pi$ by a transposition will change the maximum value. We will formalize this by studying the behavior of the random variable $T(\mathbf{\Delta})$, which draws a permutation $\pi$ randomly and sets $T(\mathbf{\Delta}) = T(\mathbf{R}(\mathbf{\Delta}, \pi))$.

To study the behavior of $T(\mathbf{\Delta})$, we need to analyze the expected height of a random binary tree. It is known that the height of a random binary tree with distinct elements (e.g. such that every permutation is equiprobable) has height $\Theta(\log n)$ with high probability:

**Theorem 1** ([Ree03], Theorem 1)**.** *Let* $\mathbf{\Delta}$ *have unique elements. Then* $\mathbf{E}[\mathsf{height}(T(\mathbf{\Delta}))] = \alpha \log n - \beta \log\log n$ *and* $\mathbf{Var}[\mathsf{height}(T(\mathbf{\Delta}))] = O(1)$

If we can guarantee that the elements of $T(\mathbf{\Delta})$ are distinct (e.g. such that every permutation of $\mathbf{\Delta}$ is equiprobable) then combining this result with (4) yields

$$\begin{aligned}
\mathbf{E}[\max_i |p^\pi(i) - p(i)|] &\leq \mu \left( \mathbf{E}[R_1(\mathbf{\Delta}, \pi)] + \max_{i,j} \left| \Delta_i - \frac{\kappa}{\mu}\Delta_j \right| \mathbf{E}[\mathsf{height}(T(\mathbf{\Delta}))] \right) \\
&\leq \mu \left( \mathbf{E}[R_1(\mathbf{\Delta}, \pi)] + \max_{i,j} \left| \Delta_i - \frac{\kappa}{\mu}\Delta_j \right| (\alpha \log n - \beta \log\log n) \right) \\
&\leq \mu \left( \max_{i,j} \left| \Delta_i - \frac{\kappa}{\mu}\Delta_j \right| \right) (\alpha \log n - \beta \log\log n + 1) \qquad (6)
\end{aligned}$$

where we used the upper bounds $\max_j \left| \Delta_{\pi(j)} - \frac{\kappa}{\mu}\Delta_j \right| \leq \max_{i,j} \left| \Delta_i - \frac{\kappa}{\mu}\Delta_j \right|$ and

$$\mathbf{E}[R_1(\mathbf{\Delta}, \pi)] = \frac{1}{n} \sum_{j=1}^{n} \left| \Delta_j - \frac{\kappa}{\mu}\Delta_1 \right| \leq \max_{i,j} \left| \Delta_i - \frac{\kappa}{\mu}\Delta_j \right|$$

Similarly, note that $\mathbf{E}[R_1(\mathbf{\Delta}, \pi)] \geq \min_j \left( \Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j \right)$ so we have

$$\begin{aligned}
\mathbf{E}[\max_i |p^\pi(i) - p(i)|] &\geq \kappa \left| \min_j \Delta_{\pi(j)} - \frac{\mu}{\kappa}\Delta_j \right| (\mathbf{E}[\mathsf{height}(T(\mathbf{\Delta})) + O(1)]) \\
&\geq \kappa \left| \min_{i,j} \Delta_i - \frac{\mu}{\kappa}\Delta_j \right| (\alpha \log n - \beta \log\log n + O(1))
\end{aligned}$$

Therefore, provided that the following two conditions hold

$$\Delta_{\min} = \left| \min_{i,j} \Delta_i - \frac{\mu}{\kappa}\Delta_j \right| = \Omega(1) \qquad\qquad \Delta_{\max} = \left| \max_{i,j} \Delta_i - \frac{\kappa}{\mu}\Delta_j \right| = O(1) \qquad (7)$$

we have $\mathcal{E}_{SURE} = \mathbf{E}[\max_i |p^\pi(i) - p(i)|] = \Theta(\log n)$. Such a bound is ideal as it ensures that there is always a minimum price discrepancy of $\Omega(\kappa \log n)$ so that an adversary cannot determine a trade size with precision greater than $\Omega(\kappa)$. On the other hand, the upper bound on price deviation means that the mechanism will not cause too great of a price impact for users.

Note that the usage of Theorem 1 is prefaced on every permutation of the elements of $R_j(\mathbf{\Delta}, \pi)$ being equiprobable. One simple example of when this isn't true is from the threshold trades, $\mathbf{\Delta} = (T, 1, \ldots, 1) \in \mathbf{R}^T$ when $\mu \geq 100\kappa$. When this is true, neither of the conditions (7) hold and moreover, the conditions of Theorem 1 do not hold. This means that SURE only works when (a) all permutations of partial sums are unique and (b) when $\mu \leq (\max_i \Delta_i)\kappa$. In the next section, we will achieve (a) by adding noise dependent on $\mathbf{\Delta}, \mu, \kappa$ to the trades and (b) by splitting trades.

## 3.3 Uniform Random Execution

We have seen the SURE mechanism works well at providing privacy while minimizing price discrepancy when (7) holds, when elements of $\mathbf{\Delta}$ are unique, and when $\frac{\mu}{\kappa}$ is not too large. However, we're not guaranteed that both of these conditions hold in general as illustrated by the example at the end of the last section. This section will focus on using randomization to ensure that a) (7) holds with high probability and b) the elements of $\mathbf{\Delta}$ are unique. We will do this by performing two actions: splitting large trades to ensure the maximum condition holds and adding noise to trades to ensure that trades are not too close in size. Applying these two actions to $\mathbf{\Delta}$ and subsequently executing SURE is termed the *Uniform Random Execution* mechanism. There are three parameters that control the URE mechanism:

- $c_{\min}$: Lower bound on $\Delta_{\min}$
- $s \in \mathbf{R}_+$: Split threshold that controls the average chunk size for a big trade
- $k \in \mathbf{N}$: Multiple of $(1+s)\Delta_{\min}$ that requires splitting

**Lower bounding the minimum by adding Laplace noise.** Our goal is to construct random variables $\xi_1, \ldots, \xi_n$ drawn i.i.d. from a distribution that can depend on a particular $\mathbf{\Delta}$ but guarantees that $\tilde{\mathbf{\Delta}} = \mathbf{\Delta} + \xi$ satisfies the left hand side of (7) with high probability. In particular, we would like to control $\mathbf{Prob}\left[\left|\min_{i,j} \tilde{\Delta}_i - \frac{\kappa}{\mu}\tilde{\Delta}_j\right| > c_{\min}\right]$ for a constant $c_{\min} > 0$. We desire the following condition to hold bounded above by $\delta \in (0,1)$:

$$\begin{aligned}
\mathbf{Prob}\left[\left|\min_{i,j} \tilde{\Delta}_i - \frac{\kappa}{\mu}\tilde{\Delta}_j\right| \leq c_{\min}\right] &= \mathbf{Prob}\left[\left|\min_{i,j} \Delta_i + \xi_i - \frac{\kappa}{\mu}(\Delta_j + \xi_j)\right| \leq c_{\min}\right] \\
&\leq \mathbf{Prob}\left[-\left|\min_{i,j} \Delta_i - \frac{\kappa}{\mu}\Delta_j\right| + \left|\min_{i,j} \xi_i - \frac{\kappa}{\mu}\xi_j\right| \leq c_{\min}\right] \\
&= \mathbf{Prob}\left[\left|\min_{i,j} \xi_i - \frac{\kappa}{\mu}\xi_j\right| \leq c_{\min} + \left|\min_{i,j} \Delta_i - \frac{\kappa}{\mu}\Delta_j\right|\right] \leq \delta
\end{aligned} \tag{8}$$

In Appendix C, we prove the following claim:

**Claim 1.** *There exists $a \in \mathbf{R}$ dependent on $\boldsymbol{\Delta}, \mu, \kappa$ and $\xi_i \sim \mathsf{Lap}(a, |a|)$ such that* (8) *holds*

This mechanism can be naturally modified to inherit the $\epsilon$-privacy guarantees of the Laplace mechanism [DR+14, §3.2]. Note that the dependence of the noise parameter $a$ on $\boldsymbol{\Delta}$ is similar to smoothed sensitivity in differential privacy [NRS07]. We note that this added noise ensures both that the lower bound of (7) holds and ensures that the elements of $\boldsymbol{\Delta} + \xi$ are unique so that Theorem 1 holds.

**Upper bounding the maximum by splitting trades.** One way to reduce the upper bound on error in (6) is to split up a trade $\Delta_i$. This reduces $\Delta_{\max}$ and as explained in Appendix G, also increases the privacy of SURE. More precisely, we split $\Delta_i$ into $\Delta_i', \Delta_i''$ with $\Delta_i = \Delta_i' + \Delta_i''$ and then consider the pricing error associated to $p(\boldsymbol{\Delta}')$ where $\boldsymbol{\Delta}' = (\Delta_1, \ldots, \Delta_{i-1}, \Delta_i', \Delta_i'', \Delta_{i+1}, \ldots, \Delta_n)$. This process can be iterated until all trades meet a particular criteria. Instead splitting trades in two, we instead split trades into $m(\Delta_i)$ pieces, where $m(\Delta_i)$ is defined as

$$m(\Delta_i) = \max\left(1, \left\lceil \frac{|\Delta_i|}{(1+s)\Delta_{\min}} \right\rceil\right)$$

That is, the mechanism splits the trade into $m(\Delta_i)$ pieces who sizes are roughly $(1+s)\Delta_{\min}$. Let $\mathbf{1}^m = (1, \ldots, 1)$. For any trade $\Delta_i$ with $m(\Delta_i) > 1$, we draw $\pi \sim \mathsf{Dir}(\mathbf{1}^{m(\Delta_i)})$ and split $\Delta_i$ into trades $\Delta_{i,j} = \Delta_i \pi_j$. Since $\sum_{j=1}^n \pi_j = 1$, this provides a natural mechanism for splitting trades in a single step. As the Dirichlet distribution is sub-Gaussian when using uniform weights [MA17] and as the expected order statistics of a Dirichlet process decay exponentially [BJP12], $\mathbf{Prob}[\Delta_{i,j} - (1+s)\Delta_{\min} > k\Delta_{\min}]$ also decays exponentially in $k$. This ensures that we have very few chunks that are significantly greater than $(1+s)\Delta_{\min}$, which ensures that with high probability $\max_i \Delta_i < (1+s+k)\Delta_{\min}$. As described in Appendices D and G, this condition ensures that SURE is effective with high probability. We note that the precise price impact of splitting trades (as a function of curvature) is analyzed in [AEC20].

## 3.4 Differential Privacy

We are now in a position to prove that the URE mechanism satisfies $(\epsilon, \delta)$-differential privacy, where $\epsilon = O(\mu \log n + \max_i \Delta_i)$. Our proof proceeds in two steps. First, we prove the following claim in the Appendix E.

**Claim 2** (Splitting is differentially private)**.** *Suppose that we have a sequence of admissible trades $\boldsymbol{\Delta} \in \mathbf{R}^n$ and after adding noise we have $\tilde{\boldsymbol{\Delta}}$ with $\tilde{\Delta}_{\min} > 0$. For each $k \in \mathbf{N}$ define $S_k = \{j : \tilde{\Delta}_j > k\tilde{\Delta}_{\min}\}$. If $\delta^* = \max_j \frac{\Delta_j}{\Delta_{\min}} = O(n)$ and there exists $k > 0$ such that $|S_k| = O(1)$, then there exists an $(\epsilon, \delta)$-differentially private algorthm $\mathsf{Split}(\Delta)$ for splitting trades in $\tilde{\Delta}$ such that $\mathsf{height}(T(\mathsf{Split}(\tilde{\Delta}))) = O(\log n)$ where $\epsilon = O(\delta^*)$*

This claim ensures that under mild conditions on the maximum trade size, we can generate a partial sum trade tree of height $O(\log n)$. Note that we can get the claim's conditions to be satisfied by varying $s$, the scale parameter, which leads to a privacy-utility trade-off. Second, we show that when a partial sum trade tree has height $O(\log n)$, permuting the trades provides $(O(\mu \log n), \delta)$-differential privacy for the maximum price impact (Claim 3). We combine these two differentially private algorithms using standard composition theorems (see Appendix), resulting in a differentially private CFMM.

**Claim 3** (SURE is differentially private)**.** *Suppose that we have a sequence of admissible trades $\mathbf{\Delta} \in \mathbf{R}^n$ such that $\mathsf{height}(T(\Delta)) = O(\log n)$ and all trade sizes are unique. Then randomly permuting the trades $\mathbf{\Delta}^\pi$ can be made into a $(\mu \log n, \delta)$-differential private algorithm for the minimum and maximum price impact*

While the full proof of the theorem is in the appendix, we sketch the steps of the proof below. First, we show that if a set of trades satisfies (7), then we can achieve differential privacy. We do this by first bounding the local sensitivity [DR+14] of the price impact vector $p_j(\pi, \Delta)$ as a function of $\Delta$. This is done by reducing the problem to analyzing two different price trees (Appendix B). We make an analogue of smooth sensitivity [NRS07] that rounds a vector of trades to an integer lattice whose length is $\Delta_{\text{min}}$. These steps ensure that the maximum difference in price impact between neighboring sets of trades will be $O(\mu \log n)$. This immediately leads to achieving $(\epsilon, \delta)$-differential privacy, where $\epsilon = O(\mu \log n)$.

Using the composition property of differential privacy, we are able to compose these two mechanisms to achieve $(\mu \log n + \max_i \Delta_i, \delta)$-differential privacy where $\delta = F^{-1}(O\left(\frac{1}{\epsilon}\right))$ and $F^{-1}$ is the inverse Laplace CDF. While the constants can likely be improved, this suggests that permuting and splitting up trades is a simple and viable mechanism for adding differential privacy to CFMMs. Finally, note that in Appendix F we provide a convex program that can split up trades more efficiently than the Dirichlet mechanism of Theorem 2. This is likely useful to practitioners where randomness is a constrained resource (e.g. on a blockchain).

# 4 Worst-Case Bounds and Path Deficiency

In this section, we'll explore if we can do better than the URE mechanism by analyzing the curvature of the mechanism and generalizing the previous work using Generic Chaining. Our goal will be to consider classes of mechanisms, $\mathcal{F}$, that can provide $(\epsilon, \delta)$-differential privacy for CFMMs and attempt to compute worst-case bounds. We're first provide some necessary conditions that elements of such a class have to satisfy. We will also show that extending the results of §3.4 to the path-deficient (positive fee) case involves proving bounds over a class of functions $\mathcal{F}$. Finally, we'll investigate connections to private PAC learning which suggest that one cannot do significantly better than the URE unless curvature is dynamically adjusted.

## 4.1 Mechanism Curvature

Instead of directly working with a mechanism, can we say something about the set of all mechanisms that ensure that $|p^m(i) - p^t(i)| > \delta$ where $p^m(i)$ is the $i$th price of the mechanism and $p^t(i)$ is the non-private or true price? Using curvature definition analogous to those of [AEC20], we can provide a simple bounds related to this question.

Note that bounds of the form $|p_i^m - p_i^t| > \delta$ involve bounding changes between two different price processes. Suppose that we define "curvatures" of the form

$$\begin{aligned}
\kappa_t|\Delta_i| &< |p^t(i) - p^t(i-1)| < \mu_t|\Delta_i| \\
\kappa_m|\Delta_i| &< |p^m(i) - p^m(i-1)| < \mu_m|\Delta_i| \\
\kappa_{mt}|\Delta_i| &< |p^m(i) - p^t(i)| < \mu_{mt}|\Delta_i|
\end{aligned}$$

First, let's look at the difference between the mechanism price at time $i$ and the true price at time $i-1$:

$$\begin{aligned}
|p^m(i) - p^t(i)| &= |(p^m(i) - p^t(i)) - (p^t(i-1) - p^t(i))| \\
&\geq |p^m(i) - p^t(i)| - |p^t(i) - p^t(i-1)| \\
&\geq (\kappa_m - \mu_{mt})|\Delta_i|
\end{aligned}$$

This says that we can ensure that the predictive value of previous price information on a trade cannot be resolved more than a multiplicative amount of $\kappa_m - \mu_{mt}$ times the trade size. In particular, $\kappa_m > \alpha + \mu_{mt}$ ensures that an adversary never has more than a precision $\alpha$ of information about the trade size. This provides a necessary condition in terms of mechanism curvature for a class $\mathcal{F}$ of mechanisms to provide differential privacy bounds.

## 4.2 Path Deficiency

Any CFMM that has non-zero fees (e.g. $\gamma = 1 - f < 1$) is path-deficient and has strictly negative expected value for round trip trades [AEC20]. Such CFMMs have price path $p^t(i)$ that are explicitly dependent on the trade ordering. Note that almost all CFMMs that are used in practice have non-zero fees to attract liquidity, so this is an importnt scenario to study. Previous work on path-deficient CFMMs has focused on analyzing how a particular price process (such as a geometric brownian motion) interacts with the exepcted returns from fees [EAC21]. Moreover, [AEC20, §2] illustrated that when fees are present $g^f(\Delta) = \gamma g(\gamma\Delta)$, where $g^f$ is the price impact function with fees and $g$ is the feeless price impact function. This suggests that we can analyze the path-dependent case by uniformly bounding the geometric parameters of §3.2 (e.g. height and number of leaves) as a function of the fee.

Suppose that given a trade vector $\mathbf{\Delta}$, we have a bound of the form

$$\mathop{\mathbf{E}}_{\pi \in S_n}\left[\max_{i \in [n]} |p_f^\pi(\mathbf{\Delta}) - p^\pi(\mathbf{\Delta})|\right] = O(\gamma^k) \tag{9}$$

In Appendix I, we compute a lower bound that allows one to prove such a bound for Uniswap (the most commonly used CFMM). Then we can bound the deviation in height between the set of trade and price trees (see Appendix B) as a function of $\gamma$ and transfer path-independent returns to the path-deficient case with extra polylogarithmic terms in $\gamma$. Two ways of proving bounds of the form (9) are using generic chaining [Tal21, Ch. 3] and smoothed analysis [HRS20]. We discuss how this analysis can be applied to CFMMs in Appendix H.

### 4.3 Private PAC Learning and Adversarial Bounds

A number of recent results have shown that differentially private PAC learning and online learning are closely related. In particular, the finiteness of an integer-valued complexity measure known as the Littlestone dimension controls whether a particular algorithm can be learned in both an online and differentially private manner [ALMM19, BLM20]. The Littlestone dimension of a class of functions $\mathcal{F}$ from $X \to Y$ $\mathsf{LDim}(\mathcal{F})$ is defined as the maximum depth $d \in \mathbf{N}$ of a tree made up of sequences $x_1, \ldots, x_d \in X$ such that there exists $f \in \mathcal{F}$ with $f(x_i) = y_i$ for every possible $y_i \in Y$. Consider the set $\mathcal{F}^\pi(\Delta)$ which is the set of all trees constructable from any permutation $\pi \in S_n$ for a fixed $\Delta \in \mathbf{R}^n$. The results of 3.2 show that $\mathsf{LDim}(\mathcal{F}^\pi(\Delta)) = \Omega(\mu \log n)$. State-of-the-art results for blackbox constructions of online learners [GL21] show that the regret of a differentially private online learning algorithm is $O(2^{2^{\mathsf{LDim}(\mathcal{F})}})$. This implies that the best online learners can do again the URE, in a blackbox manner, is $O(2^{n^\mu})$. This means that any algorithm that has non-zero curvature is unlikely to do asymptotically better then the URE. If it were possible to construct a polynomial time algorithm to privately PAC learn trades, then there would be significantly degraded privacy guarantees for users. However, this would require a mechanism for which $\mathsf{LDim}(\mathcal{F}^\pi) = O(\log \log n)$, which appears unlikely except for constant-sum market makers that have $\mu = 0$. One other piece of evidence that Littlestone dimension is the correct complexity measure for CFMM privacy comes from the fact that the worst case instances for Littlestone dimension and CFMMs are thresholds (c.f. §3.2 and [ALMM19]).

# 5 Conclusion

In this paper, we demonstrated that there exists a novel, practical mechanism for providing differential privacy to users of constant function market makers. This mechanism, unlike previous methods such as batching, has provable guarantees on the worst case price impact and strong privacy guarantees. As a number of new blockchain protocols implement CFMMs directly in their consensus mechanism, the randomness needed to execute this algorithm will become more plentiful and easier to source. Our analysis used novel techniques combining results from stochastic processes, concentration inequalities, and differential privacy. The results in this paper can likely be improved by providing tighter bounds on the minimal amount of noise needed to achieve $(\epsilon, \delta)$-differential privacy. Moreover, numerical studies of the utility loss (e.g. worsened price impact) would justify practical usage of URE on networks such as Osmosis [AO21] and Penumbra [dV21].

# 6 Acknowledgements

The authors want to thank Yi Sun, GaussianProcess, Kobi Gurkan, Dev Ojha, and Henry de Valence for helpful comments and feedback.

# References


[AAE+21] Guillermo Angeris, Akshay Agrawal, Alex Evans, Tarun Chitra, and Stephen Boyd. Constant function market makers: Multi-asset trades via convex optimization. 2021.

[ABC20] Louigi Addario-Berry and Benoît Corsini. The height of mallows trees. *arXiv preprint arXiv:2007.13728*, 2020.

[ABF13] Louigi Addario-Berry and Kevin Ford. Poisson–dirichlet branching random walks. *The Annals of Applied Probability*, 23(1):283–307, 2013.

[AC20] Guillermo Angeris and Tarun Chitra. Improved Price Oracles: Constant Function Market Makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 80–91, New York NY USA, October 2020. ACM.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[AEC20] Guillermo Angeris, Alex Evans, and Tarun Chitra. When does the tail wag the dog? curvature and market making. *arXiv preprint arXiv:2012.08040*, 2020.

[AEC21a] Guillermo Angeris, Alex Evans, and Tarun Chitra. A note on bundle profit maximization. 2021.

[AEC21b] Guillermo Angeris, Alex Evans, and Tarun Chitra. A note on privacy in constant function market makers. *arXiv preprint arXiv:2103.01193*, 2021.

[Alb19] Mélisande Albert. Concentration inequalities for randomly permuted sums. In *High Dimensional Probability VIII*, pages 341–383. Springer, 2019.

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860, 2019.

[AO21] Sunny Agrawal and Dev Ojha. Vision for osmosis, May 2021.

[BCC+20] Jeff Burdges, Alfonso Cevallos, Peter Czaban, Rob Habermeier, Syed Hosseini, Fabio Lama, Handan Kilinç Alper, Ximin Luo, Fatemeh Shirazi, Alistair Stewart, et al. Overview of polkadot and its design considerations. *arXiv preprint arXiv:2005.13456*, 2020.

[BJP12] Tamara Broderick, Michael I Jordan, and Jim Pitman. Beta processes, stick-breaking and power laws. *Bayesian analysis*, 7(2):439–476, 2012.

[BLM20] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 389–402. IEEE, 2020.

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 634–649. IEEE, 2015.

[Buc16] Ethan Buchman. *Tendermint: Byzantine fault tolerance in the age of blockchains*. PhD thesis, 2016.

[Bun20] Mark Bun. A computational separation between private learning and online learning. *arXiv preprint arXiv:2007.05665*, 2020.

[Cha07] Sourav Chatterjee. Stein's method for concentration inequalities. *Probability Theory and Related Fields*, 138(1):305–321, 2007.

[CJK+18] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

[CXZ20] Shumo Chu, Qiudong Xia, and Zhenfei Zhang. Manta: Privacy preserving decentralized exchange. *IACR Cryptol. ePrint Arch.*, 2020:1607, 2020.

[Dia88] Persi Diaconis. Metrics on groups, and their statistical uses. In *Group representations in probability and statistics*, pages 102–130. Institute of Mathematical Statistics, 1988.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[dV21] Henry de Valence. Sealed-bid batch auctions, 2021.

[Dwo19] Cynthia Dwork. Differential privacy and the us census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 1–1, 2019.

[EAC21] Alex Evans, Guillermo Angeris, and Tarun Chitra. Optimal fees for geometric mean market makers. *arXiv preprint arXiv:2104.00446*, 2021.

[ER75] Paul Erdos and Pál Révész. On the length of the longest head-run. *Topics in information theory*, 16:219–228, 1975.

[GL21] Noah Golowich and Roi Livni. Littlestone classes are privately online learnable. *arXiv preprint arXiv:2106.13513*, 2021.

[GWH+21] Parham Gohari, Bo Wu, Calvin Hawkins, Matthew Hale, and Ufuk Topcu. Differential privacy on the unit simplex via the dirichlet mechanism. *IEEE Transactions on Information Forensics and Security*, 16:2326–2340, 2021.

[HRS20] Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis of online and differentially private learning. *arXiv preprint arXiv:2006.10129*, 2020.

[JKT20] Young Hun Jung, Baekjin Kim, and Ambuj Tewari. On the equivalence between online and private learnability beyond binary classification. *arXiv preprint arXiv:2006.01980*, 2020.

[KLN+11] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[KOR19] Sep Kamvar, Marek Olszewski, and Rene Reinsberg. Celo: A multi-asset cryptographic protocol for decentralized social payments. *DRAFT version 0.24 https://storage. googleapis. com/celo whitepapers/Celo A Multi Asset Cryptographic Protocol for Decentralized Social Payments. pdf*, 2019.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[Kyl85] Albert S Kyle. Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society*, pages 1315–1335, 1985.

[MA17] Olivier Marchal and Julyan Arbel. On the sub-gaussianity of the beta and dirichlet distributions. *Electronic Communications in Probability*, 22:1–14, 2017.

[Mar21] Fernando Martinelli. The crypto cinematic universe crossover event of the summer: Balancer-gnosis-protocol (bgp), Apr 2021.

[MRV99] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*, pages 120–130. IEEE, 1999.

[MSS20] Amani Moin, Kevin Sekniqi, and Emin Gun Sirer. Sok: A classification framework for stablecoin designs. In *International Conference on Financial Cryptography and Data Security*, pages 174–197. Springer, 2020.

[Nad07] Saralees Nadarajah. The linear combination, product and ratio of laplace random variables. *Statistics*, 41(6):535–545, 2007.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[Pow21] Benjamin Powers. Secretswap is the secret network's answer to defi privacy, Feb 2021.

[QZG21] Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? *arXiv preprint arXiv:2101.05511*, 2021.

[QZLG20] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the defi ecosystem with flash loans for fun and profit. *arXiv preprint arXiv:2003.03810*, 2020.

[Ree03] Bruce Reed. The height of a random binary search tree. *Journal of the ACM (JACM)*, 50(3):306–332, 2003.

[RST10] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. 2010.

[Tal14] Michel Talagrand. *Upper and lower bounds for stochastic processes: modern methods and classical problems*, volume 60. Springer Science & Business Media, 2014.

[Tal21] Michel Talagrand. Upper and lower bounds for stochastic processes: modern methods and classical problems (2nd edition, preprint). 2021.

[ZQT+20] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. High-frequency trading on decentralized on-chain exchanges. *arXiv preprint arXiv:2009.14021*, 2020.

# A Differential Privacy Results

We implicitly use a number of differential privacy results on composition and provide them here for convenience. First we note the serial composition theorem:

**Theorem 2** ([DR+14], Theorem 3.16)**.** *Let $\mathcal{A}_1, \dots \mathcal{A}_n$ be a sequence of $(\epsilon_i, \delta_i)$ algorithms such that* $\mathbf{Range}\, \mathcal{A}_i \subset \mathbf{Dom}\, \mathcal{A}_{i+1}$*. Then the composition $\mathcal{A}_n \circ \cdots \circ \mathcal{A}_1$ is $(\sum_{i=1}^n \epsilon_i, \sum_{i=1}^n \delta_i)$-differentially private*

Secondly, we note the parallel composition theorem

**Theorem 3.** *Let $\mathcal{A}_1, \dots, \mathcal{A}_n$ be algorithms whose domains (databases) are independent and each algorithm is $(\epsilon_i, \delta_i)$-differentially private. Then $(\mathcal{A}_1, \dots, \mathcal{A}_n)$ is $\max_i \epsilon_i$ differentially private*

Finally, we note that the serial composition rule can be improved from $(\sum_i \epsilon_i, \sum_i \delta_i)$ to $(n\epsilon^2 + \epsilon\sqrt{n\log(1/\tilde{\delta})}, n\delta + \tilde{\delta})$ where $\tilde{\delta} = O(n\delta)$ if $\epsilon_i = \epsilon, \delta_i = \delta$ for all $i$ [KOV15]. We will not need to use this result, only the generic composition rules. However, it is possible that one can improve our constants using results such as this.

# B Price Tree Height is close to Trade Tree Height

Suppose that we have an admissible trade vector $\mathbf{\Delta} = (\Delta_1, \dots, \Delta_n) \in \mathcal{A}_\varphi$. Given $\pi \in S_n$, we can write a sequence of prices in terms of the price impact function:

$$p_j(\pi) = g\left(\sum_{i=1}^{j} \Delta_{\pi(i)}\right)$$

We generate a random binary tree from the price vector by uniformly sampling $j \sim [n]$ and making $p_j(\pi)$ the root before inserting the remaining prices sequentially as per $\pi$. Under this framework, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{\pi\sim S_n}\left[\max_j p_j(\pi)\right] &\leq \mathop{\mathbf{E}}_{j\sim[n]}[p_j(\pi)] + \max_i |p_i(\pi) - p_{i-1}(\pi)| \mathop{\mathbf{E}}_{\pi\sim S_n}[\mathsf{height}(T(p_j(\pi)))] \\
&\leq \mathop{\mathbf{E}}_{j\sim[n]}[p_j(\pi)] + \mu(\max_i \Delta_i) \mathop{\mathbf{E}}_{\pi\sim S_n}[\mathsf{height}(T(p_j(\pi)))]
\end{aligned}$$

We can later remove this constraint by adding a small amount of noise to each entry, which will make the entries unique a.s.. Note that the height of the tree generated by $P_j$ represents the number of trades in the longest sequential deviation from the mean price. Let's consider when the trade tree and price tree differ in branching. On average, this occurs when the $j$th price $p_{\pi(j)}$ is a left branch whereas the $j+1$st price $p_{\pi(j+1)}$ is a right branch, but both trades $\Delta_{\pi(j)}, \Delta_{\pi(j+1)}$ are left branches. When this happens, the price tree has an average height that is 1 less than the trade tree.

We will first illustrate this when the first two elements of the permutation after the pivot (which is random) differ from the expected pivot value. Explicitly, suppose that we have

$$p_{\pi(2)} - \frac{1}{n}\sum_{i=1}^{n} p_{\pi(i)} < 0 \qquad\qquad p_{\pi(3)} - \frac{1}{n}\sum_{i=1}^{n} p_{\pi(i)} > 0$$

Using curvature bounds, the first equation gives

$$0 \geq p_{\pi(2)} - \frac{1}{n}\sum_{i=1}^{n} p_{\pi(i)} \geq \kappa\Delta_{\pi(2)} - \frac{\mu}{n}\sum_{i=1}^{n}\Delta_i$$

Similarly, the second equation gives

$$0 \leq p_{\pi(3)} - \frac{1}{n}\sum_{i=1}^{n} p_{\pi(i)} \leq \mu\Delta_{\pi(3)} - \frac{\kappa}{n}\sum_{i=1}^{n}\Delta_i$$

which when combined gives

$$\Delta_{\pi(2)} \leq \frac{\mu}{\kappa}\left(\frac{1}{n}\sum_{i=1}^{n}\Delta_i\right) = \eta_+ \tag{10}$$

$$\Delta_{\pi(3)} \geq \frac{\kappa}{\mu}\left(\frac{1}{n}\sum_{i=1}^{n}\Delta_i\right) = \eta_- \tag{11}$$

On the other hand, suppose that $R_2(\pi) - \overline{R}(\pi), R_3(\pi) - \overline{R}(\pi)$ are both greater than zero (e.g. they are both left nodes of their parent). This implies that $\Delta_{\pi(2))} + \Delta_{\pi(3)} \geq \frac{1}{n}\sum_{i=1}^{n}\Delta_i$. This means that we can only end up in a state where $\mathsf{height}(T(R_j(\pi))) > \mathsf{height}(T(p_j(\pi)))$ if the trades are within the interval $[\eta_-, \eta_+]$. For instance, when the drift $\frac{1}{n}\sum_{i=1}^{n}\Delta_i = 0$, then interval has size zero (its a mean-reverting set of trades) and we never enter this error condition. This matches intuition: if there's a lot of drift in the trades, then we shouldn't expect our price and trade vectors to 'sort' the same way. In particular, the higher the curvature of the CFMM, the less drift we can tolerate because large trades cause more noticeable price impact. The length of the interval $[\eta_-, \eta_+]$ is

$$\left(\frac{\mu}{\kappa} - \frac{\kappa}{\mu}\right)\left(\frac{1}{n}\sum_{i=1}^{n}\Delta_i\right)$$

Note that we can recurse the above argument as we go down the tree and get a set of intervals $I_1 = [\eta_-(1), \eta_+(1)], I_2 = [\eta_-(2), \eta_+(2)], \ldots, I_n = [\eta_-(n), \eta_+(n)]$. Performing the same calculation as above yields

$$\eta_-(i) = \frac{\kappa}{\mu}\left(\frac{1}{n-i}\sum_{i=i}^{n}\Delta_{\pi(i)}\right) \qquad \eta_+(i) = \frac{\mu}{\kappa}\left(\frac{1}{n-i}\sum_{i=i}^{n}\Delta_{\pi(i)}\right)$$

Given that the maximum interval size is $\mu M$ is the max trade size for which curvature is valid), we can use this to bound the probability $p_j$ that vertex $j$ has a height difference between the trade and price trees. This probability is upper bounded by ratio of the length of $I_j$ and the interval length $\mu M$, e.g. $p_j \leq \frac{|I_j|}{\mu M}$. We can upper bound the interval length by the maximum mean-drift subsequence:

$$|I_j| \leq \left(\frac{\mu}{\kappa} - \frac{\kappa}{\mu}\right)\left(\max_{J \subset [n]} \frac{1}{|J|}\sum_{j \in J}\Delta_j\right)$$

Define $R^*(\mathbf{\Delta}) = \max_{J \subset [n]} \frac{1}{|J|}\sum_{j \in J}\Delta_j$. Finally, performing a union bound gives an upper bound on the probability $p_{\text{diff}}$ of the heights of the trade tree and price tree different

$$p_{\text{diff}} \leq \sum_{j=1}^{n} p_j = n\left(\frac{1}{M\kappa} - \frac{\kappa}{\mu^2 M}\right) R^*(\mathbf{\Delta}) \tag{12}$$

If this quantity is sufficiently small (e.g. we have tight curvature bounds), then bounds on the trade tree transfer to the price tree with high probability. For the rest of the paper, we will assume that (12) is sufficiently small. We note that fee adjustments and curvature adjustments are intricately related [AEC20, §3] and in practice, this can be enforced by dynamic updates to a CFMM curve.

# C Proof of Claim 1

Suppose that $\xi_i \sim_{iid} \mathsf{Lap}(a, b)$. We need to analyze the distribution of $\xi_i - \frac{\mu}{\kappa}\xi_j$. Recall that if $X \sim \mathsf{Lap}(a, b)$ then $kX \sim \mathsf{Lap}(ka, |k|b)$. Therefore we are trying to bound the distribution of $Z(a, b) = X + Y$ where $X \sim \mathsf{Lap}(a, b)$, $Y \sim \mathsf{Lap}\left(-\frac{\mu}{\kappa}a, \frac{\mu}{\kappa}b\right)$. In particular, given $\delta < 0$ we want to choose $a, b$ such that

$$F_Z(k) \leq \mathbf{Prob}[X + Y \leq k] \leq \delta$$

where $k = c_{\min} + \left|\min_{i,j} \Delta_i - \frac{\kappa}{\mu}\Delta_j\right|$. Nadarajah [Nad07, Theorem 1] explicitly computes the CDF $F_{Z(a,b)}(k)$ and shows that it is monotone, continuous, and differentiable in $a, b$ except at one value of $k$ for all $a, b$. Moreover, it is supported on the entire real line. Therefore, $\exists a^*$ such that $F_{Z(a^*,|a^*|)}(k) = \delta$.

# D Proof of Claim 2

Our proof works by differentially privately sampling a probability distribution $\pi \sim \mathsf{Dir}(\vec{1})$ multiple times using the mechanism of [GWH+21]. The Dirichlet mechanism on $k$ nodes $\mathcal{M}_D^{(k)}(\pi)$ samples a Dirichlet distribution centered at $\pi$, where $\pi \in P_k = \{x \in \mathbf{R}^k : \sum_i x_i = 1, x_i \geq 0\}$. One can think of it as sampling a increment $d\pi$, adding it to $\pi$ and renormalizing. First, we reproduce a theorem on differentially private Dirichlet sampling

**Theorem 4** ([GWH+21], Theorem 1, Corollary 1)**.** *The Dirichlet mechanism $\mathcal{M}_D^{(k)}(\pi)$ achieves $(\epsilon, \delta)$-differential privacy where $\epsilon = O(k(1+\log(o(k)))$ and $\delta = 1 - \min_\pi \mathbf{Prob}[\mathcal{M}_D^k(\pi) - \pi > \Omega(\epsilon)]$*

Define the vector $\eta(\mathbf{\Delta})$ as follows:

$$\eta(\mathbf{\Delta}) = \left(\left\lceil \frac{\Delta_1}{\Delta_{\min}} \right\rceil, \ldots, \left\lceil \frac{\Delta_n}{\Delta_{\min}} \right\rceil\right)$$

Each coordinate represents rounding each trade to an integer lattice with width $\Delta_{\min}$. Define $S_k = \{i : \eta(\mathbf{\Delta}) > k\}$ and $S_k^c = [n] - S_k$. For each $j \in S_k$, privately sample $\pi \sim \mathsf{Dir}(\vec{1})$ where $\vec{1} = (1, \ldots, 1) \in \mathbf{R}^{\eta(\mathbf{\Delta})_j}$. Let $\hat{\Delta}_{j,k} = \Delta_j \pi_k$ with $\sum_k \hat{\Delta}_{j,k} = \Delta_j$. We can view each Dirichlet sample $\pi$ as providing a mechanism for splitting the trade $\Delta_j$. Our goal is to find $k \in \mathbf{N}$ such that the following two conditions hold

1. $\mathsf{height}(T(\Delta_{S_k^c})) = \Theta(\log n)$

2. $\mathsf{height}(T(\hat{\Delta}_{j,k})) = \Theta(\log \eta_j)$ with high probability

We can show that the latter condition holds with high probability when the distribution sampled is Dirichlet centered at the centroid $(\frac{1}{n}, \dots, \frac{1}{n})$. Constructing a partial sum tree from a Dirichlet sample is the same as drawing a sample from a Poisson-Dirichlet branching random walk [ABF13]. These walks satisfy $\mathbf{Prob}[|\mathsf{height}(T(\hat{\Delta}_{j,k})) - c \log \eta(\mathbf{\Delta})_j| \geq k] = O(e^{-k})$ for a universal constant $c$ [ABF13, Corollary 1.3]. Therefore, the probability that all of the Dirichlet constructed trees $T(\hat{\Delta}_{j,k})$ have height greater than $c \log \eta(\mathbf{\Delta})_j$ is

$$\mathbf{Prob}\left[\exists j \in S_k |\mathsf{height}(T(\hat{\Delta}_{j,k})) - c \log \eta_j| \geq c' \log \eta_j\right] \leq \left(\frac{|S_k|}{\eta_j^{c'}}\right)$$

which directly follows from the independent sampling from the private Dirichlet distribution and inclusion-exclusion. Therefore, with probabilty $p^* = 1 - \frac{|S_k|}{\delta_j^{c'}}$, we have the maximum height of a tree constructed from all $|S_k|$ vectors $\hat{\Delta}_{j,k}$ is

$$\sum_{j \in S_k} \log \eta_j \leq |S_k| \max_j \log \eta_j$$

which under our assumptions is $O(\log n)$. Our claim about differential privacy then follows immediately from Theorem 4.

# E Proof of Claim 3

We will prove differential privacy by using the smooth sensitivity framework of [NRS07]. First, we will recall definitions and introduce preliminaries on this framework before specializing it to SURE. Smooth sensitivity places an upper bound on the local sensitivity of a function $f$, which is defined as

$$LS_f(x) = \max_{d(x,y) \leq 1} |f(x) - f(y)|$$

Note that unlike the global sensitivity, which is used in the generic Laplace mechanism [DR+14], the local sensitivity depends on the particular input $x$. Often times, it is too difficult to get uniform bounds on local sensitivity and instead it is easier to use a smooth proxy. A $\beta$-smooth upper bound $S : \mathbf{Dom}\, f \to \mathbf{R}$ for $LS_f(x)$ satisfies $S(x) \geq LS_f(x)$ for all $x \in \mathbf{Dom}\, f$ and $S(x) \leq e^{\beta} S(y)$ for all $x, y \in \mathbf{Dom}\, f$ with $d(x, y) = 1$. We are now in a position to recall two results of Nissim, et. al:

**Theorem 5** ([NRS07], Lem. 2.6)**.** *Let $h$ be an $(\alpha, \beta)$-admissible noise probability density function and let $Z \sim h$. For a function $f : D^n \to \mathbf{R}^d$, let $S$ be a $\beta$-smooth upper bound in the local sensitivity of $f$, then $\mathcal{A}(x) = f(x) + \frac{S(x)}{\alpha} Z$ is $(\epsilon, \delta)$-differentially private.*

**Theorem 6** ([NRS07], Lem. 2.9). *For $\epsilon, \delta \in (0,1)$, the $d$-dimensional Lpalace distribution, $h(z) = 2^{-d}e^{-\|z\|_1}$ is $(\alpha, \beta)$-admissible with $\alpha = \frac{\epsilon}{2}$, $\beta = \frac{\epsilon}{2\rho_{\delta/2}(\|Z\|_1)}$ where $\rho_\delta(Y)$ is the $1-\delta$ quantile of $Y$*

Combined, these results illustrate that if we can construct a $\beta$-smooth upper bound, we can immediately construct a Laplace mechanism that achieves $(\epsilon, \delta)$-differential privacy. Section 3 of [NRS07] provides a mechanism for computing a $\beta$-smooth upper bound by first defining the sensitivity at distance $k$,

$$LS_f^k(x) = \max_{\substack{y \in \mathbf{Dom}\, f \\ d(x,y) \leq k}} LS_f(x)$$

A $\beta$-smooth upper bound on local sensitivity is deinfed as,

$$S_{f,\beta}(x) = \max_{k \in \{0,1,\ldots,n\}} e^{-k\beta} LS_f^k(x)$$

Therefore, we need to construct a function $f$ that represents price impact and compute an analogue of local sensitivity.

For a differentially private CFMM, we want to minimize the worst case price impact in a neighborhood of a trade $\mathbf{\Delta}$. We define $f(\mathbf{\Delta})$ as

$$f(\mathbf{\Delta}) = \max_{j \in [n]} p_j(\Delta)$$

Now we need to modify the definition of local sensitivity to account for trade admissibility and discretization. Normally, local sensitivity is defined for discrete spaces where the distance $d$ is taken to be the Hamming metric. We can discretize our trade space in terms of $\Delta_{\min}$. Recall that we ensure that $\Delta_{\min} > 0$ by adding Laplace noise to all trades (whose parameter will be tuned in accordance with the above theorem). Note that moving to such a discretization simply changes our choice of $\beta$. Using this definition, we can define the local trade sensitivity as

$$TS_f^k(\mathbf{\Delta}) = \sup_{\substack{\mathbf{\Delta}' \in \mathbf{Dom}\, f \cap \mathcal{A}(R) \\ d(\mathbf{\Delta}, \mathbf{\Delta}') \leq k\Delta_{\min}}} |f(\mathbf{\Delta}) - f(\mathbf{\Delta}')|$$

where $\mathcal{A}(R)$ is the set of admissible trades. From the results of §3.2, we know that $TS_f^k(\mathbf{\Delta}) = O(k\mu(\max_i \mathbf{\Delta}) \log n)$ since the depth of the tree quantifies the largest price impact. In particular, each element $\Delta_i'$ such that $|\Delta_i - \Delta_i'| > \Delta_{\min}$ can cause price impact of at most $\mu(\max_i \mathbf{\Delta}) \log n$ and we can add these independently over the at most $k$ coordinates that have prices changed by more than $\Delta_{\min}$. We can define an analogous smooth sensitivity bound,

$$\tilde{S}_{f,\beta}(x) = \max_\ell e^{-\ell\beta} TS_f^\ell(\mathbf{\Delta}) = \max_\ell e^{-\ell\beta} \ell\mu(\max_i \Delta) \log n$$

This is minimized when $\ell = \frac{1}{\beta}$, giving

$$\tilde{S}_{f,\beta}(x) = \frac{\mu}{e\beta}(\max_i \Delta) \log n$$

Therefore, provided that a) the partial sum tree has height $O(\log n)$ b) the noise added ensures that $\Delta_{\min} > 0$, and c) the noise is rescaled by $\frac{2\tilde{S}_{f,\beta}(x)}{\epsilon}$, we achieve differential privacy.

Note that in particular, our bound depends on $\max_i \Delta$ and the curvature upper bound. By splitting trades using Claim 2, we reduce $\max_i \Delta$ and can ensure that the noise added is reasonable. Moreover, as we saw, without splitting trades, we run into issues with trades of the form $(T, 1, \dots, 1)$. Note that algorithms that try to learn where the trade $T$ occurs (after applying a permutation $\pi$) is equivalent to privately learning threshold functions [BNSV15, ALMM19].

# F Convex Trade Splitting

When we are considering CFMM arbitrage, it can be shown that a necessary condition for stability is path-deficiency. Path deficiency ensures that no rational trader (e.g. profit optimizing) is incentivized to split a desired trade size $\Delta$ into two trades $\Delta_1 + \Delta_2 = \Delta$. However, if a trader also desires privacy, splitting up trades can become necessary. To see why, consider a trader who makes a trade of size $T$ and a sequence of trades $\Delta = (T, 1, \dots, 1) \in \mathbf{R}^{T+1}$. Using curvature, we know that the price impact is at least $\kappa T$ after a trade of size $T$ and of size $\kappa$ after each trade of size 1. This means that an adversary can easily discern where my trade is, even if $\Delta$ is randomly permuted due to the $T$ times larger price impact. Therefore, splitting up the trade of size $T$ into trades close to size 1 will make it hard for an adversary to reconstruct the total trade size.

Our goal is to split up trades such that the probability of an adversary detecting the position of a single trade is small relative to the curvature. Suppose that a trade $\Delta_1$ is split into trades $\Delta'_1, \dots, \Delta'_j$ and let $\tilde{\mathbf{\Delta}} = (\Delta'_1, \dots, \Delta'_j, \Delta_2, \dots, \Delta_n)$ A splitting adversary is a binary classifier $\ell(\Delta, \mathbf{\Delta})$ that returns 1 if $\Delta \in \{\Delta'_1, \dots, \Delta'_j\}$ and 0 otherwise. We say that a splitting mechanism is $(\delta, \epsilon)$ indistinguishable if

$$\mathbf{Prob}\left[\left|\frac{1}{n}\sum_{i=1}^{n} \ell(\tilde{\Delta}_i, \tilde{\mathbf{\Delta}}^{\pi}) - \frac{j}{n}\right| < \epsilon\right] < \delta$$

over some suitable set of splitting classifiers. The inequalities in Appendix G can directly be used to prove that this holds for the $L^2$ norm.

However, path-deficiency implies that splitting trades will cost a user an extra fee. This trade-off between best execution price and privacy can be explored via a simple, convex objective function that trades off price impact vs. improved privacy via splitting. Recall that the $L^2$ norm strictly decreases under splitting, e.g.

$$\begin{aligned}
\|(\Delta_1, \dots, \Delta_n)\|_2^2 &= \sum_{i=1}^{n} \Delta_i^2 = \Delta_1^2 + \sum_{i=2}^{n} \Delta_i^2 \\
&= a\Delta_1^2 + (1-a)\Delta_1^2 \sum_{i=2}^{n} \Delta_i^2 > a^2\Delta_1^2 + (1-a)^2\Delta_1 + \sum_{i=2}^{n} \Delta_i^2 \\
&= \|(a\Delta_1, (1-a)\Delta_1, \dots, \Delta_n)\|_2^2
\end{aligned}$$

where $a \in (0,1)$ represents the splitting fraction.

This property allows us to quantify the privacy benefit to splitting trades, as the more minimal the $L^2$ norm, the less noise that is needed to ensure that the random binary tree has height $\Theta(\log n)$ and $\Omega(n)$ leaves. In particular, the Cauchy and Gaussian mechanisms for differential privacy utilize distributions whose variances are proportional to the $L^2$ norm.

Given that we want to minimize price impact while maximizing the amount of trade splitting necessary for indistinguishable, we construct a convex optimization problem. Define the function $f$ as:

$$f(\Delta_1, \dots, \Delta_n) = \sum_{i=1}^{n} \gamma g\left(\gamma \sum_{j=1}^{i} \Delta_i\right) + \eta \sum_{i=1}^{n} \Delta_i^2$$

The first term in $f$ represents an upper bound on the price impact and the second term represents the $L^2$ splitting term. Our goal is to minimize $f$ over sequences of trades $(\Delta_1, \dots, \Delta_k) \in \sqcup_{i=1}^{\infty} \mathbf{R}^i$ such that $\sum_{i=1}^{k} \Delta_i = \Delta^*$, e.g.

$$\begin{array}{ll} \text{minimize} & f(\Delta_1, \dots, \Delta_n) \\ \text{subject to} & \Delta_1 + \dots + \Delta_n = \Delta^* \end{array} \tag{13}$$

Using curvature bounds, we can construct a simple descent algorithm to solve this. Firstly, note that the definition of curvature yields

$$\kappa\gamma^2 \sum_{i=1}^{n} \sum_{j=1}^{i} \Delta_i \leq f(\Delta_1, \dots, \Delta_n) - \eta \sum_{i=1}^{n} \Delta_i^2 \leq \mu\gamma^2 \sum_{i=1}^{n} \sum_{j=1}^{i} \Delta_i$$

Furthermore, note that we can rewrite the double sum as

$$\sum_{i=1}^{n} \sum_{j=1}^{i} \Delta_i = \sum_{i=1}^{n} (n-i+1)\Delta_i$$

Next, note that we can upper bound the split function, $f(a\Delta_1, (1-a)\Delta_1, \dots, \Delta_n)$ as

$$\begin{aligned} f(a\Delta_1, (1-a)\Delta_1, \dots, \Delta_n) &\leq \mu\gamma^2 \left((n+1)a\Delta_1 + n(1-a)\Delta_1 + \sum_{i=2}^{n} (n-i+2)\Delta_i\right) \\ &\quad + \eta \left(a^2\Delta_1^2 + (1-a)^2\Delta_1^2 + \sum_{i=2}^{n} \Delta_i^2\right) \\ &= \mu\gamma^2 \left((n+a)\Delta_1 + \sum_{i=2}^{n} (n-i+1)\Delta_i + \Delta^*\right) \\ &\quad + \eta \left(a^2\Delta_1^2 + (1-a)^2\Delta_1^2 + \sum_{i=2}^{n} \Delta_i^2\right) \end{aligned}$$

Combining these gives the following

$$f(\Delta_1, \ldots, \Delta_n) - f(a\Delta_1, (1-a)\Delta_1, \ldots, \Delta_n) \geq \gamma^2(\kappa - \mu) \sum_{i=2}^{n} (n-i+1)\Delta_i - \Delta^* \\ - \mu\gamma^2(n+a)\Delta_1 + \eta\Delta_1^2(1 - a^2 - (1-a)^2) \tag{14}$$

Maximize the right-hand side in $a$ provide a mechanism for deciding whether to split trade $\Delta_1$. Optimizing over $a$ yields

$$a^* = \max\left(\frac{1}{2} - \frac{\mu\gamma^2}{4\eta\Delta_1}, 0\right)$$

If we substitute $a^*$ into (3) and the right-hand side is position, we split the trade $\Delta_1$ into two trades of size $a^*\Delta_1$ and $(1-a^*)\Delta_1$.

# G Splitting Trades: Concentration

Chatterjee proved a concentration bound using Stein's method that provides intuition as to why spliting trades improves the effectiveness of SURE. Theorem 7 shows that the variance of concentration around the mean for a randomly permuted sum is linear in the expected value.

**Theorem 7** ([Cha07], Prop. 1.1)**.** *Let $\{a_{i,j}\}_{1\leq i,j\leq n}$ be a collection of numbers from $[0,1]$. Let $X = \sum_{i=1}^{n} a_{i,\pi(i)}$ where $\pi \sim S_n$ uniformly. Then*

$$\mathbf{Prob}[|X - \mathbf{E}[X]| \geq t] \leq 2\exp\left(-\frac{t^2}{4\,\mathbf{E}[X] + 2t}\right) \tag{15}$$

Note that unlike Bernstein-like inequalities there is no direct dependence on $n$. Moreover, unlike Talagrand-like inequalities [Tal21], we do not have terms dependent on $\epsilon$-nets. If we let $t = k\,\mathbf{E}[X]$, we have

$$\exp\left(-\frac{t^2}{4\,\mathbf{E}[X] + 2t}\right) = \exp\left(-\frac{k^2\,\mathbf{E}[X]}{2k+4}\right) \leq \exp(-k\,\mathbf{E}[X])$$

For positive trade sizes, this implies that if we can split big trades into smaller trades (which reduces in turn reduces $\mathbf{E}[X]$) we can achieve the sufficient condition. More specifically, suppose that $a_{i,j} = \Delta_j - \frac{\kappa}{\mu}\Delta_i$. Then $X = \sum_{i=1}^{n} a_{i,\pi(i)}$ is the upper bound from (7) and the theorem claims that reducing the maximum will reduce the variance of SURE's utility.

We also note that better asymptotic results results exist for non-negative sums:

**Theorem 8** ([Alb19], Corollary 2.2)**.** *Let $a_{ij}$ be a connection of any real numbers and $\pi \sim S_n$ as uniform random permutation. Let $Z_n = \sum_{i=1}^{n} a_{i,\pi(i)}$. Then for all $x > 0$*

$$\mathbf{Prob}(|Z_n - \mathbf{E}[Z_n]| \geq t) \leq 16e^{1/16}\exp\left(\frac{-t^2}{256(\mathbf{Var}[Z_n] + \max_{i,j}|a_{ij}|t)}\right)$$

This bound explicitly includes a maximum term, directly justifying the improvement to SURE provided by splitting trades.

# H Path Dependency and Generic Chaining

Suppose that we want to try to find the worst case price deviation given that we have fees, $\gamma < 1$. If we define $X_j = p^\pi(i) - p(i)$, then we want to study the extremal behavior of this process, albeit without being able to directly bound price impact using methods from §3.2. We will be most interested in the behavior of the random variable $X^* = \max_j X_j$, which quantifies the worse execution price received by a user under this mechanism. To do this, we will utilize the theory of empirical processes. Roughly speaking, one can show that for a metric space $(T, d)$, $\mathbf{E} \sup_{t \in T} X_t = \Theta(\mathsf{Diam}(T)\sqrt{\log \mathsf{card} T})$ by looking at simple bounds for empirical processes [Tal14, Tal21]. Our goal is to define a metric space $T_\gamma$ that depends on fees and such that $S_n$ acts faithfully on $T_\gamma$. We want the action to be faithful because that will be equivalent to the condition of unique elements of the form $\left|\Delta_i - \frac{\mu}{\kappa}\Delta_j\right|$ We can then attempt to bound, using chaining arguments, the worst case price deviation.

Chaining bounds rely on tail bounds on increments, e.g. showing that for some metric $d$ on our space $T_\gamma$, we have the following two conditions:

$$\forall u > 0,\ \mathbf{Prob}[|X_s - X_t| \geq u] \leq 2 \exp\left(-\frac{u^2}{2d(s,t)^2}\right) \tag{16}$$

$$\exists u > 0,\ \sum_{s \in T} \mathbf{Prob}[X_s \geq u] \geq 1 \tag{17}$$

In our case, we need to construct a metric space that takes advantage of our trading function curvature and the randomness induced by the choice of permutation.

Our goal is to construct a metric on $S_n$ that depends on both $\varphi$. We need to construct metric $d_{\varphi, R_0, \boldsymbol{\Delta}} : S_n \times S_n \to \mathbf{R}_+$ that we can use to find a formula like eq. (16). A natural metric to construct is the raw price differences:

$$d_{\varphi, R_0, \boldsymbol{\Delta}}(\pi_1, \pi_2) = \sum_{i=1}^n |p^t_{\pi_1(i)} - p^t_{\pi_2(i)}|$$

Note that if we took an infimum over one of the two permutations, we arrive at the Wasserstein distance. Suppose we have $d_{\varphi, R_0, \boldsymbol{\Delta}}(\pi_1, \pi_2) \leq f(\varphi, R_0, \boldsymbol{\Delta}) d(\pi_1, \pi_2)$ for some natural metric on the symmetric group (e.g. Mallows metric [Dia88]). Moreover, suppose there exists $\kappa > 0$ such that $\mathbf{Prob}[X_s \geq \sqrt{\log n}\,(\kappa + \sum_i \Delta_i)] \geq 1$. Then we have the lower bound [Tal21, Eq. 2.15]

$$C\left(\kappa + \sum_i \Delta_i\right)\sqrt{\log n} \leq \mathbf{E} \sup_{t \in T} X_t \leq C'\left(\kappa + \sum_i \Delta_i\right) \mathsf{Diam}_d(T) \sqrt{\log n}$$

One simple idea for a metric upper bound is:

$$d^{ub}(\pi_1, \pi_2) = \mu \sum_{i=1}^n |\Delta_{\pi_1(i)} - \Delta_{\pi_2(i)}|$$

Under this metric, we need to show that

$$\mathbf{Prob}[|X_\pi - X_{\pi'}| \geq u] \leq 2\exp\left(-\frac{u^2}{2d(\pi,\pi')^2}\right)$$

This is effectively direct from Azuma's inequality since $\Delta_i$ is in a bounded ball (in order for us to use curvature). Next, we need to show $\mathbf{Prob}[X_s \geq \sqrt{\log n}\,(\kappa + \sum_i \Delta_i)] \geq 1$. For each permutation $\pi \in S_n$, we can construct a binary tree $T_\pi$ from the partial sums $S_i \sum_i \Delta_{\pi(i)}$, where $S_i < S_j$ implies $S_i$ is in the left subtree of $S_j$ (and vice versa). Assume, first, that each $S_i$ is unique. Then, it can be shown that the expected height and the tail bounds for the height of this subtree satisfies [ABC20, Ree03]

$$\mathbf{Prob}[h(T_\pi) \geq \sqrt{\log n}] \geq \frac{c}{n}$$

Our conjecture is that $\kappa h(T_\pi) \leq X_s \leq \mu h(T_\pi)$ which would immediately imply $\sum_{\pi \in S_n} \mathbf{Prob}[X_\pi \geq u] \geq 1$. Unfortunately to find bounds of this form with fees, one needs to find universal bounds on $g(\Delta) - \gamma g(\gamma\Delta)$. We illustrate such bounds for Uniswap in Appendix I.

# I Path Dependency in Uniswap

Getting bounds such as (16) relies on bounding how far away the path-dependent case strays from the path independent case. For a fixed $\mathbf{\Delta}$, $p_n^{pi}$ only depends on $\sum_i \Delta_i$ for path-independent, whereas $p_n^{pd}(\pi)$ does depend on the path $\Delta_{\pi(1)}, \ldots, \Delta_{\pi(n)}$. However, if we can uniformly bound $\max_{\pi \in S_n} |p_n^{pd}(\pi) - p_n^{pi}|$ as a function of fees and curvature.

For Uniswap, we have $g_{uni}(\Delta) = \frac{k}{(R-\Delta)^2}$. This gives a difference between the impact of a single path independent trade and a single path dependent trade as (see [AEC20] for the formulae):

$$\begin{aligned}
g(\Delta) - \gamma g(\gamma\Delta) &= k\left(\frac{1}{(R-\Delta)^2} - \frac{\gamma}{(R-\gamma\Delta)^2}\right) = \frac{k}{(R-\Delta)^2}\left(1 - \frac{\gamma(R-\Delta)^2}{(R-\gamma\Delta)^2}\right) \\
&= g(\Delta)\left(1 - \frac{\gamma(R-\Delta)^2}{R^2}\frac{1}{(1-\frac{\gamma\Delta}{R})^2}\right) \\
&\leq g(\Delta)\left(1 - \frac{\gamma(R-\Delta)^2}{R^2}\left(1 - \frac{c\gamma\Delta}{R}\right)\right) \\
&= g(\Delta)\left(1 - \frac{\gamma(R-\Delta)^2}{R^2} - \frac{c\gamma\Delta(R-\Delta)}{R^3}\right) \\
&= g(\Delta)\left(1 - \gamma\left(\frac{R-\Delta}{R}\right)^2 (R - \left(1 + \frac{c}{R}\right)\Delta)\right)
\end{aligned}$$

where we assume that $\frac{\gamma\Delta}{R} < 1$ and use the geometric series (so $c < 1$). When $R \gg 1$ and $R - \Delta \leq kR$ for some $k < 1$, this gives us the bound

$$\frac{g(\Delta) - \gamma g(\gamma\Delta)}{g(\Delta)} \leq 1 - \gamma\frac{(R-\Delta)^3}{R^2} \leq 1 - \gamma k^3 R$$