LVR from Theory to Practice: A Survey of Next Generation DEX Designs

Daniel Contreras
Arrakis Finance
daniel@arrakis.fi

Ari Rodriguez
Arrakis Finance
ari@arrakis.fi

May 9, 2024

Abstract

In this paper we provide a comprehensive taxonomy of the current proposed solutions to reduce or mitigate LVR defined in [Mil+22]. We survey the landscape of next generation DEX designs that provide either direct reduction of LVR or novel fee mechanisms to mitigate its adverse effects. We confirm that direct LVR reduction in the DEX design space can only be achieved by introducing trusted oracles or via specific offchain sequencing rules. Finally, we propose that many of these offchain solutions are best viewed as a new type of app-specific Layer 2 construction, and we explore the relationship of this class of applications with a rollup-centric Ethereum roadmap.

1 Introduction

In the rapidly evolving landscape of decentralized finance (DeFi), decentralized exchanges (DEXs) and in particular automated market makers (AMMs) have emerged as pivotal components reshaping the dynamics of asset liquidity and trading. AMMs represent algorithmic protocols designed to facilitate decentralized exchange of assets, without the need for traditional intermediaries like centrally controlled order books or exchanges (CEXs). Leveraging smart contracts on blockchain platforms, AMMs enable users to trade assets directly with a pool of liquidity, provided by individuals termed liquidity providers (LPs). This paradigm shift from centralized to decentralized exchange mechanisms introduces novel challenges and opportunities, particularly regarding liquidity provision, price discovery, and risk management.

Liquidity providers (LPs) in AMMs incur significant costs stemming from informational advantages exploited by arbitrageurs, who capitalize on outdated AMM prices compared to those on CEXs. These costs are quantified in [Mil+22] using the loss-versus-rebalancing (LVR) metric, which is defined as follows.

For a given LP position in an AMM, we call *rebalancing strategy* the portfolio that replicates the holdings of the LP position, but at CEX prices. LVR is defined as the gap between

the value of this strategy and the LP position. By shorting this rebalancing strategy, LPs can hedge any market risks and isolate the risks due to the AMM design, in particular to *price slippage*. Therefore, we can describe the profit and loss of this hedged position at a given time $t \geq 0$ by

$$\text{delta-hedged LP } \mathsf{P\&L}_t = \mathsf{FEE}_t - \mathsf{LVR}_t, \tag{1}$$

where FEE corresponds to the accumulated fees charged by the protocol that are given back to the LP position.

Hedging such a rebalancing strategy carries its own practical complexities and potential costs. However, in this paper we will not focus on this element, but rather assume that the profit and loss of LPs can be measured by the right hand side elements of (1). It is important to note, that measuring the profit and loss against other benchmarks such as HODL includes LVR as a component, see [Mil+22, Corollary 1] or [Ber+23, Section 2.3]. Therefore, reducing or mitigating LVR is under most metrics a crucial element in LP profitability.

Our goal is to classify the existing solutions and propose new designs that make the term $\mathsf{FEE}_t - \mathsf{LVR}_t$ on (1) as big as possible without undermining the usability of the AMM. From now on, we will say that LVR is *reduced* when the term LVR_t decreases by action of the protocol. On the other hand, we will say that LVR is *mitigated*, when the FEE_t term is increased to compensate for LVR_t. In the cases the difference $\mathsf{FEE}_t - \mathsf{LVR}_t$ increases, and so does the LP profitability.

We will assume through the whole paper that price discovery always happens on CEXs. For assets with long-tail onchain liquidity distributions, for which the price is often determined onchain, further research is needed to build good models of LPs profitability.

Our main result is that LVR reduction can only be attained via offchain sequencing rules that permission the liquidity access to arbitrageurs under certain conditions. In contrast, protocols can use both offchain and/or onchain rules to achieve LVR mitigation. For a global taxonomy of these solutions, refer to Figure 2.

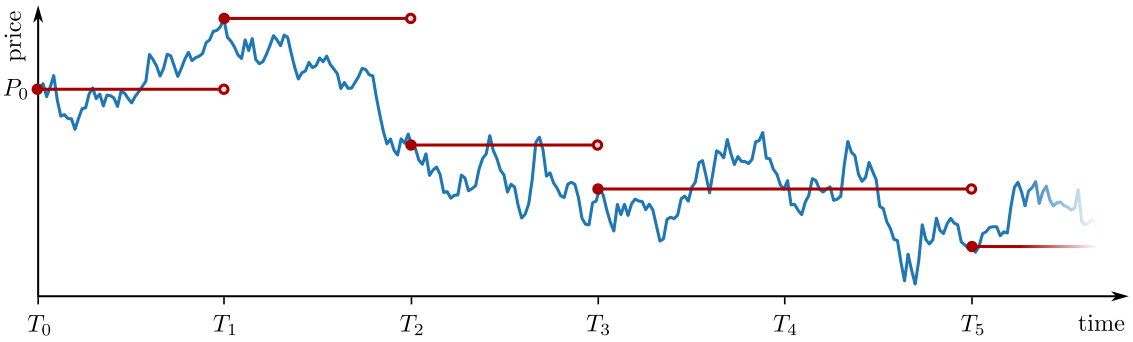
Our second contribution is to define a general framework for these protocols depending on offchain computation, but that still settle their transactions to the base layer. We define this class of protocols as *rollups** and provide extra conditions that differentiate them from application specific rollups.

1.1 Organization of the paper

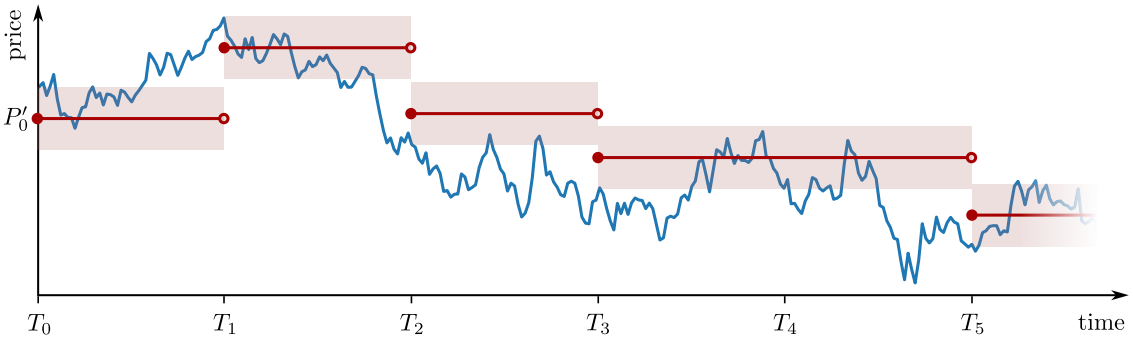
In Section 2 we describe how price moves on AMMs by the action of arbitrageurs. In Section 3 we study protocols that directly reduce LVR. In section Section 4 we study mechanisms that allow LPs to capture more fees to mitigate LVR. We also perform an empirical analysis of three major chains to test some theoretical results in Section 5. Finally, in Section 6 we establish a new class of app specific Layer 2 constructions we call *rollup** and investigate how these effect LVR and LP profitability, as well as how they fit within the larger context of Ethereum's rollup-centric roadmap.

2 AMM price evolution

In an AMM, the price of assets is determined by arbitrageurs who exploit price discrepancies between the AMM and other markets like CEXs. When the price of an asset on the AMM deviates from its market value on other platforms, arbitrageurs buy or sell the asset on the AMM accordingly. If such AMM has no swap fees, arbitrageurs move the AMM price all the way to the external price and all the LVR is extracted, see Figure 1a. On the contrary, if a fee element is added to those trades, the arbitrage will only happen when it is profitable. In Figure 1b we observe that arbitrageurs only trade if the price is out of a no-trade region (in pale red) determined by the fees charged by the AMM. For example we see that at time T_4 the price is within the no-trade region, and no arbitrage happened. Observe that the arbitrageurs move the price of the AMM until the external price is at the boundary of the no-trade region, which effectively exhausts any price differences with CEXs. This has the interesting consequence that prices at AMMs will always have a slightly different price to CEXs proportional to their trading fee.



(a) Price evolution on an AMM without fees.



(b) Price evolution on an AMM with constant fees.

Figure 1: AMM price evolution on the absence 1a and presence of fees 1b. The blue line represents the market price given by CEXs and the red line the AMM price. This price only gets updated at the block producing times T_i.

In practice there are many other factors that determine the profitability of arbitrage trades, like gas fees and sequencing rules. In Section 5 we explore arbitrage and LVR on three different chains.

3 Reducing LVR

In this section we describe possible approaches to *directly* target LVR. In the first place, we observe that LVR_t depends solely on the price differences observed between the AMM and CEXs at the time t. This means that LVR is market driven and can only be reduced if there is information leaking from the CEXs to the AMM, where sohpisticated actors then move the price of the AMM before further trades are processed.

3.1 Price Oracles

One intuitive approach to solve price differences between external markets and AMMs is to employ price oracles to dynamically define the trading curve. This is the central idea of [KFG21], where the authors prove that the use of low latency price oracles to continuously adjust a constant product AMM leaves no room for arbitrage, and LVR is converted into an equivalent gain for LPs.

An alternative method is presented in [Ber+23], where the authors propose a new AMM design in which price discovery does not exclusively rely on arbitrageurs. They propose the use of price oracles, that determine certain trading requirements and effectively block arbitrage. They show that, even when these oracles present certain delays, the LP performance is improved compared to traditional CFMM.

3.2 Batch trading

Order Flow Auctions (OFAs) are mechanisms designed to enhance the efficiency of transaction settlements while minimizing the potential for Maximal Extractable Value (MEV). OFAs employ a competitive bidding system involving third parties to ensure that users get the most value from each transaction. The process of an Order Flow Auction involves several steps. Initially, orders from multiple users are gathered and grouped together. These orders are then presented to the auctioneer, signaling the initiation of the auction. Bidders analyze the orders within the auction and optimize their bids based on predefined criteria. Subsequently, the auctioneer selects the winning bids according to specified objectives. The winning bids, along with all the orders included in the auction, are then settled onchain.

In this section we investigate the effect of trading in batches and how this could mitigate LVR. In Section 4.2 we explore protocols tax arbitrageurs for the right of extracting LVR.

Batch trading involves collecting trades over a specified time interval and then settling them collectively, as opposed to trading in continuous time. This concept is reminiscent of the approach in traditional finance, where the batching of trades is utilized to address the challenges posed by high-frequency trading (HFT) and to safeguard the interests of

regular or slower traders. When trades are batched, arbitrageurs compete based on price rather than speed. Within a batch, the priority of execution is determined by the price, allowing for a fairer and more efficient trading environment. In an AMM context, batching also eliminate sandwich attacks, since batches are settled at the same price. This also implies that if the price of the AMM moves in the right direction within a batch, LVR could be reduced by not letting arbitrageurs exploit the price differences with CEXs. Below we describe the approach of the two main protocols working around this idea on Ethereum.

- In [CF23], the authors introduce the function-maximizing AMM. In this protocol, trade intents are collected on a public mempool. At the end of the block, the trades are settled in a batch, which provides the settlement price according to the net trade of the batch. In presence of arbitrageurs, this settlement price converges to the market price. Arbitrageurs submit profitable trades on a perfect equilibrium environment, and in such equilibrium $\mathsf{LVR} = 0$. The surplus obtained by this batch is directly transmitted to the LPs by increasing the liquidity of the pool. This protocol has recently been launched as CoWAMM.

- Ångstrom [Sor24] is a Uniswap v4 Hook designed by Sorella Labs that incorporates a batch auction mechanism to offer swappers reliable settlement guarantees at a uniform settlement price. Transactions over all Uniswap v4 pools are batched together offchain via a reth module, and only a net transaction is submitted to the hook. Price discovery is provided by arbitrageurs that compete in an offchain auction. We provide more details of this protocol in Section 4.2.

These solutions rely on highly specialized solvers that look for optimal routes to trade liquidity from different sources. In practice, most solvers that perform these arbitrage transactions on traditional AMMs share most of the expected profits in form of high priority fees to block builders. This allows their trade to be included at the top of the block. By effectively blocking MEV, block builders are not bribed anymore and solvers can be directly rewarded for updating the price of the AMM. Therefore, designing solid economic incentives to attract arbitrageurs is imperative for the sustainable operation of these protocols.

3.3 Request-for-Quote

Traders in a request-for-quote (RfQ) system receive signed quotes upon submitting trading requests, with the option to finalize transactions by accepting the provided quotes at a later stage.

- HOT AMM [AV24] is a new DEX on Ethereum, by Arrakis Finance and Vaantis Labs. Its hybrid design proposes two types of execution in parallel: a permissionless AMM and an RfQ system that uses a special type of signed order quotes. The RfQ allows solvers to access deterministic quotes on liquidity in the pool at up to date prices determined offchain, while the constant function AMM ensures liveness for traders even if the RfQ quoting system censors or stops issuing quotes altogether. When

solver quotes are processed onchain, the signed quote also acts as a price oracle to the AMM updating the spot price. HOT AMM also introduces a dynamic fee that increases at each block until a new signed quote and price change is submitted by a solver, thus providing LVR mitigation whenever signed quotes with up to date price information land onchain before an arbitrage opportunity beyond the no-trade region is exposed and exploited in a block.

- Swaap v2 [Swa23] is another attempt at non-custodial RfQ market-making infrastructure. It provides liquidity services with built-in defensive modules allowing for onchain max drawdown circuit breaker, last look, and other dynamic forms of funds protection. Although this protocol aims to reduce loss-versus-holding or LVH, this is an implicit consequence of LVR reduction by means of leaking oracle prices information into the trade execution flow.

It is worth noting that all of the above designs for *directly* reducing LVR converge on introducing offchain rules that permission how the spot price of the AMM can be updated in one form or another. Whether this permissioned offchain system is a Price Oracle (feeding price information from some offchain system), a Batched OFA (permissioning access to pool liquidity to the batching and auction operators), or an RfQ (providing signed quotes offchain which also function similarly to a price oracle) – the outcome is that offchain systems or actors must be introduced to reduce the value that a DEX smart contract offering liquidity on a discrete-time blockchain naturally leaks (due to continuous-time pricing on external CEX venues).

4 Increasing FEE

As we saw in the previous section, the mechanisms to directly decrease LVR are very limited, and therefore new approaches are needed. In this section we describe different implementations of the FEE term in (1), which happen to counterbalance LVR and increase LPs profitability.

From now on, we will consider without loss of generality that all the charged fees are given back to LPs. In order to study FEE we propose to first decompose it as

$$\mathsf{FEE} = \mathsf{FEE}^{\mathsf{arb}} + \mathsf{FEE}^{\mathsf{noise}} \tag{2}$$

where, $\mathsf{FEE}^{\mathsf{arb}}$ accounts for the fees charged to arbitrageurs/toxic-flow and $\mathsf{FEE}^{\mathsf{noise}}$ are the fees collected from uninformed/noise traders. We will denote by $\mathsf{fee}^{\mathsf{arb}}$ and $\mathsf{fee}^{\mathsf{noise}}$ the corresponding fees per unit of traded liquidity.

4.1 Dynamic fees

We know from [Mil+22] that LVR is increasing in the volatility of the trading pair. In consequence, a response to LVR could also come as a FEE depending on the volatility, among other factors. This is known as *dynamic fees*.

Several blog posts have studied this approach in detail [Els23; Lab23]. Below we describe some solutions and its implementation designs.

4.1.1 Uniform dynamic fee

In this context there is no distinction between the toxic flow and the uninformed flow, as if both parties played the role of an extractor. Thus, they are both charged the same fee, i.e. $\mathsf{fee}^{\mathsf{arb}} = \mathsf{fee}^{\mathsf{noise}}$.

- Historical volatility: Several protocols like Algebra [Alg22], Trader Joe, Hypersea, HydraSwap [Hyd22] determine an approximation of the current realized volatility by looking at the current price compared to its time weighted average version. This volatility is then combined with the pool volume to determine a dynamic fee, which adds to a base fee.
- Multivariate Dynamic Fee: Ambient Finance [Amb23] suggests that volatility is a general market condition and proposes to observe volatility on several pools to determine the correct dynamic fee.

The use of weighted averages as an internal oracle diminishes the risks of adverse manipulation. However, it lacks accuracy and possibly fails to correctly tax arbitrage trades in many scenarios. On the other hand, the multivariate fees model aggregates information of several sources to avoid malicious attacks. In this case, it is necessary to identify highly correlated pairs to determine the precise volatility for each group. In general, the use of uniform dynamic fees also affects noise traders that might look for other sources of liquidity in high volatility scenarios. Moreover, with the surge of trading aggregators, this uninformed flow is efficiently routed through AMMs and other sources. Therefore, these systems can struggle to remain competitive and be forced to reduce their fees, which then makes them less effective to mitigate LVR.

4.1.2 Source-dependent dynamic fee

Another approach is to distinguish between the two order flows described above. By increasing $\mathsf{fee}^{\mathsf{arb}}$ only, we can directly reduce arbitrageurs profits without affecting retail or institutional traders. This improves the overall usability of the AMM in a highly competitive market.

- Transaction-source based: In a series of blog posts [Cro23], the author determines several factors that could help discriminate between toxic and retail traders, such as trade sizes and frequency, known past history of arbitrage and MEV extraction, aggregated vs non aggregated trades, among others. A complete solution may use this information to apply different taxes to different sources, although the complexity of such system seems not viable at the moment. In the future, coprocessors [Emp23] could take an essential role to delegate huge computational tasks without sacrificing decentralization.

- Protocol-source based: Solvers from CoWSwap can access liquidity at reduced fees on Balancer. This idea can be replicated in new modular systems like Uniswap V4 or Valantis.

- Asymmetric adaptive dynamic fee: Project Guidestar aims to distinguish between arbitrage trades from uninformed flow by charging higher fees to trades that are consistent with previous price changes and lower fees to trades that go in the opposite direction. This could eventually be improved by using external oracles to determine more effective mechanisms to adapt trading fees.

Effectively determining whether traders are arbitrageurs requires not only onchain and high-frequency offchain data, but access to reputation records and trading history. Designing reliable systems that avoid false positives and are sufficiently decentralized constitutes a major challenge. The solutions above provide a first glance at how these systems will look in the future.

4.2 Auctioned MEV redistribution

In this section we present several implementations of auctioned processes that aim to mitigate LVR by selling the right to extract this value.

- In [jos22] the author proposes a permissionned protocol called McAMM. In this design, the right to arbitrage the pool is auctioned ahead of time. The highest bid will be given back to LPs as an extra fee.

- Diamond [MDM23] is a permissionless protocol, also implemented as a Uniswap V4 Hook for simple Uniswap V2 type pools. It forces the block builder to commit to the amount of LVR in the block and tax them on it. See also [Par24].

- Ångstrom hook [Sor24] by Sorella Labs also incorporates an *ex-ante* auction for the right to arbitrage the pool at the top of the block. This provides noise traders with execution near the CEX price, and together with the batching of transactions, offers MEV protection.

- Recently, a new design was introduced in [Ada+24]. This so-called am-AMM auctions the right to set the trading fees K blocks into the future, where K is set high enough to obtain good censorship resistance properties. The auction happens totally onchain and is implemented as a Harberger lease. The winner of the auction can trade with zero fees, allowing the extraction of smaller price differences. Effectively, this transfers some of the risks incurred by LPs to solvers and builders that are more capable of pricing such variations by changing the trading fees.

Although LVR can be effectively mitigated with these designs, other MEV sources like sandwich attacks remain possible. Also, solver incentives are provided either by taking a cut on the LVR mitigation or via extra economic mechanisms like token rewards.

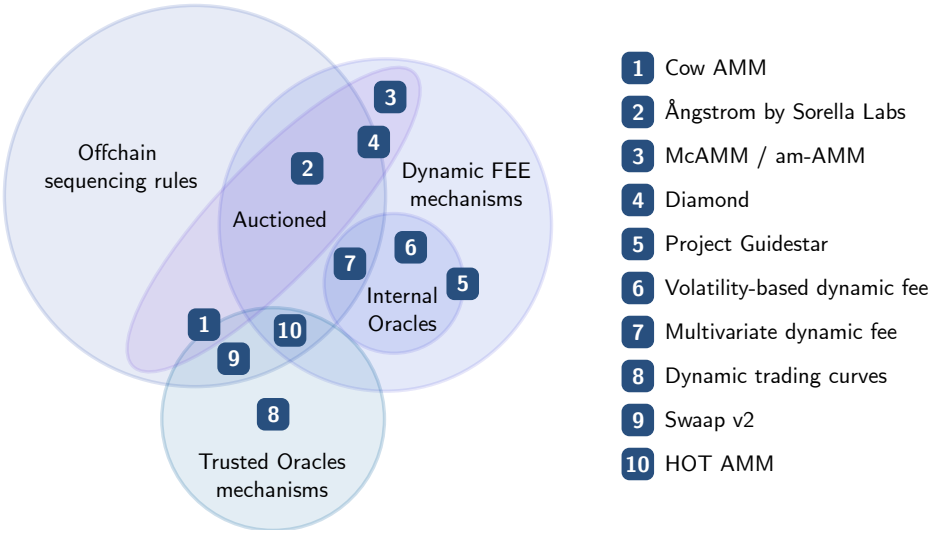


Figure 2: Diagram illustrating the different protocol approaches to LVR reduction or mitigation. Protocols are set at the boundary of certain categories when the feature is less central or still undefined.

4.3 Increasing block production speed

Intuitively, increasing block production rates shortens the time interval where the prices can change and, therefore, reduces the amount arbitrageurs can extract at each block. However, if AMMs charge no fee for these trades, the absence of memory of price processes implies that the total extractable amount by arbitrageurs is not reduced but just split on different blocks. This means that arbitrage opportunities only happen at a block with a certain probability depending on the interblock time.

In [MMR23] the authors prove asymptotic results in that direction. In particular, that under the presence of small fees, the extractable value by arbitrageurs converges to 0 as the interblock time Δt goes to 0. However, the instantaneous LVR, as defined in [Mil+22, Theorem 1] remains unchanged by faster block production. Thus, the extractable value is captured by the trading fees of the underlying AMM. If we denote by σ the volatility of the market price and by fee the trading fee of the AMM, [MMR23] shows that the value extracted by arbitrageurs ARB satisfies

$$\mathsf{ARB} \approx \mathsf{LVR} \cdot \frac{1}{\sqrt{2}} \frac{\sigma\sqrt{\Delta t}}{\mathsf{fee}}, \tag{3}$$

which implies that

$$\mathsf{FEE}^{\mathsf{arb}} \approx \mathsf{LVR} - \mathsf{ARB}. \tag{4}$$

In other words $\mathsf{FEE}^{\mathsf{arb}}$ converges to LVR as the interblock time goes to 0. More interestingly, we can estimate $\mathsf{FEE}^{\mathsf{arb}}$ as function of Δt. We observe that a reduction of 75% in

Blockchain	Block time	No. of blocks	No. of swaps	Blocks with swaps	Median swap fee
Ethereum	12s	962685	687259	48.5%	$ 10
Arbitrum	0.25s	43755889	1341997	2.1%	$ 0.25
Optimism	2s	5813715	220254	3.6%	$ 0.28

Table 1: Blockchain swap data on the observed timeframe.

block production times would imply an increase of 100% of $\mathsf{FEE}^{\mathsf{arb}}$, or a reduction of ARB by 50%. The remarkable fact of this mechanism is that trading fees do not need to increase in order to capture that value, i.e. noise traders are not affected and the LP profitability depends exclusively on $\mathsf{FEE}^{\mathsf{noise}}$.

In practice, the ARB term must also contain the fixed costs of trading, like gas fees. In [Nez23], the author proposes a correction for Equation (3) in the case of Uniswap V3, which takes the following form

$$\mathsf{ARB} \approx \mathsf{LVR} \cdot \left(1 + \frac{2 \cdot \mathsf{fee}}{a + \sigma\sqrt{2\Delta t}}\right)^{-1}. \tag{5}$$

The correcting term a is given by

$$a := 2\sqrt{\frac{F}{L\sqrt{p}}},$$

where F is the fixed cost per swap, L the liquidity in Uniswap V3 terms and p is the current price of the pool. It is shown that in a simulated Ethereum with block times of 3 seconds, ARB is reduced by 30% instead of the 50% reduction implied by (3).

5 Data Analysis

In this section we test some of the theoretical hypothesis in the previous sections by using real data. This analysis is complementary to the recent study presented in [Ada24], where the focus is given to the analysis of fees and not LVR. We compare the Uniswap V3 WETH-USDC pool with 0.05% fee on Ethereum, Arbitrum and Optimism networks. We observe these networks between October 1, 2023 and February 12, 2024, in which period they were all active. We also use Binance API data for the ETH-USD pair at 1 second resolution. It is important to mention that these 3 chains have different features, that we describe below.

- Ethereum is a Layer 1 blockchain with high gas fees and block time productions. Also, transactions on Ethereum need to be submitted to block builders up to 4 seconds prior to the settlement of the block. These builders can sequence the transactions in the

order that generates more MEV, and can also receive bribes in form of maximum gas priority fees.

- Arbitrum and Optimism are Layer 2 blockchains, in particular they are optimistic rollups, see Section 6.1. These rollups generate and post *fraud proofs* to Ethereum, which ensures the integrity of the transactions processed on the Layer 2. As we see in Table 1, Optimism has interblock time production of 2 seconds, and Arbitrum reaches sub-second production times. Gas fees on both of these chains are insignificant compared to Ethereum and both sequencers are centralized, supposedly working under a first-come-first-serve dynamic. This means that arbitrageurs compete on speed and usually send their trades at the beginning of the block.

  The main difference between these two chains is that funds are locked on Optimism during 7 days, on which anyone can challenge a fraud proof. On the contrary, Arbitrum provides instant withdrawals from its Ethereum bridge, without the need to wait for these disputes to be resolved.

These different infrastructures present risk trade-offs that can be reflected in the strategies used by arbitrageurs. In order to study LVR on these chains, we assume that for each arbitrage trade, its price impact is proportional to the underlying liquidity of the pool, with the same proportion factor for the three chains. For example this would be the case if the pools have equivalent liquidity profiles, i.e. the same profile up to a vertical scaling factor. This is a reasonable assumption for two reasons. First, the pool we are studying has a tick spacing of 10, or approximately 0.1%, and the fee of such pool is 0.05%. Second, arbitrages trades tend to move the price through a limited set of ticks, and in this case it could be enough to only look at an horizon of 3 to 5 initialized ticks, or 0.3% to 0.5%. Our initial assumption is then reduced to having locally equivalent liquidity profiles. This is in general the case, since Uniswap V3 profiles locally look as Uniswap V2 pools. In practice this means that the only quantity we need to look at is the difference in price between Binance and the pools.

Since identifying arbitrage trades is a complicated task, we consider the net trade at each block, and we compare the price of the pools at the end of the block with Binance prices at the time the arbitrage trade should be sent for sequencing. Formally, for a given block n, we denote t_n the timestamp of the block, p_0^n the price of the pool at the top of the block and p_f^n the price at the bottom of the block. For a given time t, we denote p_b^t the price at Binance at t. We also denote by s the average time delay between sequencing closes and the block is produced. We say that a block n has *price arbitrage* if either

$$\begin{cases} p_0^n(1+\mathsf{fee}) < p_b^{t_n-s} \\ p_0^n < p_f^n < p_b^{t_n-s} \end{cases} \quad \text{or} \quad \begin{cases} p_0^n(1-\mathsf{fee}) > p_b^{t_n-s} \\ p_0^n > p_f^n > p_b^{t_n-s} \end{cases} \tag{6}$$

In such case, the price arbitrage at this block is respectively given by

$$(p_f^n - p_0^n)(1+\mathsf{fee}) \quad \text{or} \quad (p_0^n - p_f^n)(1-\mathsf{fee}).$$

Blockchain	Blocks with price arbitrage	Relative cumulative price arbitrage	Expected relative LVR
Ethereum	26.9%	-	-
Arbitrum	16.7%	48.3%	14.4%
Optimism	46.3%	88.8%	40.8%

Table 2: Price arbitrage on the studied blockchains.

Notice that the second condition in (6) allows for blocks where the final price does not make economical sense for an arbitrageur, however, since we are considering the net trade of each block, this could eventually happen due to noise trades. Let us denote by I_A the set of blocks with price arbitrage. Hence, the cumulative price arbitrage between blocks i and j is given by

$$\sum_{n=i,\ n\in I_A}^{j} |p_f^n - p_0^n|(1 + \mathsf{fee} \cdot sgn(p_f^n - p_0^n)).$$

where sgn is the sign function. On Ethereum (resp. Optimism and Arbitrum) we consider $s = 4$ seconds (resp. 1 second and 0 seconds) before the timestamp of the block.

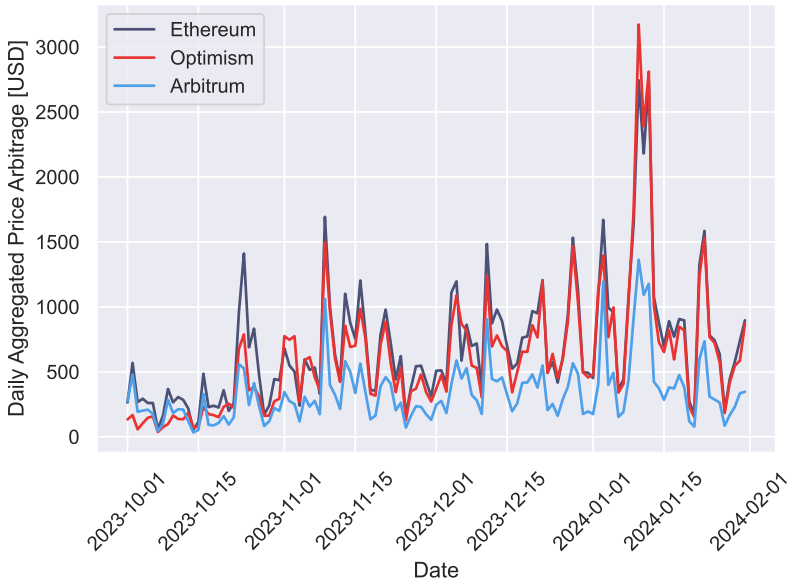
The results of this analysis can be seen in Figure 3. As we observe in Table 2, we see less blocks with arbitrage in Arbitrum compared to Ethereum. This is due to shorter block time productions on Arbitrum. However, we see that Optimism has more blocks with arbitrage, with suggests that arbitrage trades happen more often in this chain. Also, we confirm that faster interblock times reduces price arbitrage. Under assumptions, it should reduce LVR in the same proportion. However, we see that the expected relative LVR is far from being reached by this increased throughput. Security guarantees of these chains as well as withdrawal delays might be the main causes of this difference.

6 The Ethereum endgame

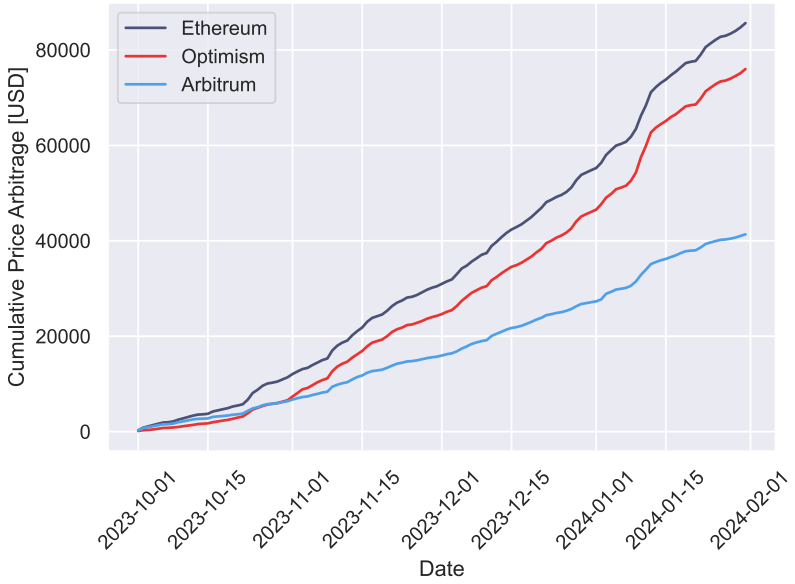
To understand the future of LVR we need to understand the mechanisms that lead this value to be leaked as MEV. In Section 3 we saw that the only possible ways to actually reduce the LVR term in the LP profitability equation (1) are either to use low latency oracles or to introduce offchain sequencing rules for transactions that touch AMM liquidity. In this section, we analyze how this second group of offchain solutions fits in the Ethereum roadmap and the impact that new sequencing designs may have on LVR and DEXs.

6.1 Scaling Ethereum Compute

The rollup-centric Ethereum roadmap [But20] seeks to address network scalability issues while maintaining compatibility with the secure and decentralized Ethereum infrastructure



(a) Daily aggregated price arbitrage.



(b) Cumulative price arbitrage.

Figure 3: Price arbitrage on studied blockchains. Under our assumptions this measure is proportional to LVR.

that exists today by introducing a layered architecture for blockchain compute and validation. Put very simply, the users of a rollup, or Layer 2 blockchain more generally, enjoy higher transaction throughput and cheaper transaction execution on a more performant but less secure and decentralized system. However, by the Layer 2 system intermittently posting commitments to its state transitions and resulting state onto the Ethereum Layer 1 blockchain, the Layer 2 inherits certain key guarantees about the security of the system from Ethereum. The term rollup just connotes a specific type of layer 2 construction that maximally inherits desirable properties from the Layer 1 blockchain (as opposed to Layer 2 which connotes the larger design space of this layered architecture of which rollups are a subset). Blockchain applications can deploy on general purpose rollup blockchains or create their own application-specific rollup blockchains to enjoy the benefits of faster and cheaper compute without sacrificing on Ethereum security, thanks to this architecture.

Application specific coprocessors are another envisioned solution for scaling the computation related to applications on Ethereum. In the coprocessor model, some computations that an Ethereum smart contract would have executed onchain are moved offchain and only the results are delivered to the Ethereum blockchain (often with accompanying proofs of correctness for the results). This reduces the amount of computation that needs to be executed by the highly compute constrained Ethereum Virtual Machine, while still allowing applications to build functionality that rely on more intensive computation. What distinguishes a coprocessor from a rollup or a Layer 2 blockchain construction is that a coprocessor is *stateless* and not a state machine. In this paper we argue that most of the relevant DEX designs that introduce reliance on offchain computation are not entirely stateless and do not actually fit neatly in the coprocessor model. Instead, we propose and define a new type of Layer 2 which has weaker properties than classic rollups but still maintains chain state that we call *rollup** (read as rollup star). By providing this new class we aim to understand the essential properties of a set of protocols that heavily rely on stateful offchain computation but retain acceptable security guarantees for users interacting with the system.

6.2 Rollup and Rollup*

First, we will depict the essential properties that define a classic rollup. Then, we will focus on specific elements that correctly shape this new *rollup** framework. Lastly we will see how current protocol designs fit into this view.

Many discussions have arisen around the definition of rollups. See for example [Cha23], [Fra23] or [Pre23] and the references therein. In this paper, we consider the following definition

Definition. *A rollup is a blockchain that posts their blocks to another blockchain, and inherits the consensus and data availability of that other chain.*

This means a rollup blockchain has its own nodes handling the state of its own state machine and execution of its own transactions that invoke its own state transition function. What makes this blockchain a *rollup* is that it intermittently posts compressed state transition data and a commitment to the resulting state in aggregated checkpoints onto the host

chain (Ethereum), in such a way that the rollup blockchain's state inherits the consensus and data availability from the host chain directly. One or more of the rollup nodes takes the role of the sequencer, which is responsible to batch the state transitions of the rollup blockchain and provide a checkpoint to the host chain. This usually consists of: ordered rollup transaction data since the last checkpoint, the resulting state root of the rollup, and a proof that gives assurances that applying the newly presented ordered transactions on the rollup pre-state (identified by the last checkpoint's state root) would indeed result in the new state root checkpointed. Crucially, on a classic rollup, users should be able to reconstruct the full state of the rollup chain up to the last checkpoint just by having observed the host chain (even if all rollup nodes spontaneously become forever unavailable).

With respect to the sequencer role and the proofs it presents to Ethereum, the rollups in production today are at differing stages. On Optimism there is only one centralized sequencer that directly posts the compressed data and resulting state root to Ethereum, without any proof system for verifying/challenging the integrity of the checkpoint. On the other hand Arbitrum transactions are processed by a centralized sequencer and posted to Ethereum by a second node called the batcher. A set of 20 validators, which are also Ethereum nodes, execute the state transitions of these batches against their local state and in case of fraud detection, they can submit a fraud proof to Ethereum within a fraud proof window. On ZkSync state transition proofs actually use zero knowledge proof cryptography to attest to the integrity of the latest state root, removing the need for state transition proofs to rely on a fraud detection window.

Definition. *A rollup* is a state machine that posts transition steps to another blockchain, and inherits the consensus and data availability of that chain for critical rollup state.*

Observe that rollups* are not necessarily blockchains and may have extra trust assumptions. Most importantly, rollup* state is split into critical and non-critical partitions, and the non-critical state does not necessarily inherit Ethereum guarantees, for instance the non-critical state transition data may not be made available on Ethereum. However, by inheriting consensus and data availability of the host chain *for critical state*, particularly for state relating to the assets belonging to users interacting with the system, these users still inherit the important security guarantees.

Let us provide an example of system that fits into this context. Consider a classical constant function AMM where two-sided liquidity is locked in a smart contract on Ethereum, but passing trades through this AMM is permissioned see Figure 4. The liquidity pool can only be accessed with a signature from the sequencer role. Users submit trading intents but solvers compete to sequence all the user intents and their own intent to trade. Notice that liquidity providers can withdraw their funds instantly, which is the guarantee provided by Ethereum. In order to protect them against excessively adversary trades, smart contracts on Ethereum can check for oracle deviations up to a certain threshold. On the other hand, traders can also inherit part of this security by imposing a limit price verified on Ethereum smart contract. This AMM qualifies then as a rollup*. If this system had history record of all the intents that were submitted and posted proofs that the auctions were fairly executed, this would be then an application specific rollup.

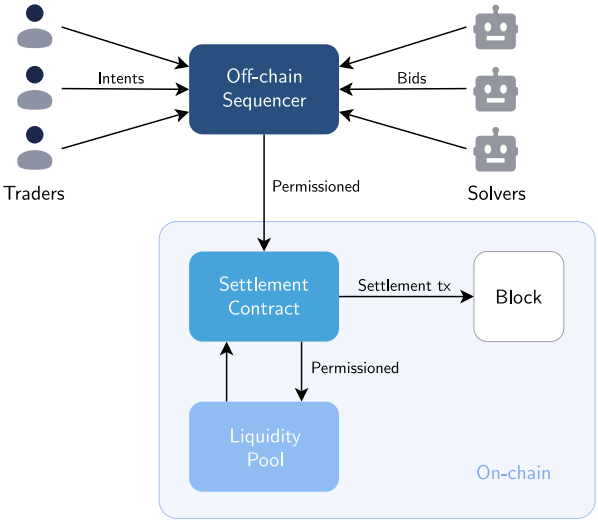


Figure 4: Simple diagram of permissioned AMM design, that also falls in the rollup* category.

Protocols like HOT AMM, CoWAMM and Ångstrom are all rollups*. By publishing verifiable proofs to Ethereum of the offchain computation they rely on, as well as providing a record of past history to recover state at any time, they would become true application specific rollups. Under this premise, they would also inherit the same security guarantees of Ethereum as classic rollups like Arbitrum and Optimism.

Other properties like censorship resistance are not inherent to rollups, and in practice they are difficult to execute. For example, on Arbitrum it is possible to force include a transaction directly on its bridge contract on Ethereum. However this is quite expensive and not practical.

7 Conclusion

In this paper, we provided an up-to-date taxonomy of the solutions that aim to reduce or mitigate LVR, which is a crucial factor impacting the profitability of LPs.

Our analysis confirms that direct reduction of LVR can only be achieved through the introduction of trusted oracles or specific offchain sequencing rules for transactions interacting with AMM liquidity. On the other hand, various mechanisms have been proposed to mitigate LVR by increasing the fees captured by liquidity providers, either through dynamic fee models, auctioned MEV redistribution, or increased block production rates.

We further propose a new framework, termed rollup*, to classify a class of protocols that heavily rely on stateful offchain computation while retaining acceptable security guarantees for users. This framework helps to understand the essential properties of these protocols and their relationship with the rollup-centric Ethereum roadmap.

Interesting further research topics include empirical validations on the LVR reduction or mitigation of the cited protocols, designs of sustainable incentives for solvers to participate into these new markets, decentralization of offchain architectures, and solutions to liquidity fragmentation across rollups and rollups*. Concerning this last point, novel ideas are emerging to modify Ethereum sequencing, like based preconfirmations [Dra23b; Dra23a] or execution tickets [Neu23]. These solutions not only have consequences on LVR by bringing lower latencies to Ethereum, but also would allow atomic composability across all the integrated layers.

References

[Ada24] Austin Adams. "Layer 2 be or Layer not 2 be: Scaling on Uniswap v3". In: *arXiv preprint arXiv:2403.09494* (2024).

[Ada+24] Austin Adams et al. "am-AMM: An Auction-Managed Automated Market Maker". In: *arXiv preprint arXiv:2403.03367* (2024).

[Alg22] Algebra Finance. *ALGEBRA Ecosystem: Decentralized exchange*. 2022. URL: https://algebra.finance/static/Algerbra%5C%20Tech%5C%20Paper-15411d15f8653a81d5f7f574bfe655ad.pdf (visited on 01/24/2024).

[Amb23] Ambient Finance. *Ambient Finance docs*. 2023. URL: https://docs.ambient.finance/users/dynamic-fees (visited on 01/24/2024).

[AV24] Arrakis Finance and Valantis Labs. *Hybrid Order Type: A New MEV Aware AMM Design*. 2024. URL: https://github.com/ArrakisFinance/research/blob/main/HOTAMM-Whitepaper.pdf (visited on 05/09/2024).

[Ber+23] Philippe Bergault et al. "Automated market makers: Mean-variance analysis of lps payoffs and design of pricing functions". In: *Digital Finance* (2023), pp. 1–23.

[But20] Vitalik Buterin. *A rollup-centric Ethereum roadmap*. 2020. URL: https://ethereum-magicians.org/t/a-rollup-centric-ethereum-roadmap/4698 (visited on 01/24/2024).

[CF23] Andrea Canidio and Robin Fritsch. "Arbitrageurs' profits, LVR, and sandwich attacks: batch trading as an AMM design response". In: *arXiv preprint arXiv:2307.02074* (2023).

[Cha23] Jon Charbonneau. *Rollups Are L1s (& L2s) a.k.a. How Rollups *Actually Actually Actually* Work*. 2023. URL: https://dba.mirror.xyz/LYUb_Y2huJhNUw_z8ltqui2d6KY8Fc3t_cnSE9rDL_o (visited on 01/24/2024).

[Cro23] CrocSwap. *Discrimination of Toxic Flow in Uniswap V3: Part 1*. 2023. URL: https://crocswap.medium.com/discrimination-of-toxic-flow-in-uniswap-v3-part-1-fb5b6e01398b (visited on 01/24/2024).

[Dra23a] Justin Drake. *Based preconfirmations*. 2023. URL: https://ethresear.ch/t/based-preconfirmations/17353 (visited on 01/24/2024).

[Dra23b] Justin Drake. *Based rollups—superpowers from L1 sequencing*. 2023. URL: https://ethresear.ch/t/based-rollups-superpowers-from-l1-sequencing/15016 (visited on 01/24/2024).

[Els23] Atis Elsts. *Dynamic Fees for Automated Market Makers: Liquidity, Volatility, and Collected Fees*. 2023. URL: https://atise.medium.com/dynamic-fees-for-automated-market-makers-liquidity-volatility-and-collected-fees-db211da18d0d (visited on 01/24/2024).

[Emp23] Emperor. *A Brief Intro to Coprocessors*. 2023. URL: https://crypto.mirror.xyz/BFqUfBNVZrqYau3Vz9WJ-BACw5FT3W30iUX3mPlKxtA (visited on 01/24/2024).

[Fra23] Bruno França. *Open letter to Jon Charbonneau (or Rollups = Bridges + Blockchains)*. 2023. URL: https://ethresear.ch/t/open-letter-to-jon-charbonneau-or-rollups-bridges-blockchains/15739 (visited on 01/24/2024).

[Hyd22] Hydra. *A window into AMM 2.0 — Introducing Volatility Adjusted Fee*. 2022. URL: https://medium.com/hydraswap/a-window-into-amm-2-0-introducing-volatility-adjusted-fee-af909b6c8ba5 (visited on 01/24/2024).

[jos22] josojo. *MEV capturing AMM (McAMM)*. 2022. URL: https://ethresear.ch/t/mev-capturing-amm-mcamm/13336 (visited on 01/24/2024).

[KFG21] Bhaskar Krishnamachari, Qi Feng, and Eugenio Grippo. "Dynamic automated market makers for decentralized cryptocurrency exchange". In: *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2021, pp. 1–2.

[Lab23] Cata Labs. *Optimising LP Performance Part 2: Dynamic Fees*. 2023. URL: https://blog.catalyst.exchange/optimising-lp-performance-part-2-dynamic-fees/ (visited on 01/24/2024).

[MDM23] Conor McMenamin, Vanesa Daza, and Bruno Mazorra. "An Automated Market Maker Minimizing Loss-Versus-Rebalancing". In: *The International Conference on Mathematical Research for Blockchain Economy*. Springer. 2023, pp. 95–114.

[MMR23] Jason Milionis, Ciamac C. Moallemi, and Tim Roughgarden. "The Effect of Trading Fees on Arbitrage Profits in Automated Market Makers". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2023, pp. 262–265.

[Mil+22] Jason Milionis et al. "Automated market making and loss-versus-rebalancing". In: *arXiv preprint arXiv:2208.06046* (2022).

[Neu23] Michael Neuder. *Execution Tickets*. 2023. URL: https://ethresear.ch/t/execution-tickets/17944 (visited on 01/24/2024).

[Nez23] Alex Nezlobin. 2023. URL: https://x.com/0x94305/status/1679247321622786048?s=20 (visited on 01/24/2024).

[Par24] Chanwoo Park. *Uniswap V4 hook: LVR-minimization with Per-block conversion vs. Futures contracts*. 2024. URL: https://ethresear.ch/t/uniswap-v4-hook-lvr-minimization-with-per-block-conversion-vs-futures-contracts/18610 (visited on 01/24/2024).

[Pre23] James Prestwich. *Defining "Rollup"*. 2023. URL: https://prestwich.substack.com/p/defining-rollup (visited on 01/24/2024).

[Sor24] Sorella Labs. *Ångstrom Documentation, private*. 2024.

[Swa23] Swaap Labs. *Swaap v2: Optimal liquidity infrastructure*. 2023. URL: https://swaap.finance/v2-whitepaper.pdf (visited on 01/24/2024).