Superposition V1 White Paper

Barter Team

March 2025

Abstract

Superposition introduces a fundamentally new approach to decentralized trading and liquidity provision, addressing key inefficiencies in existing protocols. By redesigning how liquidity is managed and utilized, the protocol enhances market efficiency while ensuring fair execution for both liquidity providers (LPs) and traders. At its core is the Distributed Liquidity Layer, a novel architecture that allows LPs to maintain full ownership of their assets while providing liquidity without the need for asset locking. This approach eliminates unnecessary capital inefficiencies, enables gas-free adjustments of strategies, and ensures seamless liquidity allocation across different provisioning models. For traders, Superposition offers a MEV-free execution environment, preventing front-running and sandwich attacks through a controlled request-for-quote (RFQ) system. Furthermore, the protocol introduces a new class of solvers dedicated to liquidity optimization, allowing LPs to dynamically adjust fees, exposure, and allocation strategies in response to market conditions. By rethinking the interaction between liquidity providers, solvers, and traders, Superposition establishes a more efficient, decentralized, and fair framework for on-chain trading, setting a new standard for the next generation of DeFi protocols.

1. Introduction

Decentralized finance (DeFi) originated as an alternative to traditional finance (TradFi), offering open, permissionless, and user-centric financial services. Nevertheless, despite its rapid growth, DeFi remains a niche market compared to centralized exchanges (CEXs) and especially TradFi. A multitude of DeFi protocols often replicate extractive mechanisms, which inhibits growth and disproportionately burdens regular users. This has led to extreme underutilization of the DeFi's whole transformational potential.

Every year dozens of new decentralized exchanges (DEXs) introduce novel automated market maker (AMM) models aimed at improving capital efficiency, trade execution, and price discovery. Yet, fundamental drawbacks persist, primarily for liquidity providers (LPs). Providing liquidity requires locking assets in smart contracts, LPs suffer far more impermanent loss and LVR than they could due to the peculiarities of existing DEXs. Furthermore, switching liquidity provisioning strategies, even within the same protocol, requires gas costs as well as a series of deposits and withdrawals. All of this leads to massive underutilization of liquidity and its isolation.

At the same time, intent-based protocols are capturing a growing share of order flow, making solvers notable actors in the industry and preserving a lot of value for traders. However, integrating new AMMs remains a bottleneck, limiting the scalability of intent-based protocols. Although a multitude of projects attempt to address this challenge by creating "Solvers' network", the problem remains unsolved.

At Barter, we believe that overcoming these limitations requires a structural shift in the very term "liquidity provision". In this paper, we introduce **Superposition**—a new paradigm that redefines liquidity provision by creating a nexus between LPs, traders, current solvers and a novel type of solvers: asset managers. Unlike current protocols, Superposition operates through the Distributed Liquidity Layer, allowing LPs to maintain full custody over their funds while at the moment of swap utilizing their liquidity to execute swap—akin to flash loans. In this way Superposition enhances capital efficiency and ensures fair distribution of value creating a mutually beneficial environment across all participants:

- **For LPs:** LPs provide liquidity by keeping tokens fully transferable on their wallets. LPs have access to a gas-free switching of pricing mechanism, which includes all existing AMMs, PMM-like pricing, and our Stable Swaps. Moreover, liquidity providers can delegate position management to asset managers, increasing their yield. Therefore, for LPs, Superposition provides a disruptive solution.

- **For traders:** Superposition provides a MEV-protected execution environment as well as order matching, enabling better execution prices.

- **For solvers:** Since LPs can select a pricing mechanism similar to any AMM or PMM, an interested solver only needs to integrate Superposition and the Internal Router and Backend will provide the best rate in a robust RFQ format.

- **For asset managers:** Gasless strategy switching and the presence of a vast number of customizable parameters in various pricing mechanisms open up a wide range of opportunities for dynamic LPs' positions management, previously unavailable on any exciting protocols.

2. Distributed Liquidity Layer

Liquidity is the cornerstone of DeFi, essential to the operation and success of any DeFi protocol. The architecture of DeFi protocols has evolved dramatically from separate pools and vaults in Uniswap V1 and Maker DAO respectively to a prominent trend in recent years - the Liquidity Layer used by Balancer V3 [1], Uniswap V4 [2], Fluid [3] and other modern protocols. Liquidity Layer is a single smart contract that stores liquidity of all pools/vaults of the protocol and keeps accounting of token affiliation to a particular pool or vault. On top of the main contract are built contracts responsible for the specific logic of using assets in the Liquidity Layer, which are called applications, hooks, etc. *(Figure 1, a))*. This approach presents several significant advantages - high gas efficiency, easy migration of DApps to new versions, and efficient use of liquidity within the entire protocol.

But what remains unchanged is how we provide liquidity. To provide liquidity users have to send tokens to the pool via the protocol UI, for which they will receive LP tokens/NFT representing their share of the pool. If a user wants to withdraw his tokens or put them in another pool, or change the pool settings, he must first burn his LP tokens and do it all over again, paying for the gas along the way. And the most frustrating thing is that the user doesn't own his tokens, can't send them or swap them at will.

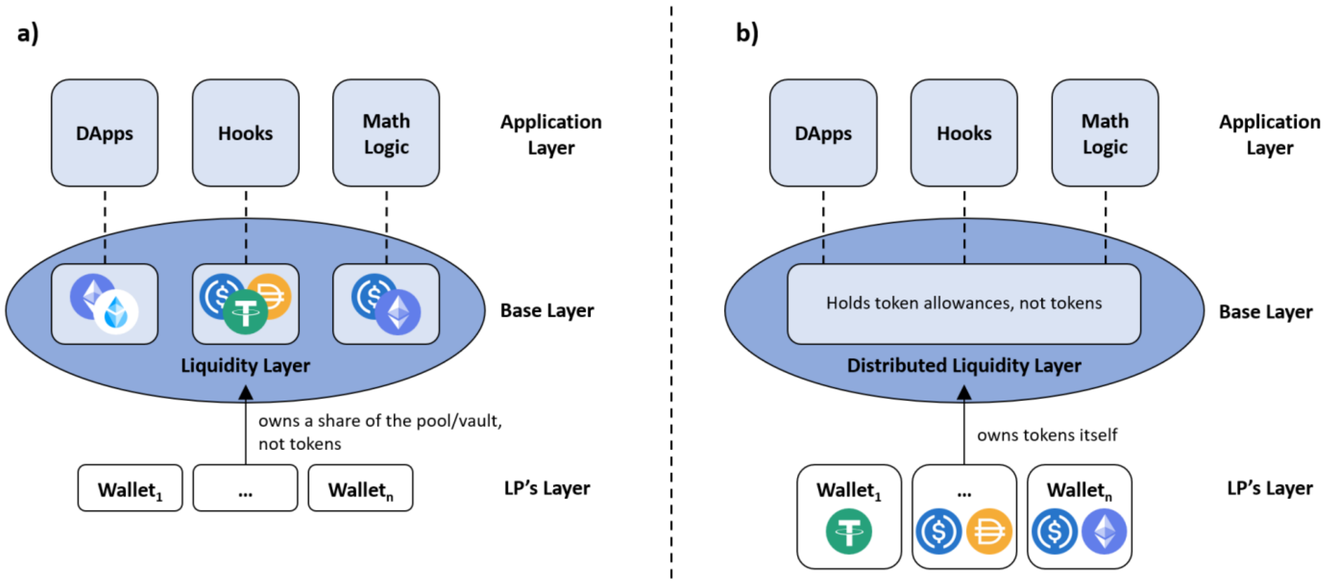

Figure 1. a) Protocol architecture with Liquidity Layer such as Balancer V3, Uniswap V4, Fluid etc. b) Superposition V1 architecture with novel Distributed Liquidity Layer

We are very excited to introduce you to the core of the Superposition Protocol - the Distributed Liquidity Layer. Unlike a traditional Liquidity Layer, Distributed Liquidity Layer contract does not hold the tokens themselves, but only infinite token allowances *(Figure 1, b))*. Such a shift in liquidity provision can be compared to liquid staking, whereby the user would hold their ETH directly on their wallet and accrue yield straight to their wallet.

The Distributed Liquidity Layer allows LP to select various liquidity provisioning models such as Superposition's Stable Swaps, Uniswap-like curve, Balancer-like curve, etc. Moreover, a user can apply several different models/applications to the same tokens and use them simultaneously as long as they do not contradict each other, and Superposition's backend itself will choose the most favorable model for a particular transaction. At any given time, the user retains full control over the tokens in its wallet, and to change the liquidity provisioning model, all the user has to do is change the settings on Superposition's UI, without any deposits/ withdrawals.

Although LP can start providing liquidity with only one token in the wallet, it still needs to provide token allowances for multiple tokens to be used in a trading pair, as the liquidity in Superposition is used for swaps. The initial token will then be gradually exchanged for other stablecoins according to the selected trading pair or for another token until the target tokens proportion is reached in the case of Stable Swaps or AMM-like pools respectively. While the concept of Distributed Liquidity Layer implies the possibility of many applications on top of wallets' liquidity, we expect the possibility of mono-token liquidity provision in further versions of Superposition, e.g. by executing flash loans directly to connected wallets.

When an application taps into a Superposition pool for a swap, the Distributed Liquidity Layer steps in. It seamlessly redirects the input tokens to a randomly selected compatible LP wallet holding the target token. The LP then sends back the target tokens to the application—minus their fee, which stays directly in their wallet.

3. High-level Architecture

Superposition Protocol in the current version consists of two parts - a smart contract holding infinity tokens allowances, aka Distributed Liquidity Layer, and an off-chain backend. The purpose of the off-chain backend is to store all connected wallets data, to select appropriate wallets when a quota is requested by Barter Solver, and to perform all calculations of the exchange rate and return a signed quota to Barter. In other words, all calculations are done off-chain, and the smart contract only distributes tokens among the selected wallets and gives out the required number of tokens outward according to the signed quota. Essentially, the Distributed Liquidity Layer can be considered as a network of wallets offering their own execution price for a particular swap, which are then selected by the Superposition Backend, keeping the swap favorable for both the trader and the liquidity provider.

Such Superposition V1 architecture provides essential safety of LP's funds, since the contract holding infinity token allowances does not contain complex logic that could be exploited to extract user's funds. Nevertheless, since the calculations are done off-chain, to ensure the safety of users' funds before execution of a swap, the contract also compares the quoted execution price with the market price obtained from the oracle and rejects the quota if there is a deviation greater than a set percentage. To provide more security for LPs' funds in the current version of the protocol, only whitelisted entities (Barter Solver) can request a quota from the Superposition backend. Off-chain calculations allow Superposition V1 to remain cost-effective in terms of gas costs, as a swap through the Distributed Liquidity Layer may require multiple ERC20 token transfers across multiple wallets of different liquidity providers. In addition, off-chain calculations allow liquidity providers to expose more custom settings, such as curve type, as less data has to be stored on-chain. However, in future versions of the protocol, our team expects full on-chain migration to achieve greater transparency and decentralization.

3.1 Superposition Execution Flow

As mentioned above, Superposition has both an off-chain and on-chain component, below is a detailed look at the lifecycle of a transaction performing a swap through Superposition's Distributed Liquidity Layer. *(Figure 2)*

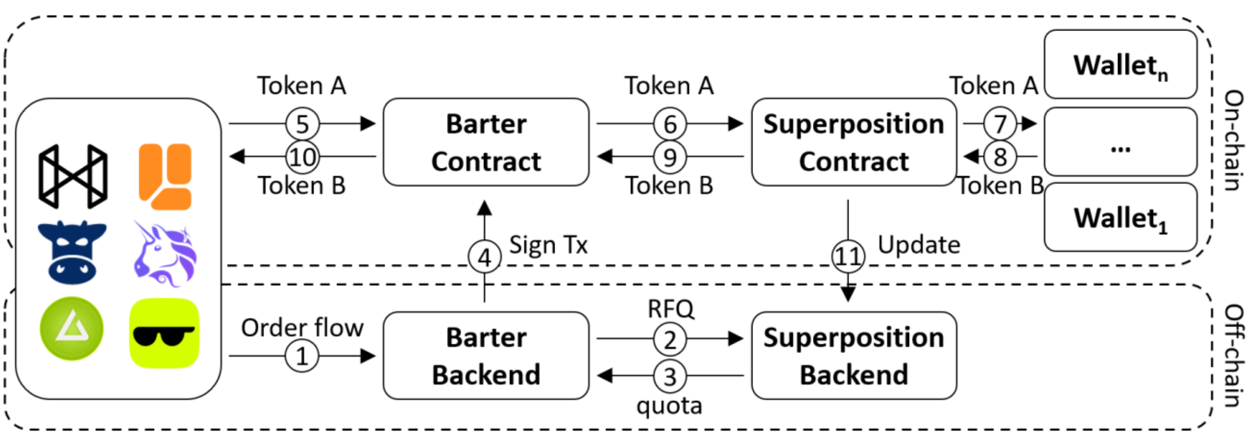

Figure 2. Superposition RFQ-type Execution Flow

1. As with all intent-based swaps, the transaction lifecycle begins off-chain, with Barter receiving the intent to swap.

2. Barter then requests a quote from the Superposition's Backend. At this point, the Superposition's Backend calculates how many tokens B can give for an input number of tokens A, based on a snapshot of the Superposition's LPs' wallets for the last block, and randomly selects the wallets whose liquidity will be utilized.

3. After Superposition Backend calculations, returns a signed quota to the Barter which contains: all selected wallets, input and output amounts of tokens for each selected wallet and a special signature of the Superposition Backend proving that the quota was issued by the Backend.

4. With the signed quota, along with the rest of the route, Barter initializes the transaction. During this step, the quoted price is compared with the oracle price as described above.

5. The Barter contract executing the transaction receives A tokens from an intent DEX end-user. (Depending on the DEX, this could also be an aggregator contract like CoW GPv2 Settlement)

6. Barter sends A tokens and calldata containing quota data to the Distributed Liquidity Layer contract.

7. The Distributed Liquidity Layer contract verifies the Backend's signature in calldata and then distributes the A tokens to the LPs' wallets selected for the quota.

8. The Distributed Liquidity Layer contract sends to itself from the selected wallets the number of tokens B calculated by the Backend, minus the LP's fee that remains on the wallets.

9. B tokens are sent to the Barter contract.

10. B tokens are sent to the intent DEX end-user.

11. The Superposition Backend updates the snapshot of the connected LPs' wallets.

3.2 Superposition Internal Router

The purpose of the Superposition Backend is to select specific wallets, using the liquidity of which the swap will be performed according to the issued quota. When Barter Solver asks a quota for swap x^{in} tokens A to y^{out} tokens B, the Internal Router is responsible for selecting the wallets, solving the problem of maximizing the output tokens according to the constraints of each wallet. The problem is described by the objective function in the *expression 1*,

$$y^{out} = \max_{S\subseteq\{1,\ldots,n\},x_i} \left(\sum_{i\in S} y_i^{used} - gas \cdot |S| \right) \tag{1}$$

where S is the set of selected wallets, n is number of connected wallets and y_i^{used} is the amount of token B utilized from wallet i. Essentially the Internal Router searches a route to multiple "pools", but the complexity is that filling the quota with wallets with the best exchange rate P_i is not always optimal, as interacting with each wallet requires a gas cost equal to two ERC20 token transfers, which is described by gas.

$$\sum_{i \in S} x_i^{in} = x^{in} \tag{2}$$

$$y_i^{used} = P_i \cdot \mathrm{x}_\mathrm{i}^\mathrm{in}, \qquad y_i^{used} \le y_i, \qquad \forall i \in S \tag{3}$$

y_i^{used} depends on the number of tokens B available on wallet i, as well as the chosen pricing mechanism that determines the exchange rate P_i from A to B. *(Expressions 2, 3)*

$$S^* = \underset{S \subseteq \{1,\dots,n\}, x_i}{\operatorname{argmax}} \left(\sum_{i \in S} y_i^{used} - gas \cdot |S| \right) \tag{4}$$

$$|S^*| > 1 \Rightarrow P(S) = \frac{1}{|S^*|} \tag{5}$$

In case the number of appropriate wallets exceeds the minimum required number of wallets, the Internal Router selects a particular wallet randomly to maintain an uniform distribution of trading volume and fees to all wallets. *(Expressions 4, 5)*

4. Mechanisms of Pricing

The exchange rate on each wallet is calculated according to the selected pricing mechanism. Three different pricing mechanisms are available to LPs in Superposition, which they can select depending on the trading pair or their own preferences.

Regardless of the selected pricing mechanism, Superposition pools offer LPs and traders unique advantages and opportunities over any other existing protocol:

- **Funds** are always transferable and **remain in the LP's wallet**
- **Fees** for each swap simultaneously **remain in the LP's wallet**
- Since only whitelisted entities have access to Superposition's RFQ system, any swap cannot be sandwiched and therefore is completely **MEV free**.
- Possibility to create **custom pools** for users providing better rates than any existing pools

4.1 Stable Swaps Issue

Stable swaps, or swaps between tokens pegged to the same asset (e.g., USDC-USDT), make up a significant amount of DEX volume with a 15% share during 2024 or $103.9b in volume only in swaps between stablecoins. [4] As with all other swaps on DEX stable swaps must use AMM logic, but obviously since a stable swap must be executed at a price close to 1 to 1 the classic constant product invariant is not applicable in this case due to a price determined by a token ratio in the pool.

Curve was the first to offer a solution to the problem of stable swaps with its Stable Math and multi-token pools, which are now used on many other DEXs. Stable Math consists in approximating constant product invariant to constant sum invariant in the central part of the curve, which allows to dramatically reduce slippage. [5]

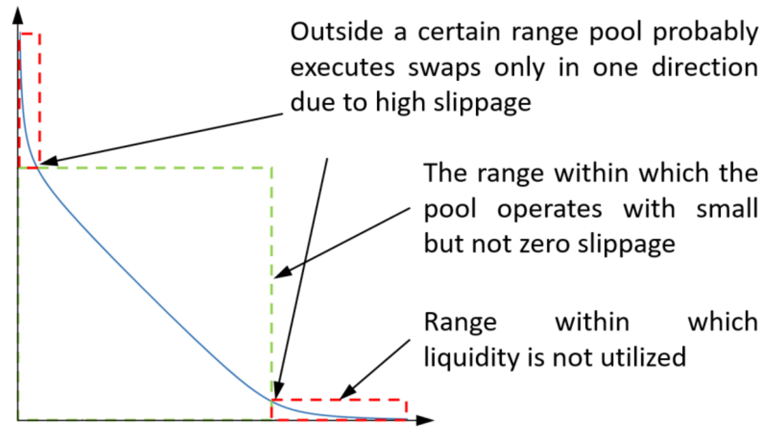

Figure 3. Curve Stable Math type pool

AMM models like Stable Math or Constant Product mitigate the risk to the liquidity provider from the devaluation of one token in the pool relative to another, meanwhile protect against the running out of one token in the pool. In reality, in the case of stable swaps, the risk of depreciation of one stablecoin against another is determined by the liquidity provider's level of trust in the issuer of stablecoin and can be embedded in the **fixed fee** rather than slippage, so a large fraction of liquidity providers do not care if their USDC in the pool is converted into USDT and then into DAI. In case the price between two stablecoins in the pool is relatively far from 1, a rational actor will not continue to exchange at that price, effectively forcing the pool to operate in one direction only. *(Figure 3)*

Furthermore, small slippage is not the same as zero slippage and can lead to noticeable loss of value in swaps, especially between unpopular stablecoins. So 60.6% share or $27b in terms of volume of total DEX trades between stablecoins generated by such extremely sophisticated entities as DEX aggregators was executed by price worse than 1 in 2024. [4]

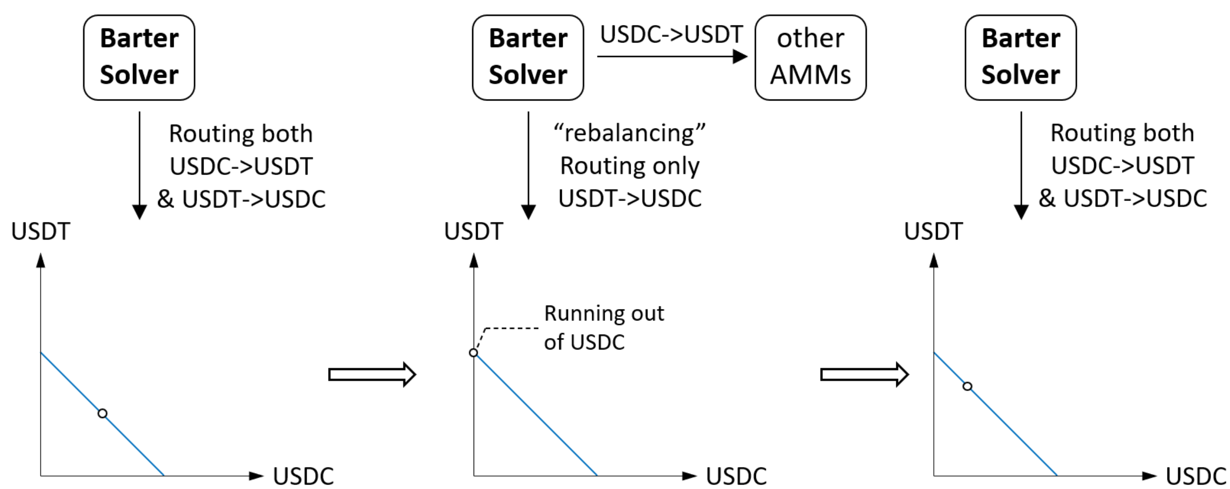

Figure 4. Natural rebalancing of the distributed pool

That's why our team is aiming to uncompromisingly eliminate slippage in stablecoin swaps, using a simple constant sum invariant and fixed fees that determine the risk associated with the depegging of a particular stablecoin. Running out of one of the stablecoins is not a problem, as after some time the distributed pool itself rebalances, executing swaps only in one direction *(Figure 4)*. This logic allows all liquidity to be used to maximum efficiency and makes it economically feasible to swap through a pool even with limited liquidity. Moreover, creating distributing pools that allow for 1-to-1 swaps of stablecoins will reduce slippage throughout the ecosystem. To protect liquidity providers' funds from a significant depegging from the usd of one of the stablecoins, the Superpositions' backend inspects the market price of a particular stablecoin before issuing a quota and, in the case of a depegging, does not perform swaps against risky stablecoin.

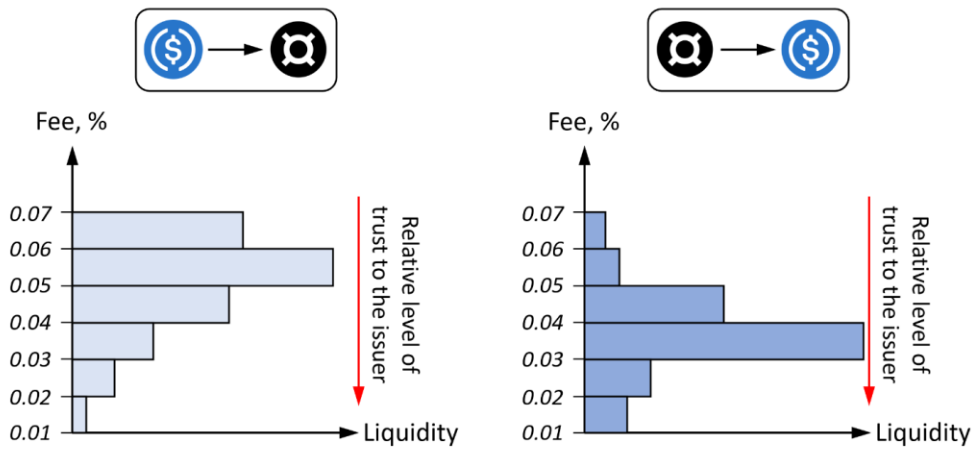

Figure 5. Asymmetric fees in Superposition Stable Swaps based on the level of confidence in a particular stablecoin

As mentioned above, the crucial element of Stable Swaps Superposition is the fixed fees based on the confidence level in the reliability of a particular stablecoin. The confidence level of a stablecoin is an empirical value based on the type of stablecoin, its capitalization, and liquidity. The swap fee is set depending on the trustworthiness of the stablecoin in the pair, and also depends on the direction of the swap. Thus, the fee for a swap of a potentially more trustworthy stablecoin for a less trustworthy one is greater than for a swap in the opposite direction, which balances a fixed exchange rate of 1-to-1. However, since the level of trust is a relative metric, it can differ between LPs with respect to identical stablecoins. Superposition therefore allows LPs to set their own fees, providing a base recommended fee for each pair. *(Figure 5)* Another important implementation of asymmetric fees is asset managers' (described below) ability to control the ratio of stablecoins in the LPs' wallets by dynamically changing the fee to one direction, which allows to achieve maximum velocity of funds.

4.2 AMM-like Pricing

In addition to providing liquidity for Stable Swaps, LPs can provide liquidity for delta non-neutral pairs, similar to traditional DEXs. To accomplish this, users select a pair and two key parameters - the type of AMM curve and the range of token proportion in the pool in dollar equivalent in which proportion of tokens will be maintained by the Superposition Backend.[1] Liquidity providers can choose the AMM curve of any existing or even non-existing

---

[1] LP has to choose a range of ratios rather than a single ratio to be maintained, as in classic AMMs' pools, due to the fact that unlike a AMM's pool where the liquidity of many wallets is aggregated, Superposition executes trades across isolated wallets with a limited amount of liquidity, which makes maintaining a particular ratio inefficient.

liquidity pool along which their tokens will be traded. Superposition pools completely mimic the selected pool, including using the liquidity reserves of the underlying pool to calculate the swap exchange rate at the Superposition Backend.

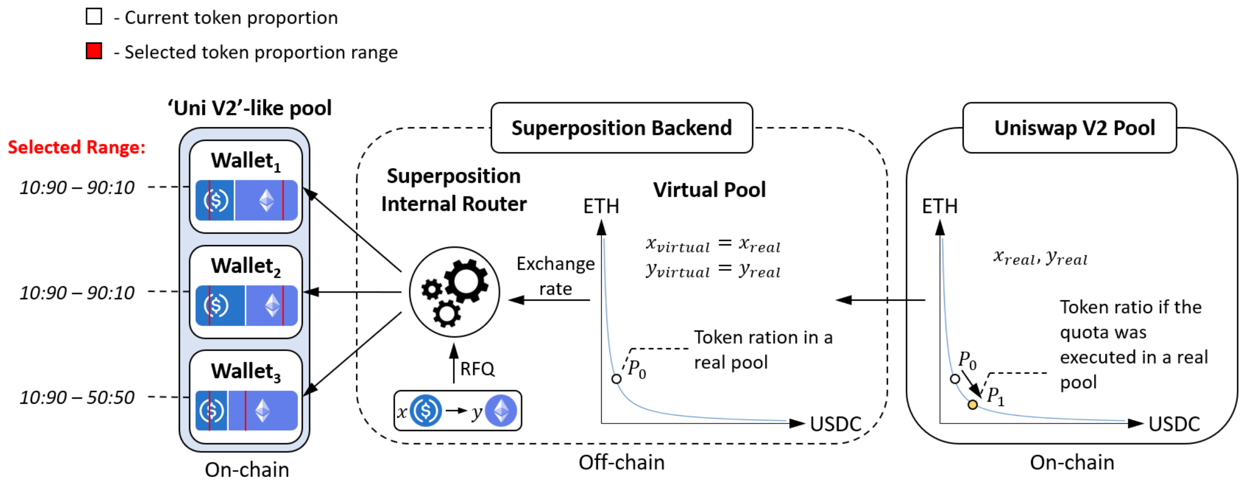

Figure 6. The working principle of AMM-like pools that mimic existing pools across various DEXs

However, the main difference between Superposition pools and classic liquidity pools is that they give DeFi users back full control over their liquidity, i.e., swaps occur directly on the LPs' wallets. So let's take a closer look at the swap process via Superposition "pools". *(Figure 6)* When the Superposition Backend receives a request for quote, the first step it takes is to select the Superposition "pool" or combination of "pools" with the best execution price. To do this, it simulates quota execution in the available virtual pools, special pools that are updated every block and are completely identical to the underlying on-chain pools.

$$R_i^{current} = \frac{y_i}{x_i} \cdot \frac{1}{P_0}, \qquad R_i^{selected} = \frac{y_i - \Delta y_i}{x_i + \Delta x_i} \cdot \frac{1}{P_1} \tag{6}$$

$$\Delta y_i = \frac{y_i - R_i^{selected} \cdot P_1 \cdot x_i}{1 + R_i^{selected} \cdot P_1 \cdot P_i^{observed}} \tag{7}$$

The Internal Router then selects wallets with the maximum number of output tokens available for swap Δy_i, obtainable at the best exchange rate P_i relative to input token.[2] *(expressions 6, 7)* P_i and token ratio after swap P_1 are determined by the underlying pool, whereas Δy_i depends on the current proportion of tokens in the wallet, as well as the proportion range set by the user $R_i^{selected}$. The selected wallets along with the input and output amounts of tokens are then recorded in a quota.

[2] The task solved by the Internal Router is described in Section 3.2.

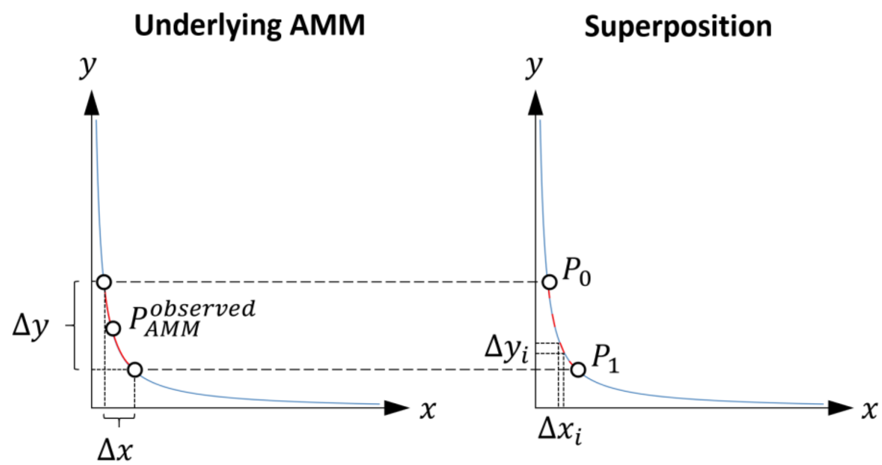

Figure 7. Skipping a part of the underlying curve in the absence of trades at the Superposition

Although Superposition pools fully replicates the underlying pools in calculating the exchange rate P_i for each swap, since the price of tokens is obtained from an external source in the absence of trades, part of the curve is skipped. *(Figure 7, blue areas of the right curve)*

$$P_{AMM}^{observed} = \frac{\Delta y}{\Delta x}, \qquad P_{AMM}^{observed} \in (P_1;\ P_0) \tag{8}$$

$$P_{Sp}^{observed} = \frac{\sum_i \Delta x_i \cdot P_i}{\sum_i \Delta x_i} \tag{9}$$

This leads to a different average execution price $P_{AMM}^{observed}$ when the price moves from P_0 to P_1 in an underlying pool with an average execution price $P_{SP}^{observed}$ in Superposition "pool", which depends on the distribution of trades executed by the Superposition between P_0 and P_1. [3]*(Expressions 8, 9)*

$$\left|P_1 - P_{Sp}^{observed}\right| \le \left|P_1 - P_{AMM}^{observed}\right| \Rightarrow IL_{Sp} \le IL_{AMM} \tag{10}$$

Eventually, this leads to the fact that the impermanent loss at AMM-like pricing in the Superposition may differ from that in the underlying pool, either upward or downward. Which depends on the difference between $P_{AMM}^{observed}$ and $P_{SP}^{observed}$.[4] *(expression 10)*

4.3 PMM-like Pricing

Nowadays, price discovery always occurs in an external source such as CEXs due to high latency and relatively low on-chain trading volume. This causes the current P_0 price inside the pool on DEX to lag behind the current market price P_1, after some time the pool has to be rebalanced to reach the P_1 price. *(Figure 8)*

---

[3] The deviation between $P_{AMM}^{observed}$ and P_1 is determined by a particular AMM's function.

[4] In superposition, the weights of the tokens on the wallets change when swaps are executed, this will also make a difference in the impermanent loss. In addition, since each trade is executed only by a few wallets, the impermanent loss may differ slightly between different wallets.

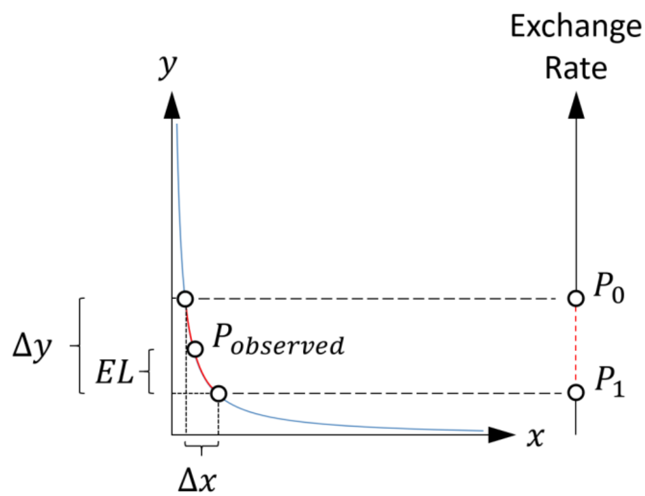

Figure 8. Excessive loss in CP-AMM vs external pricing

However, due to the nature of AMMs which obliged to execute trades on the entire price curve, when the pool rebalances to the current market price P_1, essentially a price rediscovery occurs, resulting in an average observed price lower than the known market price P_1.[5] *(Expression 11)*

$$P_{observed} \in (P_1;\ P_0) \tag{11}$$

This causes liquidity providers to incur excessive losses, which leads to increased impermanent loss when the trend is steady and high loss-vs-rebalancing (LVR) when volatility occurs. *(Expression 12)* [6] These losses are captured by sophisticated entities like PMMs. Furthermore, according to data from recent research, the income from fees for the majority of Uniswap pools is less than the associated LVR. [7]

$$excessiveLoss = \Delta x(1 - fee)(P_{observed} - P_1) \tag{12}$$

To remain profitable the fees collected by pools must be greater than the excessive losses, corresponding to *Expression 13*. Pool fees must be greater than the ratio of the difference of the observed execution price and the actual market price to the observed price.[6] *(Expression 14)* At the same time, the pool cannot use too high fees on liquid trading pairs, as this will lead to a significant reduction in trading volume, offsetting the gains from high fees.

$$\Delta x \cdot fee \cdot P_1 \geq excessiveLoss \tag{13}$$

$$fee \geq (P_{observed} - P_1)/P_{observed} \tag{14}$$

Various approaches exist to minimize and return excessive losses back to liquidity providers, such as function-maximizing AMM (FM-AMM), however, FM-AMM relies on internal token reserves in the pool to calculate the marginal price after the trade execution, which means a higher price impact than when the trade is executed at the actual market price in the case of PMMs. [8] Meanwhile, according to Flashbots' data, Wintermute

---

[5] The deviation between $P_{observed}$ and P_1 is determined by a particular AMM's function.

[6] In fact, since there are almost always trades, both buy and sell, the *charged fee* is slightly higher than the minimum necessary $\Delta x \cdot fee$ due to trading volume being higher than Δx, here we neglect it.

PMM executes a higher weekly volume of trades from aggregators than Uniswap, which indicates the favorable rate provided by PMMs for traders as well. [9]

Superposition provides liquidity providers, for the first time in DeFi, the capability to execute trades at the actual market price in the PMMs' fashion. This allows for the skipping of already irrelevant price segments, which undercuts the very basis of impermanent loss and LVR, at the same time without sacrificing executed trading volume and fees.[7] In this way, the user can choose pricing of existing PMMs or even CEXs. The Superposition Backend makes this possible because it can utilize any pricing mechanism, be it an AMM model or an external price.

5. Limit and TWAP orders in Superposition

Another important capability brought by Superposition is the new generation of limit orders. The Distributed Liquidity Layer allows users to place limit orders by holding tokens in their wallet, rather than locking them to a smart contract until the order is executed, as is currently implemented. Furthermore, the user doesn't need to cancel an old order - they can simply place a new one, and the Superposition Backend, itself, will cancel the old one.

In addition, Superposition enables TWAP orders, which allows the execution of orders at a time-weighted price, which can be particularly useful when executing large orders. Importantly, both TWAP and limit orders in Superposition are sources of liquidity for executing external swaps. When selecting the appropriate wallets to issue a quota, the Superposition Backend compares all available sources of liquidity, including AMM-like pools, PMM-like pools, limit orders and TWAP orders, selecting among them the wallets that offer the best execution price, while prioritizing limit and TWAP orders to ensure their execution as soon as possible.

When executing a limit or TWAP order, the Backend will first try to execute it according to the coincidence-of-wants principle, first of all internally, by matching it with the opposite direction order on the Superposition. Then trying to fill the order with an external swap from Barter, it is important to note that at each stage the order will be partially filled if full execution is not possible. In case a Limit or TWAP order cannot be filled by another order or external swap, Backend will try to execute it internally, using the available AMM-like and PMM-like “pools”. If after all of the above scenarios the order remains unfilled, the Backend Superposition will pass it to the Barter for execution at the best available price in the market. In this case, the trade will proceed in the opposite to the usual direction - from the Distributed Liquidity Layer to the Barter.

6. Asset Managers

In Superposition the users can select many various parameters while providing liquidity, such as direction-dependent fees and trading pairs, as well as token proportion ranges and curve type in Stable Swaps and AMM-like pools, respectively. Such unprecedented customization options for a pool settings open up a wide scope for custom and sophisticated liquidity provisioning strategies. However, while Superposition provides recommended settings for each trading pair, such an abundance of complex parameters may discourage the regular user. So here comes the opportunity to outsource the LP's position management to sophisticated entities.

[7] Assuming that over the long term, trades are equally distributed between the final and initial price, which is ensured by the profitability of the PMM's price to traders.

That opens a new paradigm in the evolution of solvers - previously solvers were focused only on traders, providing the best exchange rates, now solvers will also work on the opposite side, together with LPs. Seamless and gasless changes in curve type, fees or target tokens proportion allow novel solvers, called asset managers, to dynamically change the LPs' position settings, adjusting to fast changing market conditions. Furthermore, Superposition enables users to create their own curves with any amount of virtual liquidity used to calculate the exchange rate, resulting in more efficient pools and increased profitability for liquidity providers.

All of these benefits are available by performing off-chain calculations on the Superposition Backend. Asset managers will manage LPs' positions through the Superposition Backend, shielding users' funds from direct interactions with solvers. LPs will be able to select various asset managers to which they delegate the management of their positions based on multiple parameters such as the number of connected users, total trading volume of delegated pools, total pools' TVL and the yield of the pools. Essentially, for the end user, this would look like a fourth type of pricing, since at any given time the exchange rate on the wallet is determined by the solver.

References

[1] - Balancer.fi. Retrieved March 24, 2025 from https://docs.balancer.fi/concepts/vault/

[2] - Uniswap. Retrieved March 24, 2025 from https://docs.uniswap.org/contracts/v4/concepts/PoolManager

[3] - Fluid. Retrieved March 24, 2025 from https://fluid.guides.instadapp.io/liquidity-layer/unifying-liquidity

[4] - Internal research.

[5] - Michael Egorov. 2019. StableSwap - *Efficient Mechanism for Stablecoin Liquidity*.

Retrieved March 24, 2025 from https://docs.curve.fi/assets/pdf/whitepaper_stableswap.pdf

[6] - Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, Anthony Lee Zhang. 2022. *Automated Market Making and Loss-Versus-Rebalancing.* Retrieved March 24, 2025 from https://arxiv.org/abs/2208.06046

[7] - Robin Fritsch and Andrea Canidio. 2024. *Measuring Arbitrage Losses and Profitability of AMM Liquidity.* Retrieved March 24, 2025 from https://arxiv.org/abs/2404.05803

[8] - Andrea Canidio and Robin Fritsch. 2023. A*rbitrageurs' profits, LVR, and sandwich attacks: batch trading as an AMM design response.* Retrieved March 24, 2025 from https://arxiv.org/abs/2307.02074

[9] - Flashbots. Retrieved March 24, 2025 from https://orderflow.art/?isOrderflow=false

Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. This paper reflects current opinions of the authors and is not made on behalf of Barter or its affiliates and does not necessarily reflect the opinions of Barter, its affiliates or individuals associated with them. The opinions reflected herein are subject to change without being updated.