Target Weight Market Maker: A Solution To IL And LVR

Artamonov Artemii
badconfig@edu.misis.ru

Jul 6, 2025

Abstract

Constant Function Automated Market Makers (CF-AMMs) are a cornerstone of decentralized finance, yet they face inherent challenges such as Impermanent Loss (IL), Loss-Versus-Rebalancing (LVR), and significant price impact on large trades.

This paper introduces the Target-Weight Market Maker (TWMM), a novel system that utilizes external oracles to address these limitations. The TWMM reframes IL and LVR as active asset management challenges, implementing a sophisticated, deviation-based fee model that economically incentivizes the pool to maintain a target equilibrium.

Contents

1 Introduction

At the time of this paper's writing, on-chain liquidity provision predominantly relies on **Constant Function Automated Market Makers (CF-AMMs)**. The evolution of CF-AMMs led to the concept of **concentrated liquidity**, where liquidity is supplied only within a specific price range. This innovation significantly improves capital efficiency and offers enhanced tools for managing the risks associated with Impermanent Loss (IL) and Loss-Versus-Rebalancing (LVR), as comprehensively detailed in research from a16z and Columbia University[5].

Despite these advancements, market analysis indicates that even concentrated liquidity AMMs (CL-AMMs) face substantial risks and present considerable management complexity. Statistics from CrocSwap [1] reveal that only half of liquidity providers in the ETH/USDC trading pair—currently the highest volume pair—experience positive returns. Furthermore, the majority of Profit and Loss (PnL) outcomes are statistically clustered around zero, rendering liquidity provision a challenging and high-risk endeavor.

Alternative solutions are emerging to optimize LVR and IL. These include approaches that combine Request for Quote (RFQ) mechanisms with AMMs, such as those explored by Maverick Protocol [4], or methods that minimize harmful trade impacts through trade batching, as implemented by CoW Swap pools [2].

Another critical challenge in liquidity provision is the prevalence of paired on-chain liquidity, which restricts volume acquisition and forces Liquidity Providers (LPs) to select optimal trading pairs. Projects like Balancer [3] and Paradigm's innovative approach [6] (an advanced iteration of Curve Finance's 3-pool) address this by introducing multi-dimensional curves. These innovations enable the creation of pools that facilitate simultaneous trades across a basket of assets. Conceptually, they transition from a simple curve of $f(x, y) = L$ to a more generalized form of $f(\mathbf{X}) = L$, where $\mathbf{X} = [x_1, x_2, \ldots, x_i]$ represents a vector of i assets.

The **Target Weighted Market Maker (TWMM)** system introduces an innovative approach to on-chain market making, designed specifically to address these aforementioned challenges.

2 TWMM Description

2.1 Overview

Unlike traditional CF-AMMs, the TWMM reframes IL and LVR as asset management challenges, akin to active trading strategies.

The TWMM system integrates a weighted portfolio strategy with an on-chain liquidity pool. It facilitates the exchange of underlying assets using oracle-provided prices. These oracles are designed to be LVR-aware, with their efficiency directly influenced by price volatility and the speed of their technical implementation. A key distinction from CF-AMMs is that TWMM orders, whose prices are determined by oracles, do not incur a price impact correlated with order quantity, thereby mitigating the significant price impact observed in large trades on CF-AMMs.

TWMM introduces the concept of assets possessing **target shares (S_t)**, which are denominated in a quote asset's value. In an ideal scenario, the share of each asset within the pool would perpetually fluctuate around a predefined target. These shares continuously deviate from their targets due to two primary factors: collateral rebalancing initiated by liquidity usage (trades) or fluctuations in the market prices of the assets themselves. The mechanism that enables this share rebalancing is an innovative fee model, which incentivizes the real-time share of each asset to converge towards its predetermined target or "ethalon" value.

2.2 Fees

The fee model is centered on a deviation metric that quantifies the difference between an asset's real-time share and its target share. As previously discussed, the **current share** (S_c) of an asset can diverge from its target share due to market price fluctuations and trading activity. The current share is calculated by dividing the asset's total value by the pool's Total Value Locked (**TVL**):

$$S_c = \frac{R \cdot P}{Q_{lp} \cdot P_{lp}} \tag{1}$$

In this equation, the numerator $R \cdot P$ is the asset's total value (reserves multiplied by price). The denominator, $R_{lp} \cdot P_{lp}$, represents the TVL of the pool, which is calculated by multiplying the total supply of LP tokens (their reserve R_{lp}) by the price of the LP token (P_{lp}).

2.3 Deviation: The Core Metric

Deviation is the fundamental metric that quantifies the difference between an asset's current share and its target share. For a specific asset , its **deviation value** (D) indicates the degree to which the asset is imbalanced relative to its target value within the TWMM pool. It is calculated as:

$$D = S_c - S_t \tag{2}$$

Here, S_c and S_t are the current and target shares for asset, respectively. A negative deviation value ($D < 0$) indicates a deficit of the asset in the pool, while a positive value ($D > 0$) signifies a surplus.

With n representing the total number of assets, the deviation values adhere to two primary invariants:

$$\sum_{i=1}^{n} D_i = 0, \qquad \sum_{i=1}^{n} |D_i| = 2 \cdot I \tag{3}$$

The sum of all deviation values consistently equals zero. Concurrently, the sum of their absolute values defines the total imbalance of the pool, which is equal to twice the pool's aggregate imbalance rate (I).

This quantitative understanding of deviation is crucial for the TWMM's rebalancing mechanisms and fee model, which are elaborated upon in subsequent sections. The precise management of these deviations ensures the stability and efficiency of the system.

2.4 Fee Structure

Deviation plays a vital role in the fee structure of the TWMM. A fee is levied in the specific asset whose pool quantity is altered by an operation. These fees can be broken down into several components:

- **Base Fee (F_b)**: A constant fee that is charged unconditionally on every operation.
- **Depeg Fee (F_{dv})**: This dynamic fee serves the dual purpose of generating revenue and creating an economic incentive to maintain the pool's target allocations. It discourages operations that would increase an asset's imbalance by making them economically less appealing. The fee is a function of the asset's deviation before the operation (D_o) and after the operation (D_n). It is applied only when the absolute deviation of the asset increases as a result of the operation (i.e., $|D_n| > |D_o|$). Conversely, if an operation causes the absolute deviation to decrease, this fee is waived.
- **Cashbacks (F_{cb})**: Functioning as the inverse of the depeg fee, cashbacks are rebates that reward users for operations that reduce an asset's imbalance. A cashback is issued when a transaction moves an asset's current share closer to its target share (i.e., when $|D_{new}| < |D_{old}|$ and $\text{sgn}(D_{old}) = \text{sgn}(D_{new})$).

Synthesizing these individual components, the total effective fee (F_{total}) for any given operation is defined by the following general formula, which combines the base fee (F_b), the deviation fee (F_{dv}), and the cashback (F_{cb}):

$$F_{total} = F_b + F_{dv} - F_{cb} \tag{4}$$

The core concept is to calibrate fees based on the resulting change in an asset's deviation following a transaction. It is important to note that the **TWMM architecture supports various custom functions for calculating the deviation fee**, enabling flexible and adaptable fee structures. Examples of such functions include **linear** or **constant** models, in addition to more complex **curve-based** approaches.

2.5 Fee Distribution

Fees collected by the TWMM, excluding funds allocated for cashbacks, are distributed among three distinct parties, with configurable percentage shares for each:

- LP token holders (liquidity providers)
- The pool owner or manager
- The protocol treasury

The distribution method varies by recipient. The share allocated to liquidity providers remains in the pool, increasing the quantity of the underlying assets and thereby causing the value of their LP tokens to appreciate.

Conversely, the shares designated for the pool owner and the protocol are distributed by minting new LP tokens. If the total quote value of the fees collected for the owner and protocol is V_{col}, the quantity of new LP tokens to be minted (Q_{lpfee}) is determined by the current LP token price (P_{lp}):

$$Q_{lpfee} = \frac{V_{col}}{P_{lp}} \tag{5}$$

This minting mechanism ensures that fee distributions to the owner and protocol do not directly withdraw assets from the pool. However, it is crucial to recognize that since fees are collected as assets within the pool, their collection inherently alters the pool's asset quantities. Consequently, the impact of both collected fees and awarded cashbacks must be accounted for when calculating the final deviation after a transaction.

2.6 Deviation Limit

The **deviation limit** is a control mechanism designed to restrict large-volume swaps that could cause a severe imbalance in the pool's asset composition.

An asset's deviation can exceed this specified limit passively as a result of market price fluctuations. When this occurs, transactions that reduce the asset's deviation are permitted to proceed normally, while any transaction that would further increase its absolute deviation is restricted.

Notably, a deviation fee may be waived even for trades that cause an asset's deviation to change sign (e.g., from a surplus to a deficit). The condition for this waiver is that the final absolute deviation must be less than the initial absolute deviation. Such transactions, even if large, are considered beneficial as they decrease the pool's overall imbalance and move the portfolio closer to its target equilibrium state ($I_{new} < I_{old}$) as described in deviation invariant 3.

2.7 Deviation Fee

To model the deviation fee, we first define two auxiliary components: the initial deviation before a transaction and the change in deviation resulting from it.

First, the initial deviation (D_{old}) represents the asset's deviation state immediately before a swap. Consistent with Equation 2, it is defined using the asset's reserve (R) in TWMM pool, its price (P), its target share (S_t), and the pool's Total Value Locked ($R_{lp} \cdot P_{lp}$):

$$D_{old} = S_t - \frac{R \cdot P}{R_{lp} \cdot P_{lp}} \tag{6}$$

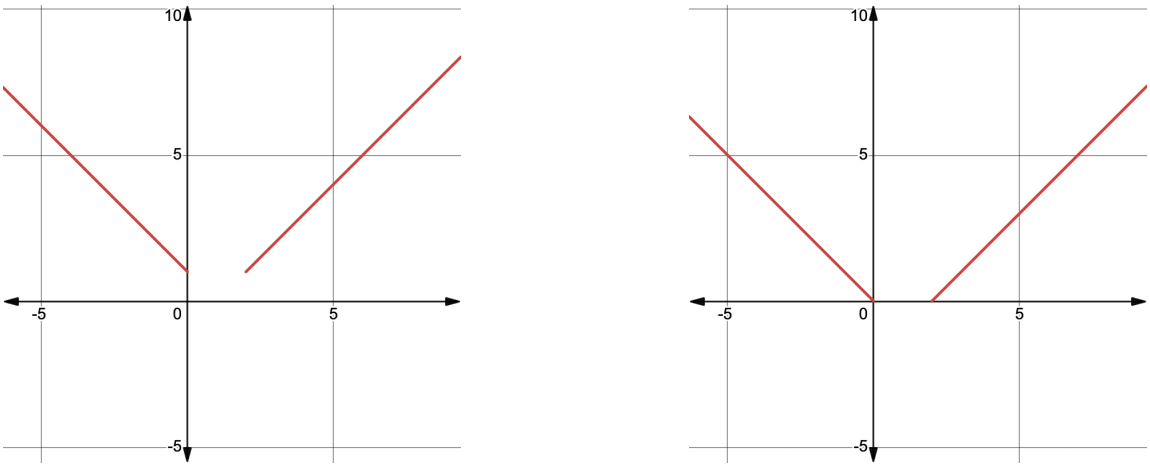
Second, the change in deviation, ΔD, is a function of the change in the asset's quantity, ΔQ, resulting from the swap. It is calculated as:

$$\Delta D(\Delta Q) = \frac{\Delta Q \cdot P}{Q_{lp} \cdot P_{lp}} \tag{7}$$

The deviation fee, F_{dv}, is modeled as a piecewise linear function. It applies a fee only if a transaction increases the absolute deviation of the asset. The general form, parameterized by a multiplier K and a constant B, is:

$$F_{dv}(\Delta D) = \begin{cases} K \cdot |D_{old} + \Delta D| + B, & \text{if } |D_{old} + \Delta D| > |D_{old}|, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The illustration of this function is:



(a) Deviation function with $B = 0$

(b) Deviation function with $B = -K|D_{old}|$

Figure 1: Linear deviation functions

To refine the fee model, we first adjust the linear function from Equation 8. The previous form results in a fee greater than B as soon as the deviation increases. To ensure the fee is exactly B at the threshold where deviation begins to increase, and grows proportionally to the *increase* in absolute deviation, the function is recalibrated as follows:

$$F_{dv}(\Delta D) = \begin{cases} K \cdot (|D_{old} + \Delta D| - |D_{old}|) + B, & \text{if } |D_{old} + \Delta D| > |D_{old}|, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

A further refinement is necessary to account for the recursive impact of the fee itself. The collected fee alters the final input or output amount, which in turn influences the deviation value used to calculate the fee. This self-referential dependency requires an implicitly defined function. We introduce a parameter, deviation fee to cashback ratio (r_d), which is related to the fee's allocation toward the cashback fund.

For an asset being added to the pool, the implicit fee function, F_{dIn}, which accounts for the portion of the fee retained by the pool, is:

$$F_{dIn}(\Delta D_{in}) = K \cdot (|D_{oIn} + \Delta D_{in} \cdot [1 - r_d \cdot F_{dIn}(\Delta D_{in})]| - |D_{oIn}|) + B \tag{10}$$

Here, the gross change in deviation, ΔD_{in}, is adjusted by a factor dependent on the final fee itself.

Similarly, for an asset being withdrawn from the pool (the "out" asset), a corresponding implicit function, F_{dOut}, is required to model the fee's effect on the final withdrawal amount:

$$F_{dOut}(\Delta D_{out}) = K \cdot (|D_{oOut} + \Delta D_{out} \cdot [1 + r_d \cdot F_{dOut}(\Delta D_{out})]| - |D_{oOut}|) + B \tag{11}$$

2.8 Cashbacks

Cashbacks are financial incentives provided to users who execute trades that shift an underlying asset's share within the **TWMM** closer to its designated target share.

The cashback mechanism is funded by the deviation fees collected from prior operations that increased the pool's imbalance. These accumulated funds are then distributed to users whose transactions reduce an asset's deviation. The magnitude of the distributed cashback is directly proportional to the extent of the deviation reduction. For instance, if a transaction reduces an asset's d The total accumulated fund available for cashback distribution for a given asset is denoted by C. The specific cashback amount, F_{cb}, is calculated using the absolute deviation before the operation, $|D_o|$, and after the operation, $|D_n|$, as described by the formula:

$$F_{cb} = \frac{(|D_o| - |D_n|) \cdot C}{|D_o| \cdot \Delta Q} \tag{12}$$

A key design consideration is that this calculation is non-recursive. While the awarded cashback could theoretically be redeposited and create a secondary impact on deviation, calculating this effect recursively is avoided as it could create unstable feedback loops that overdistribute and overincentivise trades. For this reason, the secondary impact is not considered.

Furthermore, to ensure stability and efficient value distribution, the cashback award is capped by maximum cashback (C_{max}) - a maximum ratio relative to the size of the user's input quantity, ΔQ. General cashback function derived from 12 looks like:

$$F_{\text{cb}}(\Delta Q) = \begin{cases} \min\Big(\dfrac{|D_{old}| - \big|D_{old} + \Delta D\big|}{|D_{old}|} \cdot \dfrac{C}{\Delta Q}, C_{max}\Big), & \text{if } |D_{old} + \Delta D| < |D_{old}| \\ & \text{and } \operatorname{sgn}(D_{old}) = \operatorname{sgn}(D_{new}), \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The calculation of the cashback rate, denoted as F_{cb}, requires careful constraint to ensure it properly incentivizes balancing the pool without creating instabilities. The models differ slightly for assets being deposited versus assets being withdrawn.

Cashback for Input Asset ($\mathbf{F_{cIn}}$)

For an input asset ($\Delta Q_{in} > 0$) that is reducing a deficit ($D_{oIn} < 0$), the cashback must be constrained to prevent the final deviation from overshooting past zero. To prevent the cashback from causing the final deviation to cross zero, we introduce a second constraint, C_z. This value represents the maximum cashback rate that results in a final deviation of exactly zero.

$$D_{oIn} + \Delta D_{in} \cdot (1 + C_z) = 0 \iff C_z = -\left(\frac{D_{oIn}}{\Delta D_{in}} + 1\right) \tag{14}$$

$$F_{cIn}(\Delta Q_{in}) = \min\left(\frac{|D_{oIn}| - |D_{oIn} + \Delta D_{in}|}{|D_{oIn}|} \cdot \frac{C_{in}}{\Delta Q_{in}}, C_{max}, C_z\right) \tag{15}$$

Under the specific condition that the transaction reduces the deficit but does not eliminate it (i.e., $D_{oIn} + \Delta D_{in} < 0$), the formula can be simplified:

$$F_{cIn}(\Delta Q_{in}) = \min\left(-\frac{\Delta D_{in}}{D_{oIn}} \cdot \frac{C_{in}}{\Delta Q_{in}}, C_{max}, C_z\right) \tag{16}$$

Combining these constraints yields the final piecewise function for the input asset cashback rate. The rate is the minimum of the calculated proportional rate, the global maximum rate (C_{max}), and the zero-crossing rate (C_z). A cashback is only awarded if the transaction does not push the deviation into a surplus.

$$F_{cIn}(\Delta Q_{in}) = \begin{cases} \min\left(-\frac{\Delta D_{in}}{D_{oIn}} \cdot \frac{C_{in}}{\Delta Q_{in}}, C_{max}, -\left(\frac{D_{oIn}}{\Delta D_{in}} + 1\right)\right), & \text{if } D_{oIn} + \Delta D_{in} < 0, \\ 0, & \text{otherwhise.} \end{cases} \tag{17}$$

Cashback for Output Asset ($\mathbf{F_{cOut}}$)

The logic for an output asset that reduces a surplus ($D_{oOut} > 0$) is symmetrical. The cashback is drawn from the collected reserve C_{out} and does not create secondary deviation effects. The general and simplified formulas are analogous to the input asset case:

$$F_{cOut}(\Delta Q_{out}) = \min\left(\frac{|D_{oOut}| - |D_{oOut} - \Delta D_{out}|}{|D_{oOut}|} \cdot \frac{C_{out}}{|\Delta Q_{out}|}, C_{max}\right) \tag{18}$$

Assuming the transaction reduces but does not eliminate the surplus ($D_{oOut} - \Delta D_{out} > 0$), this simplifies to:

$$F_{cOut}(\Delta Q_{out}) = \min\left(\frac{\Delta D_{out}}{D_{oOut}} \cdot \frac{C_{out}}{|\Delta Q_{out}|}, C_{max}\right) \tag{19}$$

The final function for the output asset cashback rate is therefore given as:

$$F_{cOut}(\Delta Q_{out}) = \begin{cases} \min\left(\frac{\Delta D_{out}}{D_{oOut}} \cdot \frac{C_{out}}{|\Delta Q_{out}|}, C_{max}\right), & \text{if } D_{oOut} - \Delta D_{out} > 0, \\ 0, & \text{otherwhise.} \end{cases} \tag{20}$$

2.9 Full TWMM swap invariant

The swap invariant defines the fundamental relationship between the amount of an asset deposited into the pool (ΔQ_{in}) and the amount of another asset withdrawn (ΔQ_{out}). This relationship incorporates the asset prices (P_{in}, P_{out}), the respective base fees (F_{bIn}, F_{bOut}), and the net directional fees (F_{in}, F_{out}), which represent the combined effect of deviation fees and cashbacks. The invariant is expressed as:

$$\Delta Q_{in} \cdot P_{in} \cdot (1 - F_{bIn} - F_{in}(\Delta Q_{in})) = \Delta Q_{out} \cdot P_{out} \cdot (1 + F_{bOut} + F_{out}(\Delta Q_{out})) \tag{21}$$

Because the base fee (F_{base}) is retained in the form of the asset being transacted and thus affects the pool's deviation, its collection can be split between the input and output assets. We define these user-configurable portions as F_{bIn} and F_{bOut}, which must adhere to the constraint:

$$F_{base} = F_{bIn} + F_{bOut} \tag{22}$$

The net directional fee terms, F_{in} and F_{out}, are composite values. They are derived by combining the previously defined deviation fee functions (e.g., Equation 10) and cashback functions (e.g., Equation 17). Merging these components yields the final, comprehensive fee function used in the swap invariant for the input asset:

$$F_{\text{in}}(\Delta Q_{\text{in}}) = \begin{cases} -\min\Big(\dfrac{\Delta D_{\text{in}}}{D_{o\text{In}}} \cdot \dfrac{C_{\text{in}}}{\Delta Q_{\text{in}}},\ C_m,\ -\Big(\dfrac{D_{o\text{In}}}{\Delta D_{\text{in}}} + 1\Big)\Big), & \text{if } D_{o\text{In}} + \Delta D_{\text{in}} < 0, \\ K\Big(\big|\, D_{o\text{In}} + \Delta D_{\text{in}}\big[1 - r_d\, F_{\text{In}}(\Delta D_{\text{in}})\big]\,\big| & \\ \qquad - |D_{o\text{In}}|\Big) + B, & \text{if } D_{o\text{In}} + \Delta D_{\text{in}} > |D_{o\text{In}}|, \\ 0, & \text{otherwhise.} \end{cases} \tag{23}$$

Following the same procedure, the net directional fee for the output asset, F_{out}, is derived by combining the cashback and deviation fee functions applicable to asset withdrawals:

$$F_{\text{out}}(\Delta Q_{\text{out}}) = \begin{cases} -\min\Big(\dfrac{\Delta D_{\text{out}}}{D_{o\text{Out}}} \cdot \dfrac{C_{\text{out}}}{\Delta Q_{\text{out}}},\ C_{\max}\Big), & \text{if } D_{o\text{Out}} - \Delta D_{\text{out}} > 0, \\ \Big(\big|\, D_{o\text{Out}} + \Delta D_{\text{out}}\big[1 + r_d\, F_{d\text{Out}}(\Delta D_{\text{out}})\big]\,\big| & \\ \qquad - |D_{o\text{Out}}|\Big) \cdot K + B, & \text{if } D_{o\text{Out}} - \Delta D_{\text{out}} < -|D_{o\text{Out}}|, \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

To execute a swap, a user must identify a pair of quantities, $(\Delta Q_{in}, \Delta Q_{out})$, that satisfies the swap invariant defined in Equation 21. In practice, finding the optimal trade that maximizes the output value for a given input may require iteratively solving this equation, a process that must account for potential fluctuations in market prices during transaction construction.

2.10 Liquidity Provision and Removal

Liquidity provision and removal are the operations for depositing assets into the pool to mint LP tokens, or burning LP tokens to withdraw underlying assets. Unlike swaps which exchange one pool asset for another, these operations always involve a transaction between one or more pool assets and the pool's own LP tokens. This allows to apply less deviation impact for big amount of liqudiity provision or removal

Because these operations alter the total supply of LP tokens (Q_{lp}), the denominator of the share calculation (i.e., the Total Value Locked) changes during the transaction itself. This requires a redefinition of our deviation formulas (the deviation old formula (6) and deviation delta formula (7)). For a liquidity operation involving a quantity ΔQ of a single asset and ΔQ_{lp} of LP tokens, we define the initial deviation and the change in deviation as follows:

$$D_{old}(\Delta Q, \Delta Q_{lp}) = S_t - \frac{R \cdot P}{R_{lp} \cdot P_{lp}} \tag{25}$$

$$\Delta D(\Delta Q, \Delta Q_{lp}) = \frac{\Delta Q \cdot P}{\Delta Q_{lp} \cdot P_{lp}} \tag{26}$$

It is important to note that the LP token itself is not subject to deviation fees or cashbacks. As a derivative representing a share of the entire pool, it is not considered a constituent asset with a target share.

2.11 Multi-Asset Liquidity Invariant

While liquidity can be provided with a single asset, this action inherently unbalances the pool's composition, leading to a significant deviation impact and potentially high fees for the liquidity provider. To mitigate this, the TWMM is designed to support multi-asset liquidity provision and removal. By depositing or withdrawing assets in proportions that align with the pool's target shares, users can add or remove liquidity with minimal deviation impact and associated costs.

The relationship between the assets being transacted and the LP tokens is governed by the following invariants. For minting LP tokens by depositing n assets, the invariant is:

$$\sum_{i=1}^{n}(1 - F_{base} - F_{in}(\Delta Q_{in}, \Delta Q_{lp}))\Delta Q_{in} P_{in} = \Delta Q_{lp} P_{lp} \tag{27}$$

This equation states that the value of the minted LP tokens equals the sum of the net values of all deposited assets after their respective fees are deducted.

Symmetrically, the invariant for burning LP tokens to withdraw n assets is:

$$\Delta Q_{lp} P_{lp} = \sum_{o=1}^{n}(1 + F_{base} + F_{out}(\Delta Q_{out}, \Delta Q_{lp}))\Delta Q_{out} P_{out} \tag{28}$$

These liquidity operations can be integrated into a generalized invariant that holistically describes all possible interactions with the pool.

3 Rebalancing and Maintenance

The TWMM manages its assets by defining target shares for its constituent assets and facilitating market-driven trades. These target shares can be adjusted dynamically or remain static over long periods. However, the actual shares of assets within the pool inevitably fluctuate. This fluctuation is driven by two primary factors: changes in the market prices of the assets and changes in their quantities resulting from trading activity.

The system also supports the active maintenance of the asset portfolio, including the addition or removal of assets as needed.

3.1 Managing Asset weights

To add a new asset to the pool, a non-zero target share is assigned to it. This action necessitates a proportional reduction in the target shares of existing assets to maintain a total portfolio weight of 100%. This immediately creates a significant negative deviation (a shortage) for the new asset, especially if its target share is large. This state incentivizes users to supply the new asset to the pool to reduce its deviation. Conversely, the deviation limit mechanism restricts withdrawals or swaps that would further increase the asset's shortage.

Conversely, removing an asset is achieved by setting its target share to zero. This action creates a large positive deviation (a surplus) for that asset, incentivizing users to withdraw it from the pool. Due to the deviation mechanism, the asset can effectively only be withdrawn. Any attempt to deposit more of the asset would be blocked by the deviation limit, as it would exacerbate the surplus.

3.2 Oracle Price Risk Mitigation

Oracle prices are subject to latency and potential inaccuracies, creating a discrepancy between the on-chain price and the true market price. We can term this discrepancy the "oracle error." If this error reaches a sufficient magnitude, it can create an arbitrage opportunity, allowing traders to extract value from the pool.

The TWMM employs a multi-layered defense to mitigate this risk:

1. **Base Fee:** The base fee (F_{base}) is the primary defense. By applying a fee to every trade, the system establishes a profitability threshold for arbitragers. For an arbitrage to be profitable, the oracle error must exceed the base fee.

2. **Secondary Defenses:** In scenarios where the oracle error does exceed the base fee, two additional mechanisms protect the pool from significant capital drain:

    - **The Deviation Limit:** This mechanism inherently caps the total amount of an asset that can be swapped in a single direction, limiting the total value that can be extracted via arbitrage.
    - **Deviation Fees:** If the arbitrage trade increases the pool's imbalance, progressively higher deviation fees are applied, rapidly diminishing the profitability of the exploit.

These factors work in concert to significantly reduce the net profit of such arbitrage, especially for large-scale exploits. This defense is most effective when the pool is near its target equilibrium. While a large pre-existing deviation could alter the fee dynamics, the cashback system is designed

to incentivize continuous rebalancing, keeping the pool close to its target shares and thus ensuring these protective mechanisms remain effective.

Ultimately, the most robust solution to oracle risk is to minimize the error in the first place. Employing advanced techniques, such as MEV-aware oracle designs that provide the most recent price atomically within a trade, is the preferred long-term strategy.

4 TWMM efficiency

4.1 Comparing with CF-AMM

For a CF-AMM with the invariant function $x \cdot y = L$, the amount of output asset Δy received for a given input amount Δx is calculated based on the initial reserves of the assets.

The formula, which accounts for a standard percentage trading fee, is:

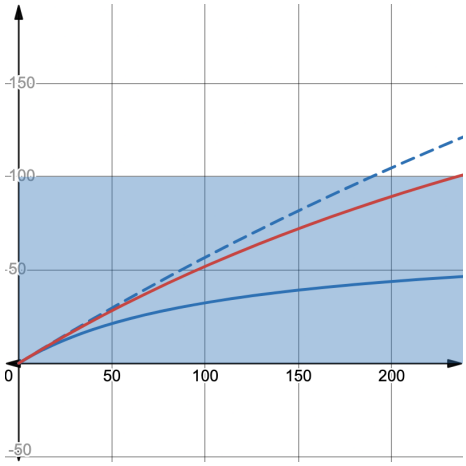
$$\Delta y = \frac{\Delta x \cdot 997 \cdot R_{in} \cdot m \cdot \dfrac{P_{in}}{P_{out}}}{R_{in} \cdot m \cdot 1000 + \Delta x \ \cdot 997 \cdot \sqrt{\dfrac{P_{in}}{P_{out}}}} \tag{29}$$

To provide a clear performance benchmark, we will utilize a standard CF-AMM formula. This formula calculates the output asset amount for a given input amount, including a 0.3% fee charged on the input. To facilitate a direct comparison, the CF-AMM model is parameterized using prices (P_{in}, P_{out}) and a liquidity level derived from the input asset's reserve (R_{in}), mirroring the variables used in the TWMM's swap invariant.

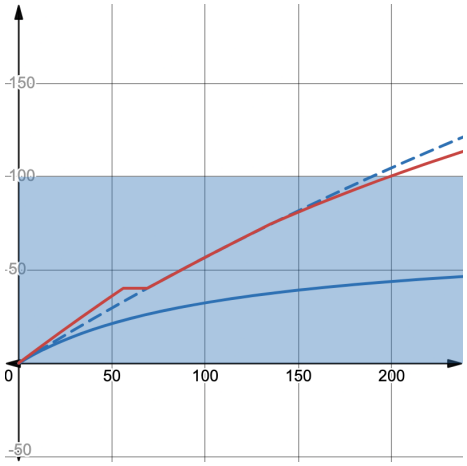
We will now compare the trade execution of this CF-AMM against the TWMM across several key scenarios. These include conditions with varying cashback reserves, near-zero deviation, and significant initial deviation.

In the following charts, the performance is illustrated as follows:

- The **red line** represents the **TWMM**.
- The solid **blue line** represents the standard **CF-AMM** ($m = 1$).
- The dashed **blue line** represents a **CF-AMM with tenfold liquidity**, providing a benchmark for capital efficiency ($m = 10$).
- The **Blue Zone** highlights a fundamental constraint of the TWMM: the output amount cannot exceed the total available reserve of the output asset ($\Delta Q_{out} \leq R_{out}$).

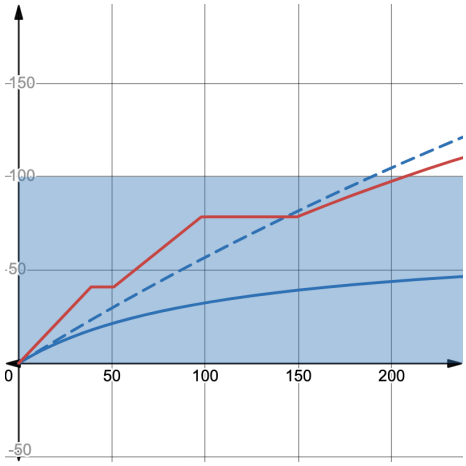


(a) Deviation hight on both assets

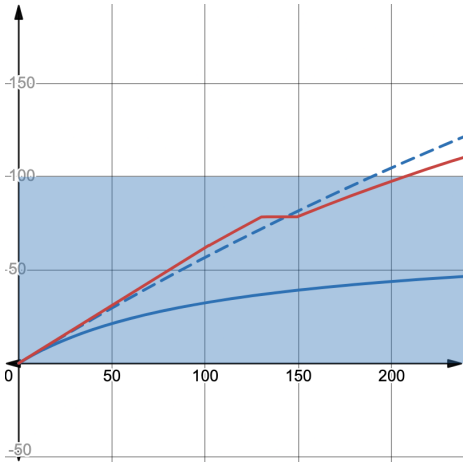


(b) Cashback on one asset

Figure 2: Comparison in deviation cases



(a) Cashback on both assets



(b) No cashbacks

Figure 3: Comparison in non deviation cases

The analysis demonstrates that the TWMM offers superior capital efficiency across a wide range of trading scenarios. Excluding cases of extreme deviation—where the pool holds a severe surplus or deficit of an asset—the TWMM's performance is comparable to that of a standard CF-AMM with as much as tenfold its TVL. This finding and ability to put multiple trading pairs into one TWMM pool indicates a profound improvement in capital utilization, allowing the TWMM to facilitate large trades with significantly less liquidity than traditional models.

4.2 Impermanent Loss Mitigation

In the context of CF-AMMs, impermanent loss represents the opportunity cost a liquidity provider incurs compared to simply holding the assets in their wallet. While the TWMM does not eliminate this risk, it provides powerful tools to actively manage and mitigate it by reframing liquidity provision as a form of active portfolio management.

The key to this mitigation lies in the ability to dynamically adjust the target weights of the assets within the pool. This flexibility enables several strategic possibilities that are unavailable in static CF-AMMs:

- **Fee Optimization:** A pool manager can increase the target share of assets anticipated to have high trading volume. This strategy aims to maximize fee generation, which can serve as a direct offset against potential impermanent loss.
- **Exposure and Risk Management:** Conversely, a manager can reduce the target share of a highly volatile asset. This allows the pool to maintain exposure to the asset's potential upside while minimizing the magnitude of impermanent loss caused by its price fluctuations.

Consequently, the TWMM LP token transcends its role as a simple claim on pooled assets. It becomes a tokenized representation of a managed financial strategy. While this strategy may not outperform every individual asset in the portfolio, it can be calibrated to offer a superior risk-adjusted return profile, effectively serving as a more stable store of value compared to a passive holding strategy.

Conclusion

The TWMM is a powerful system that successfully challenges current issues with on-chain decentralized liquidity provision. The problem of impermanent loss is addressed through the ability to actively manage asset shares, causing them to rebalance. This creates a new type of on-chain derivative that represents a portfolio strategy.

The deviation fee model provides strong incentives to keep asset shares aligned with their target values. Because the TWMM supports multiple assets in one pool, it also increases trading volume capture from liquidity provision, which can be further optimized by adjusting asset weights to provide deeper liquidity where needed.

The LVR problem is addressed as the TWMM uses frequently updated oracle prices, and the pool shows significant advantages in capital efficiency over CF-AMMs.

The model has a potential bottleneck caused by oracle price inefficiencies. If oracle prices have errors that overcome the collected fee value, an opportunity to drain the pool by extracting arbitrage profit can appear (same mechanic as LVR). This, however, is more complicated if the pool is nearly balanced, as additional volume would also involve deviation fees that grow proportionally with the trade size and would counteract the profitability of exploiting the oracle error. Therefore, effective oracles are key to the profitability of the TWMM.

References

[1] 0xfbifemboy. Impermanent loss and jit liquidity in the uniswap eth/usdc 0.3% pool. Medium, 2023. Accessed: 2025-07-05.

[2] Andrea Canidio and Robin Fritsch. Arbitrageurs' profits, lvr, and sandwich attacks: batch trading as an amm design response. *arXiv preprint*, 2025.

[3] Fernando Martinelli and Nikolai Mushegian. Balancer whitepaper, 2019.

[4] Maverick Protocol. Dynamic distribution amm: Technical introduction. Technical report, Maverick Protocol, 2023.

[5] Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing. arXiv:2208.06046v5 [q-fin.MF], 2024.

[6] Dave White, Dan Robinson, and Ciamac Moallemi. Orbital: An automated market maker for multi-asset stablecoin pools. Whitepaper, Paradigm, Jun 2025. Accessed: 2025-07-05.