Financial Innovation

RESEARCH Open Access

Predictive crypto-asset automated market maker architecture for decentralized finance using deep reinforcement learning

Tristan Lim[1]*

*Correspondence:
tristan_lim@nyp.edu.sg;
tris02@gmail.com

[1] School of Business Management, Nanyang Polytechnic, C.317, 180 Ang Mo Kio Ave 8, Singapore 569830, Singapore

Abstract

This study proposes a quote-driven predictive automated market maker (AMM) platform with on-chain custody and settlement functions, alongside off-chain predictive reinforcement learning capabilities, to improve the liquidity provision of real-world AMMs. The proposed architecture augments Uniswap V3, a cryptocurrency AMM protocol, by using a novel market equilibrium pricing to reduce divergence and slippage losses. Furthermore, the proposed architecture involves a predictive AMM capability, for which a deep hybrid long short-term memory (LSTM) and Q-learning reinforcement learning framework is used. It seeks to improve market efficiency through obtaining more accurate forecasts of liquidity concentration ranges, where liquidity starts moving to expected concentration ranges prior to asset price movement; thus, liquidity utilization is improved. The augmented protocol framework is expected to have practical real-world implications through (1) reducing divergence loss for liquidity providers; (2) reducing slippage for crypto-asset traders; and (3) improving capital efficiency for liquidity provision for the AMM protocol. The proposed architecture is empirically benchmarked against the well-established Uniswap V3 AMM architecture. The preliminary findings indicate that the novel AMM framework offers enhanced capital efficiency, reduced divergence loss, and diminished slippage, which could potentially address several of the challenges inherent to AMMs.

Keywords: Predictive automated market maker architecture, Decentralized finance, Deep reinforcement learning, Divergence (or impermanent loss) and slippage losses, Capital efficiency, Liquidity utilization, concentration and depth

Introduction

The introduction of smart contracts, backed by public blockchains such as Ethereum, has allowed an entire financial system to be created in which different parties can operate under shared data and assumptions without trust issues arising from institutional intervention. This is known as decentralized finance (DeFi).

Decentralized exchanges (DEXs) are a crucial element of the DeFi market structure. Recent studies have revealed a significant shift in the landscape of crypto trading with the advent of such exchanges, which emphasizes their growing importance in the DeFi ecosystem (Ghosh et al. 2023). Before DEXs, crypto-assets were generally traded in off-chain, centralized settlement infrastructures called centralized exchanges (CEXs), which act as trusted third parties. Examples include Binance and Bitfinex. While CEXs offer an easy-to-understand order book format execution similar to conventional financial market exchanges, they have some drawbacks, such as server downtime, uncertain fair execution, slow withdrawals, and traders being wholly dependent on trust with the exchange on their custody of assets. Over time, some semi-custodial exchanges have sought to move partial functionality on-chain, such as EtherDelta and IDEX, which deploy an on-chain custody and settlement solution with an off-chain order book and trading engine. While CEXs' original intent is to ensure improved performance, their downsides persist.

A new class of quote-driven crypto-asset trade execution systems has now been developed, which are called automated market makers (AMMs). They only require data structures and traversals and have low gas complexity (Moosavi and Clark 2021). Furthermore, AMMs allow multiple parties to interact directly in a nonrivalrous and programmatic manner with smart contracts of the DEX protocol; thus, trade is executed automatically using a hard-coded pricing function (or a bonding curve), while the matching of individual buy and sell orders is not required. Lehar and Parlour (2021) evidence the uptake of liquidity-sharing AMM protocols and empirically demonstrate that AMMs can provide liquidity more efficiently than CEXs. According to Mohan (2022), AMMs are reshaping the DeFi landscape by offering more efficient and trustless trading solutions.

As the DeFi space has rapidly evolved, various AMM protocols have emerged, each with its unique features and challenges. In terms of monthly trading volume, in September 2023, Uniswap led the AMM market by far with 19.4 billion trades, outstripping the next three highest AMM protocols' (PancakeSwap, DODO, and Curve) 6.2, 2.5, and 2.4 billion trades, respectively (Fig. 1). At its peak, Uniswap accounted for 86 billion trades. Other popular protocols include SushiSwap, Balancer, and QuickSwap.

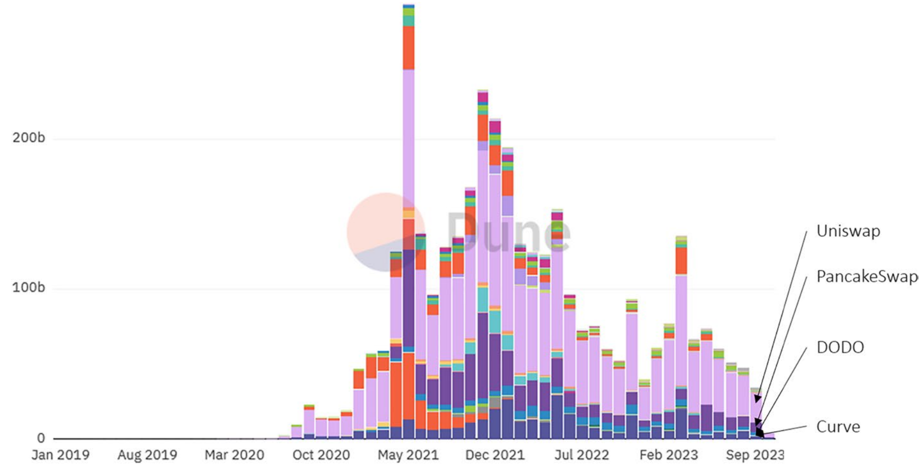


Fig. 1 Monthly DEX transaction volume by project (Dune Analytics 2023). This figure highlights Uniswap as the leading AMM market, followed by PancakeSwap, DODO and Curve, in the month of September 2023

Most key AMMs on Ethereum-based protocols implement a constant function market maker (CFMM) for executing trades (Uniswap 2022; Curve 2022; Balancer 2022; SushiSwap 2022). CFMMs are AMMs that employ a fixed bonding curve for asset price determination and liquidity provision. Angeris and Chitra (2020) demonstrate that agents who interact with CFMMs are incentivized to price assets correctly in a computationally efficient manner.

In this study, we focus on Uniswap, the most used protocol. Uniswap has two actively traded versions, namely V2 and V3. Uniswap implements the XYC constant product market maker (CPMM) function, where given x units of token X and y units of token Y, the liquidity of pool K is the product of $x \cdot y = c$. Upon choosing a pool to provide liquidity, V2 allows a liquidity provider to supply liquidity across the entire price range, whereas V3 applies a novel CPMM design that allows liquidity providers to specify the price range at which they wish to supply liquidity. Since its introduction, V3 has overtaken V2 to become the AMM with the largest trading volume. However, despite the high trading volumes, issues persist for both the liquidity pool and market participants in V3.

In addressing the challenges in Uniswap V3, our problem statement centers on the critical issues faced by AMMs—namely capital inefficiency, significant slippage, and divergence loss. An imperative demand exists for an innovative AMM design that not only addresses these challenges but also sets a precedent for the next generation of AMM platforms. The following paragraphs introduce the three key terms liquidity pool, liquidity taker, and liquidity provider:

- Liquidity pool:
  Capital efficiency is a function of the amount of capital required to provide for efficient market making. The less capital required to make the market, the more efficient the liquidity provision. This also implies that total value locked is not a useful metric for measuring the liquidity productiveness of a liquidity pool.
- Liquidity taker:
  A liquidity taker is any party that exchanges assets by taking liquidity supplied by liquidity makers from the market. Liquidity takers expect the market to reflect the true price of assets, a low price change during the execution (or slippage) of trade, and the capacity to exchange assets on demand. In AMM protocols, trade is executed through liquidity pools for each pair of tradable tokens, which are reserved in their respective smart contracts. A trader who seeks to exchange X tokens for Y tokens can deposit X tokens in the liquidity pool and receive Y tokens in an atomic swap, such that the aggregate liquidity of the pool remains unchanged, as defined by the bonding curve (Park 2022).
  Slippage is an implicit cost to a liquidity provider that occurs when the price at which a trade is executed differs from the expected price of the trade. Slippage can occur when the market is volatile or when the sizes of the trades are large relative to the size of the liquidity pool. While slippage may not be entirely eliminated, it is to the benefit of liquidity takers to reduce this market inefficiency to the lowest possible level.

- Liquidity provider:

  A liquidity provider is any party who contributes liquidity to the market. They create an efficient market in which liquidity takers can trade assets. Liquidity providers commit pairs of X and Y crypto-assets to the pool such that liquidity exists for traders to buy and/or sell X and Y crypto-assets. Liquidity providers are incentivized through market making incentive fees from the trades supported by their liquidity.

  Enabling liquidity providers to select specific price ranges for supplying liquidity alters the risk–return profile of their investments. Providers who strategically choose optimal price positions and widths to concentrate their liquidity can significantly mitigate divergence loss, thereby enhancing their potential rewards compared with those who do not employ such targeted strategies.

  Divergence loss, or impermanent loss, is an implicit cost to a liquidity provider tied to the risk of a decline in value of the liquidity position compared with the value of the initial deposited assets. Heimbach et al. (2022) demonstrate how liquidity providers' risk–return profile of selected liquidity ranges in Uniswap V3 can exhibit significant fluctuations, which may require active management strategies to circumnavigate. Furthermore, such active management of positions can affect the market depth in volatile market conditions, which goes against the interests of the AMM protocol.

Recent research has delved into the intricacies of these challenges and highlighted the need for innovative solutions (Auer et al. 2023; Frontier Research 2023; Phan 2024). However, significant challenges remain that have not been comprehensively addressed in the literature. The dynamics of liquidity pools, especially in terms of capital inefficiency, significant slippage, and divergence loss, are not yet fully optimized for liquidity pools, providers, and takers (Xu et al. 2021; Heimbach et al. 2022). Moreover, the potential of integrating deep learning predictive mechanisms (Zhang et al. 2023; Sabate-Vidales and Šiška 2022), especially for liquidity provisioning, represents relatively unexplored territory.

The present study proposes a quote-driven AMM with its original intent of on-chain custody and settlement functions, alongside off-chain predictive reinforcement learning capabilities. First, the proposed AMM architecture augments the Uniswap V3 protocol by using novel market equilibrium pricing for reduced divergence and slippage loss. Second, the proposed protocol involves a predictive AMM capability through utilizing a deep hybrid reinforcement learning framework that seeks to improve market efficiency through more accurate forecasts of liquidity concentration ranges; thus, liquidity starts moving to expected concentration ranges prior to asset price movement, such that liquidity utilization is improved.

This study introduces a transformative approach to enhancing liquidity provision in AMMs for the realm of DeFi. Our main contributions are summarized as follows:

1. *Quote-driven predictive AMM platform:* We propose a unique quote-driven predictive AMM platform that integrates on-chain custody and settlement functions. This design is unique as it synergizes off-chain predictive reinforcement learning capabilities, thus offering improved liquidity provision compared with conventional AMMs.

2. *Augmentation of Uniswap V3:* Our architecture represents a significant advancement of the renowned Uniswap V3 cryptocurrency AMM protocol. By employing a novel market equilibrium pricing mechanism, we achieve reduced divergence and slippage losses, thereby addressing major challenges faced in the current AMM landscape.
3. *Predictive capability with deep learning:* Zhang et al. (2023) and Sabate-Vidales and Šiška (2022) have used reinforcement learning to improve earnings for liquidity providers. In this study, aside from drawing liquidity providers through improved incentive fees (vis-à-vis Uniswap V3), we focus on enhancing liquidity utilization in an AMM. Central to our approach is the incorporation of a deep hybrid long short-term memory (LSTM) and Q-learning reinforcement learning framework. The said framework not only enhances market efficiency but also ensures more accurate forecasts of liquidity concentration ranges. The result is proactive liquidity movement to anticipated concentration ranges even before asset price shifts; thus, liquidity utilization is optimized.

Through empirical simulations and methodical analysis, this research aims to evaluate the proposed model's effectiveness compared with the baseline Uniswap V3 AMM architecture. The augmented protocol framework is expected to reduce (1) divergence loss for liquidity providers and (2) slippage for crypto-asset traders while (3) improving capital efficiency for liquidity provision for the AMM protocol. The proposed innovations not only set a new benchmark for AMM architectures but also hold the potential to revolutionize DeFi platforms' efficiency and effectiveness.

Background and related works

AMMs have become a cornerstone of the DeFi landscape, significantly influencing the structure and functionality of DEXs (Meyer et al. 2022). Schär (2020) notes that the evolution of DEXs and AMMs represents a pivotal shift toward more accessible and transparent financial markets. These AMMs function by converting inputs (tokens) into outputs (prices) through a defined "exchange function" (Mohan 2022). Malinova and Park (2024) posit that AMMs should be economically viable in traditional financial markets, especially with advancements in asset tokenization and the increasing regulatory recognition of crypto-tokens. Moreover, Schmitt (2023) explains that AMMs' design efficiency is why they currently handle over 95% of all DEX transactions, overshadowing other exchange models. This dominance is due to their ability to offer continuous liquidity and immediate pricing without requiring traditional order books.

Despite their successes, AMMs face significant challenges that affect their performance and reliability. Auer et al. (2023) conduct an in-depth analysis of the technological foundations of DeFi and AMMs and note that while innovative designs have been developed, many existing models still struggle with various problems, such as price slippage, impermanent loss, and capital inefficiency. Frontier Research (2023) and Phan (2024) have further highlighted the need for innovative design mechanisms that can address these challenges, suggesting that such improvements could catalyze further growth and development in the DeFi sector.

Economics of AMM DEXs

Current innovations in AMM design are primarily targeted at enhancing economic efficiency in DeFi markets. A pivotal innovation is concentrated liquidity mechanisms. Fritsch (2021) notes their profound impact on liquidity provider returns and market dynamics, especially on platforms such as Uniswap. This approach allows liquidity providers to allocate funds to specific price ranges, significantly boosting potential returns and market efficiency.

Furthermore, Xu et al. (2021) discuss the economics of AMM DEXs, including rewards such as liquidity incentive fees, and implicit costs such as divergence and slippage losses. Heimbach et al. (2022) analyze factors that influence the performance of liquidity positions in Uniswap V3, including divergence loss and the selection of liquidity positions. Moreover, Neuder et al. (2021) and Cartea et al. (2022) have introduced optimal liquidity provision strategies to maximize liquidity provisioning earnings. Additionally, Singh et al. (2023) discuss the problem of shallow liquidity in low trading volume token pairs, while Bar-On and Mansour (2023) examine the problem of determining optimal price intervals for liquidity provision using online learning.

Pricing accuracy within AMMs is a critical concern. Aoyagi (2020) proposes the use of an equilibrium valuation point to enhance pricing accuracy in AMM DEXs. Building upon this, Engel and Herlihy (2021b) analyze how the equilibrium valuation price and divergence and slippage losses can be minimized in AMM DEXs based upon the formal model, axioms, and notations in the paper of Engel and Herlihy (2021a). Engel and Herlihy (2021a; 2021b) provide the foundational work for this paper.

Deep reinforcement learning on AMM DEXs

The application of reinforcement learning to market making started as early as 2001 (Chan and Shelton 2001). More recently, Hambly et al. (2021) provide an account of the state-of-the-art of reinforcement learning's application to market making.

Market making is generally applied in market microstructure modeling research using the stochastic control or reinforcement learning approaches, where optimal bidding (e.g., pricing strategy in limit order books [LOBs]) is studied (Sun et al. 2022). This study restricts the focus to the application of reinforcement learning on AMM DEXs, which operates in an algorithmically deterministic market making manner, as opposed to using LOBs. Pourpouneh et al. (2020) provide a survey of current AMM models.

Research on this subdomain is sparse. Most crypto-asset–based research that has applied deep reinforcement learning is related to automated trading from an investment management perspective. Lucarelli and Borrotti (2019) cover this to some extent. In relation to DEXs, Sadighian (2019, 2020) proposes—and later enhances—a deep reinforcement learning framework for a crypto-asset DEX. They use a policy gradient–based algorithm to interact with data from an LOB and order flow arrival statistics to solve a stochastic inventory control problem. Later, Sabate-Vidales and Šiška (2022) use the actor-critic approach to investigate the potential earnings from liquidity provision in constant product markets, and they recommend the introduction of adjustable fee levels for liquidity providers. Moreover, Zhang et al. (2023) use the Double Dueling Deep

Q-Learning Network for optimal liquidity provisioning for enhanced return on investment for liquidity providers. Despite these recent studies, research that has applied deep reinforcement learning to crypto-asset–based AMM DEXs is limited.

Preliminaries and proposed method

Notation

We define the notations in this paper following Engel and Herlihy (2021a). Italics are used for scalars (x) and bold typography for vectors ($\boldsymbol{x}$). Constants are defined from the beginning of the alphabet (a, b, c), and variables, vectors, or scalars from the end (x, y, z). We use "=" to represent equality and ":=" for definitions. We also use subscript "*obs*" a market observed price and a subscript "*p*" for a predicted valuation.

Informally, to represent the CPMM function, an AMM in state (x, y) has custody of x units of token X and y units of token Y, subjected to $x \cdot y = c$, where $x, y > 0$ and some constant $c > 0$. For any trade to occur, liquidity invariance is achieved when a buyer purchasing the δ_X of token X deposits the δ_Y of token Y, such that $(x - \delta_X) \cdot (y + \delta_Y) = c$.

To formally represent the CPMM, an AMM state space with trading assets X and Y is represented by $(x, y) \epsilon \mathbb{R}^2_{>0}$. The state space is represented by curve $(x, f(x))$, such that $f : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$. This study assumes that the pool of assets is not exhausted, while boundary conditions are set as $\lim_{x \to 0} f(x) = \infty$ and $\lim_{x \to \infty} f(x) = 0$.

Uniswap charges fees of 0.3% for each trade back to the asset pool, which are in part used to incentivize liquidity providers. This study ignores the effects of these fees as they have a minimal impact on costs. In general, fees cause a slight reduction in divergence loss for liquidity providers and in slippage cost for liquidity takers.

Equilibrium state

For asset pricing, it is assumed that only one market valuation is acceptable to most liquidity takers at any time. *Valuation* $v \epsilon (0,1)$ is assigned, such that v units worth of X equate to $(1 - v)$ unit worth of Y. At valuation v, a profit is made when $v(x - x') + (1 - v)(f(x) - f(x'))$ is positive when the AMM state space moves from $(x, f(x))$ to $(x', f(x'))$. Otherwise, a loss is incurred.

An *equilibrium point*, or the state at which no arbitrage profits can be made, is defined as a valuation v at point $(x, f(x))$ that solves the optimization problem (Eq. 1):

$$v := \min_x \boldsymbol{v_{obs}} \cdot \boldsymbol{x} \tag{1}$$

where $\boldsymbol{v_{obs}}$ is the market observed price on the asset obtained from a trusted price oracle.

For $(x, f(x))$ to be the equilibrium point, $\frac{df(x)}{dx} = -\frac{v}{1-v}$. The exchange rate of asset Y in units of asset X is defined as $-f'(x)$. This is the negative of the curve's slope at the point.

To carry each valuation v to the equilibrium state x that minimizes the dot product $\boldsymbol{v_{obs}} \cdot \boldsymbol{x}$, or $vx + (1 - v)f(x)$, we define $\phi(v) = f'^{-1}(-\frac{v}{1-v})$, where $\phi : (0,1) \rightarrow \mathbb{R}_{>0}$. For instance, the equilibrium state for AMM at $\left(x, \frac{1}{x}\right)$ is $\phi(v) = \sqrt{\frac{1-v}{v}}$. It is useful to express ϕ in vector representation $\boldsymbol{\Phi}(v) := \left(\phi(v), f(\phi(v))\right)$, where $\boldsymbol{\Phi} : (0,1) \rightarrow \mathbb{R}^2_{>0}$. The inverse of ϕ is represented by $\psi(x) = -\frac{f'(x)}{1-f'(x)}$, where $\psi : \mathbb{R}_{>0} \rightarrow (0,1)$. The vector representation is $\boldsymbol{\Psi}(x) := (\psi(x), 1 - \psi(x))$.

Every x is the equilibrium point for some valuation v. For instance, for a CPMM AMM $:= \left(x, \frac{1}{x}\right)$, the point $\left(x, \frac{1}{x}\right)$ is the equilibrium point for $\left(\frac{1}{1+x^2}, 1 - \frac{1}{1+x^2}\right)$. To generalize, for a CPMM AMM $:= \left(x, f(x)\right)$, the point $\left(x, f(x)\right)$ is the equilibrium point for $\left(\frac{f'(x)}{f'(x)-1}, \frac{f'(x)}{1-f'(x)}\right)$ (Engel and Herlihy 2021a).

Total value of AMM holdings

Let the valuation with equilibrium point $(x, f(x))$ be defined as $(v, 1-v)$. Given $\boldsymbol{v} = (v, 1-v)$ and $\boldsymbol{x} = \left(x, f(x)\right)$, the total value (or *capitalization*) of the total AMM holding is given as follows (Engel and Herlihy 2021b):

$$cap(x, v) := vx + (1-v)f(x) = \boldsymbol{v} \cdot \boldsymbol{x}$$

When v represents the current market valuation, the AMM is in the equilibrium state $\boldsymbol{\Phi}(v) = \left(\phi(v), f(\phi(v))\right)$, giving

$$cap(v) := cap(\phi(v), v) = \boldsymbol{v} \cdot \boldsymbol{\Phi}(v)$$

In the case of a CPMM AMM $:= \left(x, \frac{1}{x}\right)$, the capitalization at the equilibrium point is given by

$$cap\left(v; (x, \frac{1}{x})\right) := 2\sqrt{v(1-v)}$$

Divergence loss, slippage loss, and load

To improve the performance of an AMM using the CPMM function, we seek to reduce divergence and slippage losses. This subsection defines divergence and slippage losses (Engel and Herlihy 2021b) and identifies a composite loss function to reduce them.

- Divergence loss:
  Divergence loss is incurred when a difference in value arises from the funds remaining in the wallet, against the initial fund amount deposited into the AMM. If the valuation $\boldsymbol{v}$ moves to $\boldsymbol{v}'$, the equilibrium state will shift from $\boldsymbol{x}$ to $\boldsymbol{x}'$. The shift away from $\boldsymbol{v}$ creates an unstable state, such that arbitrageurs will be able to profit the amount of $\boldsymbol{v}' \cdot \boldsymbol{x} - \boldsymbol{v}' \cdot \boldsymbol{x}'$.
  Divergence loss is defined as a function of liquidity pool size as follows:

$$\begin{aligned} loss_{div}\left(v, v'\right) &:= \boldsymbol{v}' \cdot \boldsymbol{\Phi}(v) - \boldsymbol{v}' \cdot \boldsymbol{\Phi}\left(v'\right) \\ &= v'\phi(v) + \left(1-v'\right)f(\phi(v)) - \left(v'\phi\left(v'\right) + \left(1-v'\right)f\left(\phi\left(v'\right)\right)\right) \end{aligned}$$

  where $\boldsymbol{\Phi}(v, 1-v) = (\phi(v), f(\phi(v)))$.
  In the case of a CPMM AMM $:= \left(x, \frac{1}{x}\right)$, the divergence loss for trade size δ is given by

$$loss_{div}(x, x+\delta) := \frac{\delta^2}{2\delta x^2 + x^3 + \delta^2 x + x}$$

- Slippage loss:
  Slippage loss is defined by how an increase in trade sizes can reduce a liquidity taker's return. Suppose that a trade of size δ is placed, where $\delta > 0$. Here, the state of the AMM changes from $(x,f(x))$ to $(x+\delta,f(x+\delta))$. In a linear rate of exchange, in an exchange of δ units of X, the trader receives $-\delta f'(x)$ units of Y. Therefore, the trader makes a loss of $-\delta f'(x) - f(x) + f(x+\delta)$, resulting in the final receipt of $f(x) - f(x+\delta)$.
  Slippage is defined as a function of liquidity pool size as follows:

  $$loss_{slip}\left(v,v'\right) := \left(\frac{1-v'}{1-v}\right)\left(\boldsymbol{v}\cdot\boldsymbol{\Phi}\left(v'\right) - \boldsymbol{v}\cdot\boldsymbol{\Phi}(v)\right)$$

  In the case of a CPMM AMM $:= \left(x, \frac{1}{x}\right)$, the divergence loss for trade size δ is given by

  $$loss_{slip}(x,x+\delta) := -\frac{\delta^2(\delta+x)}{x^2(\delta^2+x^2+2\delta x+1)}$$

- Composite divergence and slippage loss:
  To reduce the overall effect of the cost of divergence loss to liquidity providers and slippage loss to liquidity takers, a composite function known as *load* (Engel and Herlihy 2021b), which considers both divergence and slippage losses, can be useful. Load across an interval, with respect to X, is defined as the product of the interval's slippage and divergence loss and is given by

  $$load_X(v,v') := loss_{div}(v,v')\cdot loss_{slip}(v,v')$$

  Given a probability density for future valuations, one can compute an *expected load* when exchanging X tokens for Y tokens, starting in the equilibrium state for valuation v, given that $p(v')$ is the distribution over possible future valuations (Eq. 2):

  $$E_p\left[load\left(v'\right)\right] := \int_0^v p\left(v'\right)load_X\left(v,v'\right)dv' + \int_v^1 p\left(v'\right)load_Y\left(v,v'\right)dv' \tag{2}$$

Pricing and changes to liquidity provision

Suppose that for an AMM $:= (x,f(x))$, the valuation moves from v with equilibrium state (a,b) to v' with equilibrium state (a',b'). Then, an arbitrageur can make an arbitrage profit by moving from (a,b) to (a',b').

One can eliminate this arbitrage, which results in divergence loss, by moving the bonding curve in the AMM protocol, as a *pseudo arbitrage*, as Engel and Herlihy (2021b) refer to it. Suppose that $a > a'$ and $b' > b$, then the transformed AMM becomes $AMM' := \left(x,f\left(x-\left(a-a'\right)\right)-\left(b'-b\right)\right)$. The new equilibrium state v' = the new market price and continues to lie on the shifted curve with a slope of $\frac{v'}{v'-1}$.

A downside of this pseudo arbitrage is that the AMM now has more units of X and a shortage of Y to cover all possible trades (Engel and Herlihy 2021b). However, this imbalance is small as each price action is generally driven by small tick changes, assuming an efficient market, but they can add up over time and become problematic. The AMM will have to account for this shortfall by making minor liquidity provision adjustments to rebalance the pool. This implies that, as part of liquidity provision, liquidity providers will deposit an additional X or Y tokens as stated by the AMM. Incentives will be provided for all tokens deposited, including the additional X or Y tokens. This study proposes incorporating this in its configurable virtual AMM, as demonstrated in the proposed AMM architecture presented in a later section. Thus, the bonding curve will revert to its primary CPMM bonding curve formula.

Deep reinforcement learning

In instances where an interaction exists between the agent (AMM) and the environment (i.e., the financial market, including market participants such as liquidity takers and providers), one can execute actions and receive observations and rewards as a Markov decision process. At each time step$t$, the agent selects an action $a_t \epsilon \mathcal{A}$ at state$s_t \epsilon \mathcal{S}$, where $\mathcal{S}$ is the set of possible states. This step of action selection depends on the policyπ, which is a description of the agent behavior, and it guides the actions taken by the agent for each possible state. Upon the execution of each action, the agent receives a scalar reward $r_t \epsilon \mathcal{R}$ and the next state s_{t+1} is observed. This learning sequence is repeated in a (possibly infinite) horizon T until the algorithm is halted. The transition probability of possible future state s_{t+1} is given by$P(s_{t+1}|s_t, a_t)$, while the reward probability is given by $P(r_t|s_t, a_t)$. Therefore the expected reward is computed as follows:$E_{P(r_t|s_t,a_t)}(r_t|s_t = s, a_t = a)$.

- Event-driven environment:

  This study considers a state based AMM agent that acts on events as they occur. The action space is based on a typical market making strategy where the agent cannot exit the market and is restricted to executing a single order. An event constitutes an observable change in the state of the environment and can occur due to a change in price. This implies that actions are not regularly spaced in time. The agent is required to quote prices at which it is willing to buy and sell at valid timepoints unless constraints on the asset inventory prevail.

  In line with Sadighian (2020), this study proposes the use of a price-based approach for an event-driven environment, where an event is defined as a change in equilibrium valuation v', and if this is greater than or less than a threshold β_v, β_v allows the adjustment of the learning rate's sensitivity.

  These price change events are not regularly spaced in time, which reduces the time required to train the agent per episode (i.e., an executed trading action that results in a price change). Algorithm 1 presents the algorithm used to evaluate price-based events (Sadighian 2020):

Algorithm 1: Evaluating a Price-Based Event

```
Input: Valuation $v'_t$ at time $t_0 = 0$
Output: Observation at time $t_k = t_0 + k$

$\beta_v \leftarrow 0.01\%$
$k \leftarrow 0$
$upper \leftarrow v'_t(1 + \beta_v)$
$lower \leftarrow v'_t(1 - \beta_v)$
$step \leftarrow True$
while $step$ do
    if $upper \leq v'_{t+k} \leq lower$ then
        $k \leftarrow k + 1$
    else
        $step \leftarrow False$
    end
end
```

- Reward function:

  To improve market efficiency and provide optimal liquidity, this study ties the reward objective function for trading agents to the quality of forward prediction of valuation v'_p, against the equilibrium valuation v' at this future time, and implicit costs for liquidity takers and providers.

  In most state-of-the-art reinforcement learning literature for market making, the obvious reward functions selected are profit-seeking (Spooner et al. 2018; Sadighian 2020; Haider et al. 2022) or utility-maximizing (Selser et al. 2021) as the natural choices of reward functions.

  We propose the following single-step loss function ℓ (Eq. 3):

  $$\ell := |v'_t - v'_{p,t}| + E_p\left[load\left(v'\right)\right] \tag{3}$$

  The loss function in Eq. (3) computes the *prediction slippage*, or the difference between the valuation v'_p as predicted by an AMM prediction model, against the equilibrium point v' (computed from Eq. 1). The latter is a function of the actual observed valuation from a trusted price oracle. We take the modulus of this difference as we seek to identify absolute deviations between prediction and equilibrium prices; thus, we can minimize this difference using reinforcement learning. Furthermore, we add the expected load (computed from Eq. 2), which represents divergence and slippage losses. Our overall objective is to minimize this function by reducing prediction slippage and expected load, in turn improving capital efficiency.

  The cumulative reward function R is obtained as follows:

  $$R_t := \sum_{k=0}^{k=T} \gamma^k r_{t+k}$$

where $\gamma \epsilon (0,1)$ is a parameter called the discount rate, and r is defined as follows:

$$r_{t+k} := \begin{cases} -1, & if\ \ell_t > \beta_c \\ 0, & if\ \ell_t = \beta_c \\ +1, & if\ \ell_t < \beta_c \end{cases}$$

where β_c represents a threshold within which prediction slippage and expected load can be tolerated. This threshold determines the sensitivity of the reward function to the loss function (Eq. 3).

- Action space:
  The agent action space consists of the following two possible actions:

$$A_t := \begin{cases} Insert\ input\ parameter\ \varepsilon_{t+k}, & if\ \ell_{k-t} > \beta_c \\ Do\ nothing, & if\ \ell_t \leq \beta_c \end{cases}$$

where ε represents a Gaussian input parameter to the learning model, where $\varepsilon \epsilon (-1,1)$ and $\varepsilon \sim N(\mu_\varepsilon, \sigma_\varepsilon)$. This input parameter effects changes to the learning model with the goal of helping to reduce prediction slippage and expected load.

- State space observations:
  An environment state is constructed from an attribute set that describes the condition of the market and the agent. The market state comprises observations derived from the following, among others:
- A market valuation obtained from an external trusted price oracle, represented by v_{obs};
- Preprocessed alternative data that indicate price signals that effect changes in market liquidity, represented by τ;
  An example of such alternative data sources is market signals generated from Twitter data processed using natural language processing to make predictions (Abraham et al. 2018; Kraaijeveld and De Smedt 2020). In this study, we pretrain an LSTM supervised learning model and use the LSTM outputs as observation inputs for reinforcement learning (Liu 2020).
  The agent state comprises observations derived from the trading agent's own records, including the following:
- The number of units of token X, represented by x, and the number of units of token Y, represented by y.
- *Q-learning:*
  The expected discounted return at time t is defined as follows: $R_t := E[\sum_{k=t}^{k=T} \gamma^{k-t} r_{k-t+1}]$. By applying Q-learning as a recursive update procedure, the Q-value function $Q^\pi(s,a)$ is defined as follows:

$$Q_{i+1}^{\pi}(s,a) := E_\pi[r_t + \gamma \sum_{k=0}^{k=T} \gamma^k r_{t+k+1} | s_t = s, a_t = a]$$

$$= E_\pi[r_t + \gamma Q_i^\pi(s_{t+1} = s', a_{t+1} = a') | s_t = s, a_t = a]$$

Reinforcement learning learns the optimal policy π^*, whose expected value is greater than or equal to all other policies, to converge at an optimal Q-value $Q^*(s,a)$.

$$Q_{i+1}(s,a) := E_\pi[r_t + \gamma \max_{a' \epsilon A} Q_i(s',a')|s,a]$$

$$Q^*(s,a) := (\mathcal{B}Q^*)(s,a)$$

where $\mathcal{B}$ represents the Bellman operator that maps any function $\mathcal{K} : S \times A \mapsto R$ into another function $S \times A \mapsto R$. The Bellman operator is given as follows:

$$(\mathcal{BK})(s,a) := \sum_{s' \epsilon \mathcal{S}} \mathcal{T}(s,a,s')[R(s,a,s') + \gamma \max_{a' \epsilon A} \mathcal{K}(s',a')]$$

where $\mathcal{T}$ represents the function for computing the transaction value to move from s to s', given an action a.

- Deep reinforcement learning architecture:
  To perform prediction for the forward valuation v'_p, Liu (2020) finds a hybridized advantage actor-critic agent to be useful through performing the pretraining of a supervised recurrent neural network (RNN) in the form of LSTM, before using the LSTM outputs as observation inputs for reinforcement learning. This paper proposes a hybrid LSTM–Q-learning architecture, the architectural derivatives of which are proposed by Lucarelli and Borrotti (2019) and Liu (2020).
  The unique advantage of LSTM lies in its ability to remember short-term patterns for long periods. This is especially useful when one is dealing with sequences of market events, such as time-series financial data. Furthermore, LSTM's memory cell structure mitigates the vanishing gradient problem of traditional RNNs, enabling the network to learn from important experiences observed n periods ago. In addition, LSTMs can be combined with other deep learning architectures, which makes them versatile for various prediction challenges. In financial markets, numerous studies have indicated the superior performance of LSTMs in predicting stock prices, trading volumes, and other relevant metrics (Hu et al. 2021; Rundo 2019).
  Our hybrid LSTM–Q-learning approach synergizes the strengths of both LSTM and Q-learning algorithms. LSTM is used for its capability to model and predict time-series data, providing a forecast of future market valuations. Q-learning's primary role is to guide the AMM's actions to optimize liquidity provision. Given a state (predicted future market valuation), it provides the AMM with an action that aims to maximize expected future rewards. These rewards are related to enhanced capital efficiency, reduced slippage, and minimized divergence loss.
  Algorithm 2 presents the LSTM algorithm. One LSTM layer was applied with 100 neurons, with a sliding window of 50 interval inputs, including the following: (1) the market observed price from the trusted oracle v_{obs}; (2) preprocessed alternative data representing market movement signals τ; and (3) a Gaussian input parameter from the action space ε that aims to reduce prediction slippage and load.

Algorithm 2: The LSTM Algorithm

```
Input: Market observed price sequence: $v_{obs} = \{v'_{obs}, v''_{obs}, \ldots, v^{t}_{obs}\}$
    External market movement signals: $\tau = \{\tau_1, \tau_2, \ldots, \tau_t\}$
    Gaussian input parameter: $\varepsilon$
Output: Predicted valuation: $v'_{p,t}$

Initialize LSTM network with weights $W$.
Set epoch count $e$ to predefined value.
For $i$ from 1 to $e$ do
  For each time step $t$ in sequence length do
    Read the market price $v^{t}_{obs}$, external signals $\tau_t$, and Gaussian input parameter $\varepsilon$.
    Update LSTM's internal state based on input and previous state.
    Predict the next valuation $v'_{p,t}$.
  Update weights $W$ based on prediction error using backpropagation through time.
 Output the predicted valuation $v'_{p,t}$ for the next time step.
end
```

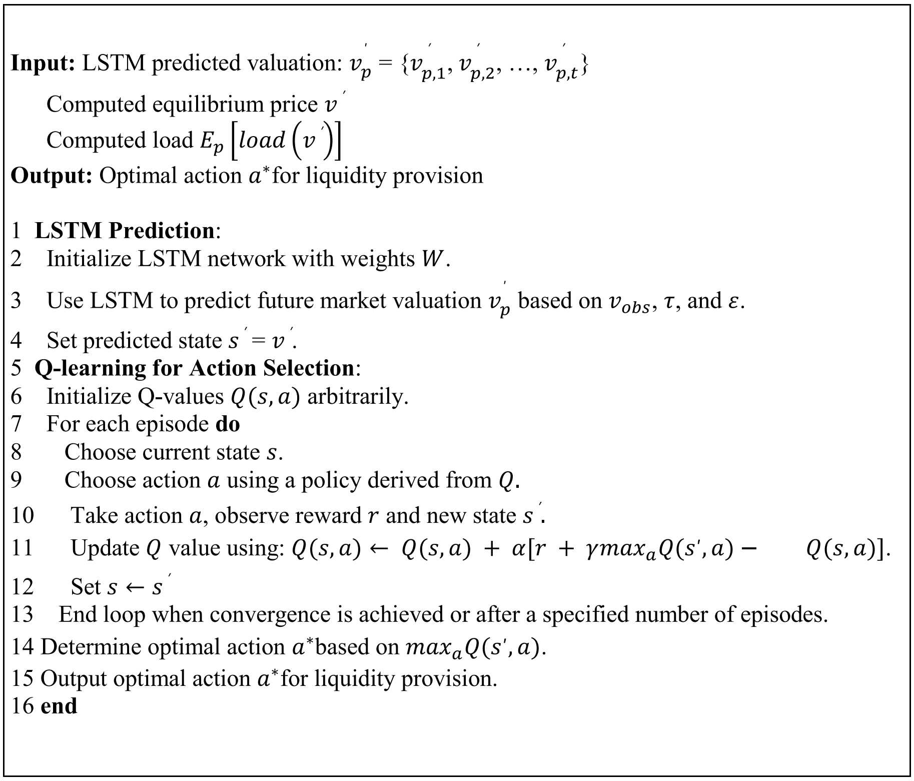
Q-learning is employed to derive optimal action policies for the agent. In this context, it determines the optimal liquidity provision strategy based on the predicted market valuations from LSTM. The hybridization works as follows: (1) LSTM predicts future market valuations; (2) these predicted valuations serve as the state input for the Q-learning algorithm; and (3) based on these states, Q-learning determines the optimal action to maximize the expected reward.

In addition, this study uses the Dueling Double Deep Q-Learning Network (DD-DQN) architecture. DD-DQN combines the benefits of Double Deep Q-Network (D-DQN), which mitigates overestimations of Q-values in traditional DQNs for more stable and robust learning. Moreover, the dueling network divides the Q-value estimation into two streams—one for estimating the state value function and the other for estimating the advantage function for each action. The separation of the latter provides a more nuanced estimation of Q-values, especially in situations where the difference between actions is minimal.

Algorithm 3 presents the hybrid LSTM–Q-learning algorithm. The predicted output of v'_p with a sliding window of 10 interval inputs, computed equilibrium price v', and computed load $E_p\left[load\left(v'\right)\right]$ are used as inputs for the Q-learning model. In the DD-DQN architecture, two CNN layers are applied, each with 100 neurons, followed by two fully connected layer streams—one with 50 neurons for estimating the value function and another with 50 neurons for estimating the advantage function. Both epochs and batch sizes are set to 50. For weight optimization, the Adam algorithm is applied (Kingma and Ba 2015), while for the activation function, the Leaky Rectified Linear Units (Leaky ReLU) algorithm is applied (Maas et al. 2013), where γ is set to 0.98 (Lucarelli and Borrotti 2019). Given Eq. 3, the loss function is defined as follows (Eq. 4):

$$\mathcal{L} := \frac{1}{n}\sum_{k=0}^{k=T}\{|v'_t - v'_{p,t}| + E_p\left[load\left(v'\right)\right]\} \tag{4}$$

Algorithm 3: The Hybrid LSTM-Q-learning Algorithm



```
Input: LSTM predicted valuation: $v_p^{'} = \{v_{p,1}^{'}, v_{p,2}^{'}, \ldots, v_{p,t}^{'}\}$
    Computed equilibrium price $v^{'}$
    Computed load $E_p\left[load\left(v^{'}\right)\right]$
Output: Optimal action $a^*$for liquidity provision

LSTM Prediction:
    Initialize LSTM network with weights $W$.
    Use LSTM to predict future market valuation $v_p^{'}$ based on $v_{obs}$, $\tau$, and $\varepsilon$.
    Set predicted state $s^{'} = v^{'}$.
Q-learning for Action Selection:
    Initialize Q-values $Q(s,a)$ arbitrarily.
    For each episode do
        Choose current state $s$.
        Choose action $a$ using a policy derived from $Q$.
        Take action $a$, observe reward $r$ and new state $s^{'}$.
        Update $Q$ value using: $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_a Q(s',a) - Q(s,a)]$.
        Set $s \leftarrow s^{'}$
    End loop when convergence is achieved or after a specified number of episodes.
Determine optimal action $a^*$based on $max_a Q(s',a)$.
Output optimal action $a^*$for liquidity provision.
end
```

Figure 2 depicts the proposed recursive LSTM–Q-learning DD-DQN reinforcement learning architecture:

Figure 3 presents details of the neural network layers:

Table 1 presents the key parameters used in both LSTM and Q-learning:

Predictive liquidity distribution

At present, liquidity concentration ranges, in which incentives to liquidity providers are distributed, are created on a look-back basis by relying on observed market values (Uniswap 2022). A market maker is responsible for providing liquidity for trade execution. Liquidity pooling requires time to form. Through the advanced prediction of valuation v'_p, incentivization for liquidity provision can be altered in n intervals in advance (e.g., 1, 5, or 10 intervals); therefore, liquidity shifts before the actual market change. The shifting of incentive fee distribution can help to motivate liquidity

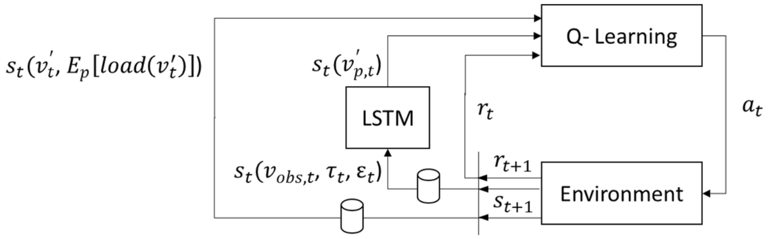


Fig. 2 Recursive LSTM-Q-learning reinforcement learning architecture. This figure shows how a Q-learning network with LSTM units can be used to model and predict forward valuation v_p' for optimal liquidity pooling

providers who seek higher yields to support predicted new liquidity concentration ranges; thus, pooled capital efficiency can be achieved.

Furthermore, the current incentivization program for Uniswap V3 liquidity providers is binary in nature. Thus, fees are only earned if liquidity providers provide liquidity within a certain range in the bonding curve, and they are not compensated if their liquidity provision falls outside of that range. However, research demonstrates that the proportion of time where asset prices remain within a liquidity position relative to a liquidity width is not uniformly distributed (Heimbach et al. 2022). While active liquidity providers benefit from range targeting to earn the best possible fees in a uniform distribution fee structure in Uniswap V3, it is useful to consider a different distribution structure that can help to insure against sharp price movements. This can in turn help to improve the attractiveness of liquidity provision.

We use v_p' to help to determine the position of the new liquidity concentration range on the bonding curve. The distribution of the incentive fee φ is proposed to be Gaussian in nature (Fig. 4), such that $\varphi \sim N\left(\mu v_{\varphi}, \sigma_{\varphi}\right)$ and $\mu_{\varphi} = v_p'$. It is given as follows:

$$\varphi(x) := \frac{1}{\sigma_{\varphi}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-v_p'}{\sigma_{\varphi}}\right)^2}$$

In effect, the LSTM predicted v_p' formulates the new liquidity concentration region on the bonding curve in n intervals in advance; thus, the liquidity pool rebalances its liquidity before an actual market change occurs. Active yield seekers who shift liquidity to new liquidity concentration positions will be rewarded positively. Furthermore, as incentivization distribution is Gaussian in nature, liquidity providers continue to be incentivized, albeit to lesser amounts, even if they do not correctly identify the optimal prices and length of time for positioning their liquidity provision. Compared with Uniswap V3, this relative lowering of "incentive penalization" due to incorrect liquidity positioning seeks to help to draw liquidity providers.

For this purpose, this study also proposes including transparency in v_p' and the historical shifts in v_p' in the AMM design to liquidity providers, thereby positively improving the market's ability to analyze and preposition resource allocation.

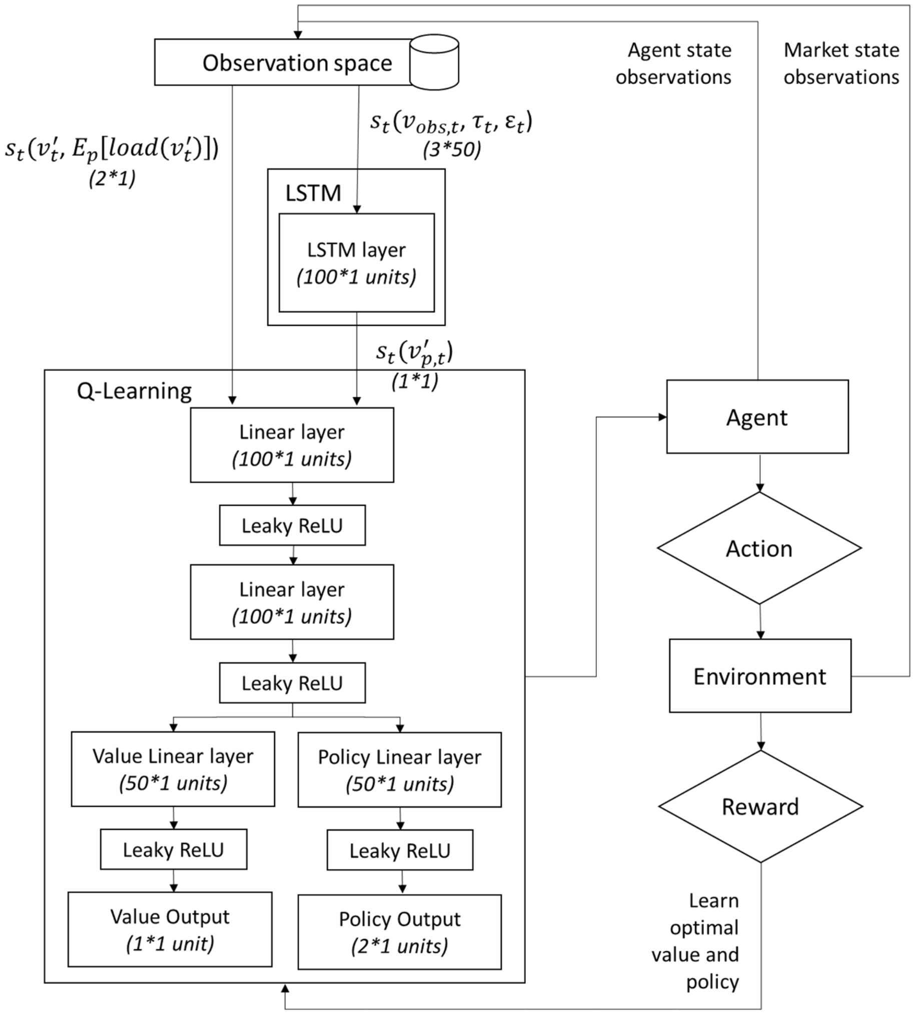


Fig. 3 LSTM-Q-learning reinforcement learning architecture layers. Sliding window input observations are first directed to a linear layer with a size of 100. Following this, a Leaky ReLU activation function is applied. The resulting output is then channelled separately to both the policy and value components. The policy is optimized using the policy gradient method, with the loss function defined by Eq. (4)

Table 1 Key parameters of LSTM and Q-learning

Parameter	Description	Algorithm	Value/range
W	LSTM network weights	LSTM	Initialized randomly
e	Epoch count for training LSTM	LSTM	50
α	Learning rate for Q-learning	Q-learning	0.1 (typical starting value)
γ	Discount factor for future rewards in Q-learning	Q-learning	0.98
ϵ	Exploration rate for epsilon-greedy policy in Q-learning	Q-learning	Starts at 1.0, decays over time
Q (s,a)	Q-value for state-action pairs	Q-learning	Initialized to zeros
s and s′	Current and next state, respectively	Q-learning	Based on LSTM predictions and environment
*a and a**	Chosen and optimal action, respectively	Q-learning	Determined by policy and Q-values
r	Reward received after taking an action	Q-learning	Computed based on state transition and reward function

Adjustments to the values stated herewith can be made based on specific use cases or further tuning

Experiment

To ascertain the efficacy of the proposed AMM architecture, this study performs an experiment that aims to systematically evaluate and compare the architecture's capital efficiency, divergence loss, and slippage reduction against those of the baseline Uniswap V3 AMM architecture. We compare the baseline and proposed architectures by integrating LSTM and Q-learning reinforcement.

A synthetic dataset of trading prices and volumes is simulated, including 7,000 trading hours of data for training, 2,000 h for testing, and 1,000 h for validation. The simulations operate under efficient market conditions, with price actions driven by small tick changes. Trade volume variations in the simulations are reflective of the real-world trading variations in AMM architectures.

To ascertain capital efficiency in terms of performance metrics, this study examines liquidity utilization, liquidity concentration, and liquidity depth as follows:

- Liquidity utilization:
  Liquidity utilization measures how effectively the deposited capital (liquidity) is used in facilitating trades. It is computed as follows:

$$Liquidity\,Utilization = \frac{Trade\,Volume}{Average\,Liquidity}$$

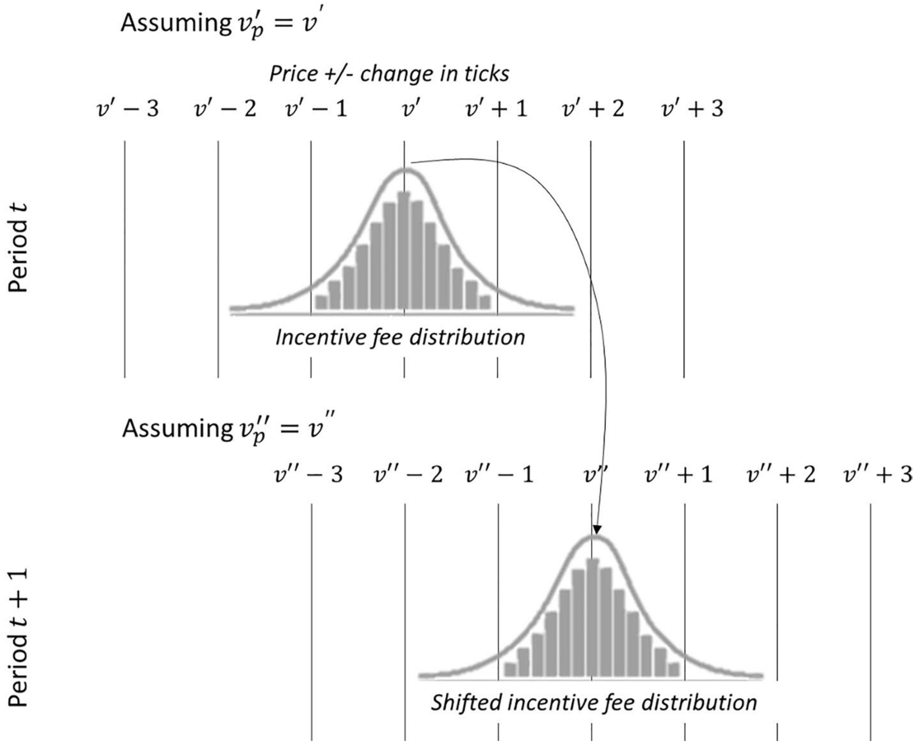


Fig. 4 Incentive distribution for liquidity providers. This figure shows that LSTM-predicted v_p' formulates the new liquidity concentration region with mean v_p'' on the bonding curve in 1 interval in advance, so the liquidity pool rebalances its liquidity before actual market change occurs

For comparative purposes, this study computes liquidity utilization for both the proposed AMM architecture and the baseline Uniswap V3. Then, it compares the results to determine which architecture has a better liquidity utilization rate.

- Liquidity concentration:

  Liquidity concentration indicates how liquidity is distributed within an AMM. Liquidity should ideally be concentrated around the current price, ensuring that trades close to the current price experience minimal slippage.

  For comparative purposes, this study simulates the liquidity distribution over a range of prices for both AMM architectures to understand how liquidity is distributed within an AMM.

- Liquidity depth:

  Liquidity depth refers to the amount of an asset that can be bought or sold at a particular price point without causing a significant change in its price. A deeper liquidity pool means that larger trades can be executed without a significant impact on prices.

  For comparative purposes, this study computes trade sizes that result in a price impact closest to a reasonable acceptable threshold of 1% for both AMM architectures to ascertain the liquidity depth.

To ascertain the divergence loss, this study compares the difference in values arising from the funds remaining in the AMM with the initial fund amount deposited. This is done for the baseline Uniswap V3 AMM architecture, with its constant product formula, and also the proposed AMM architecture, which makes use of pseudo arbitrage to adjust the bonding curve and reduce divergence loss. Given that large trades can lead to more significant divergence, this study introduces more diverse trading patterns with variable price fluctuations and varying trade sizes to simulate both small and large trades as well as rapid price changes to stress-test the system.

To ascertain the slippage loss, this study compares the difference between the expected trade execution price (based on a model's prediction or provided liquidity) and the actual trade execution price.

Results and discussion

This section discusses the (1) efficacy of the proposed model and then presents the (2) proposed architecture to be used for practical implementations.

Empirical results: capital efficiency, divergence loss, and slippage reduction

The results for capital efficiency, in terms of liquidity utilization, concentration, and depth, are presented as follows:

- Liquidity utilization:

  The proposed AMM architecture demonstrates enhanced capital efficiency, as evidenced by its significantly higher liquidity utilization rate compared with the baseline Uniswap architecture. The simulated liquidity utilizations for the baseline and proposed architectures are 56% and 93%, respectively (Figure 5). This indicates that, on average, 93% of the provided liquidity is effectively used to facilitate trades. This underlines the capital efficiency of the proposed architecture.

- Liquidity concentration:

  Next, this study simulates the liquidity distribution across a range of prices for both architectures. The results are presented in Figure 6. For the baseline Uniswap V3 AMM architecture, the liquidity is more evenly spread across the price range, ranging mostly between 40 and 60 units. For the proposed AMM architecture, the liquidity is highly concentrated around the current price (around 100 in our simulation), where we observe values close to 90–100 units. Away from the current price, the liquidity drops significantly, with values of mostly 10–20 units.

  Moreover, the baseline architecture exhibits a relatively uniform liquidity distribution across the entire price range. By contrast, the proposed architecture exhibits a pronounced peak around the current price, which indicates a high liquidity concentration at that point. Away from the current price, the liquidity significantly decreases, which is advantageous because it implies that less capital is "wasted" where it is not immediately required. The pronounced peak for the proposed AMM architecture signifies that it can handle larger trades around the current price with minimal slippage, thus enhancing capital efficiency. These results affirm the premise that the proposed AMM architecture offers better capital efficiency by concentrating liquidity around the most traded (or current) price levels. This provision of adaptive liquidity is crucial for AMMs to be effective and efficient in real-world trading scenarios.

- Liquidity depth:

  For both AMM architectures, the trade sizes that result in a price impact closest to a reasonable acceptable threshold of 1% are described as follows:

  A trade size of 1 unit results in the closest price impact to the acceptable threshold for the baseline Uniswap architecture, whereas a trade size of 100 units results in the closest price impact for the proposed AMM architecture. This indicates that the proposed architecture has a significantly deeper liquidity pool around the current price compared with the baseline. Specifically, the proposed AMM architecture can handle trades 100 times larger than the baseline before the price impact reaches 1%.

  Figure 7 visualizes the price impact for various trade sizes in both architectures:

  Furthermore, the baseline Uniswap V3 AMM architecture starts with a slightly higher price impact even for small trade sizes, and it exhibits a steeper increase in price impact as the trade size increases. By contrast, the proposed AMM architecture begins with a smaller price impact for lower trade sizes and maintains a more gradual increase in price impact as the trade size grows. At a trade size of 100 units, the price impact of the proposed architecture is still around the acceptable threshold, while the baseline architecture's price impact exceeds it with a much smaller trade size. These findings underscore the enhanced capital efficiency of the proposed AMM architecture, which can accommodate larger trades with minimal price impact. This is crucial for maintaining a stable and efficient market, especially during periods of high trading volume.

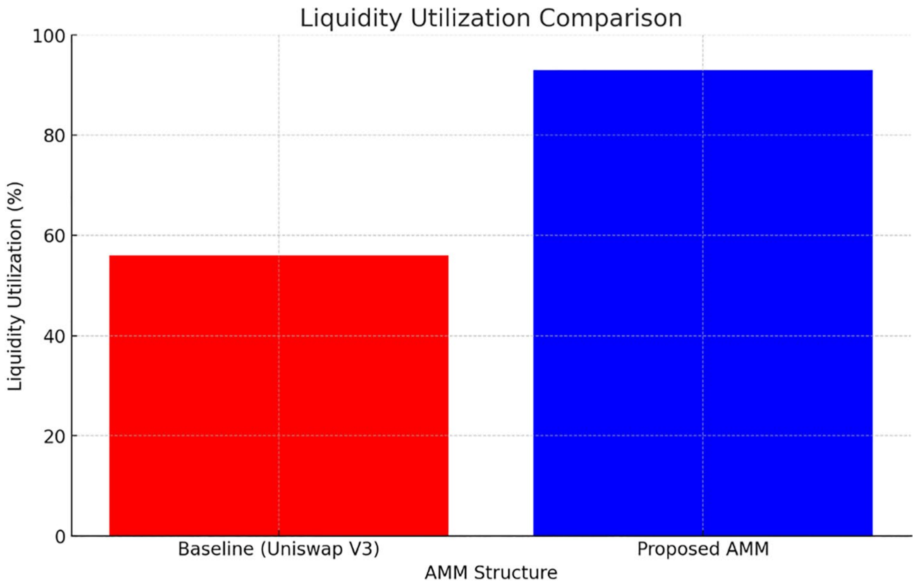


Fig. 5 Liquidity utilization comparison between the proposed AMM and baseline Uniswap V3 models. The bar chart provides a visual comparison of liquidity utilization between the baseline Uniswap V3 AMM (in red) and the proposed AMM structure (in blue)

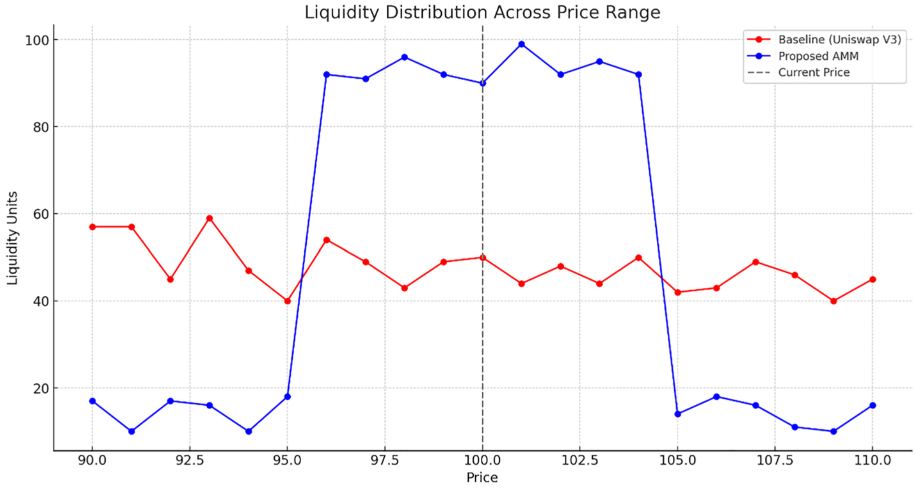


Fig. 6 Comparison of liquidity distribution across price range between the proposed AMM and baseline Uniswap V3 models. The graph illustrates the liquidity distribution across a range of prices for both the baseline Uniswap V3 AMM (in red) and the proposed AMM structure (in blue). The vertical dashed line represents the current price

Regarding divergence loss, the average losses are 1.465 units for the baseline Uniswap V3 AMM architecture and 0.482 units for the proposed AMM architecture (Fig. 8). The proposed architecture thus significantly reduces the average divergence loss compared with the baseline architecture. This demonstrates not only the efficacy of the proposed architecture's adaptive liquidity management but also its ability to handle diverse trading patterns with reduced divergence loss.

Regarding slippage loss, the average losses are 0.4779 units for the baseline architecture and 0.2389 units for the proposed AMM architecture (Fig. 9). The proposed architecture has a slippage approximately half that of the baseline architecture. Evidently, the proposed AMM architecture can potentially offer a considerable reduction in slippage for traders. This would translate into better trade execution prices, thus improving the overall efficiency of the AMM architecture and potentially attracting more traders and liquidity providers due to the enhanced trade quality.

In summary, the innovative design and predictive capabilities of the proposed AMM architecture offer considerable improvements in terms of divergence loss reduction, enhanced capital efficiency, and slippage reduction compared with the baseline Uniswap V3 AMM architecture. The proposed architecture could thus potentially directly address several of the challenges faced by existing AMM architectures.

Proposed architecture

The proposed architecture, depicted in Fig. 10, comprises the following protocol layers (Xu et al. 2021), which augment the deployment by Shrivastava (2022):

- Aggregator layer:
  This layer extends the application layer and is designed to create user-centric platforms to connect several protocols and applications. Thus, users can connect to multiple protocols and perform tasks, such as trading across services and comparing services.
- Application layer:
  The application layer comprises two components, namely the user interface and the blockchain layer interaction service:

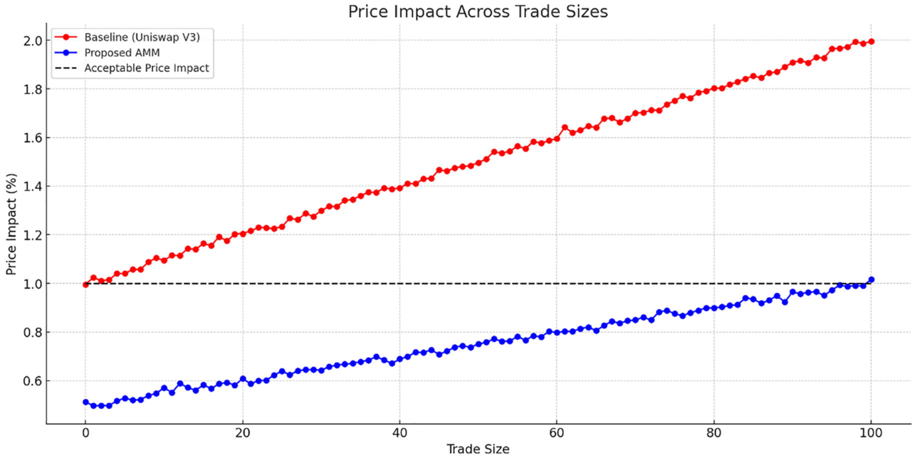


Fig. 7 Comparison of price impact across trade sizes between the proposed AMM and baseline Uniswap V3 models. The graph illustrates the price impact across different trade sizes for both the baseline Uniswap V3 AMM (in red) and the proposed AMM structure (in blue). The dashed line represents the acceptable price impact threshold of 1%

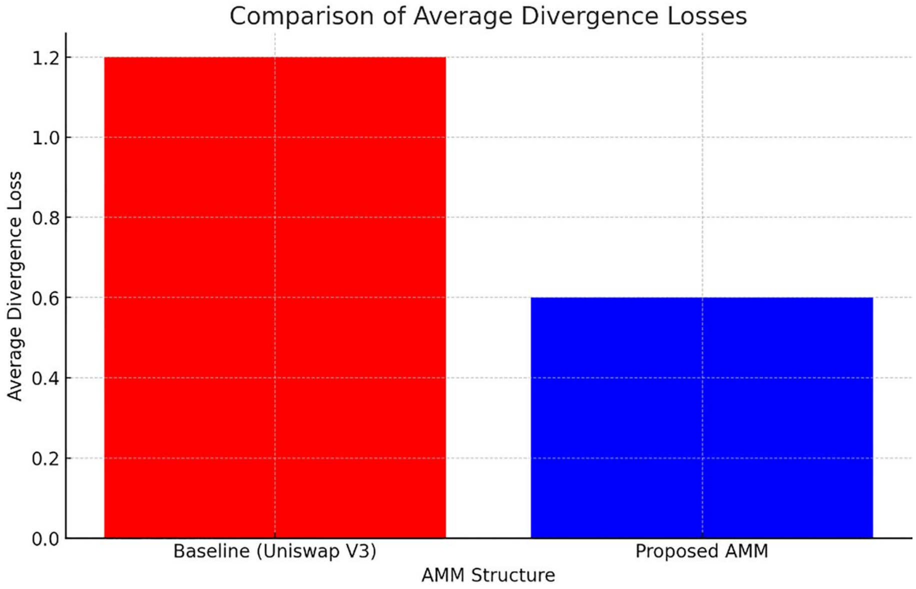


Fig. 8 Comparison of average divergence loss between the proposed AMM and baseline Uniswap V3 models. The graph illustrates the average divergence losses between the baseline Uniswap V3 (in blue) and the proposed AMM structure (in green). The proposed AMM structure demonstrated a significant reduction in divergence loss compared to the baseline, aligning with the findings

- *User interface:* This is designed to allow AMM users to interact with the various system functions provided by the AMM architecture. This is usually abstracted by a web browser or mobile app–based front end.
- *Blockchain layer interaction service:* This is designed to communicate and interact with the smart contract protocol layer. The interaction service allows function calls to be applied in the clearing house to perform specific actions that a liquidity provider or taker conducts.

- *Blockchain protocol layer:*

  This is the protocol layer for asset pooling and transaction settlement. The system logic is contained in the smart contracts deployed on a blockchain (e.g., Ethereum blockchain). A layer 2 solution may be implemented to allow off-chain transactions that can be rolled up to the layer 1 Ethereum main chain, thus lowering Ethereum gas fees and improving the processing rate.

- *Clearing house:* This is designed to securely execute trade positions and facilitate the deposit and returning of funds when called upon by a liquidity provider or taker. It is also responsible for the returning of details about the vault, price of the token pair, and token reserves in the AMM architecture.
- *Configurable virtual AMM (cAMM)*: This protocol allows the flexibility to adjust the token pair price ν' based on spot market prices ν_{obs} to minimize the expected load. Liquidity is also recalibrated based on ν'_p for the determination of the liquidity concentration range and the distribution of the incentive fee φ.
- *Vault*: This smart contract vault holds deposits securely for liquidity providers and takers.
- *Oracle*: This protocol allows the discovery of the spot price for a token pair.

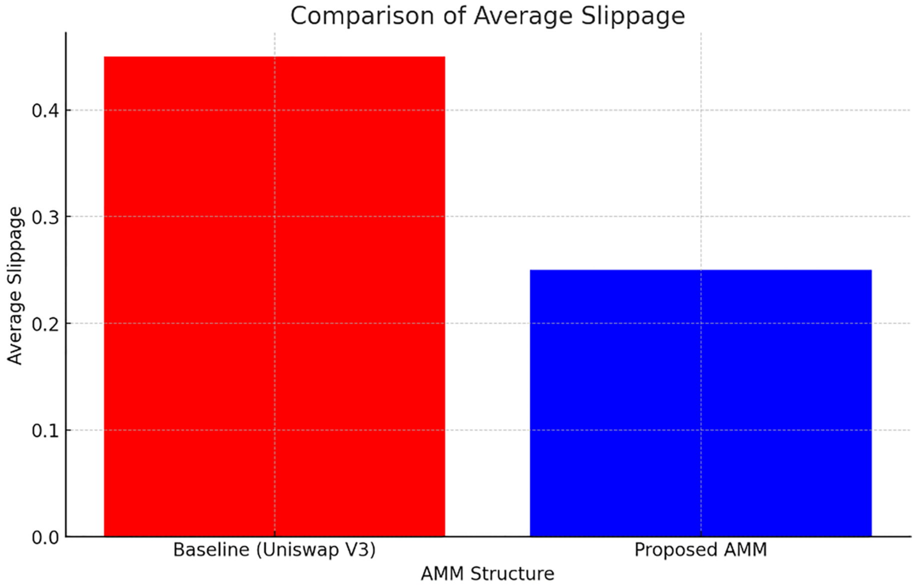


Fig. 9 Comparison of average slippage loss between the proposed AMM and baseline Uniswap V3 models. The proposed AMM structure (in green) demonstrates a significantly reduced average slippage when compared to the baseline Uniswap V3 structure (in red)

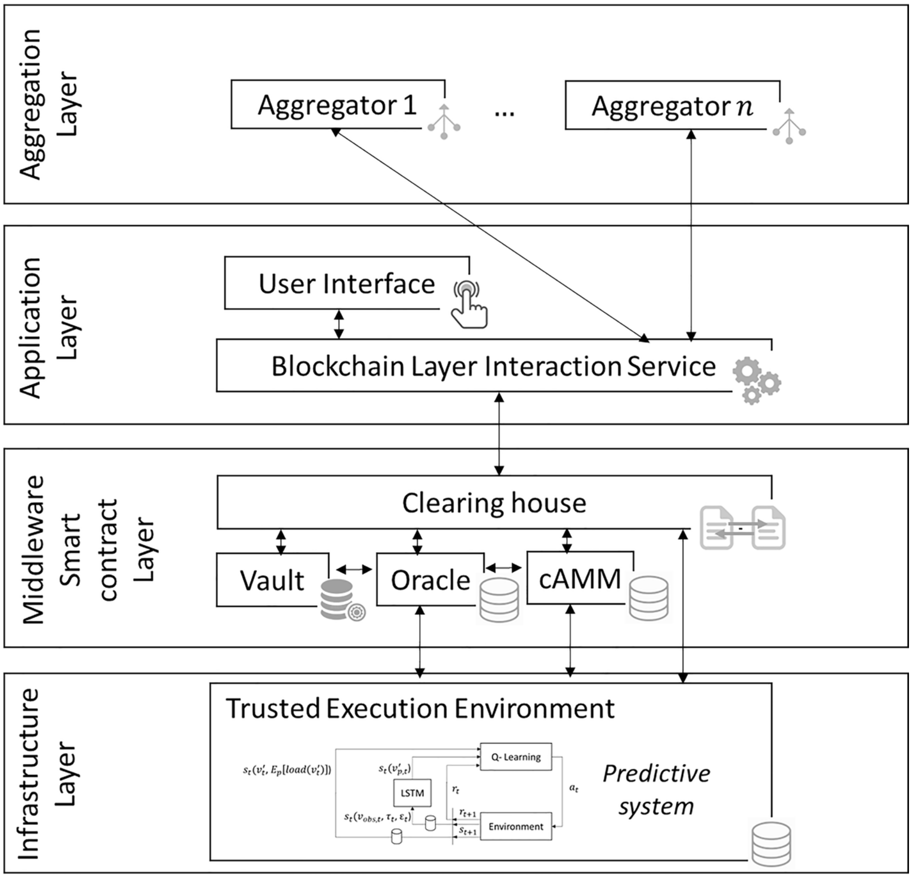


Fig. 10 Architecture layers of proposed AMM. The proposed architecture comprises the following protocol layers—Aggregation Layer, Application Layer, Middleware Smart Contract Layer, and Infrastructure Layer

- *Infrastructure layer:*
  This layer contains the trusted execution environment (TEE), which provides an enclave for secure intensive computation (e.g., the proposed LSTM–Q-learning reinforcement learning model), where external applications outside the enclave will not be able to interfere with the state or control flow shielded by the TEE (Pandl et al. 2020).
  - *TEE:* This physical server environment is designed to enable the computation of resource-intensive machine learning applications while preserving the integrity and security of data throughout. Through smart contracts, protocols can be designed to define policies on how data are shared. These policies may include requests for reward and differential privacy requirements (Hynes et al. 2018). As the deep reinforcement learning model is shielded by the smart contract and inference executions count toward the contract policies, this improves privacy against potential inference attacks, which aims to execute the predictive system to extract the model or underlying data (Cheng et al. 2019).

Interactions between key architecture layer components are summarized in Fig. 11.

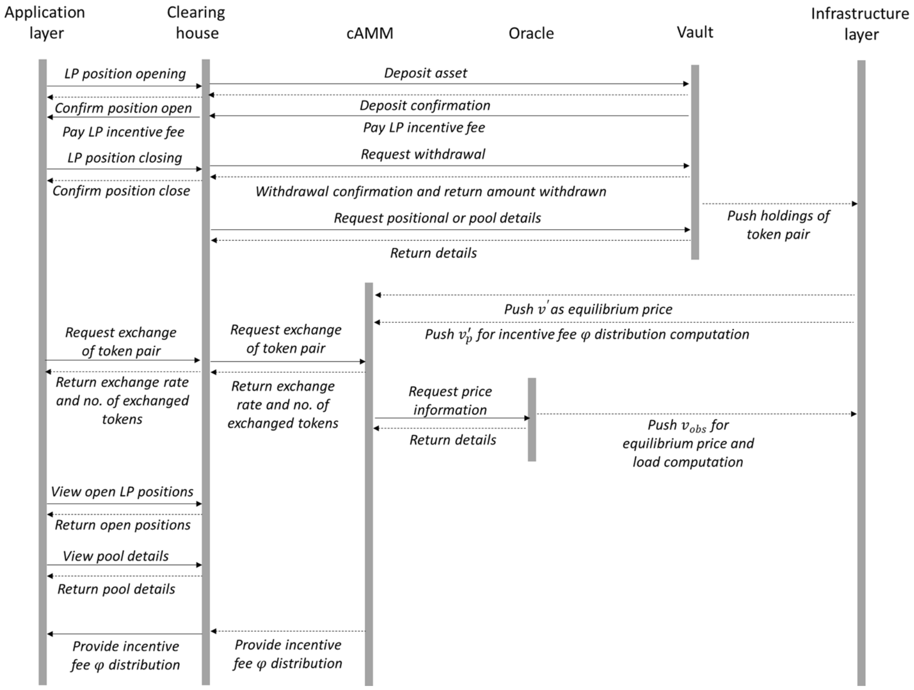

Application layer
Clearing house
cAMM
Oracle
Vault
Infrastructure layer
LP position opening
Deposit asset
Confirm position open
Deposit confirmation
Pay LP incentive fee
Pay LP incentive fee
LP position closing
Request withdrawal
Confirm position close
Withdrawal confirmation and return amount withdrawn
Request positional or pool details
Push holdings of token pair
Return details
Push v' as equilibrium price
Push v_p' for incentive fee φ distribution computation
Request exchange of token pair
Request exchange of token pair
Return exchange rate and no. of exchanged tokens
Return exchange rate and no. of exchanged tokens
Request price information
Return details
Push v_{obs} for equilibrium price and load computation
View open LP positions
Return open positions
View pool details
Return pool details
Provide incentive fee φ distribution
Provide incentive fee φ distribution

Fig. 11 Interaction between key architecture layer components of proposed AMM. Herein shows the key interactions between the Application Layer, Middleware Smart Contract Layer and Infrastructure Layer

Conclusion

AMM DEXs are a recent development in their early stages of growth. Present market solutions, while innovative in nature, can be further optimized.

This study introduces a predictive AMM architecture that utilizes a loss-minimizing market pricing mechanism as well as a deep reinforcement learning architecture that seeks to reduce divergence and slippage costs. The objective is to enhance the capital efficiency of liquidity provision. This paper formalizes and analytically expounds the implicit costs to a liquidity taker and provider, as well as the deep reinforcement learning mechanism for market making, thus benefiting both research and industry uses.

The empirical analysis based on the experimental simulations showcases the proposed AMM architecture's superiority in several key metrics. With a 93% liquidity utilization rate, a pronounced liquidity concentration around current prices, and a deeper liquidity pool, the new architecture outperforms the baseline Uniswap V3 AMM architecture. Furthermore, it demonstrates a significant reduction in both divergence and slippage losses. These findings underscore the potential of the proposed AMM architecture to revolutionize the DeFi space by directly addressing the challenges inherent to existing AMM architectures. The results pave the way for further refinements and real-world implementations of the proposed framework, promising a more efficient and effective DeFi trading landscape.

While our research offers a deep dive into the technical aspects of AMMs, it also bridges the gap between theory and practice, offering clear implications across various domains. It thus offers profound implications for both academia and the industry at large. We highlight these implications across various spectrums as follows:

- *Academic implications:* Our approach to enhancing liquidity provision in AMMs can serve as a foundation for future research, encouraging explorations into deeper integrations of machine learning in DeFi. The proposed architecture and its novel market equilibrium pricing mechanism also open avenues for studies into optimizing other related financial protocols, which could provide innovative solutions to existing problems.
- *Practical implications:* For practitioners in the DeFi domain, our findings offer actionable insights into optimizing AMM performance, reducing divergence, and minimizing slippage losses. Decision-makers in financial institutions can harness the findings to refine their trading strategies, thereby optimizing liquidity provision and maximizing returns. Managers who oversee AMM platforms can implement the proposed architecture to enhance the efficiency of trading platforms as a differentiator in an increasingly competitive market.
- *Policy and societal implications:* As DeFi grows in prominence, policymakers can use our research as a basis to frame regulations that foster innovation while ensuring market stability. At a societal level, improved AMM architectures can lead to more efficient and transparent financial transactions, which will promote trust and the wider adoption of decentralized platforms.

An AMM DEX liquidity provision optimization strategy is an attractive topic for both practitioners and researchers. In terms of future work, practitioners could include a physical implementation of the proposed AMM DEX grounded in a profitable business model, aiming to bridge the gap between theoretical innovation and practical utility. This includes simplifying the presentation of the proposed model to enhance its accessibility to a broader audience. Future work can enrich the study's endeavors with practical examples, case studies, and—potentially—the development of a software toolkit designed to lower the barriers to adoption. The goal would be to transform the research into valuable, actionable resources for industry application.

For researchers, future work could involve the following avenues: (1) conducting evaluations of a broad spectrum of preprocessed alternative data to enhance the precision of market liquidity predictions; (2) evaluating incentive fee distribution structures beyond the Uniswap V3 uniform distribution and the Gaussian distribution mechanisms proposed in this paper; (3) investigating an integrated TEE architecture within the infrastructure layer, augmented with relevant specific security protocols; (4) exploring the integration of the predictive AMM model with layer 2 solutions and assessing its performance and feasibility in a scaled environment; (5) expanding the empirical testing framework by including rigorous stress testing against a wide variety of market scenarios (including those marked by extreme volatilities); (6) improving the proposed hybrid LSTM–Q-learning reinforcement learning framework to enhance predictions of liquidity concentration ranges; (7) modeling and learning complex relations in transaction networks within AMMs (Zhao et al. 2023a; Zhang and Kou 2022); and (8) performing representation learning of transaction networks within AMMs, thereby enabling more accurate and dynamic predictions of liquidity and price movements, such as through the use of graph-based dual attention networks (Zhao et al. 2023b) or contrastive learning (Chen and Kou 2023).

Through these multifaceted future research initiatives, practitioners and researchers can bridge the gap between theoretical innovation and practical utility, providing contributions that not only enhance the academic discourse but also deliver significant, measurable benefits to the DeFi space.

Abbreviations

AMM	Automated market maker
cAMM	Configurable virtual automated market maker
CEX	Centralized exchange
CFMM	Constant function market maker
CPMM	Constant product market maker
DD-DQN	Dueling double deep Q-learning network
D-DQN	Double deep Q-network
DeFi	Decentralised finance
DEX	Decentralized exchange
DQN	Deep Q-network
LOB	Limit order book
LSTM	Long short-term memory
ReLU	Rectified linear unit
RNN	Recurrent neural network
TEE	Trusted execution environment

Acknowledgements
Not applicable.

Author contributions
Author is the sole author of the publication. Author has read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The author declare that he has no competing interests.

Received: 17 May 2023 Accepted: 2 August 2024
Published online: 12 September 2024

References

Abraham J, Higdon D, Nelson J, Ibarra J (2018) Cryptocurrency price prediction using tweet volumes and sentiment analysis. SMU Data Sci Rev 1(3):1

Angeris G, Chitra T (2020) Improved price oracles: Constant function market makers. In Proceedings of the 2nd ACM conference on advances in financial technologies, pp 80–91

Aoyagi J (2020) Lazy liquidity in automated market making. SSRN Electron J. https://doi.org/10.2139/ssrn.3674178

Auer R, Haslhofer B, Kitzler S, Saggese P, Victor F (2023) The Technology of Decentralized Finance (DeFi). BIS Working Papers, No. 1066. Bank for International Settlements. Retrieved: https://www.bis.org/publ/work1066.pdf [Accessed 17 June 2024]

Balancer (2022) Retrieved: https://balancer.fi/ Accessed 17 June 2024]

Bar-On Y, Mansour Y (2023) Uniswap liquidity provision: an online learning approach. arXiv preprint arXiv:2302.00610

Cartea A, Drissi F, Monga M (2022) Decentralised finance and automated market making: predictable loss and optimal liquidity provision. SSRN Electron J. https://doi.org/10.2139/ssrn.4273989

Chan NT, Shelton C (2001) An electronic market-maker. Technical Report AIMEMO 2001–005. MIT, AI Lab

Chen J, Kou G (2023) Attribute and structure preserving graph contrastive learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 37, no 6. Washington, DC, pp 7024–7032

Cheng R, Zhang F, Kos J, He W, Hynes N, Johnson N, et al (2019) Ekiden: a platform for confidentiality-preserving, trustworthy, and performant smart contracts. In: 2019 IEEE European symposium on security and privacy (EuroS&P), IEEE, pp 185–200

Curve (2022) Retrieved: https://curve.fi/ [Accessed 17 June 2024]

Dune Analytics (2023) DEX Tracker - Decentralized Exchanges Trading Volume. Retrieved: https://defiprime.com/dex-volume [Accessed 10 October 2023]

Engel D, Herlihy M (2021a) Composing networks of automated market makers. In: Proceedings of the 3rd ACM conference on advances in financial technologies, pp 15–28

Engel D, Herlihy M (2021b) Presentation and Publication: Loss and Slippage in Networks of Automated Market Makers. arXiv preprint arXiv:2110.09872

Fritsch R (2021) Concentrated liquidity in automated market makers. In: Proceedings of the 2021 ACM CCS workshop on decentralized finance and security, pp 15–20

Frontier Research (2023) Designing a DEX in 2023: Addressing MEV, AMMs, and Liquidity [Video]. YouTube. Retrieved: https://www.youtube.com/watch?v=6K-QX6HtoiM [Accessed 17 June 2024]

Ghosh B, Kazouz H, Umar Z (2023) Do automated market makers in DeFi ecosystem exhibit time-varying connectedness during stressed events? J Risk Financ Manag 16(5):259

Haider A, Wang H, Scotney B, Hawe G (2022) Predictive market making via machine learning. In: Operations Research Forum, vol 3, no 1 Springer, pp 1–21

Hambly B, Xu R, Yang H (2021) Recent advances in reinforcement learning in finance. arXiv preprint arXiv:2112.04553

Heimbach L, Schertenleib E, Wattenhofer R (2022) Risks and Returns of Uniswap V3 Liquidity Providers. arXiv preprint arXiv:2205.08904

Hu Z, Zhao Y, Khushi M (2021) A survey of forex and stock price prediction using deep learning. Appl Syst Innov 4(1):9

Hynes N, Cheng R, Song D (2018) Efficient deep learning on multi-source private data. arXiv preprint arXiv:1807.06689

Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of the 3rd international conference of learning representations (ICLR'15). San Diego, CA, pp 1–15

Kraaijeveld O, De Smedt J (2020) The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. J Int Finan Mark Inst Money 65:101188

Lehar A, Parlour CA (2021) Decentralized exchanges. Working paper

Liu C (2020) Deep reinforcement learning and electronic market making (Doctoral dissertation, Imperial College London)

Lucarelli G, Borrotti M (2019) A deep reinforcement learning approach for automated cryptocurrency trading. In: IFIP international conference on artificial intelligence applications and innovations. Springer, Cham, pp 247–258

Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th international conference on machine learning. Atlanta, GA, pp 1–6

Malinova K, Park A (2024) Learning from DeFi: Would automated market makers improve equity trading? SSRN Electron J. https://doi.org/10.2139/ssrn.4531670

Meyer E, Welpe IM, Sandner P (2022) Decentralized finance—a systematic literature review and research directions. In: ECIS 2022 research papers 25. [Accessed 17 June 2024]

Mohan V (2022) Automated market makers and decentralized exchanges: a DeFi primer. Financ Innov. https://doi.org/10.1186/s40854-021-00314-5

Moosavi M, Clark J (2021) Lissy: experimenting with on-chain order books. arXiv preprint arXiv:2101.06291

Neuder M, Rao R, Moroz DJ, Parkes DC (2021) Strategic liquidity provision in Uniswap V3. arXiv preprint arXiv:2106.12033

Pandl KD, Thiebes S, Schmidt-Kraepelin M, Sunyaev A (2020) On the convergence of artificial intelligence and distributed ledger technology: a scoping review and future research agenda. IEEE Access 8:57075–57095

Park A (2022) Conceptual flaws of decentralized automated market making. Working paper, University of Toronto

Phan C (2024) Decentralized exchanges: current limitations of AMM models & exploring the future of DEX mechanics. The Tie Research. Retrieved: https://www.thetie.io/insights/research/decentralized-exchanges-current-limitations/ [Accessed 17 June 2024]

Pourpouneh M, Nielsen K, Ross O (2020) Automated Market Makers. IFROWorking Paper 2020/08. University of Copenhagen, Department of Food and Resource Economics. Retrieved: https://www.econstor.eu/bitstream/10419/222424/1/IFRO_WP_2020_08.pdf [Accessed 17 June 2024]

Rundo F (2019) Deep LSTM with reinforcement learning layer for financial trend prediction in FX high frequency trading systems. Appl Sci 9(20):4460

Sabate-Vidales M, Šiška D (2022) The case for variable fees in constant product markets: an agent based simulation. In: International conference on financial cryptography and data security. Springer, Cham, pp 225–237

Sadighian J (2019) Deep reinforcement learning in cryptocurrency market making. arXiv preprint arXiv:1911.08647

Sadighian J (2020) Extending deep reinforcement learning frameworks in cryptocurrency market making. arXiv preprint arXiv:2004.06985

Schär F (2020) Decentralized finance: on blockchain- and smart contract-based financial markets. SSRN Electron J. https://doi.org/10.2139/ssrn.3571335 [Accessed 17 June 2024]

Schmitt M (2023) The next steps in DEX design. Frontier Research. Retrieved: https://www.youtube.com/watch?v=6K-QX6HtoiM [Accessed 17 June 2024]

Selser M, Kreiner J, Maurette M (2021) Optimal market making by reinforcement learning. arXiv preprint arXiv:2104.04036

Shrivastava AK (2022) Dynamic virtual automated market makers and their limitations in decentralized finance. (Master dissertation, Trinity College Dublin)

Singh SF, Michalopoulos P, Veneris A (2023) DEEPER: enhancing liquidity in concentrated liquidity AMM DEX via sharing. In: 2023 IEEE international conference on blockchain and cryptocurrency (ICBC). IEEE, pp 1–7

Spooner T, Fearnley J, Savani R, Koukorinis A (2018). Market making via reinforcement learning. arXiv preprint arXiv:1804.04216

Sun T, Huang D, Yu J (2022) Market making strategy optimization via deep reinforcement learning. IEEE Access 10:9085–9093

Sushiswap (2022) Retrieved: https://sushi.com/ [Accessed 17 June 2024].

Uniswap (2022) Retrieved: https://uniswap.org/ [Accessed 17 June 2024].

Xu J, Paruch K, Cousaert S, Feng Y (2021) SoK: decentralized exchanges (DEX) with automated market maker (AMM) protocols. arXiv preprint arXiv:2103.12732

Zhang H, Kou G (2022) Role-based multiplex network embedding. In: Proceedings of the 39th international conference on machine learning. Baltimore, Maryland, PMLR, pp 26265–26280

Zhang H, Chen X, Yang LF (2023) Adaptive liquidity provision in uniswap v3 with deep reinforcement learning. arXiv preprint arXiv:2309.10129

Zhao Y, Du H, Liu Y, Wei S, Chen X, Zhuang F, Li Q, Kou G (2023a) Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. In: IEEE transactions on knowledge and data engineering, vol 35, no 8, pp 8559–8571

Zhao Y, Wei S, Du H, Chen X, Li Q, Zhuang F, Liu J, Kou G (2023b) Learning bi-typed multi-relational heterogeneous graph via dual hierarchical attention networks. In: IEEE transactions on knowledge and data engineering, vol 35, no 9, pp 9054–9066

Publisher's Note