Article

Not peer-reviewed version

ESS-LP: An Effective Slippage Scheme Based on Liquidity Pools for Data Trading

Huayou Si , Mengyang Li , Yuanyuan Qi * , Neal N. Xiong * , Wei Chen , Loc Nguyen The , Shichong Wang

Posted Date: 1 May 2026

doi: 10.20944/preprints202604.2221.v1

Keywords: data trading; automatic market maker; slippage optimization; trading systems

doi:10.20944/preprints202604.2221.v1

Article

ESS-LP: An Effective Slippage Scheme Based on Liquidity Pools for Data Trading

Huayou Si [1], Mengyang Li [1], Yuanyuan Qi [1,*], Neal N. Xiong [2,*], Wei Chen [1], Loc Nguyen The [3] and Shichong Wang [1]

1 The School of Infomation Engineering, China Jiliang University, Hangzhou 310018, China
2 The Dept. of Computer Science, Southern New Hampshire University, Manchester, NH, USA
3 The School of Information and Communication Technology, Hanoi University of Science and Technology
* Correspondence: qiyuanyuan@cjlu.edu.cn (Y.Q.); xiongnaixue@gmail.com (N.N.X.)

Abstract

This paper proposes a decentralized data trading approach based on the Automated Market Maker (AMM) mechanism, aiming to break through the bottlenecks in data trading efficiency and fairness via the collaborative innovation of market-oriented pricing mechanisms and automated trading processes. We focus on constructing an automatic pricing and matching mechanism based on liquidity pools. Subsequently, mathematical modeling and simulations reveal slippage generation mechanisms in data liquidity pools under trading shocks and imbalances. To address these issues, a novel dual slippage optimization mechanism integrating dynamic trade splitting and alternating order sorting is proposed, which decomposes orders into sub-orders and reorganizes their timing, establishing a dynamic equilibrium model. Experiments show the method reduces average slippage amplitude from 2.1% to 0.5% and representing a 76.2% reduction, significantly enhancing price stability and market efficiency. This research explores the mechanism of applying AMM to data asset trading and overcomes AMM's limitations, providing a theoretical and empirical foundation for building low-cost, high-fairness data trading systems through mechanism innovation and technical optimization.

Keywords: data trading; automatic market maker; slippage optimization; trading systems

---

1. Introduction

In recent years, the digital economy has grown at an unprecedented scale and speed. It has become a driving force in reshaping the global economic structure and competitive landscape. Data is gradually transformed from a supportive resource to a core resource driving production and innovation. The efficient circulation and value realization of data increasingly rely on standardized and orderly trading mechanisms, and the establishment of a data assets market allocation system has become a critical pathway for advancing the digital economy. To facilitate data circulation, various data trading platforms have been successively established both domestically and internationally [1], such as the domestic Guiyang Global Big Data Trading Platform, the Central China Big Data Trading Platform, and the Shanghai Data Trading Platform, etc., alongside the international data trading platforms such as Factual, BDEX, and Data plaza [2]. These platforms aim to bridge data suppliers and customers. While data exchanges promote cross-hierarchical data circulation across society, their centralized architectures frequently manifest single point of failure vulnerabilities. Persistent challenges include security risks from centralized data storage, suboptimal pricing mechanisms, and inefficient matching capabilities, collectively rendering these platforms inadequate to satisfy the flexible, real-time, and high-frequency trading requirements characteristic of data assets markets.

Decentralized digital currency trading platforms based on the AMM model, such as Bancor [3], Uniswap [4], Curve [5], etc., have been widely applied in blockchain financial systems recently. These platforms utilize smart contracts on the blockchain to enable liquidity pool trading, allowing users to complete trades in a decentralized, trustless way that partially resolves issues inherent in traditional exchanges. The AMM mechanism relies on liquidity pool and predefined mathematical functions for dynamic pricing, facilitating decentralized trading without order matching, which provides a novel idea to overcome the liquidity bottleneck and inefficient matching in centralized platforms. Although the mechanism has been validated in homogeneous financial assets, its applicability to data asset trading remains under-explored. Few existing studies have introduced the AMM mechanism into data trading, and the non-homogeneity, non-standardization and sparsity of data assets pose new modeling and trading challenges. However, the trading slippage problem is particularly prominent under the liquidity pool-based trading mechanism. Slippage typically denotes the discrepancy between executed and expected prices [6], influenced by factors such as trade volume, market depth, and supply-demand dynamics. Data assets exhibit strong time-sensitivity and uniqueness [7], with slippage amplification frequently occurring during block trades or asymmetric trades. This phenomenon significantly increases trading costs and undermines market fairness.

Existing research on slippage control mainly focuses on financial trading scenarios, typically employing mitigation strategies such as increasing market depth, limiting trading volume, optimizing trading strategies for dynamic curve market makers, or introducing price oracles, e.g., Bancor, Uniswap, etc. However, these methods mainly rely on the structure of high-frequency and highly symmetric financial assets, and it is difficult to adapt them to the highly heterogeneous, low-frequency and supply-demand disequilibrium characteristics of data trading. The new mechanistic causes underlying slippage in data trading scenarios still lack systematic research and effective solutions. Within the domain of data trading platforms, some studies have attempted to optimize the price discovery and data quality assessment mechanisms, such as automatically negotiating pricing through smart contracts, supply-demand matching based on data quality priorities [8], fair exchange agreements incentivizing honest behavior [9] and privacy protection based zero-knowledge proofs [10], yet a comprehensive slippage analysis framework and control mechanism remain unrealized. Crucially, the precise optimization of trade execution pathways in response to real-time platform liquidity conditions, particularly within dynamic trading environments, still constitutes an unexplored research frontier.

To tackle these challenges, this paper proposes a decentralized data trading approach based on the AMM mechanism, pioneering the application of the AMM mechanism to heterogeneous data asset trading, breaking through the bottlenecks in data trading efficiency and fairness via the collaborative innovation of market-oriented pricing mechanisms and automated trading processes. In response to the slippage issues, we propose a novel dual slippage optimization mechanism integrating dynamic trade splitting and alternating order sorting to effectively alleviate block trade impacts and matching disequilibrium. The following are the main contributions of this paper:

- Proposes a decentralized data trading approach based on the AMM mechanism, aiming to break through the bottlenecks in data trading efficiency and fairness via the collaborative innovation of market-oriented pricing mechanisms and automated trading processes. This establishes a novel decentralized trading paradigm for data assets, extending the application boundaries of the Decentralized Finance (DeFi) key mechanism to non-financial trading scenarios.
- Innovatively constructs an automated pricing and trading matching mechanism based on a liquidity pool. Through mathematical modeling and simulation experiments, it quantitatively analyzes the generation mechanism of slippage in data liquidity pools when coping with transaction shocks and buy-sell disequilibrium. The findings reveal that liquidity shocks induced by block trades and temporal mismatches arising from buy-sell disequilibrium constitute the primary root causes of pronounced slippage volatility in data trading.
- Innovatively proposes a dual slippage optimization mechanism integrating dynamic trade splitting and alternating order sorting, featuring a liquidity pool state-aware dynamic trade splitting and alternating order sorting execution engine. This achieves stable control of slippage in block trades scenarios.

The rest of the paper is organized as follows: Section 2 introduces and discusses related work in the data trading field. In Section 3, we propose a data trading model based on liquidity pools and analyze the factors influencing trading slippage. Building upon this analysis, we introduce our slippage optimization methodology for liquidity pool-based data trading. In Section 4, we evaluate the efficacy of the proposed slippage optimization approach through comprehensive experiments. Finally, Section 5 presents the principal conclusions and research contributions.

2. Related Work

2.1. Development of Data Trading Market

Currently, most domestic and international data trading platforms adopt a centralized architecture, facilitating trades between data buyers and sellers. While these platforms improve the efficiency of data resource aggregation and sharing, they commonly rely on a high degree of trust in the platform, are plagued by structural deficiencies such as the absence of pricing mechanisms and low negotiation efficiency, and fail to meet the market-oriented allocation needs of data assets. For example, China's Zhongguancun Data Sea Big Data Trading Platform and the international platform RapidAPI [11] both utilize service brokerage or API interface authorization to circumvent data storage issues, yet they still face fundamental challenges like a lack of pricing mechanism flexibility. Some platforms offer data packaging and authorized distribution services, standardizing raw datasets into structured commodities with clear usage rights, allowing buyers to acquire data packages under specific license terms [12]. Other platforms act solely as intermediaries of information, facilitating discovery and trade matching between data buyers and sellers without directly storing or processing data. Subscription-based trading models are also prevalent in data markets, where buyers continuously receive updated data streams through regular payments. Crowdsourcing market models aggregate user-contributed data assets and centrally sell them to interested enterprise clients [13,14]. Despite the variations in their implementation, these trading models fundamentally depend on a centralized architecture. Consequently, they are inherently susceptible to a range of issues, including data leakage, single points of failure, opaque trading processes, and a high reliance on trust among participants. These limitations have prompted researchers to explore more open and decentralized data trading mechanisms.

Recently, the academic community has extensively explored decentralized data trading. The emergence of blockchain decentralized technology has effectively addressed issues such as data theft by trading platforms and single points of failure [15–17]. For example, Li et al. [18] applied blockchain technology to virtual power plant trades, verifying the feasibility of decentralized dispatch and smart contracts. Dai et al. [19] proposed a data secure trading method based on blockchain, which prevents both the trading platform and data buyers from obtaining the original data of sellers, only allowing access to the analysis results of the required data, thereby ensuring the security of original data. In [8], Hu et al. designed a blockchain-based big data trading system architecture called DataTBC. In DataTBC, data requestors publish data demands via contracts, and data providers can respond to these demands. During the data matching process completed by the contract, high-quality data has a higher priority when participating in data demands. Furthermore, trading parties complete the price negotiation process through smart contracts, and data rewards are automatically distributed according to the negotiated price. In [9], Jiang et al. proposed a blockchain-based data trading system called BDTS, implementing a fair exchange protocol where benevolent behavior of traders is rewarded, and dishonest behavior is punished. This encourages rational traders to faithfully participate in data trading activities to obtain maximum incentive benefits. In this research, Jiang et al. also analyzed the strategies of sellers, buyers, and dealers in trades, pointing out that if every participant is honest and rational, the trading market can reach a Nash equilibrium, ensuring the maximization of traders' interests. In [10], Xue et al. proposed a fair and accurate blockchain-based medical data trading scheme. Using zero-knowledge proofs and encryption techniques, data sellers can sell partial data without disclosing personal information, and data buyers can only obtain the necessary data, thus ensuring fairness and privacy protection in trades.

The aforementioned methods primarily focus on ensuring fairness, protecting privacy, and designing incentive mechanisms in the process of data trading. These methods alleviate problems such as data leakage, information asymmetry, and dishonest trading behavior to some extent. However, these methods generally suffer from a lack of practicality, relying on fixed rules or static negotiation processes that fail to meet the market-oriented allocation needs of data assets. Especially in scenarios with block trades, high frequencies, or strong heterogeneity among participants, they often exhibit limitations such as low matching efficiency and long response delays. At the same time, current mechanisms largely lack dynamic pricing capabilities based on market conditions, making them unable to respond sensitively to real-time supply and demand relationships, which can lead to pricing deviations and market disequilibrium. In this context, there is an urgent need to introduce decentralized trading mechanisms with automatic pricing and continuous matching capabilities to enhance the flexibility and efficiency of data trading systems. The AMM model, due to its mechanistic characteristics, offers a new possible path to address these issues.

2.2. AMM Mechanism

The market maker mechanism originated in the stock and foreign exchange markets, aiming to maintain market liquidity by providing buy and sell quotes. As trading scales expanded and markets became high-frequency, the traditional manual quoting methods gradually failed to meet efficiency demands, leading to the emergence of algorithmic market making. AMMs [20,21] as representative mechanisms, were initially used in prediction markets to mitigate pricing distortions and insufficient liquidity caused by irrational participants. The core features of AMMs lie in their ability to achieve continuous liquidity provision and automatic dynamic pricing without order book matching, through predefined pricing functions and liquidity pool mechanisms, thus significantly simplifying the trading execution process. A typical example is the AMM model built on the Logarithmic Market Scoring Rule [22], which not only controls maximum losses but also maintains continuous trading capability under any market conditions, demonstrating high stability and flexibility.

In recent years, with the popularity of cryptocurrencies and the growing demand for decentralized exchanges, AMMs have again become more important as a viable alternative to centralized order books. In the AMM model, liquidity providers supply single or multiple types of tokens to designated liquidity pools, and traders trade based on these token pools rather than relying on traditional order matching. The liquidity pools in these AMM models track predefined mathematical functions, which dynamically adjust token prices based on the supply and demand of assets in the pool, thereby determining the provision of various token types to traders in exchange for a certain quantity of another token. Compared to traditional order book models, curve-based AMMs provide a continuous supply of liquidity [23]. Additionally, depending on the mathematical function used, trading prices can better reflect market supply and demand [24]. To meet the various trading demands and asset characteristics, various AMM models have been developed in the industry. As shown in [3], Bancor implemented an AMM-based DEX called Bonding Curve, which provides continuous liquidity for buying and selling individual tokens. In a curve-based AMM, any two currencies can be traded directly with each other. The common curve is a mathematical function of the token supply used to calculate the token price, satisfying the law of supply and demand and ensuring unlimited liquidity. Drawing on solutions from the prediction market model, Buterin [25] proposed an AMM for decentralized exchanges, with the constant product curve presented in Eq. (1):

$$x \times y = k \tag{1}$$

where x and y represent the total supply of the two tokens in the liquidity pool, and k is a constant. Subsequently, as shown in [4], Adams et al. created Uniswap based on this. Since then, a series of constant function AMM variants have emerged. For example, as presented in [5], Curve combines constant sum and constant product AMMs for stablecoin exchange. Balancer [26] uses a constant geometric mean AMM, which is the first DEX to allow liquidity pools with more than two tokens.

Liquidity pools are a core component of an AMM and an innovative market structure in DeFi that allows assets to be traded freely without the need for traditional market makers. By constructing homogeneous liquidity pools, they provide dynamic pricing and liquidity support for both trading parties, and automatically adjust asset prices through algorithms, maintaining equilibrium asset ratios and improving data trading efficiency. Regarding research on liquidity pools, scholars have proposed a series of new methods and mechanisms to address market liquidity issues and malicious trading. Othman et al. [27] pointed out that traditional market makers, unable to adapt to liquidity changes, caused their trading prices to fluctuate similarly in high-volume and low-volume markets, and their trades were always at a loss in typical markets. Therefore, a liquidity-sensitive market maker was constructed based on a variant of LMSR. This market maker generates bounded losses at any initial liquidity level, and in the worst case, its loss is zero even when the initial liquidity level is close to zero. Thus, for any initial liquidity level, a boundary can be established in the market state space, and if the market terminates within this boundary, the market maker can achieve profitability. Mohan et al. [28] proposed a neoclassical black-box-based decentralized AMM algorithm analysis method. Their method can not only visually analyze the geometric shape of a given AMM algorithm but also examine the differences caused by manipulating a given exchange function. For instance, analysis shows that any monotonic transformation of the exchange function in the Constant Product Market Maker (CPMM) does not change a given reserve price. Jiang et al. [29] proposed a Dynamic Curve Automated Market Maker (DCAMM) model that uses a Price Oracle to track market prices in real time and dynamically adjust the price curve. These models are used to solve arbitrage and sandwich attack problems in traditional AMM models and ensure trading fairness.

Despite the superior performance of AMM mechanisms in the digital asset domain, they rely on assumptions such as asset standardization, high-frequency trading, and strong symmetry, making them difficult to adapt to scenarios involving complex, illiquid, and heterogeneous data assets. This also means that the exploration of the AMM mechanism in the data trading field is still in a preliminary stage, and the adaptability of its key mechanisms and risk control issues require further in-depth research.

2.3. Slippage in Data Trading

Slippage is one of the main factors prompting the proposal and adoption of different AMMs, and it is directly related to the traders' losses. Slippage is the difference between the expected trading execution price and the actual trading execution price [6]. AMMs exhibit highly volatile, complex, and dynamic issues similar to futures prices [30]. The asset exchange price users see during exchanges often deviates from the actual exchange price, and slippage is commonly used to measure this difference. Slippage occurs whenever the token price changes during a trade. In liquidity pool-based data trading, the slippage problem is particularly prominent, especially with block trades or insufficient liquidity. For instance, in the constant product model, the larger the block trade, the more significant the impact on the asset ratio in the pool, resulting in price deviations from expectations and increased slippage [31].

In Bancor, Bancor v2 relies on the Chainlink Price Oracle to reduce divergence losses caused by arbitrage, and dynamically updates the pricing curve of the liquidity pool, allowing the asset price in AMM liquidity pool track its market price, significantly reducing slippage within a specific price range by tracking market prices [3]. In Uniswap, similar to Bancor, prices in the liquidity pool are unstable, and trading volumes related to the liquidity pool size affect the liquidity pool price to varying degrees, larger block trades may result in greater slippage [4]. A Constant-sum curve has zero slippage, but due to its fixed price and limited liquidity, and the ease with which one of their pooled assets can be exhausted, it is only suitable for stablecoins and has mainly theoretical significance [32]. Furthermore, Wang et al. [33] proposed a dynamic curve AMM model to derive optimal trading strategies for dynamic curve market makers to minimize slippage.

 doi:10.20944/preprints202604.2221.v1

StableSwap/Curve [34] introduced a hybrid invariant that allows trading on a constant-sum curve when the portfolio is relatively equilibrium and switches to a constant-product curve when disequilibrium. This design achieves lower slippage, but is only suitable for stablecoins because the price of the desired trading range remains close to 1. Wang et al. [35] proposed the Constant Elliptic Curve AMM, whose general form is shown in Eq. (2):

$$(x-a)^2+(y-a)^2+b\times xy=C \tag{2}$$

where a b and C are constants. Traders can choose between concave and convex curves in the first quadrant. Compared with traditional AMM models, the proposed constant elliptical curve supports a fixed price interval; thus, the slippage range is also bounded.

With the rapid development of deep learning technology, an increasing number of fields are leveraging this advanced tool to optimize solutions to complex problems [36,37], especially in data fusion [38,39] and prediction models [40–42]. Lim et al. [43] proposed a predictive AMM architecture based on deep reinforcement learning. By combining deep learning algorithms and market dynamics prediction, this architecture can dynamically adjust the price curve during trading to reduce slippage losses, while improving the stability of liquidity pools. It effectively optimizes the liquidity provision and price adjustment mechanisms of AMMs.

Unlike financial asset scenarios, data assets trading is characterized by high heterogeneity, discontinuity, and value sensitivity, where block trades and buy-sell disequilibrium are more likely to cause severe price fluctuations. Slippage in such scenarios not only occurs frequently but also stems from complex causes, making traditional AMM slippage control methods ineffective. Therefore, there is an urgent need to develop a slippage identification and control mechanism specifically for the characteristics of data asset trading.

2.4. Research Gaps and Breakthroughs

Compared to existing studies that primarily focus on crypto assets or data pricing, this paper addresses the key challenge of slippage control in data asset trading. It applies the AMM mechanism to heterogeneous data trading scenarios for the first time and proposes a dual slippage optimization mechanism integrating dynamic trade splitting and alternating order sorting. This mechanism actively controls trading slippage based on awareness of the liquidity pool state, significantly improving trading efficiency and market stability.

3. Data Trading Model and Slippage Mechanism Analysis Based on AMM

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3.1. Data Trading Liquidity Pool Model Construction

The essence of the data trading market lies in establishing a networked trading ecosystem that supports multi-agent participation, with its core characteristic manifested as a many-to-many trading model. This paradigm aggregates massive homogeneous data providers and geographically dispersed data consumers within a unified marketplace. Driven by dynamic demands, consumers typically procure large-scale homogeneous data assets to fulfill domain-specific analytical objectives. A representative application scenario involves constructing high-precision supermarket consumption behavior models using customer purchase records. To achieve this, data consumers such as market researchers, suppliers, advertisers, and academic institutions require multidimensional, large-scale datasets to support machine learning training, in-depth data mining, and analytical activities. Concurrently, numerous supermarkets capable of supplying such data act as data providers.

In this operational framework, data generated by Supermarket A is fragmented and distributed across multiple consumers, each imposing distinct purposes, processing methods, retention periods, and re-licensing terms for data usage. Such heterogeneity complicates the traceability of data flows and post-transaction usage for providers like Supermarket A. Conversely, individual consumers must consolidate fragmented datasets from multiple supermarkets to assemble comprehensive, application-specific datasets. This operational complexity underscores the intrinsic attributes of data trading markets: dynamic supply-demand alignment and interwoven data circulation pathways. This scenario exemplifies how decentralized data trading requires both fine-grained contractual governance and cross-source integration mechanisms to reconcile scalability with compliance. Specific scenario is illustrated in Figure 1.

To address the multi-agent dynamic trading scenario illustrated in Figure 1, this study proposes a multi-party data trading mechanism based on an AMM liquidity pool architecture, leveraging smart contracts to achieve decentralized pricing and trading. Furthermore, we design a slippage optimization algorithm to mitigate the market impact costs caused by block data trading.

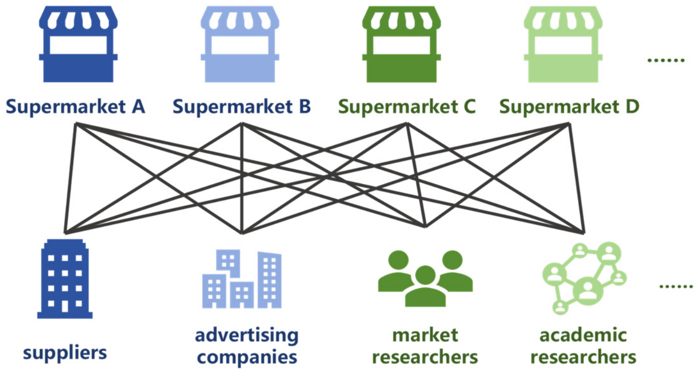


Figure 1. Data Trading Ecosystem.

3.1.1. Trading Model

Under the liquidity pool-based data trading model, the data trading process is divided into three stages: data assetization, asset certification, and entitlement transactionalization. Data assetization refers to transforming raw data into a standardized asset object, enabling the management of corresponding data assets in the form of digital assets. Asset certification refers to converting the ownership rights of a data asset into a set of entitlement objects, enabling the management of data rights in the form of digital assets according to the ownership system. Entitlement transactionalization refers to the actual trading of data through the exchange of data rights, with digital assets representing data asset rights as the object of exchange. In this process, we define the data meta-certificate as the subject matter of the original data for trading. The data warrant is the subject matter of the ownership of the data asset represented by the data meta-certificate. This design facilitates the separation of exercisable data ownership from the underlying data asset used as the transaction subject, and simplifies the governance of data assets. As shown in Figure 2.

 doi:10.20944/preprints202604.2221.v1

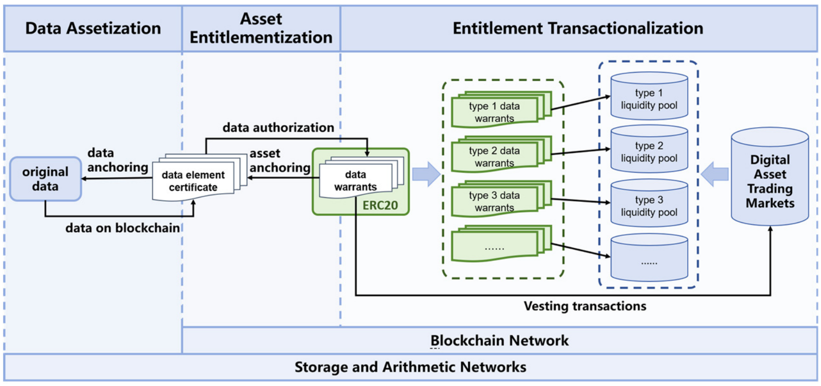


Figure 2. Warrant-oriented data trading ideas.

Based on this concept, this paper utilizes ERC20-based data warrants as the equity targets for data assets. ERC20 is a widely used token standard on the Ethereum blockchain, defining a set of standardized rules for the creation, transfer, and management of tokens on the Ethereum network. By employing ERC20-based digital warrants, the existence and uniqueness of data asset ownership can be represented, while also providing cross-platform interoperability and flexibility, thereby facilitating data trading management. During data trading, similar types of data can generate corresponding data warrants of the same type, which are then traded as trading objects in specific types of liquidity pools.

In the data trading model proposed in this paper, a data market maker liquidity pool model is constructed to isolate the trading parties, forming an indirect data circulation and trading model through interactions between data buyers/sellers and the liquidity pool. For example, a smart device collects supermarket sales records (i.e., customer shopping receipt datasets containing N purchase records) for a specific period and publishes them for sale as a data asset. A data meta-certificate can be created to characterize its asset attributes. For a data meta-certificate, a data warrant or a set of data warrants can be automatically generated, allowing the supermarket as a data seller to deposit these warrants into the supermarket receipt liquidity pool. If no such liquidity pool exists, a supermarket shopping receipt warrant liquidity pool can be created. At the same time, the supermarket deposits supermarket receipt data warrants and tokens to form a supermarket shopping receipt data warrant market maker, and also receives the corresponding reward tokens. Data analysts or market researchers as data buyers will enter the liquidity pool to purchase data warrants based on their needs and achieve data trading. The data trading process is automatically executed by smart contracts, where purchasers pay corresponding data tokens to obtain the required data. The liquidity pool dynamically adjusts data prices based on supply, demand, and trading volume. For example, if shopping receipt data for a certain period is highly sought after and in high demand, the liquidity pool will raise its price accordingly to balance supply and demand. Furthermore, it is also necessary to design optimized trading strategies based on the slippage issues inherent in liquidity pools to ensure trading efficiency and fairness. Figure 3 illustrates the workflow of the proposed data market maker liquidity pool model.

 doi:10.20944/preprints202604.2221.v1

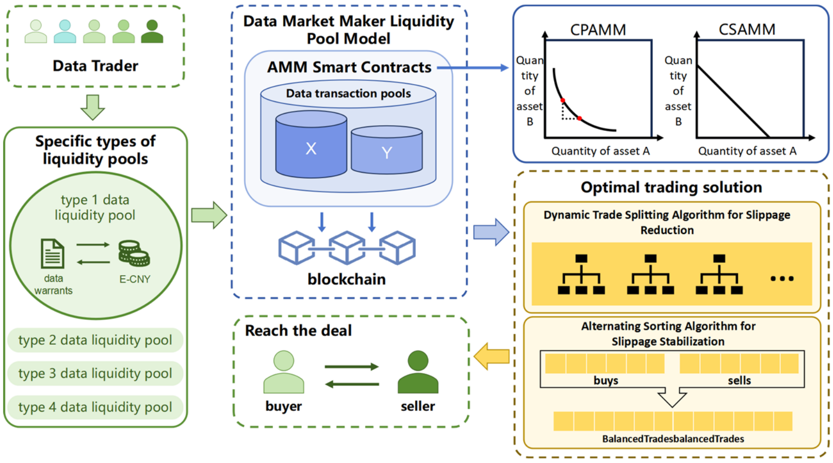


Figure 3. The system architecture of our proposed ESS-LP.

Data trading is supported by different types of liquidity pools, and data traders interact with specific liquidity pools via smart contracts to buy or sell data assets. This model adopts commonly used liquidity pool models, such as Constant Product Automated Market Maker (CPAMM) and Constant Sum Automated Market Maker (CSAMM), as references and applies them to liquidity management in the data market. To optimize the trading process, the model introduces a dynamic trade splitting algorithm and an alternating order sorting algorithm to find the optimal trading solution, thereby reducing slippage and improving trading efficiency.

The data trading pool is a liquidity pool consisting of data warrants, Digital Currency Electronic Payment (e-CNY), and a data trading access validation mechanism, as shown in Figure 4. When a new data trading pool is created, data access verification rules are first set, and the initial ratio of data warrants to payment currency in the liquidity pool is also determined. To incentivize liquidity providers to provide liquidity and ensure the depth of data and monetary assets in the data liquidity pool, the liquidity pool implements a strategy that rewards liquidity providers with trading fees. Once data trading is completed, the liquidity pool calculates the trading price through an AMM algorithm, thus completing the data trading activity.

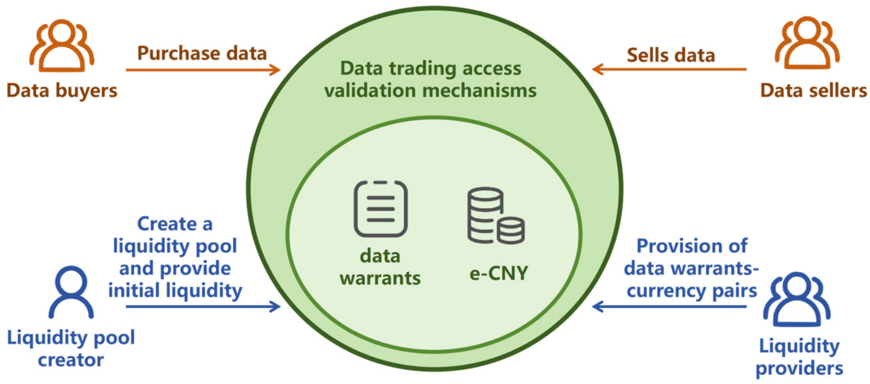


Figure 4. Data trading pools.

3.1.2. Pricing Mechanisms

Consider a data trading pool containing two types of tokens: the data warrant and e-CNY, referred to as MTK and ECNY, respectively, with quantities denoted x_{MTK} and y_{ECNY}. The AMM allows one token to be exchanged for the other according to a given function, this paper uses constant product curves, as shown in Eq. (3):

$$x_{MTK} \times y_{ECNY} = k \tag{3}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2026 doi:10.20944/preprints202604.2221.v1

Users holding data warrants can choose to simultaneously deposit MTK and ECNY asset pairs into the liquidity pool to provide liquidity and earn fee income directly. Data sellers can generate MTK from their data assets and then sell them to obtain ECNY, while data buyers pay ECNY to the trading pool to purchase MTK, and through MTK they can access the corresponding data assets. In the process of mutual exchange between ECNY and MTK, when a user purchases data assets (ignoring trading fees), the amount of data assets purchased is as shown in Eq. (4). Similarly, when a user sells data assets, the obtained e-CNY is as shown in Eq. (5).

$$\Delta x_{MTK} = x_{MTK} - \frac{k}{y_{ECNY} + \Delta y_{ECNY}}, \tag{4}$$

$$\Delta y_{ECNY} = y_{ECNY} - \frac{k}{x_{MTK} + \Delta x_{MTK}}. \tag{5}$$

In these equations, Δx_{MTK} represents the data warrants purchased, Δy_{ECNY} represents the e-CNY spent.

3.1.3. Definition of Slippage

For curve-based AMMs, slippage refers to the loss incurred by a trader due to a mismatch between the pool price at which a trade is initiated and the effective price obtained during the trade. The AMM mechanism in the trading pool is based on the constant product formula, and each trade changes the number of tokens in the pool, which in turn affects the price of the subsequent trades. Therefore, slippage directly influences the costs and benefits of both parties involved in data trading. Taking the purchase of MTK as an example, the actual price of purchasing MTK is shown in Eq. (6):

$$P_{trade} = \frac{\Delta y_{ECNY}}{\Delta x_{MTK}}, \tag{6}$$

where P_{trade} denotes the actual trading price.

The expected price of purchasing MTK is shown in Eq. (7):

$$P_{expected} = \frac{y_{ECNY}}{x_{MTK}}, \tag{7}$$

where $P_{expected}$ denotes the expected trading price.

The calculation of the slippage H is detailed in Eq. (8):

$$H = \left(\frac{P_{trade} - P_{expected}}{P_{expected}}\right) \times 100\%. \tag{8}$$

The lower the slippage, the more favorable it is for users. In data trading, lower slippage means that traders can trade closer to the expected price, which will boost users' enthusiasm for participating in data trading and further improve the efficiency and appeal of the data trading market.

3.2. Analysis of Slippage Mechanisms and Influencing Factors

In this study, we quantitatively and qualitatively analyze the impact of different factors on slippage in the data trading pool through simulation experiments. The experiment simulates a real trading environment, assuming the existence of the aforementioned data trading pool. In the initialization phase of the data trading pool, a user deposited 1000 data warrants and 1000 tokens into the pool to provide initial liquidity, and a certain amount of initial funds is allocated to each trading user's account. The data trading pool adopts a constant product pricing model, meaning that the product of the number of data warrants and the number of tokens is constant, which can effectively reflect market demand. The total trading duration is 10 minutes. During this period, three data traders continuously perform buy and sell operations at specific intervals: User 1 trades every 10 seconds, User 2 every 30 seconds, and User 3 every 1 minute. During the trading process, users' trading behavior occurs randomly and is determined by a random generator. This random decision

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2026 doi:10.20944/preprints202604.2221.v1

model is designed to simulate the uncertainty of user behavior in real trading. During the experiment, key data for each trade, including trading time, price, volume, and slippage, is recorded. Appropriate statistical analysis methods are used to analyze the data to study the impact of different factors on slippage.

3.2.1. Impact of Block Trade on Slippage

To gain a deeper understanding of the impact of block trades on slippage, we designed experiments to observe slippage variation over different trading volume intervals, and we quantified six different trading volume intervals ranging from 0.1% to 20% of the number of tokens in the liquidity pool, as shown in Table 1.

Table 1. Range of trading volumes.

Serial Number	Volume Ranges
1	0.1–0.5%
2	0.5–1%
3	1–5%
4	5–10%
5	10–15%
6	15–20%

Within these intervals, users' buying and selling behavior occurs randomly with equal probability to simulate a more realistic data trading market. Figure 5 illustrates the variation in slippage over time for different volume ranges. It can be observed that for smaller volume ranges, particularly at the 1% level, the market absorbs trades more easily, resulting in lower slippage. However, as the trading volume increases, slippage also increases, indicating that block trades significantly affect slippage, with larger trades leading to higher slippage.

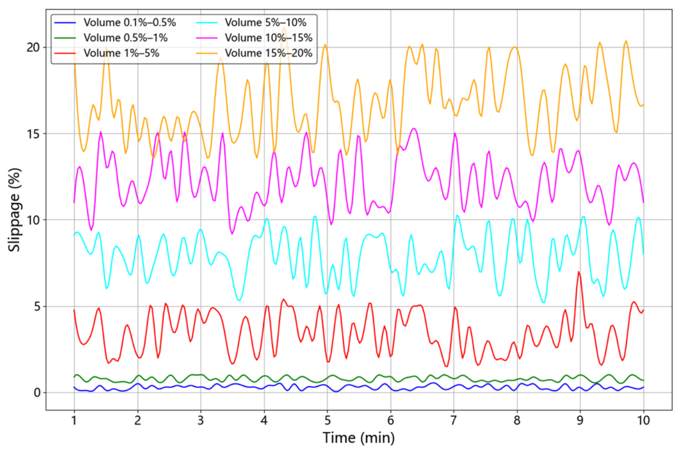


Figure 5. Changes in slippage over time for various volume ranges.

In addition, we use standard deviation as a metric for slippage volatility and analyze the data for trades in different volume ranges. Figure 6 shows the results comparing the standard deviation of slippage across different volume ranges. A significant positive correlation is observed between the standard deviation of slippage and the trading volume range. Specifically, the standard deviation of slippage increases significantly as the trading volume range grows, showing a clear increasing trend. The relatively small standard deviation in lower volume ranges (e.g., 0.1–0.5% and 0.5–1%) indicates more stable slippage changes. In contrast, higher volume ranges (e.g., 15–20%) have larger standard deviations, reflecting more dramatic fluctuations in slippage changes. Therefore, in block trades, the

 doi:10.20944/preprints202604.2221.v1

uncertainty of slippage changes is stronger, and the stability of the trading environment decreases, which means higher risks and potential unfair trading environments for data traders.

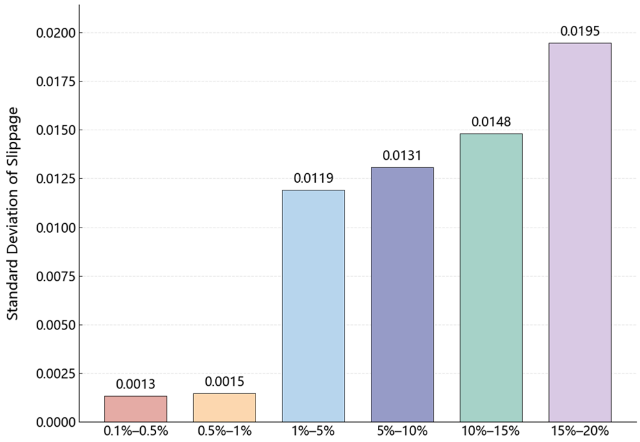


Figure 6. Standard deviation of slippage over a range of volumes.

Figure 7 further quantifies the impact of block trades on slippage, showing the trend in average slippage across trading volume ranges. It can be observed that as the trading volume range increases, the average slippage shows a clear upward trend, consistent with theories of market liquidity and price impact of trading pairs. Within the 0.1–1% volume range, average slippage is close to 0%, indicating that at smaller trade volumes, the market can efficiently absorb trades with low slippage, benefiting traders. In the 0.5–1% range, slippage remains relatively low, but has begun to show a tendency to increase as the trading volume increases. However, once the trading volume reaches 1%, the market feels the pressure of block trades, causing average slippage to rise significantly, with an accelerated growth rate. When the trading volume reaches the 15–20% range, average slippage increases significantly to nearly 17%, highlighting the substantial impact of block trades on market prices and slippage. In addition, the increase in slippage is non-linear: it grows more slowly for smaller ranges of trading volumes and faster for a larger block trade. This suggests that for block trades, the liquidity of the pool appears insufficient, and slippage significantly increases.

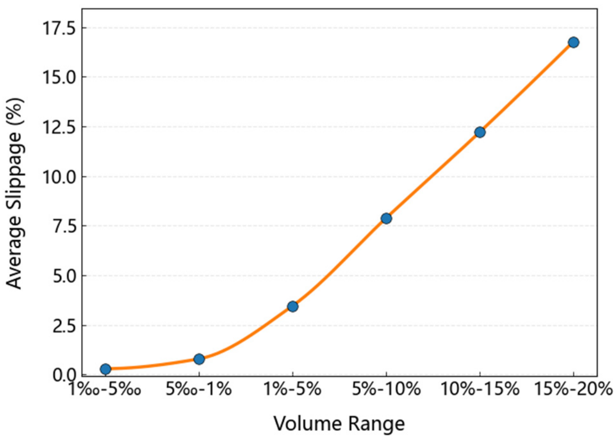


Figure 7. Standard deviation of slippage over a range of volumes.

3.2.2. Impact of Buy-Sell Equilibrium on Slippage

In financial markets, buy-sell equilibrium refers to a state where the trading intentions and capabilities of buyers and sellers are balanced, meaning the buyers' willingness to purchase matches the sellers' willingness to sell. In the second part of the experiment, we examine the impact of buy-

 doi:10.20944/preprints202604.2221.v1

sell equilibrium on slippage by simulating both equilibrium and disequilibrium trading environments.

To simulate a buy-sell equilibrium market environment, we set the probability of traders buying and selling to be equal, with each accounting for 50%, thereby ensuring an equilibrium power between buyers and sellers. In contrast, for buy-sell disequilibrium, the probability of buying or selling activity is set to 80%, simulating situations of oversupply or undersupply in the market. Through multiple repeated simulations, we obtain slippage data under different conditions.

Figure 8 illustrates the trend of slippage over time under both buy-sell equilibrium and disequilibrium conditions. The blue curve represents slippage in the equilibrium state, where the probabilities of users buying and selling are both 50%. It can be observed that the fluctuation amplitude is moderate, with overall stability in the range of 10–20%, indicating that slippage is relatively smooth under equilibrium conditions. The green curve corresponds to buying-dominated behavior, where the buying probability is 80% and the selling probability is 20%, while the red curve corresponds to selling-dominated behavior. It can be seen that the magnitude of slippage fluctuations is significantly higher under disequilibrium conditions, indicating increased slippage risk.

Figure 9 shows the comparison of the standard deviation of slippage under different conditions through a bar chart. The lowest standard deviation of slippage is 0.0394 in the buy-sell equilibrium condition, suggesting low slippage volatility and relatively smooth market trading in the equilibrium condition. The slippage standard deviation under disequilibrium conditions is significantly higher than that under equilibrium conditions, further confirming the significant impact of buy-sell disequilibrium on slippage volatility.

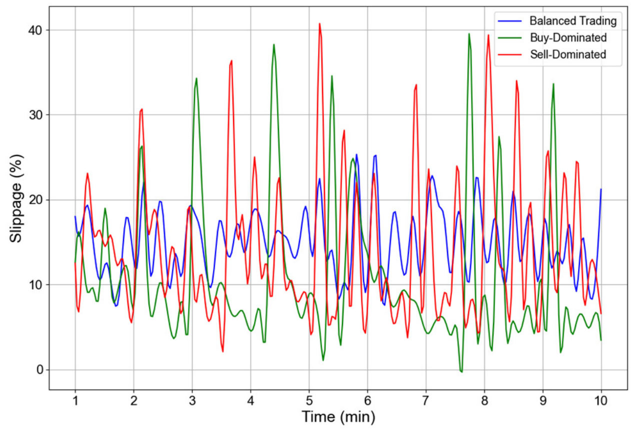
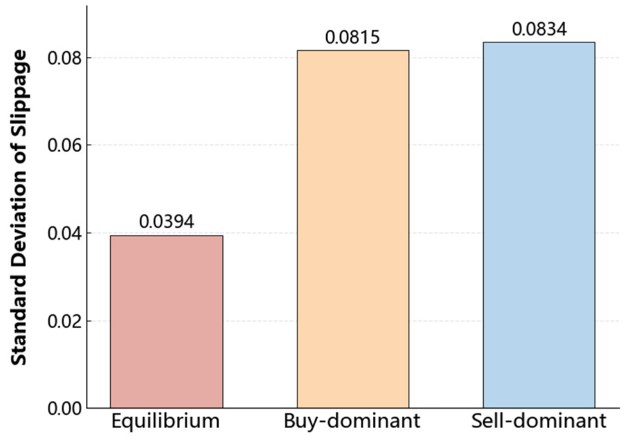


Figure 8. The variation of the slippage points over time for the equilibrium and disequilibrium conditions of buying and selling.



Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2026 doi:10.20944/preprints202604.2221.v1

Figure 9. Standard deviation of buying and selling.

In summary, block trades and the degree of market buy-sell equilibrium are key factors influencing slippage. Block trades, due to their impact on market liquidity, often trigger significant price movements, leading to increased slippage. Similarly, the degree of market buy-sell equilibrium directly affects slippage, with implications for the fairness of the trading environment. Therefore, it is crucial for traders to incorporate these factors when formulating and implementing trading strategies, so as to minimize the adverse effects of slippage and effectively improve the overall efficiency and fairness of trading.

3.3. Design and Implementation of Slippage Optimization Algorithm

3.3.1. Design Methodology

Building upon the experimental analysis presented earlier, we identify the key factors that influence slippage and accordingly propose an Effective Slippage Scheme Based on Liquidity Pools (ESS-LP) to minimize slippage. As illustrated in Figure 10, the data trading pool is divided into a pending trading pool and an execution pool. When a trader initiates a new trading, it first enters the pending trading pool. This pool continuously collects trades, performing splitting and sorting operations, and then enters the execution pool to achieve the optimal trading. The pending trading pool and the execution pool operate alternately. This paper focuses on the splitting and sorting mechanisms within the pending trading pool, which include the following key steps:

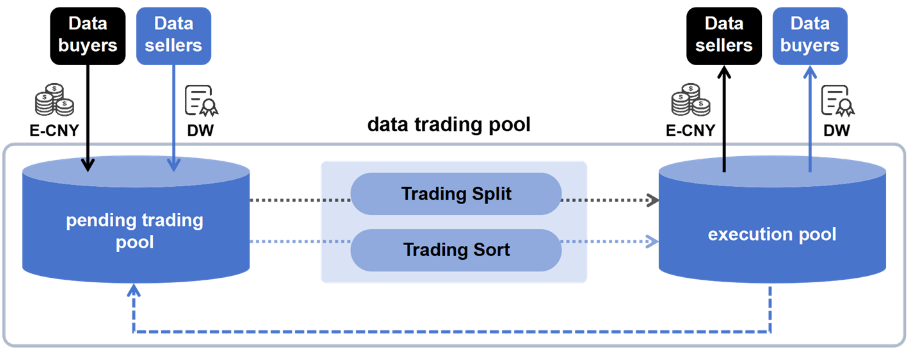


Figure 10. The method of optimizing the slippage.

- Collection of Trading Requests: Data traders may continuously submit trading requests to the data trading pool within a specified trading time window, and all requests initiated within this window will be collected.
- Processing of Trading Requests: Once the time window closes, the system will begin processing all collected trading requests. The processing involves two main steps: block trade splitting and trading sorting. First, block trades, defined as those exceeding 1% of the current token supply in the trading pool, are split. Each split will be made according to 1% of the liquidity of the trading pool to minimize the impact of block trades on the market price. Second, all trades will be sorted to achieve buy-sell equilibrium. This is achieved by using two arrays to store buy and sell orders respectively, and the system alternately selects trades from the buy and sell order arrays to populate the result arrays.
- Execution of Trading: After processing the trading requests, the system executes all trades in the result array sequentially, ensuring orderly trade execution while minimizing slippage.

3.3.2. Dynamic Trade Splitting Algorithm for Slippage Reduction

After the trading time window closes, we first identify and process block trades. We define block trades as those where the trading volume exceeds 1% of the token quantity in the trading pool. For these block trades, this paper adopts a splitting strategy, dividing them into multiple smaller trades,

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2026 doi:10.20944/preprints202604.2221.v1

each not exceeding 1% of the pool's balance, to reduce the impact of a single block trade on market prices.

Algorithm 1 calculates the ratio of trading volume to the token pool balance and compares it with the threshold. If the trading volume exceeds 1% of the pool's balance, it enters the splitting process. During splitting, the algorithm iteratively reduces the trading volume through a loop until the remaining volume does not exceed 1% of the pool balance. In each iteration, the algorithm first obtains the current token balance and calculates the current 1% pool balance. Then, it determines the split volume, which is the lesser of the remaining trading volume and the current 1% pool balance. The split trading information is then added to the trading list, and the remaining trading volume is updated. If the trading volume does not exceed 1% of the pool balance, the original trading information is added directly to the trading list. Finally, the algorithm terminates and returns the updated trading list. This algorithm effectively balances the impact of block trades on trading pool slippage by dynamically adjusting the trading volume.

Algorithm 1: Check and Split

Input: *simpleSwapAddress, token, tokenAddress, user, volumeToTrade, balanceToken*
Output: *trades*
$onePercentPoolBalance \leftarrow \frac{balanceToken}{100}$
if $volumeToTrade > onePercentPoolBalance$ **then**
  $remainingVolumeToTrade \leftarrow volumeToTrade$
  While $remainingVolumeToTrade > 0$ **do**
    $currentBalanceToken \leftarrow token.balanceof(simpleSwapAddress)$
    $currentonePercentPoolBalance \leftarrow \frac{currentBalanceToken}{100}$
    $splitVolume \leftarrow min(remainingVolumeToTrade,\ currentOnePercentPoolBalance)$
    $trades.push\begin{pmatrix} user, \\ token, \\ tokenAddress, \\ splitAmount \end{pmatrix}$
    $remainingVolumeToTrade \leftarrow remainingVolumeToTrade - splitVolume$
  end while
else
    $trades.push\begin{pmatrix} user, \\ token, \\ tokenAddress, \\ splitAmount \end{pmatrix}$
    end if

3.3.3. Alternating Order Sorting Algorithm for Slippage Stabilization

Next, Algorithm 2 sorts all trades, including split trades, to achieve buy-sell equilibrium, thereby reducing the volatility of slippage within the trading pool.

Algorithm 2: Balance Buys and Sells

Input: *trades*
Output: *balancedTrades*
$buys \leftarrow [\,],\ \ sells \leftarrow [\,]$
for *trade in trades* **do**
  if *trade.tokenAddress = ENCYTokenAddress* **then**
    buys.append(trade)
  else if *trade.tokenAddress = mtkTokenAddress* **then**
    sells.append(trade)

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 May 2026 doi:10.20944/preprints202604.2221.v1


```
        end if
    end for
    balancedTrades ← []
    while volume(buys) > 0 and volume(sells) > 0 do
        balancedTrades.append(buys.pop(0))
        balancedTrades.append(sells.pop(0))
    end while
    balancedTrades.append(buys)
    balancedTrades.append(sells)
    return balancedTrades
```

Initially, the algorithm creates two empty lists to store buy and sell orders respectively. Then, through a loop, trades are alternately taken from the buy and sell order lists and added to a new list called BalancedTrades. This step ensures the equilibrium execution of both buy and sell orders. This process minimizes slippage caused by one-sided market pressure. If any trades remain in either the buy or sell lists, the algorithm adds them all to the BalancedTrades list, ensuring that all trades are executed, even in the case of disequilibrium. Finally, the algorithm returns the BalancedTrades list with the optimized sequence of trades.

4. Experimental Evaluation

To validate the efficacy of the proposed ESS-LP slippage optimization mechanism, this study conducts a comparative analysis of system slippage under unified trading environments and simulation parameters. We systematically examine before and after optimization variation trends in slippage, volatility characteristics and overall controllability. This section includes three parts: Experimental settings, Comparative slippage results and Evaluation conclusions, which demonstrate the comprehensive performance of the optimization algorithm in improving trading stability and reducing trading costs.

4.1. Experimental Setting

This experiment replicates the data trading environment described in Chapter 3. We set a 10 minutes trading window, during which traders submit trading requests at fixed intervals. Trade volumes and directions are generated to simulate real world behavioral differences probabilistically. The trading mechanisms are divided into two cohorts: the control group executes trading via conventional AMM mechanism with no optimization strategy; the experimental group: Implements the proposed dual slippage optimization mechanism, incorporating liquidity pool state-aware dynamic trade splitting and alternating order sorting.

Slippage evaluation employs three key metrics: Average Slippage (AS), Slippage Standard Deviation (SSD) and Maximum Slippage (MS). The formulas are given in equations (8)-(10):

$$AS = \frac{1}{n}\sum_{i=1}^{n}|H_i|, \tag{9}$$

$$SSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(H_i - AS)^2}, \tag{10}$$

$$MS = \max(|H_1|, \ldots, |H_n|). \tag{11}$$

Specifically, AS measures systematic cost through mean price deviation from expectations; SSD quantifies the volatility and stability of slippage; MS identifies extreme deviations in individual trades to reveal risk exposure. Here, n represents the total number of trading orders in the evaluation period, i denotes the transaction index, and H_i refers to the slippage deviation of each individual trade.

doi:10.20944/preprints202604.2221.v1

4.2. Comparative Slippage Results

Figure 11 illustrates the temporal evolution of slippage for both control and experimental cohorts throughout the trading session. As can be seen, the pre-optimized mechanism (blue trace) exhibits pronounced slippage spikes with substantial volatility, particularly during the 4-5 min, 6-7 min, and 8-9 min intervals where slippage exceeds the 5.0% threshold. In the data trading context, such elevated slippage signifies heightened price volatility and increased trading costs for block trades executions. In contrast, the proposed dual slippage optimization mechanism (red trace) demonstrates significantly attenuated fluctuations over the same trading session. The maximum observed slippage is 1.1% throughout the whole trading window, and there is no absence of statistical outliers. Consequently, the optimized platform achieves minimal price volatility during block trade executions, significantly reduces trading costs and improves the efficiency of the data trading market.

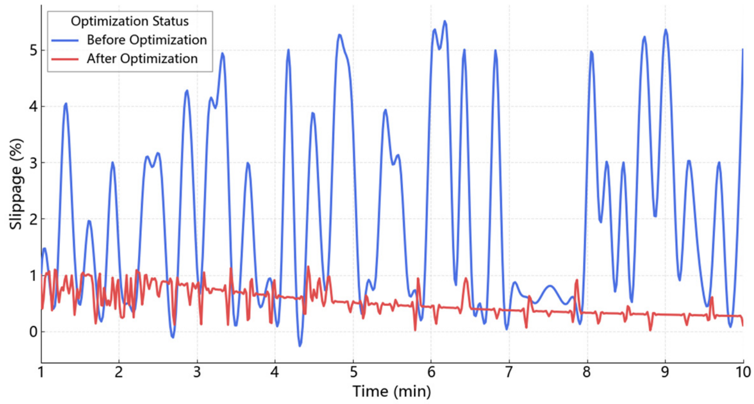


Figure 11. Slippage comparison before and after optimization.

Further statistical results are shown in Table 2, demonstrate substantial improvements. First of all, the AS decreases from 2.1% pre-optimization to 0.5% post-optimization, representing a 76.2% reduction. Secondly, the SSD decreases from 1.66% to 0.25%, indicating 84.9% volatility reduction. Thirdly, the MS is decreased from 5.5% to 1.1%. These quantitative metrics confirm the optimization algorithm's significant efficacy in controlling both slippage magnitude and uncertainty.

Table 2. Experimental statistical results.

Indicator name	Unoptimized	Optimized	Level of improvement
Average Slippage (%)	2.1	0.5	↑76.2%
Slippage Standard Deviation (%)	1.66	0.25	↑84.9%
Maximum Slippage (%)	5.5	1.1	↑80%

4.3. Evaluation Conclusions

The experimental findings indicate that the proposed dual slippage optimization mechanism not only significantly reduces trading costs throughout the trading session, but also achieves excellent stability in controlling slippage volatility. This enhances the predictability and fairness in the data trading environment. In block trade scenarios, the traditional AMM mechanism suffers from catastrophic slippage amplification due to liquidity pool impacts. However, the proposed optimization algorithm efficiently mitigates slippage accumulation by splitting block trades into micro-trades and staggering their bidirectional execution between buyer and seller. This mechanism is proven to substantially strengthen platform robustness and increase market attractiveness in data trading.

In summary, the dual slippage optimization mechanism proposed in this paper effectively mitigates the slippage problem caused by impacts of a single trading and matching disequilibrium. This mechanism integrates "Dynamic Trade Splitting" and "Alternating Order Sorting", which

 doi:10.20944/preprints202604.2221.v1

demonstrates theoretical coherence with data asset characteristics and provides empirically validated adaptability. This mechanism provides robust technical support for automated pricing systems in high-frequency or block trade scenarios.

5. Conclusion and the Future Work

At present, the digital economy has become an important engine to lead the scientific and technological revolution and industrial change, and drive economic growth. As a foundational strategic resource, the efficient trading of data is paramount for unleashing its full value and driving digital economy development. Addressing the limitations of traditional centralized data trading platforms in efficiency, transparency, and dynamic pricing, this study pioneers the integration of AMM mechanisms into data trading through our proposed ESS-LP framework. We establish an innovative liquidity pool-based data trading paradigm that breaks through the bottlenecks in data trading efficiency and fairness via the collaborative innovation of market-oriented pricing mechanisms and automated trading processes.

This study innovatively constructs an automated pricing and trading matching mechanism based on a liquidity pool. Through mathematical modeling and simulation experiments, it quantitatively analyzes the generation mechanism of slippage in data liquidity pools when coping with trading shocks and buy-sell disequilibrium. The findings reveal that liquidity shocks induced by block trades and temporal mismatches arising from buy-sell disequilibrium constitute the primary root causes of pronounced slippage volatility in data trading. This analysis fills a critical research gap in understanding slippage mechanisms within data trading contexts. Building upon these insights, we propose and implement a dual slippage optimization mechanism integrating "Dynamic Trade Splitting" and "Alternating Order Sorting". This mechanism mitigates slippage caused by single-trade shocks and aggregation disequilibrium. It does so by intelligently splitting block trades and executing them as alternating buy and sell operations. The experimental results demonstrate that the ESS-LP method proposed in this paper yields significant efficacy. In the simulated data trading contexts, the average slippage magnitude decreased from 2.1% to 0.5%, representing a 76.2% reduction. This effectively reduces trading costs for data traders, enhances price stability and resource allocation efficiency in the overall market. It provides a fairer, more predictable trading environment for participants in the data trading market.

In the future, we will further investigate research on the dynamic slippage optimization strategy by exploring more precise adjustment methodologies based on real-time market conditions and liquidity perception. At the same time, we will thoroughly analyze the issue of Impermanent Loss (IL) within liquidity pools and examine the mechanism that combines slippage optimization. The goal is to provide a trading environment that minimizes economic losses, enhancing fairness-efficiency equilibrium, and accelerating data asset circulation, ultimately advancing frictionless value realization across data asset markets.

Author Contributions: Writing-original draft, Mengyang Li; Writing-review & editing, Huayou Si, Yuanyuan Qi, Neal N. Xiong, Wei Chen, Loc Nguyen The and Shichong Wang; Experimental design, Huayou Si and Mengyang Li; Methodology, Huayou Si and Wei Chen; Conceptualization, Yuanyuan Qi and Neal N. Xiong; Funding acquisition, Huayou Si and Yuanyuan Qi.

Funding: This study was supported by Zhejiang Province Leading Geese Plan (Grant No. 2024C01107 and Grant No. 2025C02028).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ren M, Wang W, Zhou W, Liu Y, Yao A, Wen L, Gong Y. Promoting the Data Element Marketization: Progress, Problems and Solution. *Documentation, Information & Knowledge* **2024** *41(5)*, 25-35. 10.13366/J.DiK.2024.05.025.
2. Wang W, Zhang M, Wang, J. Research and Analysis of Big Data Trading Platforms at Home and Abroad. *Journal of Intelligence* **2019**, *38(2)*, 181-186+194.
3. Hertzog E, Benartzi, G. Bancor protocol Continuous Liquidity for Cryptographic Tokens through their Smart Contracts. Available online: https: //cryptopapers.info/assets/pdf/ bancor.pdf (accessed on 2017).
4. Adams H, Zinsmeister N, Salem M, Keefer R, Robinso D. Uniswap v3 core. Available online: https://github.com/Uniswap/v3-core
5. Egorov M. Stableswap: Efficient mechanism for stablecoin liquidity. Available online: https://www.stableswap.org/ (Available online: 2021).
6. Auer R A, Haslhofer B, Kitzler S, Saggese P, Victor F. The technology of decentralized finance (DeFi). *Digital Finance* **2024**, 6, 55-95. https://doi.org/10.1007/s42521-023-00088-8.
7. Zhu Y, Ye Y. Defining data assets based on the attributes of data. *Big Data Research* **2018**, *4(6)*, 65-76.
8. Hu D, Li Y, Pan L, Li M, Zheng S. A blockchain-based trading system for big data. *Computer Networks* **2021**, 191, 107994.
9. Jiang E, Qin B, Wang Q, Wu Q, Li S, Shi W, Bi Y, Tang W. BDTS: Blockchain-Based Data Trading System. In Proceedings of Information and Communications Security (ICICS 2023), Berlin, Heidelberg, 20 Oct 2023; pp. 645-664. 10.1007/978-981-99-7356-9 38.
10. Xue L, Ni J, Liu D, Lin X, Shen X. Blockchain-Based Fair and Fine-Grained Data Trading With Privacy Preservation. *IEEE Transactions on Computers* **2023**, *72(9)*, 2440-2453. 10.1109/TC.2023.3251846.
11. RapidAPI. API Marketplace. Available from: https://rapidapi.com/.
12. Huang J, Li J, Tang K. Data trust: a trustworthy data transaction model. *Big data research* **2023**, *9(2)*, 67-78.
13. Azcoitia S A, Laoutaris N. A Survey of Data Marketplaces and Their Business Models. *ArXiv* 2022, arXiv: 2201.04561.
14. Amazon Mechanical Turk. Amazon Mechanical Turk. Available from: https://www.mturk.com/.
15. Tsai W -T, He J, Wang R, Deng E. Decentralized Digital-Asset Exchanges: Issues and Evaluation. In Proceedings of 2020 3rd International Conference on Smart BlockChain (SmartBlock), Zhengzhou, China, 23-25 Oct 2020; pp. 1-6. 10.1109/SmartBlock52591.2020.00024.
16. Sang Y, Shen H, Tan Y, Xiong N. Efficient protocols for privacy preserving matching against distributed datasets. In Proceedings of Information and Communications Security: 8th International Conference (ICICS 2006), USA, 4-7 Dec 2006; pp. 210-227. https://doi.org/10.1007/11935308_15.
17. Chen X, Zhang Z, Qiu A, Xia Z, Xiong N, Novel coverless steganography method based on image selection and StarGAN, *IEEE Transactions on Network Science and Engineering* **2020**, *9(1)*, 219-230.
18. Li D, Guo Q, Bai D, Zhang, W. Research and Implementation on the Operation and Transaction System Based on Blockchain Technology for Virtual Power Plant. In Proceedings of 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS), Huaihua, China, 15-17 Jul 2022; pp. 165-170. 10.1109/ICBCTIS55569.2022.00046.
19. Dai W, Dai C, K Choo -K R, Cui C, Zou D, Jin H. SDTE: A Secure Blockchain-Based Data Trading Ecosystem. *IEEE Transactions on Information Forensics and Security* **2020**, *15*, 725-737.
20. Xu J, Paruch K, Cousaert S, Feng Y. SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols. *ArXiv* 2023, arXiv: 2103.12732v7.
21. Park A. The Conceptual Flaws of Decentralized Automated Market Making. *Management Science* **2023**, *69(11)*, 6731-6751.
22. Hanson R. Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation. *The Journal of Prediction Markets* **2007**, *1*, 3-15.
23. Heimbach L, Wang Y, Wattenhofer R. Behavior of Liquidity Providers in Decentralized Exchanges. *ArXiv* 2021, arXiv: 2105.13822v2.
24. Krishnamachari B, Feng Q, Grippo E. Dynamic Curves for Decentralized Autonomous Cryptocurrency Exchanges. *ArXiv* 2021, arXiv: 2101.02778.

doi:10.20944/preprints202604.2221.v1

25. Buterin V. Let's run on-chain decentralized exchanges the way we run prediction markets. Available online: https://www.cleardao.com/docs/Clear%20Litepaper.pdf.
26. Finance B. Available online: Balancer. http://balancer.fi.
27. Othman A, Pennock D M, Reeves D M, Sandholm T. A Practical Liquidity-Sensitive Automated Market Maker. *The Journal of Prediction Markets* **2013**, *1(3)*, 1-25.
28. Mohan V. Automated market makers and decentralized exchanges: a DeFi primer. *Financial Innovation* **2022**, *8(20)*.
29. Jiang S, Chen J, Li F, Geng H, Chi, H. DCAMM: Dynamic Curve-Based Automated Market Maker. In Proceedings of GLOBECOM 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 04-08 Dec 2023; pp. 4491-4496. 10.1109/GLOBECOM54140.2023.10437668.
30. Gu Q, Chang Y, Xiong N, Chen L. Forecasting Nickel futures price based on the empirical wavelet transform and gradient boosting decision trees. *Applied Soft Computing* **2021**, *109*, 107472. https://doi.org/10.1016/j.asoc.2021.107472.
31. Heimbach L, Schertenleib E, Wattenhofer R. Risks and Returns of Uniswap V3 Liquidity Providers. *ArXiv* 2022, arXiv: 2205.08904v2.
32. Berenzon D. Constant function market makers: DeFi's "zero to one" innovation. Available online: https://medium.com/bollinger-investment-group/constant-function-market-makers-defis-zero-to-one-innovation-968f77022159.
33. Wang S, Krishnamachari, B. Optimal Trading on a Dynamic Curve Automated Market Maker. In Proceedings of 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Shanghai, China, 02 May 2022; pp. 1-5, 10.1109/ICBC54727.2022.9805489.
34. Egorov M. Stableswap-efficient mechanism for stablecoin liquidity. Available online: Stableswap Foundation. https://stableswap.org.
35. Wang Y. Automated Market Makers for Decentralized Finance (DeFi). Available online: https://api.semanticscholar.org/CorpusID:221652658 (accessed on 2024).
36. Xiong N, Vasilakos A, Yang L, Wang C, Kannan R, Chang C, Pan Y. A novel self-tuning feedback controller for active queue management supporting TCP flows. *Information Sciences* **2020**, *180(11)*, 2249-2263.
37. Li S, Chen Y, Chen L, Liao J, Kuang C, Li K, Liang W, Xiong N. Post-quantum security: Opportunities and challenges, *Sensors* **2023**, *23(21)*, 8744.
38. Sun C, Zhang C, Xiong, N. Infrared and Visible Image Fusion Techniques Based on Deep Learning: A Review. *Electronics* **2020**, *9(12)*, 2162.
39. Guo J, Liu A, Ota K, Dong M, Deng X, Xiong N. ITCN: An intelligent trust collaboration network system in IoT, *IEEE transactions on network science and engineering* **2021**, *9(1)*, 203-218.
40. Shen X, Yi B, Liu H, Zhang Z, Liu S, Xiong N. Deep Variational Matrix Factorization with Knowledge Embedding for Recommendation System. *IEEE Transactions on Knowledge and Data Engineering* **2021**, *33(5)*, 1906–1918.
41. Yang Y, Xiong N, Chong N Y, Défago X. A decentralized and adaptive flocking algorithm for autonomous mobile robots, In Proceedings of 2008 The 3rd International Conference on Grid and Pervasive Computing, Kunming, China, 25-28 May 2008; pp. 262-268. 10.1109/GPC.WORKSHOPS.2008.18.
42. Wang X, Li Q, Xiong N, Pan Y. Ant colony optimization-based location-aware routing for wireless sensor networks, *Wireless Algorithms, Systems, and Applications* **2008**, 109-120.
43. Lim T. Predictive crypto-asset automated market maker architecture for decentralized finance using deep reinforcement learning. *Financial Innovation* **2024**, *10(144)*.