University of California
Santa Barbara

Essays in FinTech and Behavioral Economics

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Economics

by

Dingyue Liu

Committee in charge:

Professor Erik Eyster, Chair
Professor Sevgi Yuksel
Professor Rod Garratt

September 2024

The Dissertation of Dingyue Liu is approved.

Professor Sevgi Yuksel

Professor Rod Garratt

Professor Erik Eyster, Committee Chair

July 2024

Essays in FinTech and Behavioral Economics

Acknowledgements

This dissertation would not have been possible without the love and support of my mentors, friends, and family. Thank you all for being part of this journey.

I would like to start by expressing my deepest gratitude to my advisors. Erik, your unwavering support, guidance, and belief in me have been invaluable. I cannot express how much I appreciate your help in fostering my growth, both in my research and personal life. Sevgi, your sincere insights and dedication have shown me what it takes to become a great researcher. Rod, thank you for introducing me to the fascinating world of FinTech. I would not have reached this point without your guidance and encouragement along the way.

Thank you to all the faculty members in the experimental theory group and the department who generously offered me their advice on my ideas, papers, and career path.

I'm grateful to everyone in the graduate program office and the administrative office for their great support since day zero.

Nir, thank you for being the best co-author and co-organizer one could ever dream of. It has been an amazing journey with you, and I look forward to many more years of collaboration.

I also want to thank all my friends at UCSB. Thank you for all the memories, laughs, tears, celebrations, and condolences. Shoutout to Caroline, Yuanzhe, Yang, Lei, Micah, Sarah, Danny, Vincent, and Jing.

Finally, I'm deeply grateful and fortunate to have my family's unconditional love, encouragement, and support. I love you all.

Dingyue Liu

Curriculum Vitae

Education

Ph.D. in Economics (expected), University of California, Santa Barbara, 2024.

M.A. in Statistics - Emphasis in Data Science, University of California, Santa Barbara, 2022.

M.A. in Economics, University of California, Santa Barbara, 2020.

B.S. with Highest Distinction in Mathematics/Statistics, Purdue University, 2019.

B.S. with Highest Distinction in Economics, Purdue University, 2019.

Research Interests

Primary: Behavioral Economics, FinTech

Secondary: Labor Economics, Economics of Education

Research Experience

PhD Research Fellow at Uniswap Labs, 2024.

Research Assistant for Professor Sevgi Yuksel (UCSB), 2022 - 2023.

Research Assistant for Professor Erik Eyster (UCSB), 2021.

Publication

"The Suitability of Using Uniswap V2 Model to Analyze V3 Data." (joint with Nir Chemaya)
Finance Research Letters 59 (January 2024): 104717.

"The Power of Default: Measuring the Effect of Slippage Tolerance in Decentralized Exchanges" (joint with Nir Chemaya, Robert McLaughlin, Nicola Ruaro, Giovanni Vigna, and Christopher Kruegel)
Financial Cryptography and Data Security (FC 2024)

Working Papers

"Estimating Investor Preferences for Blockchain Security" (joint with Nir Chemaya)

"Motivating Academic Success: The Role of Leaderboards in Shaping Student Study Behaviors" (joint with Anna Jaskiewicz, Ruth Morales, and Caroline Zhang)

"College Basketball Game Day and Sexual Assault" (joint with Yixin Chen)

"Uniswap Daily Transactions Index by Network" (joint with Nir Chemaya, Lin William Cong, Emma Joergensen, and Luyao Zhang)

Presentations

by coauthor*, poster session†

2024: CfC St. Moritz (CFC-SMART)*, Uniswap Labs, Financial Cryptography and Data Security*, UC San Diego†, National Conference on Teaching and Research in Economic Education (CTREE)*

2023: UC Irvine*, WEAI Annual Meeting*, The Science of Blockchain Conference (Lightning session), ESA North-American Conference, Tokenomics, Crypto and Blockchain Economics Research Forum*

2022: Southern California Graduate Conference in Applied Economics, UCSB-ECON DeFi Seminar, The Science of Blockchain Conference (Lightning session), Blockchain Accelerator Foundation Research Symposium

Conferences Organized

UCSB-ECON DeFi Weekly Seminar (jointly organized with Nir Chemaya), 2022 - Present.

Economic Science Association (ESA) North-American Conference (jointly organized with Gary Charness and Nir Chemaya), 2022.

World Congress on Teaching, Learning, and Curriculum in Physical Education (jointly organized with Xiaozhan Wang et al.), 2019.

Refereeing Experience

Finance Research Letters, Southwestern Finance Association Annual Meeting, FMA European Conference, FMA Applied Finance Conference.

Contributions to Crowd-Sourced Research Projects

Fišar, M., Greiner, B., Huber, C., Katok, E., Ozkes, A., and the Management Science Reproducibility Collaboration (forthcoming). Reproducibility in Management Science. Management Science.

Workshop Participation

CBER-CDFT-DSF Summer School, 2024.

Yale Summer School in Behavioral Finance, 2024.

The Learning, Computational and Bayesian Methods in Experimental Economics, 2024.

FAIR NHH and Rady Spring School in Behavioral Economics at UC San Diego, 2024.

Caltech Summer School in Theroy-Based Experiments & Workshop, 2023.

27th Visiting Graduate Student Workshop in Experimental Economics at Chapman University, 2022.

IFREE Undergraduate Workshop in Behavioral and Experimental Economics, 2018.

Honors and Awards

University of California, Santa Barbara

Academic Senate Doctoral Student Travel Grants, 2024.
GSA Conference Travel Grant, 2024.
FC24 Travel Grant, 2024.
Economics Graduate Research Quarter Fellowship, 2023.
Academic Senate Doctoral Student Travel Grants, 2023.
GSA Conference Travel Grant, 2023.
Outstanding Undergraduate Course Teaching Assistant, 2023.
Economics Department Travel Grants, 2023.
GSA Conference Travel Grant, 2022.
Outstanding Undergraduate Course Teaching Assistant of the Year, 2022.
Bryan Conley Fellowship, 2019.

Purdue University

RecWell Employee of the Year, 2019.
Dennis J Weidenaar Memorial Scholarship, 2019.
Outstanding Krannert Senior award, 2019.
Mu Sigma Rho Award, 2019.
Omicron Delta Epsilon, 2018.
Big Move Short Term Scholarship, 2017.
Devos International Scholorship, 2017.
Dean's List and Semester Honors, 2015 - 2019.

Teaching

Teaching Associate

ECON5 Statistics for Economics, Winter 2022, Summer 2022, Summer 2023, Spring 2024.

Lecturer

UCSB Academic Research Program, UCSB Extension Program, Summer 2022.
Introduction to Economics - Micro, UCSB Extension Program, Summer 2021.
Introduction to Economics - Macro, UCSB Extension Program, Summer 2021.

Head Teaching Assistant

ECON5 Statistics for Economics, Fall 2021, Fall 2022, Winter 2023, Fall 2023.
ECON134A Financial Management, Spring 2022.

Graduate Teaching Assistant

ECON134A Financial Management, Spring 2021.

ECON2 Principles of Economics - Macro, Winter 2020, Spring 2020, Winter 2021.

ECON106 Managerial Economics, Fall 2020.

ECON10A Intermediate Microeconomic Theory, Fall 2019.

Mentor

Research Mentorship Program, Summer 2022, Summer 2023.

Undergraduate Teaching Assistant

ECON360 Econometrics, Purdue University, Spring 2019.

ECON451 Game Theory, Purdue University, Fall 2018.

ECON422 Public Finance & Taxation, Purdue University, Spring 2018.

Skills

Language: Mandarin (native), English (fluent), French (basic).

Primary Programming Language: R, Python, LaTeX.

Secondary Programming Language: STATA, MATLAB, JavaScript, HTML.

Other

Citizenship: People's Republic of China.

Permissions and Attributions

Parts of the content of Chapter 2 and Appendix B is published in the Finance Research Letter.

Other parts of the content of Chapter 2 and Appendix B is published through proceeding of Financial Cryptography and Data Security 2024 in Springer LNCS 14744 and 14745.

Abstract

Essays in FinTech and Behavioral Economics

by

Dingyue Liu

This dissertation consists of two essays studying decision making in FinTech, and one essay on decision making in education.

The first chapter explores swapping decisions on decentralized exchanges (DEX). With the increasing adoption of DEX platforms, new Layer 2 (L2) blockchain alternatives offer better scalability and lower fees than the Ethereum blockchain (L1), yet the relative security of L2 is uncertain. Using a structural model and a novel dataset, we estimate investor preferences for blockchain security between mainnet (L1), and two main L2 networks, Polygon and Optimism. We find that traders anticipate a 0.68% (3.29%) chance of losing transaction value when trading on Polygon (Optimism) compared to L1, significantly higher than the transaction fees (0.01%-0.3%) charged on each trade. Our findings provide empirical evidence of the trade-off between scalability, security, and decentralization, a major challenge for blockchain networks.

The second chapter investigates default settings on DEX. DEX prices update continuously after each swap, causing potential price shifts for users awaiting execution. Users set a slippage tolerance to limit acceptable price increases, but this can either expose them to sandwich attacks or cause transaction failures. We analyze the impact of slippage tolerance settings on the health of Uniswap and Sushiswap ecosystems. A recent Uniswap change replaced the static default slippage setting (0.5%) with a dynamic one based on market conditions to reduce sandwich attacks. We find that this change significantly reduces trader losses by approximately 54.7%, with an even more pronounced effect (90%)

for traders following the default settings. We also propose further improvements for these settings.

The final chapter examines student decision-making, specifically the impact of a gamified leaderboard on engagement and procrastination. Procrastination is common among students, particularly with assignments. Gamification, incorporating game-like elements into education, shows promise in addressing this issue. Our results indicate that students in the treatment group complete assignments faster, suggesting the leaderboard positively influences study behaviors. While the overall class performance effect is not significant, transfer students and male students exposed to the leaderboard achieve higher course grades than their peers in the control group.

Contents

Chapter 1

Estimating Investor Preferences for Blockchain Security

with Nir Chemaya

1.1 Introduction

Liquidity pools are innovations in decentralized finance (DeFi). They allow for the exchange of crypto assets without the traditional centralized limit-order-book mechanism. Investors deposit tokenized assets into smart contracts.[1] They then exchange tokens from these pools according to the terms prescribed by a mechanism that determines the swapping price of each transaction. Uniswap is currently the largest liquidity pool protocol in DeFi, with a daily volume of roughly $2 billion, and total liquidity of $5 billion.[2]

Traditionally, most liquidity pools operate on the Ethereum blockchain, also known

[1] A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code on the blockchain.

[2] Data source: https://defillama.com/dexs as of July 5, 2024.

as the Layer 1 (L1). However, the DeFi landscape is evolving, and liquidity pool protocols like Uniswap have expanded their support to Layer 2 (L2) blockchains, such as Polygon and Optimism. These L2 blockchains address Ethereum's scalability challenges by offering lower gas fees and faster transaction processing,[3] making them an appealing option for traders.

Ethereum's scalability is limited to processing 15-30 transactions per second, resulting in high gas fees. Gas fees are paid for the validation service made by the validators (miners). When only a small number of transactions can be validated in a given block, this can potentially creating blockchain congestion and generate high fees, as described by Sokolov (2021). In December 2021, the gas fees for swapping transactions in Ethereum were on average \$93.30.[4] Conversely, L2 blockchains like Polygon can handle up to 10,000 transactions per second with minimal gas fees. Traders are highly sensitive to network fees and may postpone or abandon transactions when fees are high, a phenomenon documented in Easley et al. (2019). Similarly, the study by Cong et al. (2023a) reveals that L2 scaling solutions offer substantial reductions in operating costs (gas fees), enhanced data accuracy, and promote decentralization by decreasing market concentration and fostering increased participation.

L2 blockchains offer traders an alternative blockchain network to execute transactions with improved conditions that could incentivize them to transition exclusively to these platforms. However, the degree to which traders migrate to L2 will depend significantly on their perceptions of the relative security provided by L2 and the original L1. A major concern in moving from L1 to L2 blockchain is the security of transactions and asset ownership. Assessing the actual risks of trading in L2 networks compared with L1

[3] Speed of settlement/validation of the transactions also known as Finality.

[4] Gas fees for swapping are determined by the gas price and the number of gas units the smart contract uses to execute the transaction. In December 2021, the average gas price was 94 gwei. Data source: Etherscan data from swapping transactions on Uniswap.

involves many different aspects, so this is not as straightforward to identify.

We divide the risks into three main categories. First, there are smart contract risks – there could be a bug in the code or hacking that affects the contract, or admin key access, all of which could contribute to a centralization problem. The second risk relates to the use of wrapped ether tokens when trading in L2 pools. Wrapped tokens represent blockchain native tokens issued on a non-native blockchain, and the use of warped tokens thus includes liquidity risk.[5] Finally, there are validation risks that depend on the particular blockchain's validation (consensus) technology. The main risk of the validation process, known as a 51% attack,[6] occurs when someone or a group of people takes control of more than half of the validation authority of a blockchain network, thereby enabling them to create and manipulate transactions. To tackle Ethereum's scalability challenge, L2 solutions employ distinct validation mechanisms that expedite the validation process. However, this increased speed comes with potential risks.

It is difficult to determine how much riskier L2 blockchains are compared to L1 blockchains. One approach to estimating traders' perception of this risk is through surveys, but this method can be costly and may encounter validity issues due to the anonymous nature of users within the blockchain ecosystem. In our research, we propose an alternative approach by analyzing trading data from liquidity pools. This method captures traders' behavior and decisions, offering insights into their beliefs about risk.

Our inspiration for this approach comes from previous studies that have used market prices to reveal subjective beliefs. The core idea is that prices convey valuable information about people's perceptions, and by employing economic models, we can estimate these

[5] Native tokens are often used to pay gas fees or stake in DPoS systems. Ether (ETH) on Ethereum is an example of a native token.

[6] There have been several 51% attacks on blockchain networks. For example, there was an attack described in Garratt and van Oordt (2023), on Bitcoin Gold in May 2018. A more detailed explanation of these attacks in different blockchain validation technologies are provided in Sayeed and Marco-Gisbert (2019).

subjective views. For instance, past research has evaluated the value of statistical life (VSL) by comparing wages between riskier and safer jobs as discussed in Viscusi and Aldy (2003). The wage difference between these jobs reveals workers' beliefs about the value of their lives and the compensation they require to undertake risks. Likewise, prediction markets leverage prices to reveal subjective beliefs about the likelihood of events as seen in Wolfers and Zitzewitz (2006). This approach has also been applied to financial inquiries, such as explaining the equity premium puzzle by incorporating agents' subjective beliefs as Cecchetti et al. (2000) discuss.

By adopting this approach, we have developed a model that allows us to estimate traders' preferences for blockchain security using trading data. Our results shed light on how traders may consider risk and adjust their behavior.

We use detailed data with more than five million transactions on L1 pools and more than 14 million transactions on L2 pools. The total swapping value of these transactions add up to more than $358 billion dollars. The data includes different kinds of pools with different token types (WETH/ETH, WBTC, UCSD, USDT, DAI) and L2 networks (Polygon and Optimism). We collected more than one year's worth of data with a significant variation in gas prices. Transactions largely sort in a systematic pattern; specifically, we observed larger transactions in L1 and smaller transactions in the L2 network. We wished to understand why traders still use L1 if the L2 has higher scalability and lower fees. And why did transactions sort in this way?

We first checked whether these results are due to the pool size,[7] and we found that this does not fully explain the sorting pattern. With the liquidity pools pricing mechanism, each transaction directly impacts the exchange rate based on the size of the transaction relative to the pool's size. As the pool's liquidity increases, this effect decreases. However, a larger transaction leads to greater impact. L1 pools exist longer than L2 pools; they also

[7]Pool size refers to the amount of liquidity in the pool.

have higher liquidity during this time, thus offering better exchange rates in relatively large transactions. In relatively small transactions, the price effect is low in both pools, and it is less expensive to trade in L2. We calculate the optimal monetary switching point for traders to trade on the L1 network instead of the L2 network, considering the exchange rate and gas fees. Empirical data supports the notion that traders switch to L1 earlier than predicted.

As security considerations related to L2 could significantly influence traders' behavior, we employ a structural model to capture these concerns. This model helps bridge the gap between monetary predictions about traders' transition to L2 and the empirical evidence, particularly regarding the threshold for switching. We've determined that other explanations, such as price accuracy, liquidity concentration, adoption costs, and the benefits of holding assets on the original blockchain (L1), fall short in fully accounting for the observed divergence between theoretical predictions and actual behavior.

According to our model, traders anticipate a 0.68% and 3.29% probability of incurring a transaction value loss when trading on Polygon and Optimism, respectively, compared to L1. This risk perception is considerable, especially when juxtaposed with the transaction fee range of 0.01% - 0.3% levied by Uniswap for each trade. To our best knowledge, this is the first study that quantifies traders' beliefs about these security considerations using trading data from DeFi platforms. Our methodology can be expanded to estimate trader perceptions on other DeFi or payment platforms.

The rest of the paper is organized as follows: Section 1.2 introduces Decentralized exchanges, L2 Implementations and Constant Product Market Maker (CPMM). Section 1.3 and Section 1.4 describe the proposed model and methodology. Section 1.5 introduces our data and provides summary statistics, Section 1.6 provides results, and Section 1.7 concludes.

1.2 Decentralized Exchanges (DEX)

Most trading markets in the financial system are based on the traditional limit-order-book mechanism, in which buyers and sellers bid prices via a centralized organization that matches their bids. For years, cryptocurrencies and digital assets have mainly traded in centralized exchanges (CEX) such as Coinbase, which works with the limit-order book. Decentralized exchange (DEX) platforms have recently entered the crypto market, offering traders new decentralized options to trade. Since then, there has been a significant surge in the use of DEX protocols, as observed by Makridis et al. (2023). The most common DEX protocols are liquidity pools.

Liquidity pools are contracts that enable agents to provide liquidity (tokens/assets) to a smart contract on the blockchain. Traders can trade tokens/assets from these pools using a pricing rule written in the code. Most of these pools use a "bonding curve" pricing rule, which is a function of the supply of tokens/assets in the pool and is also known as Constant Product Market Maker (CPMM). These pools incentivize agents to provide liquidity and become liquidity providers by giving them a swapping fee for each swapping action from the pool. These swapping fees are around 0.01% - 1%, depending on the protocol and tokens/assets of the pool. Most pools have two tokens/assets that traders can swap.

The most prominent DEX platforms are Uniswap, Sushiswap, Balancer, and Bancor. This paper will focus on the Uniswap protocol, which is the largest one available. Most of these protocols work on the Ethereum blockchain (the L1 network). Recently, some liquidity-pool protocols such as Uniswap have started to support L2 networks, such as Polygon, Optimism, Arbitrum and Celo.[8] The Uniswap protocol was initiated in November 2021 and December 2021 to support swapping on the Optimism and Polygon

[8]Due to data limitations, we could not collect data from Arbitrum network. Celo is in its early stage, so we only show our analysis from Polygon and Optimism networks.

networks. We use Polygon and Optimism for our analysis as the alternative L2 networks for Ethereum (L1).

Recently, many researchers have explored DEX platforms in various directions. Some works (see, e.g., Lehar and Parlour, 2021; Barbon and Ranaldo, 2021) compare the various aspects of CEX and DEX platforms, such as liquidity provision, absence of arbitrage, price efficiency, and transactions cost. Additionally, many papers (e.g., Park, 2021; Capponi and Jia, 2021) have introduced the CPMM mechanism and discussed its properties and conceptual flaws. We instead explore how agents decide which network to use on DEX platforms, as well as the security aspect of those decisions.

1.2.1 L2 Implementations and Security

Measuring the actual risks of trading in L2 networks compared to L1 involves many different aspects and so it is difficult to identify. First, to trade in DEX, traders need to use a smart contract that involves some risks of having a bug in the code, hacking into the smart contract, and admin keys access, which could create a centralization problem, as mentioned in Tsankov et al., 2018; Schär, 2021. Integrating Uniswap methods (codes) with different blockchain networks and token types can create different security risks.

The second risk of trading in L2 compared with L1 is the use of wrapped ether tokens when trading in L2 pools. Wrapped tokens represent blockchain native tokens issued on a non-native blockchain. While using the L2 network, traders must use the wrapped tokens of Ether (Ethereum native token) to trade this token in L2. These wrapped tokens include liquidity risk, which depends on the wrapped token-issuing mechanism (Caldarelli, 2021). The recent case of the Ronin network hack, which led to the loss of more than $600M, contributed to shedding light on these risks.[9]

Finally, the validation risks depend on the blockchain's validation (consensus) tech-

[9] Data Source: BBC: https://www.bbc.com/news/technology-60933174

nology. To address the scalability problem of Ethereum (L1), L2 solutions use a different validation mechanism, which allows them to provide higher scalability and lower gas fee. Vitalik Buterin, one of the co-founders of Ethereum, already has identified that the biggest challenge of blockchain networks is achieving a decentralized payments system with high scalability and security. The main problem is that there is a trade-off between the three (decentralized, scalability, and security), and there is no technology that includes all the features together (known as the blockchain trilemma or scalability trilemma).[10]

With that in mind, L2 implementations try to provide higher scalability and lower fees, but this has some drawbacks. There are many different L2 solutions, each using a different approach. In our paper, we will focus on Polygon and Optimism, given our data set. Polygon is a side chain network with its native token (Matic) and validation mechanism (Proof-of-Stake), which means that the security is separate from the L1 network. Polygon is pegged to the Ethereum blockchain system, and users can transfer tokens from Polygon to Ethereum and vice versa using a bridge (see Thibault et al. (2022)).

Optimism uses a different L2 solution approach called optimistic rollups. In a rollup system like Optimism, transaction execution is moved to L2, and the data from these transactions are published on L1. Every Optimism transaction has two costs: An L2 (execution) fee and an L1 (data posting) fee. Most of the time, these fees are significantly lower than on the L1. Optimistic rollups use an "optimistic" validation approach in which the aggregators (who execute transactions on L2 and post them on L1) do not ask for proof of validity for each transaction execution. It means that the network supposes that the aggregators' transactions are valid. Another group of players, called verifiers, are monitoring the data published by the aggregators to deter any issues. A more detailed explanation of L2 implementations is provided in Thibault et al. (2022).

[10] See BIS (2022); Makarov and Schoar (2022)

The bottom line is that L2 solutions use a different validation process than L1; therefore, it is difficult to tell how much riskier they are than L1. This paper aims to shed some light on how trades react to the trade-off between scalability, security, and decentralization.

1.2.2 Constant Product Market Maker (CPMM)

Another difference between CEX and DEX, besides being decentralized, is the pricing mechanism; in CEX, the asset price is determined by the bids of the buyers and sellers, while in most DEX platforms it is determined by the pricing formula called the constant product market maker (CPMM). The CPMM formula works so that the product of the amount of tokens X and Y in the pool must remain the same. Let's consider a liquidity pool that contains x tokens of token X and y tokens of token Y [following the notation of Barbon and Ranaldo (2021)]. The CPMM pricing rule means that for any time t the product of the available tokens (X and Y) in the pool equals a constant k, which can be expressed as

$$x_t y_t = k$$

The amount of both tokens in the pool at time t determines the current pool price P_t which can be expressed as

$$P_t = \frac{y_t}{x_t}$$

Let f denote the protocol swapping fee which goes for the liquidity providers and $\varphi = 1 - f$ is what left for the trader to swap. If at time t+1 a trader wants to swap $\Delta(x)$ tokens X for getting Y tokens, we can calculate how many tokens Y $\Delta(y)$ she will get. CPMM states that

$$k = (x_t + \varphi\Delta x)(y_t - \Delta(y))$$

Solving for $\Delta(y)$:

$$\Delta(y) = y_t \frac{\varphi \Delta x}{x_t + \varphi \Delta x}$$

Further we can calculate the price of this swap,

$$P^t(\Delta x, y_t, x_t) = \frac{\Delta y}{\Delta x} = \frac{\varphi y_t}{x_t + \varphi \Delta x}$$

This formula's convexity relation implies that once traders have more demand for token X relative to token Y, the supply of this token in the pool will decrease, and thus its swapping price will increase. Additionally, this also implies that larger transactions have a larger price impact. However, the price impact would be small when the pool size is relatively large to the transaction size, as shown in Lehar and Parlour (2021).

These are essential properties of the liquidity pools that traders need to know. Once a trader can trade the same tokens X and Y in different networks, L1 or L2, the pool's size on each network could have a different price effect, one factor which will determine where the trader will choose to trade. The following section provides an extended model which allows the trader to pick which network they are willing to trade in.

1.3 Model

We follow the notations of Barbon and Ranaldo (2021) with an extension where trades can choose which network (i) they are willing to trade on.

Let i denote the blockchain network type, $i = 1$ is Ethereum (L1) network and $i = 2$ is Polygon or Optimisim (L2) network. Let X denote token 1, and Y denote token 2. $f_i =$ The protocol swapping fees at the network i and $\varphi_i = 1 - f_i$ is what left for the trader to swap. $T_{i,t} =$ the gas fee of swapping in network i at time t.

Given that the gas fee is paid by native tokens (Matic for Polygon and Ether for

Ethereum and Optimisim) for each network,[11] we will calculate these fees in US dollars units in our data analysis to have one unit of account. By the CPMM, we can calculate how many tokens Y Δy_i trader will get when she trades on network i, which can be expressed as:

$$\Delta y_i = y_{i,t} \frac{\varphi^i \Delta x}{x_{i,t} + \varphi^i \Delta x} \tag{1.1}$$

Let $P_{i,t}$ represents the price of making a transaction of value Δx for swapping token Y on network i.

$$P_{i,t}(\Delta x, x_{i,t}, y_{i,t}) = \frac{\Delta y_i}{\Delta x} = \frac{\varphi_i y_{i,t}}{x_{i,t} + \varphi_i \Delta x} \tag{1.2}$$

To scale our model so that we have only one unit of account for each transaction (token 1 - token 2) or (token 2 - token 1), we calculate the total value left for the trader after the swapping in token 2 units,[12] which can be expressed as,

$$P_{i,t}(\Delta x, x_{i,t}, y_{i,t}) \cdot (\Delta x)$$

To explain how agents behave in an environment where they can choose which network they are willing to trade, we specify a model in which agents need to maximize their utility when choosing between swapping in the L1 network (ETH) or L2 network (POLY or Optimisim). This maximization problem should consider two main aspects: how many token Y(X) traders get from swapping token X(Y) on each network and how many gas fees they pay. On top of that, we can add a behavioral parameter of traders' beliefs about the security of each network.

[11] Gas fees in the Optimisim are paid by Ether tokens. For more info: https://www.optimism.io/
[12] We will later convert them to US dollar values to have one unit of account.

Representative agent maximization problem:

$$\max_{i=0,1}\{i\cdot\pi_1\cdot u(P_{1,t}(\Delta x,x_{1,t},y_{1,t})\cdot\Delta x-T_{1,t})+(1-i)\cdot\pi_2\cdot u(P_{2,t}(\Delta x,x_{2,t},y_{2,t})\cdot\Delta x-T_{2,t})\} \quad (1.3)$$

Where T^{it} is the gas fees in each network at time t, $P_{i,t}(\Delta x, x_{i,t}, y_{i,t})$ represents how many token Y(X) traders get from swapping token X(Y) on each network (a function both of the transaction size Δx and the pool size $(x_{i,t}, y_{i,t})$ in each network), and π_i traders' beliefs of the probability of not losing ones' transaction wealth in network i, everything is scaled to be in US dollars units.[13]

Thus, our representative agent would choose network i if and only if

$$\pi_i \cdot u(P_{i,t}(\Delta x, x_{i,t}, y_{i,t}) \cdot \Delta x - T_{i,t}) \geq \pi_j \cdot u(P_{j,t}(\Delta x, x_{j,t}, y_{j,t}) \cdot \Delta x - T_{j,t})$$

We assume our representative agent is risk-neutral and maximizes the expected payoff.[14]

$$u(v) = v$$

There are two networks, L1 (Ethereum) and L2 (Polygon or Optimism), the agent chooses L2 network if and only if

$$\pi_{L2} \cdot (P^{L2,t}(\Delta x, x^{L2,t}, y^{L2,t}) \cdot (\Delta x) - T^{L2,t}) \geq \pi_{L1} \cdot (P^{L1,t}(\Delta x, x^{eth,t}, y^{L1,t}) \cdot (\Delta x) - T^{L1,t})$$

[13] Our data resource allow us to convert everything to US dollars value and have one unit of account.

[14] There is strong evidence from many different researchers, as summarized in BIS (2022), that most of the traders in the crypto markets are risk-seeking. Assuming that the representative agent is risk neutral is a conservative assumption for our belief elicitation.

Let $w = \Delta x$ and $P^{i,t} = P^{i,t}(\Delta x, x^{i,t}, y^{i,t})$ we can further write:

$$\begin{aligned}
\pi_{L2} \cdot (w \cdot P_{L2,t} - T_{L2,t}) &\geq \pi_{L1} \cdot (w \cdot P_{L1,t} - T_{L1,t}) \\
(\pi_{L2} \cdot P_{L2,t} - \pi_{L1} \cdot P_{L1,t})w &\geq \pi_{L2} \cdot T_{L2,t} - \pi_{L1} \cdot T_{L1,t} \\
w &\leq \frac{\pi_{L2} \cdot T_{L2,t} - \pi_{L1} \cdot T_{L1,t}}{\pi_{L2} \cdot P_{L2,t} - \pi_{L1} \cdot P_{L1,t}} \\
w &\leq \frac{\pi_{L1} \cdot T_{L1,t} - \pi_{L2} \cdot T_{L2,t}}{\pi_{L1} \cdot P_{L1,t} - \pi_{L2} \cdot P_{L2,t}}
\end{aligned}$$

$$w = \frac{\pi_{L1} \cdot T_{L1,t} - \pi_{L2} \cdot T_{L2,t}}{\pi_{L1} \cdot P_{L1,t} - \pi_{L2} \cdot P_{L2,t}} \tag{1.4}$$

This is the representative agent's threshold transaction size in which she will switch from the L2 network to L1.[15]

Consider when π_i in each network are equal, meaning there's no security concerns of L2 relative to L1, $w^\star$ represents the optimal threshold at which the representative agent should switch from trading in the L2 network to Ethereum. At any given time t with given pools sizes and gas fees, we can calculate the theoretical $w^\star$, which we will discuss in more detail in Section 1.6.1.

$$w^* = \frac{T_{L1,t} - T_{L2,t}}{P_{L1,t} - P_{L2,t}} \tag{1.5}$$

When the representative agent's empirical threshold, $\hat{w}$, is smaller than $w^\star$, it means agents are switching to Ethereum even though it is less profitable. Section 1.6.3 will provide a comparison between $\hat{w}$ and $w^\star$ and a robustness test to check if $\hat{w}$ is statistically significantly smaller than $w^\star$. This deviation will be captured by the security parameter in our model. We can estimate the representative agent's beliefs on security, the chance

[15] Alternative interpretation for the representative agent will be that traders have different trading needs, but the representative agent can capture as the marginal trader that all traders with larger (smaller) trading need trade on L1 (L2) networks.

of losing the transaction value when trading on L2 compared to on L1, as,

$$S_{L2,L1} = 1 - \frac{\hat{\pi_{L2}}}{\hat{\pi_{L1}}} = 1 - \frac{P_{2,t}(\hat{w}, x_{2,t}, y_{2,t}) \cdot \hat{w} - T_{2,t}}{P_{1,t}(\hat{w}, x_{1,t}, y_{1,t}) \cdot \hat{w} - T_{1,t}} \tag{1.6}$$

this will be discussed in Section 1.6.3.

1.4 Methodology

This section delves into our methodological approach when estimating agents' beliefs about security using trading data from a decentralized exchange platform. First, we focused on traders' decisions from decentralized exchanges to estimate preferences for blockchain security, mainly because this is the most extensive data set available on DeFi with millions of transactions.[16] This data set allows us to get the most updated information based on high-frequency decisions from the traders that capture some hidden information about their preferences for blockchain security. Using our model and empirical data, we will measure traders' preferences. This section covers our main methodological choices and the step-by-step estimation process of traders' security beliefs.

1.4.1 Representative agent

In our model, we have incorporated a representative agent. The primary reason for this selection is our classification model, which estimates the agents' empirical switching points using a logistic classification model. This will be covered in more detail in Section 1.6.2. Given that our classification model gives us the aggregate empirical swathing

[16] Another possible way to estimate security concerns can be through long-term investments, such as providing liquidity to the liquidity pools. However, this will also mean having fewer data points (much fewer liquidity providers than traders). Additionally, liquidity providers update their position less frequently than trading data, which could be less informative when evaluating that security measurement on a daily basis.

point, we need our model to follow the same logic.

To ensure robust insights from traders' decisions, we require an extensive data set of transactions that can provide reliable results for the logistic model with high accuracy (low false positives and false negative rates). This generates a trade-off, as a large number of observations is crucial for the success of the logistic model, but on the other hand, can reduce accuracy in different parts of the model. Therefore, as explained below, we needed to make methodological decisions about our data set and estimation method.

1.4.2 Pools and Blockchains Networks Selections

We followed this selection process: first, we collected trading data from all networks and pools in Uniswap V3 protocol,[17] and we identified some key features of the pools, like average transactions per day, tokens involved, pool transaction fee, etc. We decided only to focus on Polygon and Optimism as the L2 networks because they were the biggest available during our sample period. Then, we sorted pools that were only available both on L1 and L2, shared the same liquidity pool fee, and shared the same token types. Finally, we chose from only pools with more than 200 average daily transactions and a minimum of 70 or more transactions per day. This selection process allows us to focus on pools with enough transaction data to give reliable results for our logistic classification model to estimate the empirical switch point. Also, this selection process allows drop pools with less stable/familiar token types that have a lot of price manipulation and generate a lot of noise.

[17] Uniswap V2 only operates on Ethereum (L1), and there are no pools available on L2 networks.

1.4.3 Frequency of Estimation

We decided to estimate the security beliefs daily, given our date set. We observed that most of the pools have low amount of transactions per hour to run the logistic model hourly and get robust results. However, this means that we needed to use the average daily gas price for our model, which can reduce the model's accuracy given that those prices vary over the day; this is one of the limitations and trade-off decisions we were forced to make. In addition, we took the liquidity available for trade as a sticky during the day without allowing this to change when liquidity providers change their positions (use the starting liquidity in a given day for calculation), which also reduced the model accuracy. Given the consistency of the gap between the representative agent's threshold and the empirical one in our result, this suggests that this methodology decision has a low impact on accuracy.

1.4.4 Uniswap V2/V3

Our model is based on Uniswap V2 (Adams et al., 2020), where traders can trade from the pool without liquidity restrictions. However, Uniswap V3 (Adams et al., 2021) works in different mechanisms, providing liquidity with some price limits. Unfortunately, we cannot analyze our data with the new V3 mechanism due to data limitations. However, Chemaya and Liu (2024) shows that the V2 model can provide highly accurate results for V3 data, especially in the top big pools with high accuracy, 98% of the transactions have less than 0.01% price deviation. In Section 1.6.4, we test our data set accuracy using the V2 model and get very high predictive power to ensure this will not impact our main results.

1.4.5 Estimation process

1. We collected all Uniswap V3 trading data for our sample period.
2. Following the selection methodology in Section 1.4.2, we only analyze pools that meet our selection requirements, guaranteeing the accuracy of our analysis and reducing noise.
3. Each day, we calculate the average sweeping fees in US dollars on all our networks, representing the daily average gas costs to swap that traders paid in each network. Our model captures these gas costs as $T_{L2,t}$ for Polygon and Optimism and $T_{L1,t}$ for Ethereum.
4. For each day, we collected the initial liquidity available in each pool at the beginning of the day, which is captured as $x_{i,t}, y_{i,t}$.
5. Based on the daily gas prices data and the liquidity available, we use our model for every two identical pools on L1 and L2 and calculate the representative agent's theoretical threshold w^*.
6. Using empirical data that capture traders' decisions about which network they decided to trade, we use our classification model covered in Section 1.6.3 to calculate the representative agent's empirical threshold $\hat{w}$. We also use bootstrap to construct the 95% CI for $\hat{w}$.
7. Finally, using Equation 1.6 from our model we estimate the representative agent's daily beliefs on security, which is the relative chance of losing the transaction value when trading on L2 compared to L1.

1.5 Data

This paper collects transactions from a total of 21 liquidity pools on the Uniswap V3 protocol,[18] including 8 in the L1 network (ETH) and 13 in L2 network (6 in POLY, 7 in OPT), following the selection process covered in Section 1.4.2.

These pools allow traders to trade the same token types in L1 and L2 (POLY and OPT) and have a sufficient amount of transactions per day. These pools jointly contribute 63% of the transactions on the Uniswap when considering pools that are available for trades to trade the same pair of tokens on L1 and L2.[19] Six different tokens are swapped in these pools (DAI, USDC, USDT, WMATIC/MATIC, WBTC, WETH/ETH), and the pool fee ranges from 0.01% to 0.3%.

Our main analysis will focus on three liquidity pools, with one from the three networks (ETH, POLY, OPT). Each of the three pools have the same pair of tokens (USDC and WETH/ETH),[20] and have the same protocol swapping fees $f_1 = f_2 = 0.05\%$. Those pools are the biggest ones in our data set and contribute more than 50% of the daily transactions. Utilizing blockchain explorer services (Uniswap Data Extractooor),[21] we are able to track each and every Erc-20 tokens transactions that happened in the liquidity pools.[22] We collected data from December 22, 2021, the launch date of the Polygon network pool, until December 31, 2022.[23]

[18]The Uniswap protocol is a peer-to-peer system designed for exchanging cryptocurrencies (ERC-20 Tokens).https://docs.uniswap.org/protocol/introduction

[19]Many pairs of tokens are network specific and can be traded only on one of the networks. For example, on 12/05/2022, only 53.36% of the transactions on Ethereum pools were with tokens that were available on L2 (Data source Uniswap Data Extractooor); there was a similar situation with L2 pools: only 55.39% (46.56%) of the transactions on POLY(OPT) pools were with tokens that were available on L1.

[20]The Ether (ETH) tokens on the L2 blockchains are wrapped tokens (WETH).

[21]https://www.uniswap.shippooor.xyz/

[22]The ERC-20 introduces a standard for Fungible Tokens, in other words, they have a property that makes each token be exactly the same (in type and value) as another token. https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[23]Optimism pool lunching day was one month before on 11/12/2021.

Within this 12 months period, we obtained a total of 2,789,976 swapping transactions (exchange between USDC and WETH/ETH) from L1 ETH network, 4,991,764 swapping transactions from L2 POLY network, and 4,323,672 swapping transactions from L2 OPT network.[24] That is a total of more than 12 million swapping transactions, which resulted in a sum of $237 billion. The distributions of the amount swapped in the three platforms differ during the time of interest (Figure 1.1). We modified the magnitude of the values for large numbers, so this graph is more readable. All transactions with a value greater than 10,000 dollars are over-written to 10,000 (for this graph only). We observe that the L2 distribution is right-skewed for most smaller transactions (on POLY less than $923, on OPT less than $64). On the other hand, the majority of the transactions on L1 ETH are larger than $5,002, five to a hundred times more than on L2.

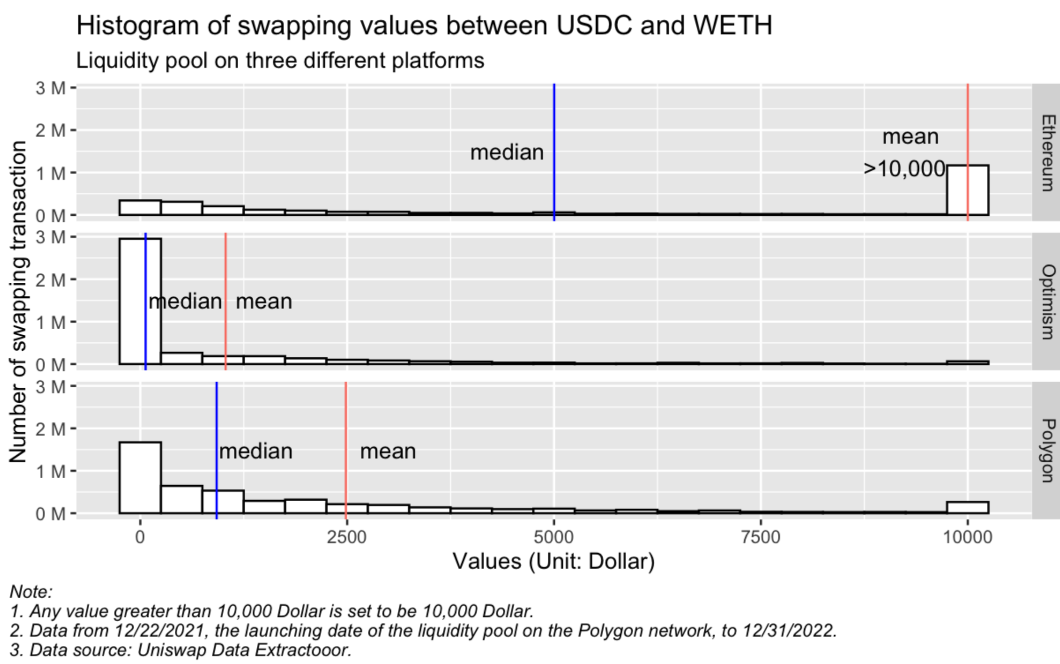
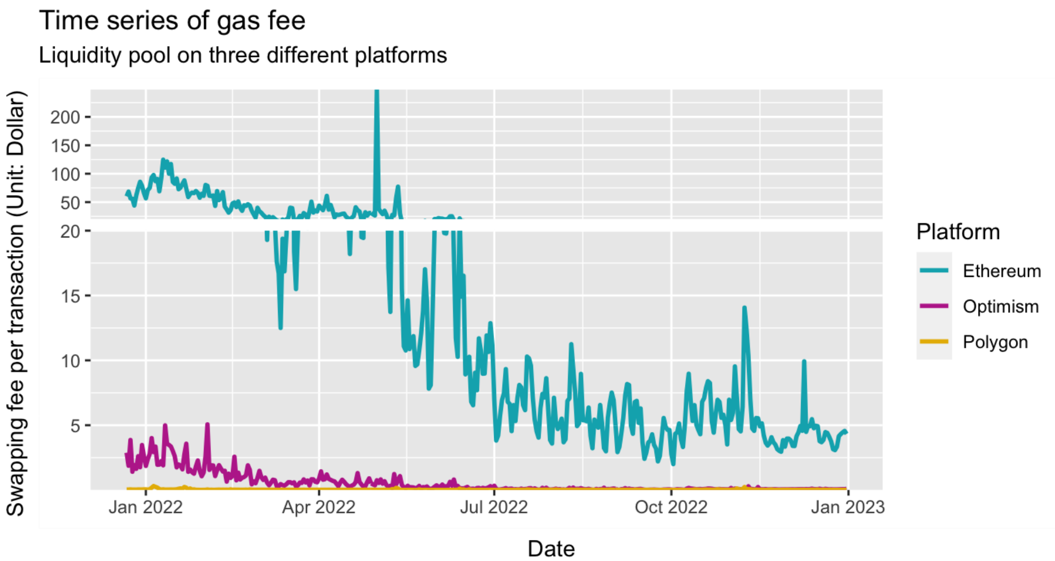


Figure 1.1: Histogram of swapping values between USDC and WETH on three platforms

[24] ETH Contract address: 0x88e6a0c2ddd26feeb64f039a2c41296fcb3f5640.
POLY Contract address: 0x45dda9cb7c25131df268515131f647d726f50608.
OPT Contract address: 0x85149247691df622eaf1a8bd0cafd40bc45154a9.

We also calculated the daily mean gas fee for swapping in each network.[25] During this time period, the average daily mean gas fee for swapping was \$22.93 on L1, \$0.559 on OPT, and \$0.030 on POLY.



Note:
1. Swapping fee (Gas prices) in L2 (Polygon/Optimism) are always much less than in L1 (Ethereum).
2. There are some fluctuation of the fee in L2 (Polygon/Optimism) too.
3. Data source: Etherscan, Polyscan, & Optimistic.etherscan.

Figure 1.2: Time series of gas fee

We also collected data on the daily size of the three liquidity pools from Uniswap. ETH's pool had a higher pool size during our observation dates, with an average of \$277.6M, while the average size of POLY was \$13.40M, and the average size of OPT was only \$4.72M. Figure 1.3 is the time series presentation of the three liquidity pools' size during this time period.

1.6 Results

This massive trading data from liquidity pools capture traders' behavior and decisions. In this section, we will use our model to analyze that data and estimate traders'

[25] Data Source: Etherscan, Polyscan & Optimistic.etherscan.

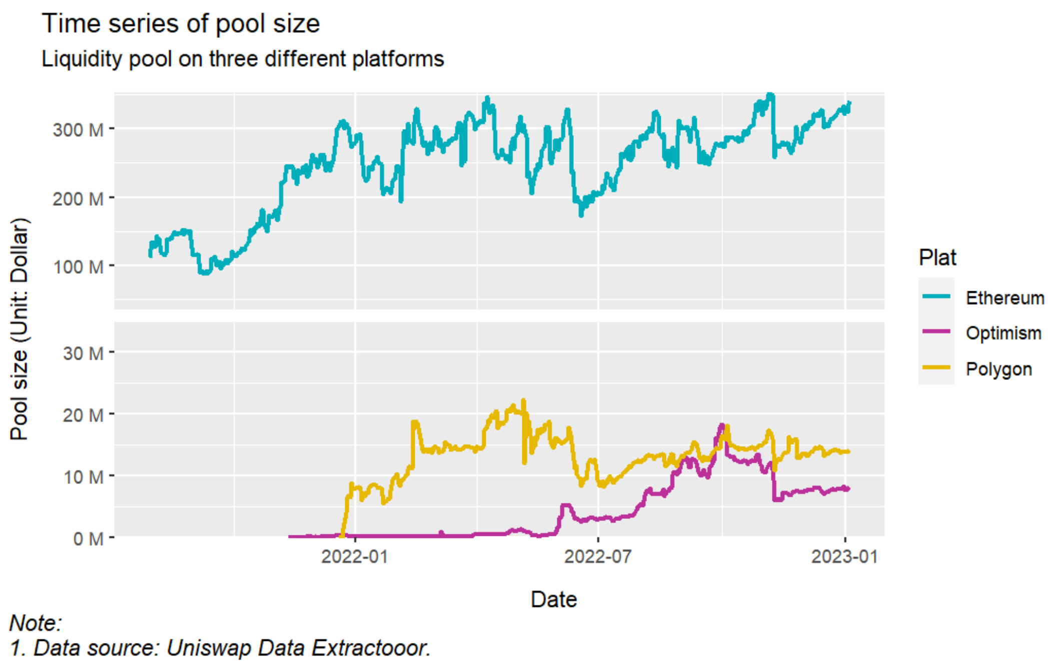


Figure 1.3: Time series of pool size

preferences for blockchain security. We will mainly discuss the result from the three pools introduced in Section 1.5, which are the biggest ones in our data set. We will first show details of estimating the security of the POLY (L2) network relative to ETH (L1) using POLY & ETH, USDC/WETH 0.05% pools; then we will show results from OPT (L2) network and other pools.

We estimate traders' belief in security by studying behavior deviation from monetarily optimal decisions. In Section 1.6.1, we calculate the monetarily optimal switching point W^*, as a function of liquidity pool size and transaction fee. Section 1.6.2 presents our empirical strategy for finding the actual switching point $\hat{W}$ from the data. Section 1.6.3 summarizes the security concern from W^* and $\hat{W}$ based on our model while Section 1.6.4 cover why alternative explanations are unlikely to explain our main results. Finally Section 1.6.5 further shows our model's generalizability to other networks and pairs of tokens in different pools.

1.6.1 Monetarily optimal switching point W^*

Why do traders still use L1 if the L2 has higher scalability and lower fees? The exchange rate of a swapping transaction in the liquidity pool is determined by the liquidity pool size and the size of the transaction itself (recall Equation 1.2 in the model section). The higher liquidity in L1 makes L1 pools have a lower price impact, offering better exchange rates for relatively big transactions. On the other hand, if the transaction size is relatively small, the price effect in both pools is low; considering L1 has a higher gas fee, it would then be cheaper to trade in L2. Given pool size and gas fee at a low swapping size, it would be cheaper to swap on L2; at some switching point, it would be optimal to switch to swap on L1.

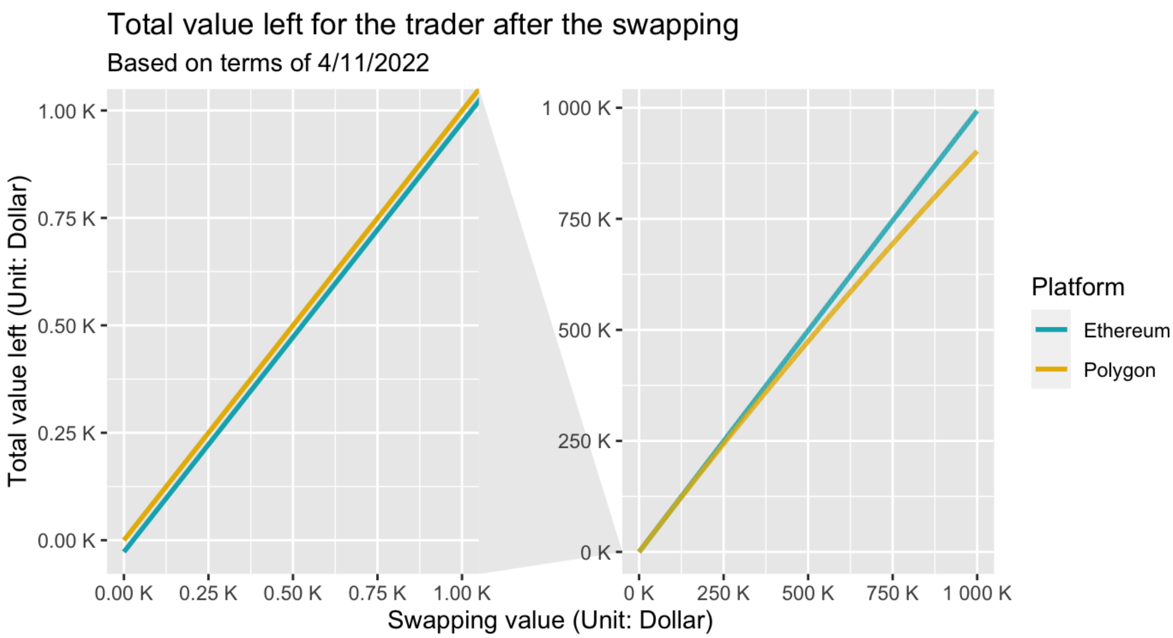
Take April 11th, an arbitrary day, as an example: the ETH, USDC/WETH 0.05% pool size is \$322.93M, and the POLY, USDC/WETH 0.05% pool size is \$18.58M. The mean gas fee is \$27.37 per swapping transaction on ETH, and \$0.02 per swapping transaction on POLY. Following Equation 1.2, while also taking into consideration the gas fee, we calculate the total value left for the trader after the swapping. As shown in Figure 1.4,[26] it is better to swap on Polygon at first, and then better on Ethereum once the swapping value becomes larger (to be exact, once the swapping value is larger than $\$16,442$).

We calculate this monetarily optimal switching point W^* for all dates in our data, and get Figure 1.5.

1.6.2 Empirical switching point $\hat{W}$

The monetarily optimal switching point W^* can explain some reasoning behind traders trading on both platforms and separated in a certain way, yet empirical data

[26] Using R package *ggforce* (Pedersen, 2021).



Note:
1. Ethereum gas fee $27.37, Polygon gas fee $0.02, Ethereum pool size $322.9M, Polygon pool size $18.6M.
2. Notice that at low value, swapping in Polygon would be better than in Ethereum, but as the swapping value increase, at some switching point, swapping in Ethereum would be better.

Figure 1.4: Total value left for the trader after swapping

supports that traders are switching to L1 for much lower transactions.

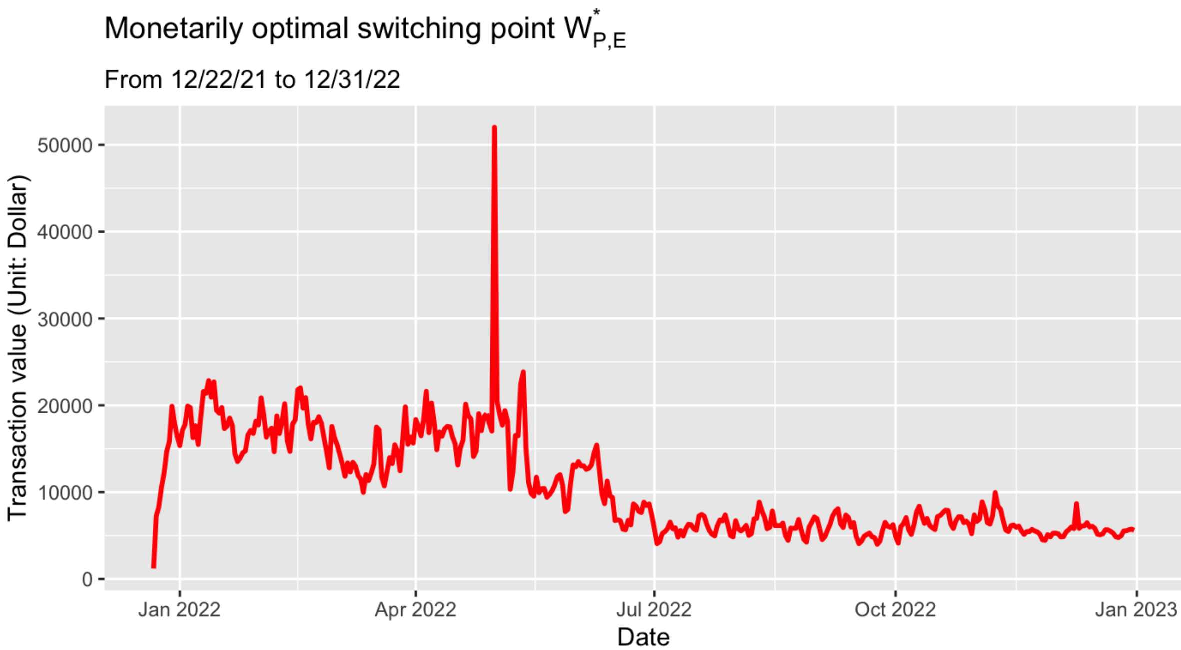
In order to find the representative agent's empirical threshold from the data, given that we are facing a binary classification problem, we implement a binary logistic model.[27]

A binary logistic model states that the probability of outcome Y belongs to class y given predictor W equal to a logistic function.

$$Pr(Y = 1|W) = \frac{e^{\beta_0+\beta_1 W}}{1 + e^{\beta_0+\beta_1 W}}$$

There are two classes, transaction is on L2 ($Y = 1$), and transaction is on Ethereum ($Y = 0$). Our predictor W is the transaction value (in Dollar unit). It is a linear model

[27] A competing method is linear discriminant analysis, a linear method in classification, see Appendix A.1 for more discussion.



Note:
1. W[P-E] represents the monetarily optimal threshold at which the representative agent should switch from trading in the Polygon network to Ethereum.*

Figure 1.5: Monetarily optimal switching point $W^*_{P,E}$

as the logit, or log-odds, is linear in W.

$$log(\frac{Pr(Y=1|W)}{1-Pr(Y=1|W)}) = \beta_0 + \beta_1 W$$

We report the the summary of the logit regression result using 2022/4/11 data in Table 1.1.[28] The coefficient for W (Transition value) is negative, meaning the higher the transaction value is, the lower the log odd, that is, the lower the probability of this transaction is on L2 (and higher probability is on Ethereum). Both the interception and the coefficient for W are statistically significant.

To find the empirical threshold value $\hat{W}$, we obtain $\hat{\beta}_0$ and $\hat{\beta}_1$ from the regression,

[28] Using R (R Core Team, 2020), and R package *texreg* (Leifeld, 2013).

	Logit Regression
(Intercept)	1.72^{***}
	(0.02)
Transaction value	$-1.278e^{-4***}$
(in Dollar)	$(2.822e^{-6})$
AIC	18401.68
BIC	18417.62
Log Likelihood	−9198.84
Deviance	18397.68
Num. obs.	21364

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1.1: Binary Logistic Model result: 2022/4/11 data

and the best threshold probability $\hat{Pr}(Y = 1|W)$.[29]

$$\hat{W} = \frac{1}{\hat{\beta}_1}(log(\frac{\hat{Pr}(Y = 1|W)}{1 - \hat{Pr}(Y = 1|W)}) - \hat{\beta}_0)$$

Figure 1.6 shows a time series of the calculated $\hat{W}$. On 2022/4/11, this empirical threshold is $\$3,469$.

1.6.3 Estimating belief on security

Our structure model captures traders' security concerns about L2. This security concern can explain the gap between the switching point from the pure monetary prediction W^* and empirical $\hat{W}$. Figure 1.7 is a direct comparison of the two time series in one graph. The shaded area around the blue line represent the area of 95% confidence interval obtained by running bootstrap on the transaction data.[30]

Notice that the monetarily optimal switching point W^* is always above the empirical threshold $\hat{W}$. This is consistent with our prediction of the model. The intuition is

[29] See Appendix A.2 for more details.

[30] Using R package *boot* (Davison and Hinkley, 1997).

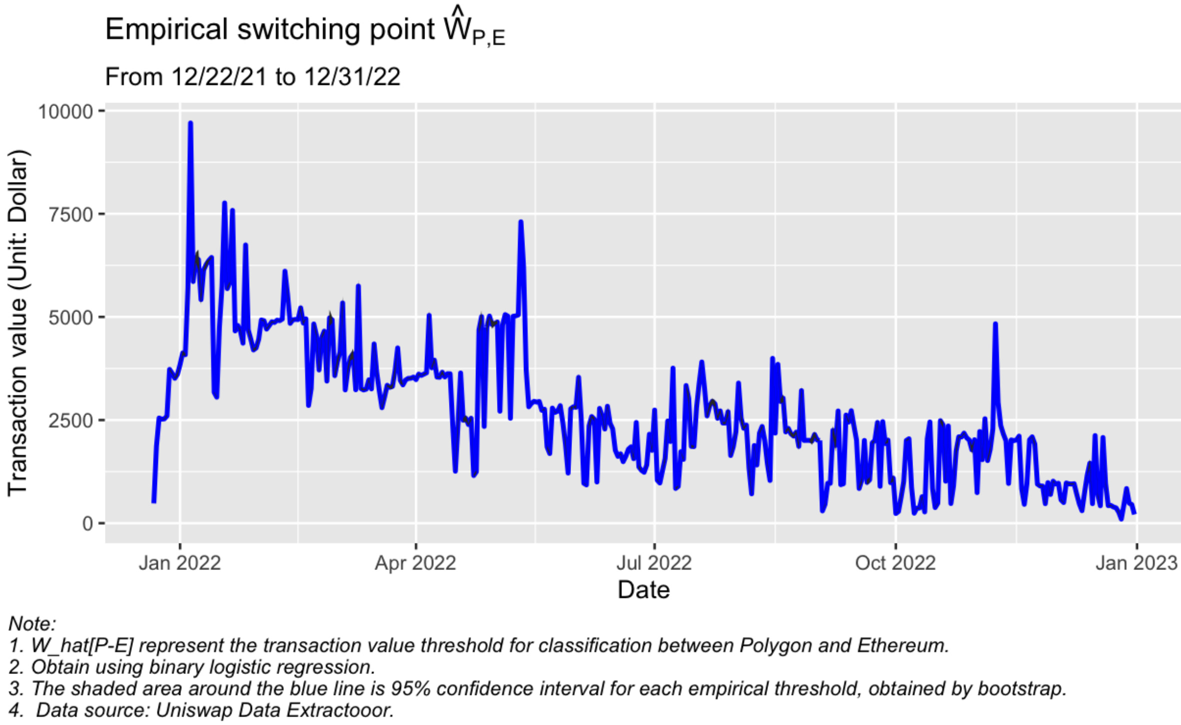
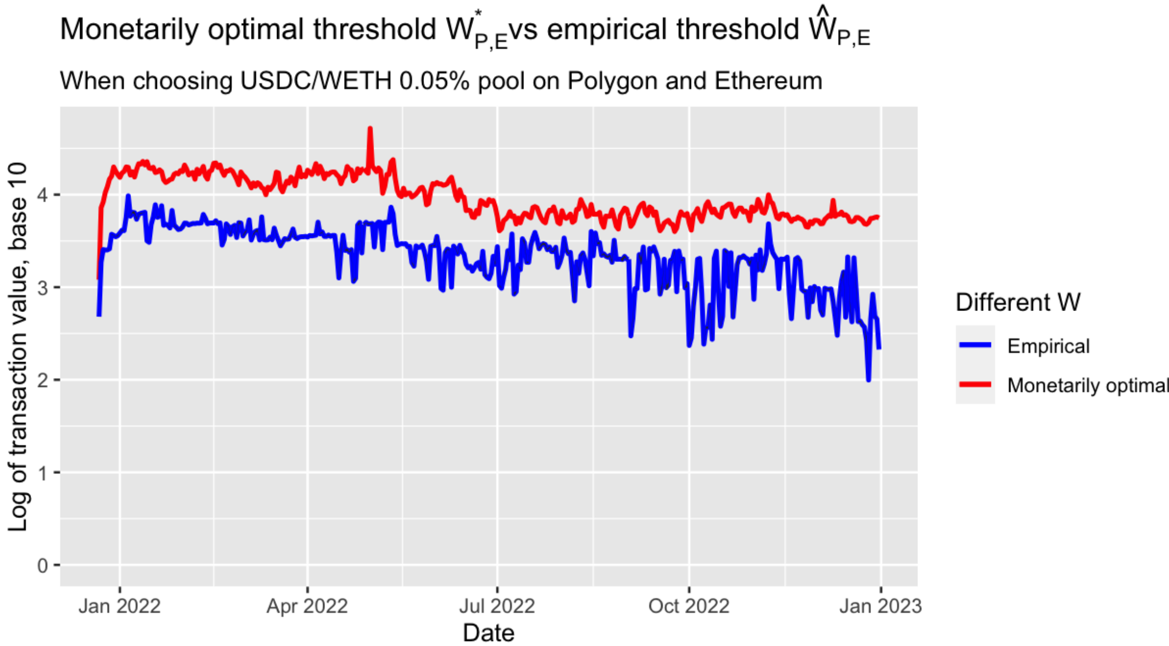


Figure 1.6: Empirical switching point $\hat{W}_{P-E}$

as follows, due to the security concerns, people would switch from trading in Polygon network to Ethereum network earlier, thus the gap we observe from empirical evidence and model prediction, that is, the gap of $\hat{W}$ and W^*.

As defined in Section 1.3, $S_{L2,L1}$, our estimator of the representative agent's beliefs on security, is the chance of losing the transaction value when trading on L2 compared to on L1. $S_{L2,L1}$ should be greater than 0, as the probability of not losing ones' transaction wealth in L1 network should always be smaller than on L2 network, since Layer-2 network building on Layer-1 network. The calculation confirms this as the estimator is always greater than 0.

Here in Figure 1.8, the y axis is $S_{L2,L1} = S_{P,E}$; the higher the estimator, the more security concerns our representative agent holds on the Polygon network. The mean of the analysis time period is 0.751%, suggesting that in this period, on average, agents think



Note:
1. The shaded area around the blue line is 95% confidence interval for each empirical threshold, obtained by bootstrap.
2. Data source: Uniswap Data Extractooor.

Figure 1.7: Monetarily Optimal Threshold $W^{*}_{P,E}$ vs Empirical Threshold $\hat{W}_{P,E}$

there is 0.751% more chance of losing transactions on Polygon compared to Ethereum. The median is 0.554%.

The ratio is significantly different from 0 (greater than 0), as its 95% confidence interval, obtained by running bootstrap on the transaction data, never cover 0.

1.6.4 Investigating Alternative Factors to Explain the Results

In this section, we will explore alternative explanations that could account for the observed gap between the monetary optimal switching point and the empirical one. Specifically, we will closely examine potential explanations related to factors such as price accuracy, adoption cost, and the advantages of owning assets on L1. However, we find that these explanations are less likely to account for the observed gap, as the available

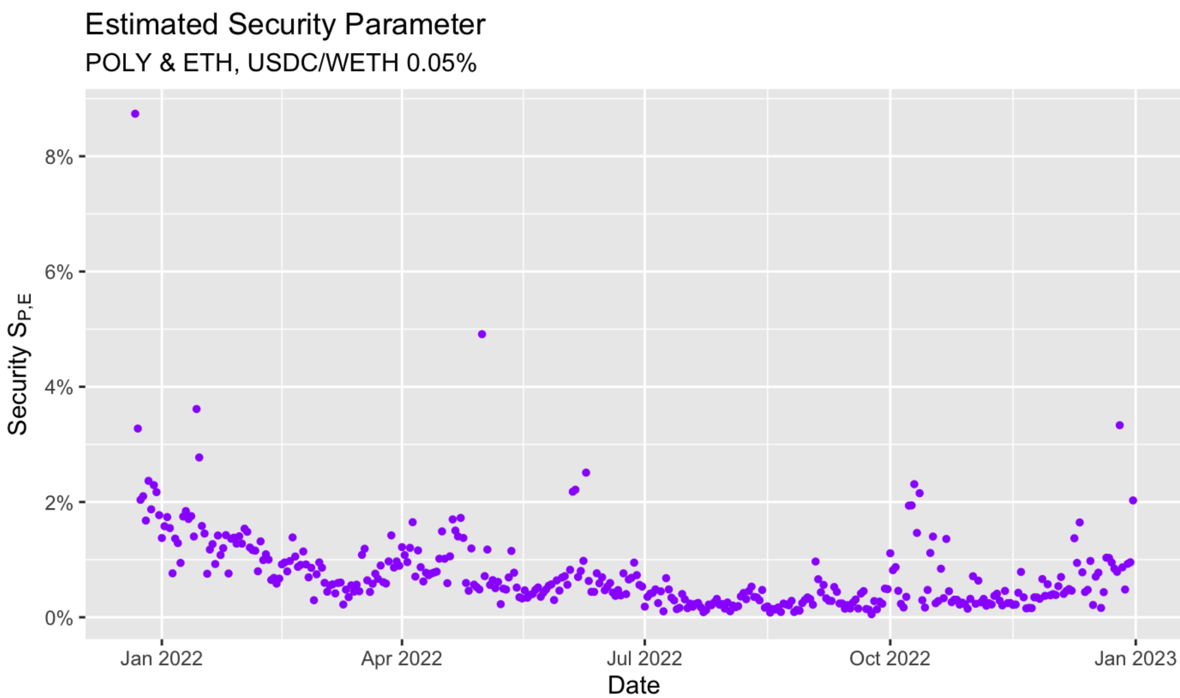


Figure 1.8: Estimated Security Parameter

data does not provide strong support for them.[31]

Moreover, taking into account the blockchain trilemma, as discussed in Section 1.2.1, which highlights the trade-off between scalability and security, users who transition to more scalable networks like L2 should be mindful of this trade-off. In light of this, we assert that security assumes a pivotal role in explaining the observed gap and our research findings.

[31] An additional argument could be that our data includes some trade-washing and price manipulation transactions, which can generate noise that might explain the observed gap, as described in Amiram et al. (2021); Cong et al. (2023b); Victor and Weintraud (2021). To address this, we conducted a robustness check, focusing only on users who trade two or fewer transaction per day, as they are less likely to engage in this price manipulation act. This analysis yielded similar results, as detailed in the Appendix A.6.

Price Accuracy

One argument suggests that the gap between the monetary optimal switching point and the empirical one may be attributed to different prices across networks. However, our data indicates that prices across networks, particularly in large pools with well-known tokens, are equal between L1 and L2. Figure 1.9 plots the time series of WETH price relative to USDC in the 0.05% fee liquidity pool for all three platforms. Prices across L1 ETH, L2 POLYGON, and L2 OPTIMISM are almost identical such that the time series overlaps.

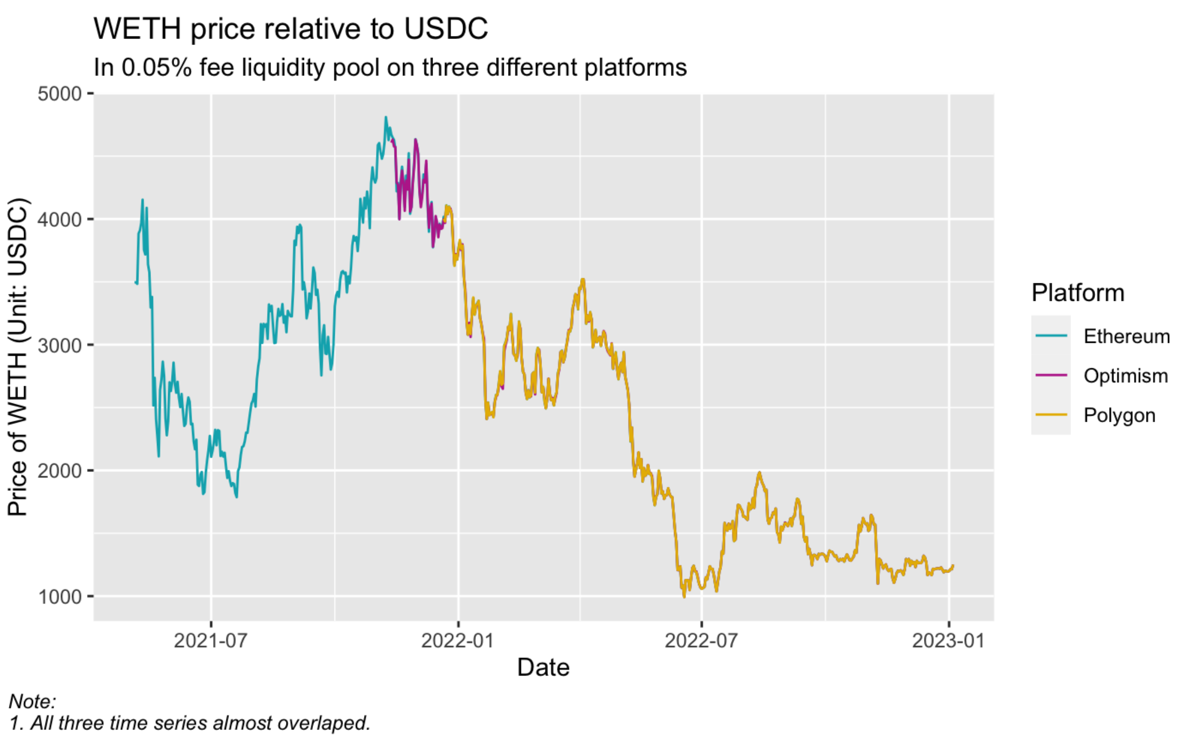


Figure 1.9: WETH/USDC price in 0.05% fee liquidity pool

This suggests that arbitrageurs operate across layers, potentially holding L2 and L1 accounts and periodically transferring funds between networks to avoid incurring cross-chain costs. This concept resembles arbitrageurs trading on both centralized exchanges (CEX) and decentralized exchanges (DEX), where moving funds from DeFi to CeFi can be costly. The presence of arbitrageurs across networks weakens the argument that traders do not monitor prices across networks, and the potential gap could be due to monitoring costs.

However, if this were the case, we would anticipate greater price volatility between networks, and the security parameter might exhibit more noise or even negative values. In light of our model, when prices on L1 are more favorable than L2, price inaccuracy should manifest as a lower security parameter and potentially even as negative values, signifying that L2 is more secure than L1. Nevertheless, our data consistently demonstrates positive security parameters, indicating the accuracy and consistency of exchange prices on L2 in comparison to L1.

Liquidity concentration

Lehar et al. (2023) suggests that lower gas fees lead liquidity providers (LPs) to manage their positions on V3 pools more actively on L2 networks. This will reduce the liquidity available for traders on L2 pools in a given price range.[32] Caparros et al. (2023) shows that lower gas fees on Polygon allow LPs to update their positions more frequently than on Ethereum (L1). Given our data and methodology restrictions, we could not consider this in our model. Still, we run the following robustness test to ensure that this is not the primary driver of our main results. Given the large number of transactions we observe, we can check the predictive power of our V2 model vs. the empirical execution price on V3 data considering the LP positions, following the methodology in Chemaya and Liu (2024).

[32] Another direction could be for sophisticated traders to strategically execute small transactions on L2 to push LPs and arbitrageurs to update prices in the pool and reduce the trader's price impact. If this is the case, our results will contradict this argument. We observe a much higher gap between the empirical and theoretical switching points on Optimism than Polygon. If gas costs are much lower on Polygon, it should yield the opposite results based on this argument.

Network	Size of Txn	Txn [34]	$\tilde{d}$ within [33] 0.5%	1%	5%
Ethereum	all	2,745,838	99.96%	99.97%	99.98%
Polygon	$\leq \$3,994$	3,989,514	99.7%	99.8%	99.85%
	$> \$3,994 \cap \leq \$52,010$ [35]	914,176	99.7%	99.7%	99.82%
	$> \$52,010$	5,568	99.3%	99.6%	99.80%
Optimism	$\leq \$983$	3,288,508	96.4%	98.4%	99.20%
	$> \$983 \cap \leq \$11,826$ [36]	991,095	92.9%	95.9%	98.5%
	$> \$11,826$	34,520	75.7%	84.6%	93.6%

Table 1.2: Percentage of transactions' $\tilde{d}$ within threshold.

Table 1.2 suggests that liquidity was available for traders to trade in L2 networks even for more significant transactions than the theoretical switching point range value (we calculate the min and max of those values during our sample period). Some transactions (the ones that fail in the classification model) indeed execute significant transactions on L2, and the predicted price is still high in those cases, with most of the transactions having less than $\tilde{d} = 0.5\%$ price deviation. This suggests that liquidity supply effects due to LPs managing their positions can't explain the observed gap.

Adoption Cost

The use of L2 solutions entails an adoption cost initially, which may influence users' transition from L1 to L2. To adopt L2 solutions, users need to transfer funds from L1 to L2 (involving bridging mechanisms) and create a new wallet on L2, requiring familiarity with the L2 network. Although L2 solutions aim to streamline the process by enabling the use of the same digital wallet and wallet ID across L1 and L2 networks, the adoption

[33] $\tilde{d}$ is the absolute percentage deviation of V2 model predictions for swap outcome from the actual exchange outcome of the V3 data. Methodology adopted from Chemaya and Liu (2024).

[34] Taking transactions between Jan 1^{st}, 2022, and Dec 31^{st}, 2022.

[35] $\min w^*_{P,E}$ and $\max w^*_{P,E}$ during the analysis period.

[36] $\min w^*_{O,E}$ and $\max w^*_{O,E}$ during the analysis period.

cost may still be substantial for certain users, depending on users' level of sophistication and familiarity with these systems.

Given that our model employs a representative agent framework, it might overlook this adoption cost concern, and it is possible that small users find the adoption cost relatively affordable while wealthier users face higher barriers. To assess this argument, we conduct analyses on a subset of the data, namely, on wallets that traded on both L1 and L2 during the period of analysis. This allow us to examine whether users who hold funds in both L1 and L2 exhibit the same pattern of smaller transactions on L2 and relatively larger ones on L1, as well as whether their switching point is lower than the optimal one.

Table 1.3 gives an overview of the subset data. 17,246 unique wallet addresses swapped on both ETH and POLY for the USDC/WETH 0.05% pools. Together they contribute to about 5%-7% of the total transactions we observed in this period, and about 1%-6% of the total transaction value on each platform.

We then conduct the same analysis as in Section 1.6.3 to obtain the mean (median) estimated security parameter. Our findings not only support that individual users follow the same patterns as our representative agent, but also highlight that the representative agent result is a lower bound of this security estimate, thereby weakening the adoption cost argument. A similar plot to Figure 1.8 can be found in Appendix A.3 for this subset data, Figure A.6.

Finally, we test whether the observed gap captures the slow adoption of the L2 networks, which should disappear over time when more traders adopt those networks. Figure A.13 in Appendix A.7 shows data that while the security estimation on Polygon has slow movement, there was massive adoption on Polygon with many new users joining the network.

	DEX	# TXN (%)	Total Volume (%)	Mean Estimated Security Parameter (Median)
Full	ETH	2,789,976	$220,065,992,854	
	POLY	4,991,764	$12,401,731,851	0.751% (0.554%)
Subset	ETH	142,538 (5.1%)	$2,119,468,116 (1.0%)	
	POLY	346,088 (6.9%)	$ 697,737,558 (5.6%)	1.672% (1.250%)

Table 1.3: Subset Data for users that swap on both ETH and POLY

Benefit of Owning Assets on L1 vs. L2

In addition to the liquidity risk associated with holding tokens on L2, which is one of the security concerns we highlight, our model assumes that tokens on L2 and L1 are essentially the same. However, it is plausible that tokens possess different utility values. Users may utilize their tokens on L1 in other DeFi applications that generate higher returns compared to L2, thereby providing additional value to L1 tokens and potentially explaining the observed gap.

Nevertheless, it is worth noting that numerous DeFi applications are currently available on L2, including DEX platforms and lending protocols, offering a diverse range of financial options that can occasionally yield higher returns than holding tokens on L1. For example, our data reveals that providing liquidity on L2 generates higher daily returns compared to L1 (Figure 1.10). Despite this, liquidity providers still exhibit a preference for providing liquidity on L1. This behavior reinforces our security concern, as it suggests that the switching cost from L1 to L2 may be relatively low compared to the higher returns from providing liquidity on L2, yet liquidity providers still choose L1 as their preferred option.

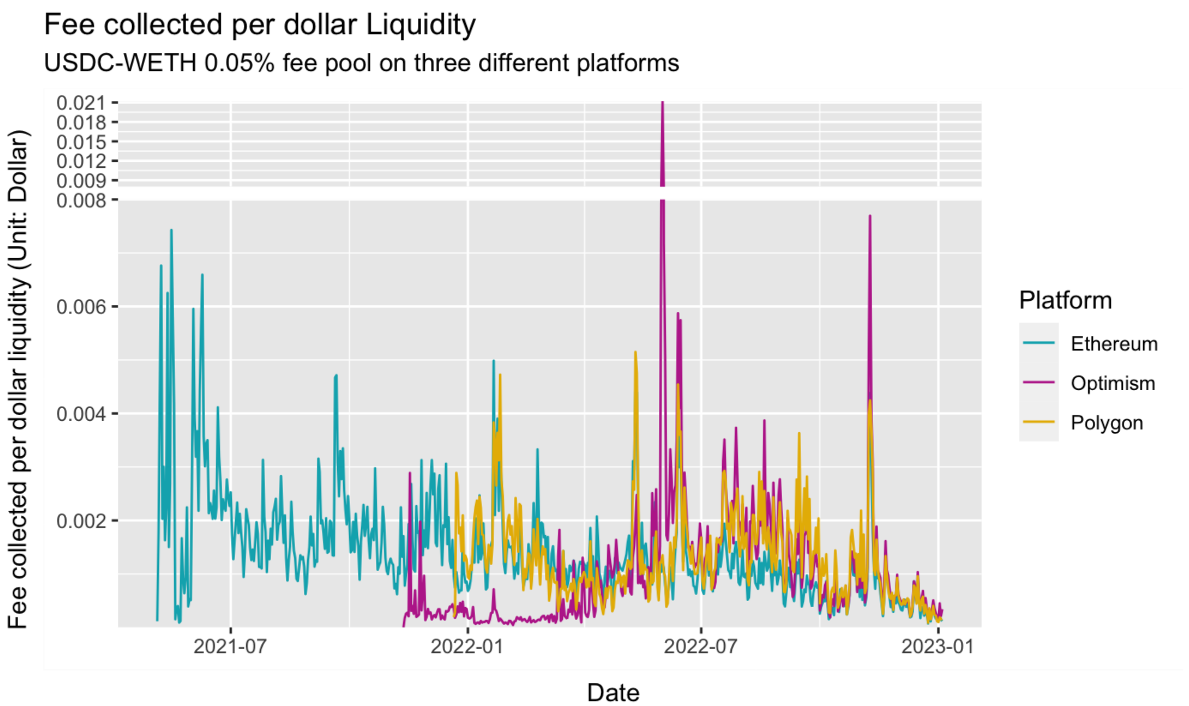


Figure 1.10: Fee collected per dollar liquidity for USDC-WETH 0.05% liquidity pool

1.6.5 Generalization

The analysis conducted in the above sections can be generalized to other L2 network, and also other liquidity pools with different tokens.

We first look at the corresponding pool in Optimism mentioned in Section 1.5. Appendix A.4 documented the corresponding Figure 1.7 and Figure 1.8 for this same pair of tokens and same protocol swapping fees, but comparing Optimism and Ethereum. Similar to our results from swapping behavior in the Polygon pool case, again, there's always a gap between the empirical threshold and monetarily optimal threshold in Figure A.7. The estimated security parameter for Optimism of this token pair (Figure A.8) has more fluctuation, and on average, higher security concerns from the representative agent compared to Polygon (mean is 3.53% and median is 2.64%). We will discuss this difference between Polygon and Optimism later.

To further verify our model, we performed the same exercise to 5 other token pairs on Polygon and 6 other token pairs on Optimism. These pools have different tokens involved, and also have various protocol swapping fees. Table 1.4 summarizes all the liquidity pools we have analyzed.

Pool fee	L2 Polygon	L2 Optimism
DAI/USDC	0.01	0.01
DAI/WETH		0.05
MATIC/WETH	0.3	
USDC/USDT	0.01	0.01
USDC/WETH	0.05	0.05
USDC/WETH	0.3	0.3
USDT/WETH		0.05
WBTC/WETH	0.05	0.05

Table 1.4: List of liquidity pool analyzed

Some of these pools were not available until recently; in order for us to compare our model result across different pools and networks, we take the intersection of period of time for all pools, which is August 5, 2022 to December 31, 2022, a period of 149 days. Appendix A.5 provides the histogram of swapping values and the estimated security parameter for these additional 11 pools. All previous analysis is replicated, as we observe the histogram of swapping values sort in the same way as we previously introduced, with small transactions in L2 and large transactions in L1. Moreover, the gap between the empirical threshold and monetarily optimal threshold are all robust, and the estimated security parameters are also similar (within the same network).

Based on each day's trading volume in each pool, we assign a weight to calculate the weighted mean of the security parameter $S_{L2,L1}$ for the two L2 platforms. The weighted mean for Polygon is 0.68%, and for Optimism is 3.29%, meaning that the representative agent believed that there is 0.68% more chance of losing transactions on Polygon compared to Ethereum, while there is 3.29% more risk on Optimism compared to Ethereum.

As one might notice from Appendix A.5, there's more fluctuation and worse security estimator for Optimism than Polygon; indeed, as shown in Figure 1.11, the Optimism's weighted mean is significantly greater from Polygon's at 95% level. This suggests traders believe that the Optimism network is less secure than Polygon. A security incident where attackers stole $15 million in OP governance tokens in June 2022 could support this result.[37]

Alternative explanations are that perhaps the "optimistic" approach in the validation process of optimistic rollups reduces the reliability from a trader's point of view. Another possible explanation is that the Community of the Polygon network is much greater than the Optimism, which generates more reliability of the traders in the Community and less about the technology behind it. But this is speculation and a topic for future research.

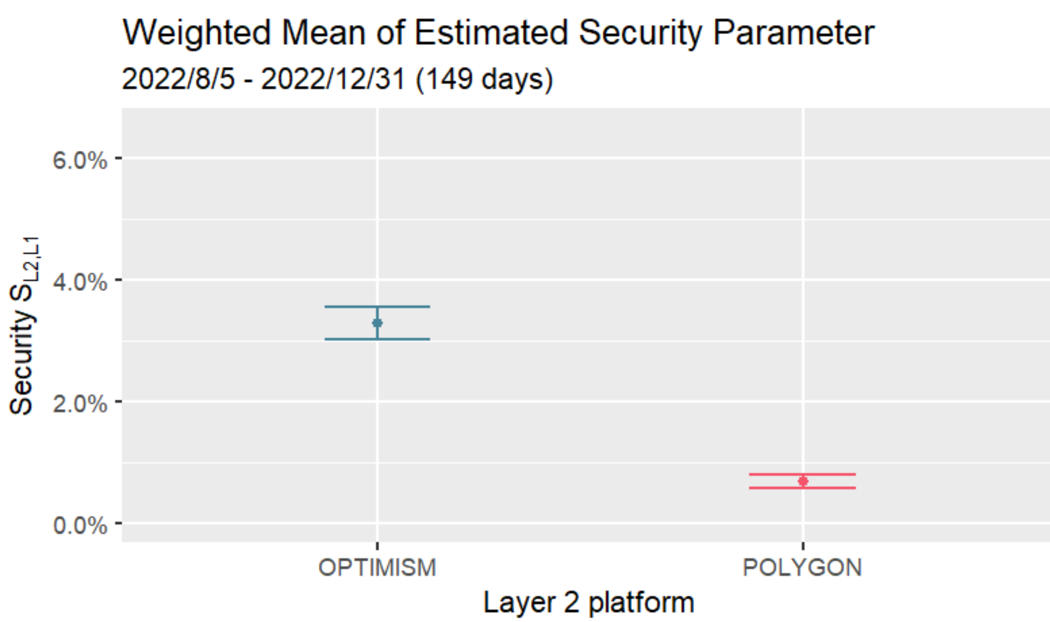


Figure 1.11: Weighed Mean of Estimated Security Parameter

[37] Data source: https://www.coindesk.com/tech/2022/06/09/15m-of-optimism-tokens-stolen-by-an-attacker-after-wintermute-sent-wrong-wallet-address/

1.7 Summary

The primary focus of our analysis of this novel trading environment in DEX platforms is to quantify traders' belief about security issues regarding L2 compared to L1. To do so, we analyzed trading data using a structural model. Our model calculates the monetarily-optimal switching point for traders to trade on the L1 network instead of L2. Empirical data supports the idea that traders use L1 for lower transactions even though it is less expensive to trade on L2. We argue that security concerns have a critical role in explaining this gap.

Our model reveals that, on average, traders anticipate a 0.68% (3.29%) chance of losing transaction value when trading on Polygon (Optimism) compared to L1, which is a substantial risk considering the (0.01%-0.3%) transaction fees charged per trade. Our analysis utilized a large and diverse dataset that incorporated various gas prices, different types of tokens, and two L2 networks (Optimism and Polygon). Despite this variation, we consistently obtained similar results, which highlight the robustness of our findings. Moreover, we have rigorously established that alternative explanations such as price accuracy, liquidity concentration, adoption costs, and the advantages of holding assets on L1 are less influential in explaining the observed preference for L1 over L2. We also develop preliminary insights on the impact of L2 solutions to the financial inclusion of DEX. L2 solutions allow traders with low stakes to enter a market with a low-gas-fee environment. The number of swapping transactions on L2 is much higher than on L1, and these are mostly small-size transactions. The high gas fees in L1 do not allow traders with a small budget to trade when gas fees are high relative to their small transactions. Our work can be seen as empirical evidence of the trade-off between scalability, security, and decentralization, which is the biggest challenge of blockchain networks.

Looking forward, our novel methodology can be applied to other L2 net- works, allowing researchers to estimate traders' security concerns across different networks in DEX platforms. The long-term effect of the introduction of the L2 networks is yet to be explored. Will concerns about security be reduced when L2 has been in existence longer? DeFi markets should be explored further, and we invite more researchers to study this new environment.

Chapter 2

The Power of Default: Measuring the Effect of Slippage Tolerance in Decentralized Exchanges

with Nir Chemaya, Robert McLaughlin, Nicola Ruaro, Christopher Kruegel, & Giovanni Vigna

2.1 Introduction

Nudge theory suggests that small environmental features can capture attention and influence behavior and decision-making, as argued by Thaler and Sunstein (2008). The phenomenon of the *default effect*, whereby people tend to stick with the default decision, is a well-known example of this theory. This effect occurs when individuals are presented with a choice, and the decision architecture suggests a default option. When individuals do not actively choose an alternative, the default option is the one they ultimately end up with. As a result, the likelihood of that particular decision being selected is higher in comparison to the other available options. Given that the default decision can significantly impact people's choices, policymakers must select the optimal default to promote greater social welfare. There is a wealth of evidence on the default effect, particularly in the realm of financial decisions. To enhance welfare, policymakers have implemented measures such as adjusting default choices, as demonstrated by the "Save More Tomorrow" program (Thaler and Benartzi, 2004).

We can find many default settings in many different Decentralized Finance (DeFi) platforms. For example, many Decentralized Exchanges (DEX) platforms have a default slippage tolerance setting for traders. This slippage tolerance protects the user in case there is a change in the expected exchange rate, i.e., the smart contract will execute the transaction only if the traders obtain, at a minimum, a specified quantity of tokens. Additionally, it can protect users from "sandwich attacks," which is a price manipulation tactic wherein an attacker strategically influences market prices to exploit traders on DEX platforms. We will describe this in detail in the following sections.

The slippage tolerance setting must be chosen carefully. By choosing higher slippage tolerance values, traders increase the potential profit for sandwich attacks, which makes them more vulnerable to such attacks. Consequently, it is imperative that traders choose a slippage tolerance that is not too high. Conversely, choosing lower slippage tolerance values raises the likelihood that the transaction fails because of price movement caused by normal, but unanticipated, trades from other users. When the the slippage tolerance threshold is exceeded, no swap will be performed, but the user must still pay gas fees. Therefore, traders must carefully consider these trade-offs when deciding on their preferred slippage levels.

Ethereum currently carries the highest Total Value Locked (TVL) among blockchains in the DeFi ecosystem, and on which Uniswap is the most prominent DEX platform, as measured by TVL (defillama, 2023). Initially, Uniswap set the default slippage tolerance at 0.5%. However, on March 16th, 2023, the Uniswap protocol implemented a modification to its default slippage tolerance mechanism. This new approach takes into account estimated network fees and the size of traders' transactions to compute a customized slippage for each trade (Uniswap, 2023b). Depending on the specific trade and market conditions, this new default can range from 0.1% to 5%. The primary objective of this modification is to reduce the impact of sandwich attacks that leverage price manipulation. Fortunately, other DEX protocols like Sushiswap (Sushiswap, 2021), have maintained the default slippage tolerance at 0.5%. This allows us to use Sushiswap as a control when we are conducting our analysis for the impact of Uniswap's new default settings.

To understand how this dynamically determined slippage tolerance impacts the ecosystem, we performed a large-scale measurement. More precisely, we analyzed all the transactions executed on Uniswap and Sushiswap protocols on the Ethereum blockchain in the month of March 2023 – amounting to approximately 5 million swap transactions – from which we infer the slippage tolerance setting of 1,187,141 Uniswap and 19,111 Sushiswap swaps.

The results of our analysis using new measurement approaches provided several insights, which are the contributions of this paper:

1. We developed a methodology for utilizing data from the Ethereum public transaction queue (the *mempool*) to infer traders' decision-making regarding slippage in trading. By employing this technique, we examine the existence of a default effect in DEX platforms. Our findings reveal that 24% of Uniswap transactions and 67% of Sushiswap transactions adhere to the default slippage tolerance offered by the protocols. This demonstrates the substantial impact of the protocols' default settings on trading behavior.

2. We evaluate the effectiveness of Uniswap's recently introduced slippage mechanism

in terms of the benefits it provides to the traders' welfare. Notably, our findings indicate that the adoption of Uniswap's new default slippage setting leads to a substantial reduction in traders' losses by approximately 54.7%. We conducted a robustness check to ensure that this effect is primarily observed in users who follow the default, i.e., defaulters. We found that this is indeed the case, with an approximate 90% reduction in losses for defaulters' welfare. Considering that default users are still suffering losses from sandwich attacks, this underscores the potential for further improvement of the default settings.

2.2 Background

2.2.1 Decentralized Exchanges

DEX platforms have emerged as a new option for traders in the cryptocurrency market, enabling them to engage in decentralized trading activities without relying on the conventional centralized order book mechanism such as the one that Coinbase operates (Coinbase, 2023). Many DEX platforms use liquidity pools, which are smart contracts that allow agents to provide liquidity in the form of tokens or assets on the blockchain. Most pools have liquidity for two tokens/assets and enable users to exchange between them. Traders exchange tokens or assets from these pools using a pricing rule specified in the smart contract code. One of the most commonly used pricing rule in these liquidity pools is the Constant Product Market Maker (CPMM), which is determined by the supply of tokens or assets in the pool (Xu et al., 2023). Liquidity providers are incentivized to engage in these pools by receiving a fee from each swap action, which typically ranges from 0.01% to 1%, depending on the protocol and the tokens/assets in the pool. Considering the predominant trading volume observed on the Ethereum blockchain, this work focuses exclusively on the analysis of DEX data on the Ethereum blockchain. Uniswap is currently the largest DEX protocol in DeFi, with a daily volume of roughly $1 billion and total liquidity of $2 billion (Uniswap, 2021) on the Ethereum blockchain.[1]

2.2.2 Sandwich Attacks

Despite their popularity, users of DEX platforms regularly suffer from *sandwich attacks*, an issue which is endemic to the ecosystem. A sandwich attack is a price manipulation tactic wherein an attacker strategically influences market prices, which forces unassuming traders into accepting higher-than-necessary prices.

In Ethereum, pending transactions are first sent into a public queue, the *mempool*, before they are executed. A malicious user (Mallory) can observe these pending transactions and exploit them using the sandwich attack pattern. During a sandwich attack, Mallory monitors public transactions queued in the mempool and strategically inserts two of her own transactions immediately before and after the victim's transaction. These transactions are commonly referred to as front-running and back-running, respectively. Attackers, by using this tactic, can gain a massive profit depending on the trade size and market conditions. The total annual profit extracted by sandwich attackers is approximately over $100 million annually (Qin et al., 2022).

For instance, consider a scenario where Alice intends to trade 10 of token A for 1 token B, which, in this example, is the expected exchange price in the liquidity pool if her transaction is executed next. Alice's transaction is sent to the mempool and awaits execution by a block producer. Mallory observes Alice's pending transaction, and strategically inserts a large swap immediately before Alice's, which raises the exchange's price significantly. Alice's transaction then executes after Mallory's, but with a worse

[1] Data source: `https://uniswap.org/` on March 16, 2023 (only Ethereum).

price due to the CPMM pricing rule (for example, assume that she now receives only 0.8 B tokens). Finally, Mallory will execute another large swap in the opposite direction of exchange, restoring the un-manipulated exchange price. This results in Mallory earning a profit of 0.2 B tokens at Alice's expense. A prototypical sandwich attack is shown in more detail in Figure B.3 in Appendix B.4. Mallory may also choose to generate "multi-meat" sandwich attacks, wherein a single attack, i.e., a pair of front-running and back-running transactions, targets multiple victims simultaneously. This is illustrated in Figure B.4 in Appendix B.4.

2.2.3 Related Work

Extensive research has been conducted on DEX platforms from various perspectives, providing valuable insights into their functioning. Notably, several studies (Lehar and Parlour, 2021; Barbon and Ranaldo, 2021) have focused on comparing centralized exchange (CEX) and decentralized exchange (DEX) platforms, analyzing crucial factors such as liquidity provision, absence of arbitrage, price efficiency, and transaction costs. Additionally, a significant body of literature (Park, 2021; Capponi and Jia, 2021) has been devoted to investigating the Constant Product Market Maker (CPMM) mechanism, examining its properties, and identifying conceptual flaws.

Among the identified flaws, one prominent concern highlighted in Park's work (Park, 2021) is the vulnerability of traders to sandwich attacks. Numerous papers have delved into the concept of sandwich attacks, discussing their empirical existence (Daian et al., 2020; Zhou et al., 2021; Lehar and Parlour, 2023; Züst, 2021; Qin et al., 2022; Torres et al., 2021).

The work by Heimbach and Wattenhofer (2022) is particularly relevant to our research. They introduced a model called the "Sandwich Game," which is based on game theory and allows traders to protect themselves against sandwich attacks by selecting an optimal slippage protector. The authors propose a dynamic slippage approach, wherein the algorithm calculates an optimal slippage protector that reduces the probability of sandwich attacks without significantly increasing the likelihood of transaction failures. Building upon this, our paper focuses on the actual slippage decisions made by traders and their default effects within this environment. Specifically, we explore the behavior of users adhering to the default slippage decision and examine the effects of altering this default within the Uniswap protocol. In line with the spirit of Heimbach and Wattenhofer's model, Uniswap transitioned from a constant slippage approach to a more dynamic one. We shed light on the effectiveness of this new slippage methodology and propose potential avenues for further improvement.

2.3 Model

In this section, we summarize some key properties of the Constant Product Market Maker (CPMM) pricing formula, which is used by several DEX applications, including Uniswap and SushiSwap. This will allow us to formalize traders' slippage decisions in DEX CPMM applications and to provide a straightforward model for estimating slippage decisions based on mempool data.

2.3.1 Transaction Model

Let's consider a liquidity pool that contains x tokens of token $\mathbf{X}$ and y tokens of token $\mathbf{Y}$ (following the notation of Barbon and Ranaldo (2021)). The CPMM pricing rule specifies that for any time t, the product of the available tokens ($\mathbf{X}$ and $\mathbf{Y}$) in the pool equals a constant k, which can be expressed as

$$x_t y_t = k \tag{2.1}$$

Let f denote the percentage fee that is subtracted from each swap's payment and remitted to the liquidity providers. Then, $\varphi = 1 - f$ is the percentage left for the trader to swap. If at time $t+1$ a trader wants to swap Δx quantity of tokens $\mathbf{X}$ for $\mathbf{Y}$ tokens, we can calculate how many tokens $\mathbf{Y}$ that trader will receive, $\Delta(y)$. CPMM states that

$$k = (x_t + \varphi \Delta x)(y_t - \Delta(y)) \tag{2.2}$$

For illustration, the amount $\Delta(y)$ received when paying $\Delta(x)$ is:

$$\Delta(y) = y_t \frac{\varphi \Delta x}{x_t + \varphi \Delta x} \tag{2.3}$$

DEX applications commonly allow users to specify either Δx, the exact amount to pay, or Δy, the exact amount desired. The unspecified quantity is computed on-demand, according to the equations above. In order to disambiguate the user-specified value and the computed value, we write the given value as either Δx or Δy and the computed value as either $\Delta(x)$ or $\Delta(y)$.

To protect the trader from sandwich attacks and price manipulations, DEX platforms, such as Uniswap, allow traders to define a maximum slippage percentage ϕ. From this percentage, the application derives either a minimum amount received, $\underline{\Delta(y)}$, or a maximum amount paid, $\underline{\Delta(x)}$.

This means that if there is any change in the expected exchange rate due to a change in the pools – possibly due to sandwich attacks – the smart contract will execute the transaction only if the trade meets this threshold. For illustration, the minimum amount received ($\underline{\Delta(y)}$) is computed as:

$$\begin{aligned}\underline{\Delta(y)} &= (1+\phi)^{-1}\Delta(y) \\ &= (1+\phi)^{-1} * y_t * \frac{\varphi * \Delta x}{x_t + \varphi * \Delta x}.\end{aligned} \tag{2.4}$$

This CPMM pricing formula is used by both Uniswap V2 and Sushiswap (Adams et al., 2020). Some liquidity pools in Uniswap use the newer V3 model (Adams et al., 2021), which is slightly more nuanced. This model allows liquidity providers to bound the price range in which their liquidity is usable by the pool – if the price moves outside of the user's configured range, their deposits are removed from market-making activities. We will address this difference more in Section 2.4.

A user who wants to perform a swap on a DEX typically proceeds as follows: The user first accesses the DEX user interface – in the case of Uniswap and Sushiswap, this UI is a (regular) web application. The user specifies the token they wish to sell, the token they wish to receive, and either an exact amount paid, Δx, or an exact amount to receive, Δy.

At this point, the user can either accept the default option and follow the slippage decision suggested by the DEX protocol, or manually specify their preferred slippage percentage, ϕ. The DEX application will then compute $\underline{\Delta(y)}$ or $\underline{\Delta(x)}$ using the current DEX state.

Finally, the user reviews, signs, and transmits the transaction to the Ethereum mempool, where it queues for execution. This is typically done with the help of a browser extension, such as Metamask (2023). Note that the transaction includes the computed value $\underline{\Delta(y)}$ or $\underline{\Delta(x)}$, but the value ϕ is not encoded within the transaction. In the following section, we explain how we determine (estimate) the value ϕ by observing the Ethereum mempool.

2.3.2 Estimating the Slippage Tolerance Decision

A user's choice of ϕ has an important impact on the transaction's execution. However, this value is not broadcast to the network. Thus, we use the following method to recover the user's original choice.

We derive the slippage percentage ϕ using Equation 2.4:

$$\phi = \Delta(y)/\underline{\Delta(y)} - 1 \tag{2.5}$$

While the values of $\underline{\Delta(y)}$ and $\underline{\Delta(x)}$ can be immediately read from the transaction, the values of $\Delta(y)$ and $\Delta(x)$ are not encoded within the transaction.

For the CPMM pricing rule, computing $\Delta(y)$ and $\Delta(x)$ requires knowing what the state of the DEX was *when the application generated the swap transaction*.

Consequently, we first estimate *when* the transaction was generated. By monitoring the Ethereum mempool, we record the timestamp t at which the transaction was first broadcast.[2] We then look up the DEX state as of timestamp t, which we use to compute either $\Delta(x)$ or $\Delta(y)$, and finally compute ϕ.

2.4 Methodology Evaluation

Researchers often deal with a trade-off between data accuracy and the cost of achieving it (Hummel et al., 2011; Evans and Crawford, 2000; Muradian et al., 2019; Bastos et al., 2021). Sometimes, due to limitations in data availability, computation power, or labor resources, researchers need to compromise on accuracy to conduct their research.

[2]Prior work observed that the amount of time to propagate a transaction across the peer-to-peer mempool network is small (200 milliseconds) compared to the block interval (12 seconds) (Wang et al., 2021). As we only require accuracy to within the block interval, time t will suffice as an estimate for the transaction generation time.

This trade-off can be observed in various data sets and research environments, including financial markets (Abowd and Schmutte, 2019).

The initial versions of Uniswap (V1 and V2) use the simplest form of the Constant Product Market Maker, as described in Adams et al. (2020). These versions have been the main focus of theoretical studies on liquidity pools, as evident in works like Lehar and Parlour (2021); Capponi and Jia (2021); Park (2023).

Uniswap V3, introduced on May 5th, 2021, brought significant updates, allowing liquidity providers to offer liquidity with specific price limits. Consequently, the bonding curve is now only locally defined, and the impact of trade size on prices depends on the overall distribution of liquidity positions, rather than just the aggregated liquidity levels, as described in detail in Adams et al. (2021) and Barbon and Ranaldo (2021). Researchers analyzing transaction prices and applying economic models for V3 data must collect not only transaction data, but also the distribution of liquidity positions for each pool, which poses challenges due to the additional resources and time required.

This chapter, among other empirical paper in this rapidly growing literature (Lehar and Parlour, 2021; Lo and Medda, 2022; Xia et al., 2021; Malinova and Park, 2023; Chemaya and Liu, 2023) needs to address an important question, to find optimal analytical approaches that ensure the most effective analysis balancing accuracy and costs of analysis.

To simplify data analysis and overcome limitations, some studies, such as Chemaya and Liu (2023), have employed V2 models to analyze V3 data. Using V2 models for V3 data analysis offers advantages in terms of simplicity and data collection. These models enable straightforward estimation of pool liquidity using solely historical trading data (without the necessity for liquidity positions), a concept elaborated upon in the next section with the introduction of our model. However, a vital question remains: while it is simpler and easier to analyze V3 data using a V2 model, what level of accuracy do we potentially sacrifice by employing this approach?

This section aims to assess the effectiveness of using V2 models for V3 data analysis and estimate the associated measurement error. We argue that the V2 model showed remarkable accuracy, accurately predicting the V3 data in 97.1% of transactions, with a deviation of less than 0.1%. Moreover, we observed that highly active pools with a significant number of trades, such as the top 10 pools, demonstrated even higher accuracy, with 99.6% of transactions falling within a 0.1% deviation. On the other hand, less active pools, like the bottom 4000, performed less optimally, with only 89.5% accuracy. This highlights the limitations of using V2 models for smaller pools or markets.

2.4.1 Measurement setup

Following the notation described in Section 2.3.1. Imagine a researcher has a dataset of V3 transactions of a particular pool, but without information about the liquidity positions in the pool. We will use this transaction data and apply the V2 model, assuming that the liquidity available in the pool has no limitations and follows V2 rules.

To do this, we take pairs of consecutive transactions from the V3 dataset. Suppose we have three consecutive transactions T_1, T_2, T_3. Using the pricing rule of V2, we can calculate the amount of liquidity available in the pool with information of T_1, T_2. This allows us to predict the exchange rate for the next transaction in line (T_3) – meaning, if a trader were to trade a certain amount of token X or Y, what they would receive in exchange for the transaction. This prediction is based on V2's assumptions.

Next, we compare these V2 predictions for T_3 with the actual exchange rates from the V3 data. By doing this comparison, we gain insights into how well the V2 model performs when applied to V3 data.

Suppose at time $t = 0$, a liquidity pool containing x_0 amount of token X and y_0

amount of token Y. Then three transactions were executed in this block with the following amounts: $\{\Delta x_1, \Delta y_1\}$, $\{\Delta x_2, \Delta y_2\}$, $\{\Delta x_3, \Delta y_3\}$. $\Delta x_t < 0$ & $\Delta y_t > 0$ means traders are giving up Δy_t for Δx_t from the liquidity pool,[3] and vice versa.

Using the CPMM pricing rule and the transactions, we can estimate the initial pool values assuming v2 model $\hat{x_0}$, $\hat{y_0}$, and $\hat{k}$ as follows:

$$k = (x_0 + \Delta x_1)(y_0 + \Delta y_1) \tag{2.6}$$

$$k = (x_0 + \Delta x_1 + \Delta x_2)(y_0 + \Delta y_1 + \Delta y_2) \tag{2.7}$$

Given $\{\Delta x_1, \Delta y_1\}$, $\{\Delta x_2, \Delta y_2\}$, one can solve this system of equations, and estimate x_0, y_0, and k. The close form solution can be written as:

$$\begin{aligned}
\hat{x_0} &= -\frac{\Delta y_2 \Delta x_1(\Delta x_1 + \Delta x_2)}{\Delta y_2 \Delta x_1 - \Delta y_1 \Delta x_2} \\
\hat{y_0} &= \frac{\Delta y_1 \Delta x_2(\Delta y_2 + \Delta y_1)}{\Delta y_2 \Delta x_1 - \Delta y_1 \Delta x_2} \\
\hat{k} &= -\frac{\Delta y_2 \Delta y_1 \Delta x_1 \Delta x_2(\Delta y_2 + \Delta y_1)(\Delta x_1 + \Delta x_2)}{(\Delta y_2 \Delta x_1 - \Delta y_1 \Delta x_2)^2}
\end{aligned}$$

Without loss of generality, assume $\Delta x_3 < 0$ & $\Delta y_3 > 0$, that is, the third transaction from the trader prospective is giving up token Y, and swap out token X from the liquidity pool.

$$k = (x_0 + \Delta x_1 + \Delta x_2 + \Delta x_3)(y_0 + \Delta y_1 + \Delta y_2 + \Delta y_3) \tag{2.8}$$

Given $\{\Delta x_1, \Delta y_1\}$, $\{\Delta x_2, \Delta y_2\}$, and $\{\Delta y_3\}$, one can estimate $\hat{\Delta x_3}$ using the estimated

[3] Please note that traders paid an additional liquidity pool fee directly to the liquidity provider, but this fee is not included in the Δx value in this model.

$\hat{x_0}$, $\hat{y_0}$, and $\hat{k}$ from the above calculation.

$$\Delta \hat{x}_3 = \frac{\hat{k}}{(\hat{y_0} + \Delta y_1 + \Delta y_2 + \Delta y_3)} - (\hat{x_0} + \Delta x_1 + \Delta x_2) \tag{2.9}$$

Denote the distance of $\Delta \hat{x}_3$ from the realized Δx_3 as d,

$$d = \Delta \hat{x}_3 - \Delta x_3 \tag{2.10}$$

and define $\tilde{d}$ as the absolute percentage deviation.

$$\tilde{d} = \frac{|d|}{\Delta x_3} = \frac{|\Delta \hat{x}_3 - \Delta x_3|}{\Delta x_3} \tag{2.11}$$

Where the estimated $\Delta \hat{x}_3$ is the result from assuming V2, and the realized Δx_3 is from V3.

$\tilde{d}$, measuring how well using V2 models towards V3 data, is the metric of interest. A higher $\tilde{d}$ denotes worse estimation, while a low $\tilde{d}$ shows effectiveness of using V2 models for V3 data analysis.

2.4.2 Measurement result

We collect one year of swapping transaction data on Ethereum chain on Uniswap V3, from January 1st, 2022, to December 31st, 2022.[4] 14,257,238 transactions have been collected, resulting in a $467,289,312,828 total transaction value.

For each of these transaction, our dataset contains transaction hash, timestamp, log index within a block, Pool ID, wallet hash that originate the transaction, and information about Token 0, Token 1. We then conduct our calculation of estimated $\hat{x_0}$, $\hat{y_0}$, and $\hat{k}$ for each transaction, based on the antecedent two transactions validated in the same pool.

[4] Data source: https://www.uniswap.shippooor.xyz/

There are 6,147 distinct pools that host transactions in this one year period. However, the majority of the transactions concentrated in a limited number of pools. In fact, 60% of of the transactions occurred in the top 50 pools; this number increase to 90% when we consider the top 500 pools, ranked by pool activeness. A more detailed break down can be found in Table 2.1, and a CDF is provided in Figure 2.1. Notice that these top pools with most transactions are the ones with a higher liquidity, especially those in the top 10 to top 50, as shown in Figure 2.2.

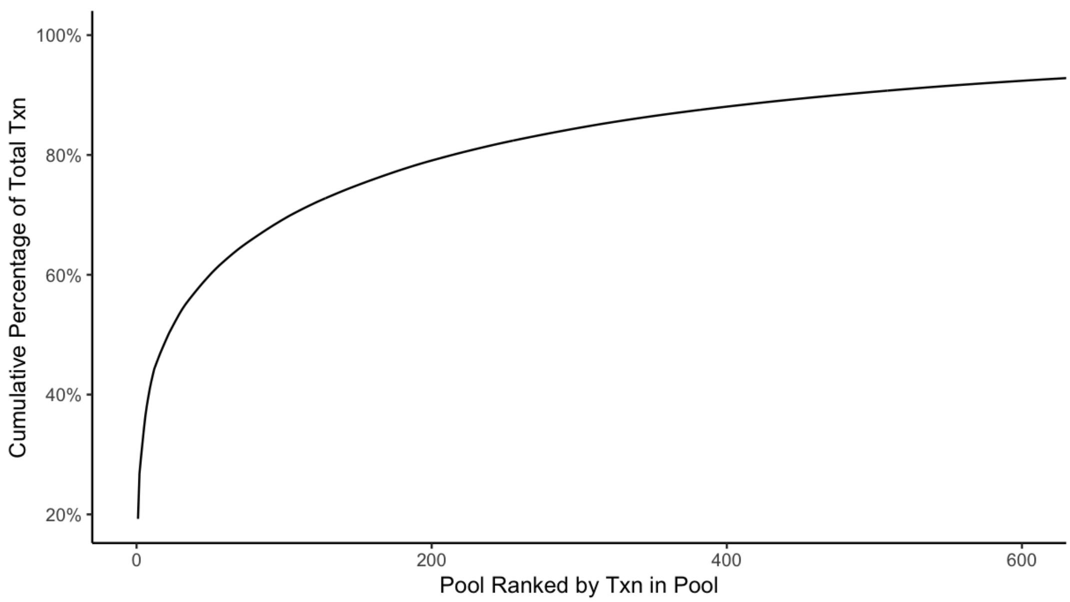


Figure 2.1: CDF of transactions, ordered by activeness of Pool

For each transaction that has two antecedents, one can calculate $\tilde{d}$, the deviation of the estimated $\hat{\Delta x_3}$ from the realized Δx_3. Figure 2.3 shows the CDF of deviation from all transactions, for which almost all the deviation is less than 0.1%. The median of $\tilde{d}$ is $8.20e^{-10}$, meaning the majority of $\tilde{d}$ is extremely small. As shown in Table 2.1, 97.1% of the deviation is less than 0.1%, and 99.1% of the deviation is less than 1%.

Furthermore, given that the more active pools are the ones with higher liquidity, we

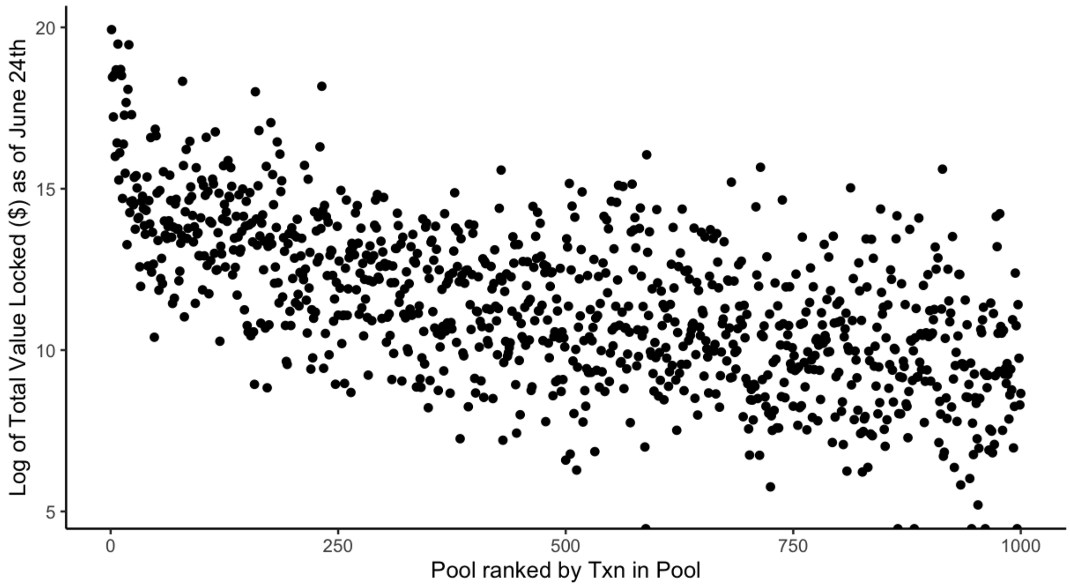


Figure 2.2: Pools with more txn are the ones with higher liquidity

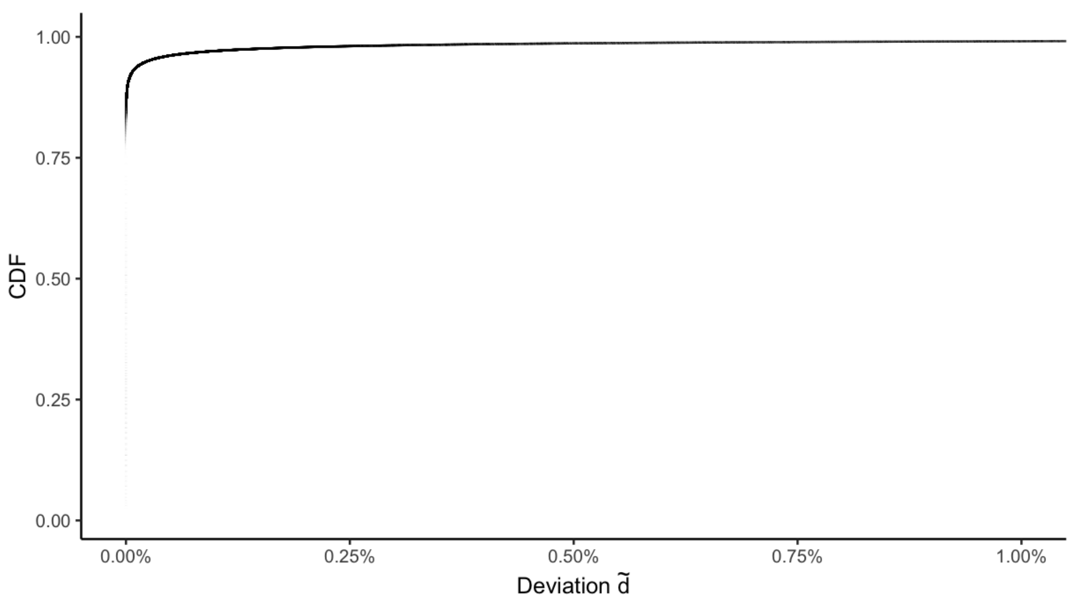


Figure 2.3: CDF of deviation for all transaction in 2022 on ETH on Uniswap

plot the same CDF by activeness of pool in Figure 2.4. The line represents the Top 10 pools, first-order stochastic dominance the Top 50 pools (exclude Top 10), which first-order stochastic dominance the Top 200, and so on. Indicates accuracy was higher in active pools with substantial transaction volume and liquidity, while inactive pools performed less effectively.

Table 2.1 provides a more in depth look at the accuracy rate for each case. Moving from the left to right of the table, the percentage of transactions that satisfy the maximum threshold deviation increases as expected. From the top of the table to the bottom, the accuracy rate drops. Once more, these findings reveal that accuracy was greater within pools displaying higher levels of transaction volume and liquidity.

			$\tilde{d}$ within				
	Txn	%	0.0001%	0.01%	0.1%	1%	5%
All Pool	14,257,238	100%	82.6%	93.4%	97.1%	99.1%	99.7%
Top 10 Pool	6,014,850	42.2%	88.3%	98.1%	99.6%	99.9%	99.96%
Top 50 Pool	8,562,026	60.1%	85.3%	96.5%	99.0%	99.8%	99.9%
Top 200 Pool	11,273,324	79.1%	83.4%	94.9%	98.2%	99.6%	99.9%
Top 500 Pool	12,909,947	90.6%	82.7%	94.0%	97.6%	99.4%	99.8%
Bottom 4000	87,493	0.6%	81.6%	86.6%	89.5%	93.4%	96.4%

Table 2.1: Percentage of transactions' $\tilde{d}$ within threshold.

In summary, using simpler V2 models for analyzing Uniswap V3 data has clear advantages in terms of ease and accuracy. The V2 model showed impressive accuracy, correctly predicting V3 data in 97.1% of transactions, with deviations of less than 0.1%. The model performed even better in active pools with high transaction volumes and liquidity. For the remainder of the chapter, we will be utilizing the V2 model on analysis.

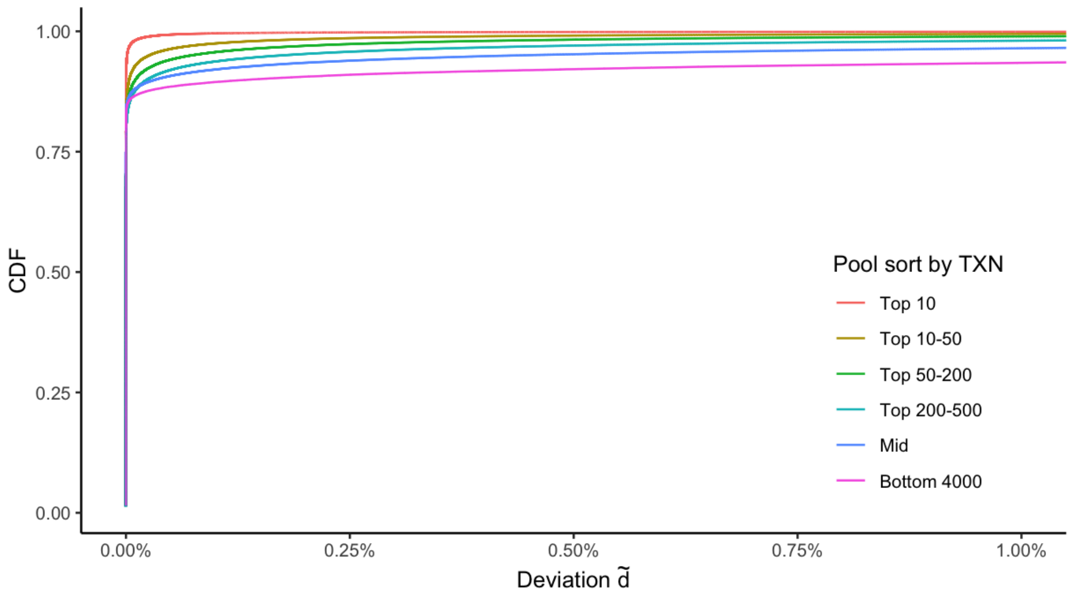


Figure 2.4: CDF of deviation, by activeness of pool

2.5 Empirical Evaluation

2.5.1 Experimental Setup

To perform our study, we require two types of data: first, the timestamp when a transaction entered the public mempool, and, second, a complete record of all Uniswap and Sushiswap DEX swaps that were executed on the blockchain.

To address both needs, we run an Ethereum node in archive mode using Go-Ethereum (Foundation, 2023) version 1.11.2. The node is hosted in the western United States and is served by a high-bandwidth Internet connection. In order to increase mempool visibility and decrease transaction propagation latency, we increased the number of peer-to-peer connections from 100 to 500.

We ran our data collection from March 1, 2023, 00:00 UTC to April 1, 2023, 00:00 UTC, and examined all 33,108,538 transactions within blocks 16,730,072 through 16,950,602.

We used a simple script to record the arrival time of each transaction in the mempool, which satisfies our timestamping needs outlined in Section 2.3.2.

Swaps generated through the Uniswap and Sushiswap web apps begin by invoking a *router* contract. As implied by the name, the router contract routes a user's swaps through the protocol, making use of at least one, but possibly several, liquidity pools. Importantly, the router enforces a DEX user's slippage tolerance setting. Third-party DeFi applications and bots commonly perform swaps by directly interacting with liquidity pools, foregoing the Uniswap or Sushiswap router entirely. We find all router-based swaps by retrieving all confirmed transactions within the study window, which we then filter down to only transactions sent to one of the known router contracts. We then use our knowledge of the router's application binary interface (ABI) to extract the swap's parameters – Δx or Δy and $\underline{\Delta(y)}$ or $\underline{\Delta(x)}$. We find that 81.7% of swaps on Uniswap use exactly one liquidity pool, so, for simplicity of calculation, we only consider router-based swaps that use exactly one liquidity pool. We also infer the total dollar value of the amount swapped by aggregating several on-chain price oracles.

2.5.2 Mempool Observation

Our Ethereum node observed 32,586,069 transactions queued in the mempool. After filtering these down to only transactions within the study window that invoke the Uniswap or Sushiswap routers, we uncover 1,640,182 router-based transactions. We present the arrival rate in Figure 2.5.

Our Ethereum node captured 1,617,820 (98.6%) of the router-based transactions while they were queued for execution. The remaining router-based transactions were not observed in the mempool, but did get included in the blockchain. This may be due to either failure to propagate through the mempool network, or use of a private transaction relayer

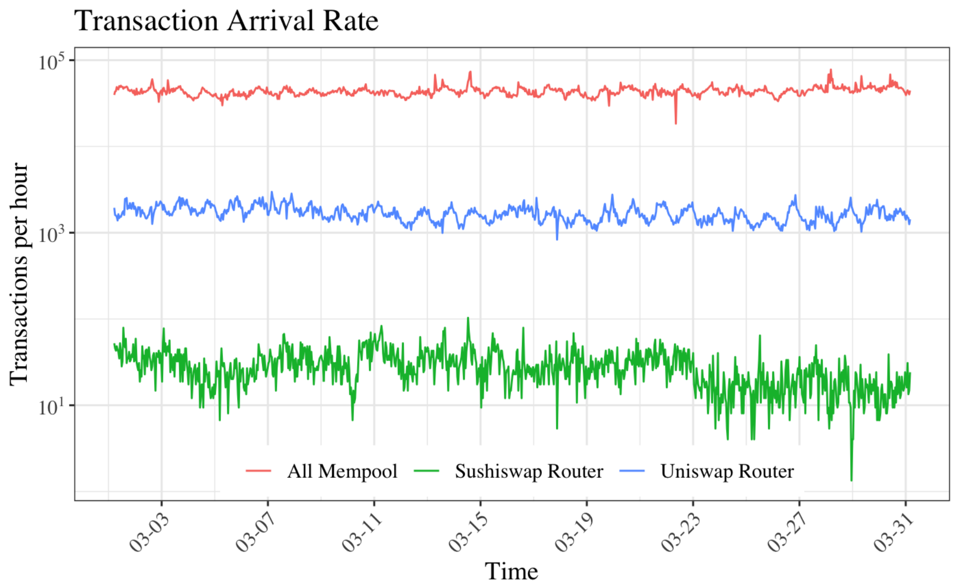


Figure 2.5: Observed hourly transaction rate. Note the use of a log scale to show detail.

such as Flashbots-Protect (Flashbots, 2023). Transactions broadcast through private relayers circumvent the mempool and are privately sent directly to block producers. This shows that our mempool observation is broadly effective and, thus, our analysis is truly representative of the ecosystem.

2.5.3 Inferring Slippage Tolerance Decisions

Router-based swap transactions include either Δx (the exact amount the user would like to pay) or Δy (the exact amount the user would like to receive).[5]

We begin with the subset of transactions observed in the mempool – 1,586,378 Uniswap and 31,442 Sushiswap. The following transactions are then also removed. First, for simplicity, we discard transaction that use multiple liquidity pools – 294,412 Uniswap and 8,103 Sushiswap. Second, we discard transactions that transfer tokens which interfere with the liquidity pool the trader would like to use. For illustration, some tokens

[5] We find that it is much more popular to specify Δx – accounting for 91.0% and 96.3% of all Uniswap and Sushiswap router-based transactions, respectively.

charge a tax upon each transfer from one address to another. A token can then immediately sell the tax on Uniswap to immediately convert it a more stable currency, which is deposited into a treasury. When this occurs it is difficult to infer the slippage, as we would need to reason about each token's arbitrary behavior. We discard any transactions which demonstrate this behavior – 64,682 Uniswap and 0 Sushiswap. Third, we discard transactions that are not using the router to perform a swap – for example, providing liquidity to a pool, buying NFTs, etc. – 173 Uniswap and 3,611 Sushiswap. Fourth, we discard malformed transactions – 279 Uniswap and 129 Sushiswap. This leaves 1,231,615 Uniswap and 19,765 Sushiswap transactions.

In total, we can infer the slippage decisions for 1,187,141 (96.4%) Uniswap and 19,111 (96.7%) Sushiswap router-based transactions observed in the mempool. In swap transactions for which slippage inference failed, we were unable to derive an expected swap price, $\Delta(y)$ or $\Delta(x)$, at the time of transaction generation. This was either because the pool did not have enough liquidity to support the user's desired quantity of tokens, or because, upon inspection, our timestamp of the transaction's arrival in the mempool fell after the timestamp of the block in which it was executed – possibly due to network latency or clock drift. We draw the most popular slippage settings over the time window in Figure 2.6, trimmed to 0% to 2% to show detail. For a wider range of percentages, see Figure B.1 in the Appendix B.3.

We immediately observe the following two oddities. First, we occasionally infer a *negative* slippage setting – this accounts for 2.6% of Uniswap and 4.4% of Sushiswap transactions. To illustrate how this might occur, consider a transaction that specifies Δx. In this situation, we use Equation 2.5 to derive the slippage. Negative values occur whenever $\underline{\Delta(y)} > \Delta(y)$ – i.e., whenever the user requires a minimum amount received that exceeds what the DEX can remit. These transactions have a very low success rate – only 6.9% and 14.7% for Uniswap and Sushiswap, respectively. Second, we

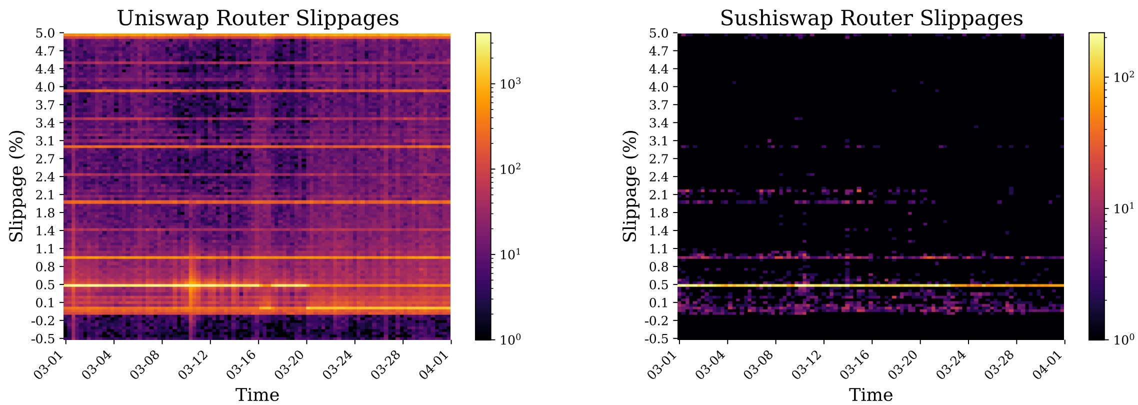


Figure 2.6: Inferred router slippages over time. Color is drawn on a log scale to show detail. Each colored cell counts the number of swaps in a slippage range at a segment of time.

occasionally infer slippage settings that are extremely high (50% and over) – however, the Uniswap and Sushiswap user interfaces both cap slippage to 50%. This accounts for 1.0% of Uniswap and 0.6% of Sushiswap transactions. Both of these oddities may plausibly indicate either a flaw in the transaction generation software, an incorrectly inferred transaction generation time, or simply spam. Nevertheless, these transactions have relatively small prevalence, and are likely not representative of a web app user – so we are confident that their exclusion does not significantly impact the results.

2.5.4 Default Effect on Slippage Tolerance

We now examine the *default effect* on slippage choices in Uniswap and Sushiswap exchanges. This can be seen intuitively in Figure 2.6: we can see a clear, bright line at the default slippage setting, 0.5%. This default was active for Uniswap up to March 16^{th}, and for Sushiswap throughout the entire study window.

Table 2.2 details the percentage of inferred slippage choices at various percentages. We see a preference for the default setting in both cases.

In the case of Uniswap, we also observe a preference for round numbers (10%, 5%, etc.), both in Table 2.2 and in Figure 2.6. Figure B.1, in the Appendix B.3, plots a version of Figure 2.6 with a wider y-axis range, from 0% up to 25% – here we can also observe a clear round number preference.

We do not attempt to distinguish between human- and bot-originated transactions, so we cannot be sure of exactly which transactions are representative of human decision-making when faced with a default. However, we can observe a significant aggregate change in slippage settings after Uniswap altered its default on March 16, 2023.

Initially, the implementation was carried out in a soft manner, resembling a "testing period" that lasted from March 16th to the 20th. This testing period is evident in our data and can be observed in Figure 2.6. On March 16, there was a noticeable decrease in decisions with a 0.5% slippage rate. This is shown by the yellow line disappearing. Afterward, it appeared again for a few days during the soft launch, but then it disappeared once more.

	Slippage Setting					
	10%	5%	2%	1%	**0.5%**	0.1%
Uniswap [6]	7.3%	7.1%	2.2%	4.7%	**24%**	2.1%
Sushiswap	1.1%	1.0%	1.9%	5.1%	**67%**	2.5%

Table 2.2: Prevalence of various slippage settings in router-based swaps.

From March 21^{st} onward, after the default fully was changed, we see only 4.7% of Uniswap transactions use the old default: 0.5%. Moreover, we see a six-fold increase (to 11%) in the percentage of transactions using 0.1% slippage, a soft minimum in the new default's "auto" calculation. This gives confidence that users are, in fact, following the default.

Following this, we label which transactions are most likely using the default slippage

[6] Before March 16, 2023.

setting. This must be done in two parts, before and after the default was changed. In the case of the former, the computation is straightforward: we label all transactions with a slippage between 0.45% and 0.55% as using the default. The latter case is more involved. In brief, Uniswap's default slippage setting is computed by taking the gas cost the user pays to perform the swap, in US dollars, divided by the transaction volume, in US dollars. The resulting value is then bounded to within the range [0.1%, 5%]. Both the numerator and denominator are subject to market fluctuations: gas fees are natively paid in Ether, not dollars, and the transaction volume is specified by the user in units of tokens. Uniswap converts both values to dollars by consulting the Uniswap DEX for spot prices. For each transaction, we derive an estimated default slippage setting by recreating the computation using historical data and market conditions. Then, we label a transaction as using the default if its inferred slippage setting is within 20% of our estimated default. The technical details of this approach are described in Uniswap (2023a).

2.5.5 Traders' Welfare Post Modification of Uniswap's Default Slippage Setting

In the following analysis, we assess the adjustment made to the default slippage setting within the Uniswap decentralized exchange (DEX) protocol on March 16, 2023. Our main goal is to evaluate whether traders are benefiting or suffering from this modification, i.e., if their welfare is increased or decreased. This evaluation encompasses two key aspects: the losses of traders affected by sandwich attacks and the losses incurred due to unnecessary failed transactions, both of which are considerations when determining the optimal slippage tolerance.

To conduct this evaluation, we gathered data from both the Uniswap and Sushiswap protocols for the month of March 2023 – not limiting ourselves to only the "official" routers – which yielded approximately 5 million transactions. We divided our data analysis into three distinct time periods to assess the welfare impact of this new method.

The first period covers the time before the modification, from March 1st to 15th. The second period corresponds to the "testing period" from March 16th to 20th. Finally, the third period focuses on the time after Uniswap adopted the new slippage protector, from March 21st to 31st.

Notably, during our data collection period, Sushiswap maintained a fixed 0.5% default slippage option, whereas Uniswap implemented the dynamic one. This distinction enables us to use Sushiswap as a reference point for control purposes in our analysis, especially due to the major similarity of both platforms.[7]

We focused exclusively on pools with tokens that had high liquidity to avoid price inaccuracies that could affect our aggregate analysis. We discovered that tokens with low liquidity tend to have very inaccurate price oracles,[8] leading to erratic results when converting to US Dollar value ($). We further filter the data set to include only the most active pools that had swap activity every day in March. This approach enables us to minimize the influence of noise generated by pools that were only available before or after the change, and it minimizes the influence of bots conducting pump-and-dump schemes (Cernera et al., 2023). Finally, our filtered data covers approximately 2.25 million transactions, accounting for around 47% of the total transactions. It includes 824 unique pools, which is approximately 4.4% of the total unique pools 18,877. This indicates that the majority of the transactions on the protocols are conducted pools with high liquidity and consistent trading activity. A more detailed information about our swap data set

[7]Sushiswap is a forked version of Uniswap, and both platforms work very similarly. More information can be found here:Liu et al. (2022); Yaish et al. (2023)

[8]We calculate the minimum Total Value Locked (TVL) within the specified window for each token. We then select only pools that have both tokens ranked in the top 1,000 from this list, which we classify as tokens with high liquidity.

can be find in Table B.4 in Appendix B.2

2.5.6 Loss from Sandwich Attacks

Our detection method for sandwich attacks – detailed in Appendix B.1 – draws on earlier studies by Lehar and Parlour (2023); Züst (2021); Qin et al. (2022); Torres et al. (2021).

Using this method we identify total of 36,158 successful attacks in our data, with 33,612 attacks occurring on Uniswap and 2,546 attacks on Sushiswap. Table 2.3 presents the quantity and daily rate of successful attacks detected.

Next, we analyze the financial impact and quantify the losses incurred due to the attacks on both Uniswap and Sushiswap. By examining the value of assets lost during these attacks, we can gain a comprehensive understanding of the implications of the default slippage change and its effectiveness in mitigating losses.

We then conducted the following analysis in order to assess the financial impact of the attacks. For each attacked swap, we calculated the difference between the trader's outcome with and without the attack, and standardized to a dollar amount. Following our previous discussion on methodology, we employed the V2 model due to its straightforwardness. Collectively, all attacks on Uniswap and Sushiswap resulted in a total loss of \$16,584,438 for traders in March. On average, each swap that was attacked incurred a loss of \$458.7, with a median loss of \$69.6. The breakdown of these loss figures by DEX and time period can be found in Table 2.4.

In order to better evaluate the effect of the implementation of dynamic slippage, we aimed to directly compare Uniswap and Sushiswap across all periods, while mitigating the influence of differing trade sizes in different periods. To achieve this, we standardized the loss by the total value swapped. For each day in March, we calculated the total loss

	Time Period		Number of Attacks Detected	Average Attacks Per Day
Uniswap	Mar 1st	- 15th	$23,299$	$1,553.27$ (100%)
	Mar 16th	- 19th	$3,214$	803.5
	Mar 20th	- 31st	$7,099$	591.58 (38.09%)
Sushiswap	Mar 1st	- 15th	$1,297$	86.47 (100%)
	Mar 16th	- 19th	464	116.0
	Mar 20th	- 31st	785	65.42 (75.66%)

Table 2.3: Attacks Detected.

	Time Period		Mean	Percentile			Daily Total
				25^{th}	50^{th}	75^{th}	
Uniswap	Mar 1st	- 15th	\$565	\$33	\$89 (100%)	\$298	\$877,936 (100%)
	Mar 16th	- 19th	\$292	\$21	\$47	\$138	\$234,874
	Mar 20th	- 31st	\$277	\$23	\$51 (56.8%)	\$134	\$163,807 (18.7%)
Sushiswap	Mar 1st	- 15th	\$250	\$22	\$50 (100%)	\$146	\$ 21,596 (100%)
	Mar 16th	- 19th	\$119	\$16	\$30	\$73	\$ 13,764
	Mar 20th	- 31st	\$167	\$18	\$36 (72.9%)	\$75	\$ 10,934 (50.6%)

Table 2.4: Loss from Attacks.

from attacks per $100 swapped on each DEX, which we plot in Figure 2.7.

The y-axis in the figure represents the loss per $100 swapped, where a value of 0.02 signifies that the loss from an attack would account for 0.02% of the swapped value. On the left-hand side of the figure, we have Uniswap, while Sushiswap is on the right-hand side. It is important to note that the range of the y-axis differs between the two plots, with values for Uniswap mostly below 0.05% and Sushiswap reaching up to 0.25%.

The colored lines in the figure represent the weighted mean for each period. The red line corresponds to the pre-launch period from March 1^{st} to March 15^{th}, while the purple line represents the period after the full launch. For Sushiswap, the two colored lines are nearly level, suggesting no significant impact from the change of the default slippage setting on Uniswap. However, a noticeable gap is observed for Uniswap. This gap is statistically significant as shown with the non-overlapping dashed line, which represent the 95% confidence interval. This result confirms that after the deployment of dynamic slippage, traders experienced reduced losses due to sandwich attacks, resulting in a more favorable trading environment on Uniswap.

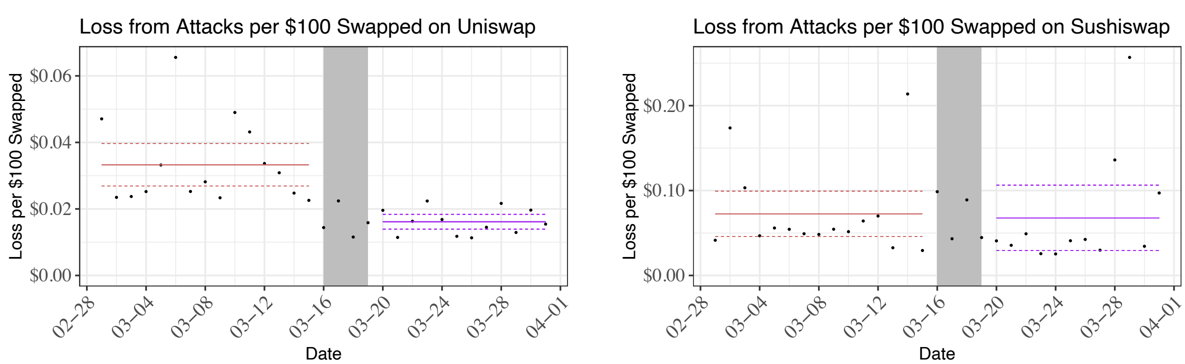


Figure 2.7: Loss from sandwich attacks per $100 on Uniswap vs Sushiswap.

2.5.7 Loss from Unnecessary Failures

We further investigated the impact of dynamic slippage from the perspective of additional losses resulting from unnecessary failures due to potential tighter slippage. By scanning the mempool information, we identified all transactions in our data set that failed due to triggered slippage protection, recording the gas losses incurred from these swaps. An overview of the number of swaps that failed due to triggered slippage and the total gas losses in March can be found in Table 2.5.

	Failed Swaps	Failed due to slippage protector	Gas Loss
Uniswap	18,136	7,928 (43.7%)	$77,813.8
Sushiswap	876	777 (88.7%)	$ 1,643.0
All	19,012	8,705 (45.8%)	$79,456.8

Table 2.5: Failed Swaps that triggered Slippage Tolerance.

Interestingly, losses due to attacks measures 200 - 300 times greater than gas paid upon failure.

2.5.8 Cumulative Impact on Traders' Welfare

Finally, we calculate the total welfare loss, i.e., taking into account both the loss from sandwich attacks and gas losses for unnecessary failed transactions, for each protocol before and after the modification of Uniswap as shown in Table 2.6. We find that overall, Uniswap's new default slippage setting leads to a substantial reduction in traders' losses (54.7%) while in Sushiswap, the reduction is only 4.9%.

Next, we investigate the impact on users who follow the default, i.e., defaulters.

We managed to classify defaulters for 20,477 Uniswap and 7,919 Sushiswap transactions. [9] We will use these groups of transactions that follow the default settings to check if we can observe the same effect or even a greater one in the Uniswap default users. For robustness checks, we will compare it to Sushiswap default users as our control group. This allows us to conduct the same analysis as before, but only on the default users, as shown in Table 2.6. Before the change, default users on both platforms experienced a similar loss, approximately 0.13% of transaction value, which is much higher than the losses of the average user, which are 0.037% in Uniswap and 0.081% in Sushiswap. This implies that attackers may target these 0.5% default users regardless of which protocol they are using. The results show that the effect on Uniswap defaulters was much greater, with a 90% reduction in traders' losses, while for Sushiswap defaulters, there was almost no change at all, with only a 4.9% reduction. These findings suggest that DeFi protocols, such as Sushiswap, should consider changing their default settings to a new mechanism similar to the one utilized by Uniswap.

Loss per $100	Users	Before	After	Change
Uniswap	All	0.0371	0.0168	54.7% decrease
	Defaulter	0.1366	0.0137	**90.0% decrease**
Sushiswap	All	0.0816	0.0706	13.5% decrease
	Defaulter	0.1369	0.1302	4.9% decrease

Table 2.6: Cumulative Impact on Traders' Welfare

[9] We were only able to classify transactions for which we managed to infer their slippage tolerance decisions (as described in section 4.3), and these were executed in pools that met our consistency and liquidity availability restrictions, (as described in section 4.5).

2.5.9 Potential For Further Improvement

Selecting the correct slippage tolerance for a transaction requires balancing the risk of overpaying and the risk of failure due to unexpected, small price movement. We have shown that Uniswap's new automatic default slippage tolerance setting improves the risk of the former with little-to-no additional risk to the latter. Research on further improving this setting must answer the following questions:

1. Exactly how close to zero can the slippage setting be set before being *too small*? We provide a partial answer to this through Figure 2.8, which shows the CDF of the realized slippage percentage in which each transaction incurred. The figure shows that 81.0% of all transactions actually experienced *zero slippage* – i.e., no unexpected price movement occurred whatsoever. Moreover, this percentage grows slowly. This suggests that unexpected price movement may be less common than previously believed, potentially allowing for even smaller slippage settings than the soft minimum of Uniswap's new default (which is 0.1%).

2. Which dynamic slippage mechanism is the optimal one to achieve the highest welfare for traders? The idea of having a dynamic slippage mechanism to determine a customized slippage for each trade is important and could indeed improve traders' wealth and reduce sandwich attacks, as we showed in our analysis. Given that default users are still experiencing losses from sandwich attacks with little-to-no additional cost from unnecessary transaction failures, it can highlight the potential for future improvement.

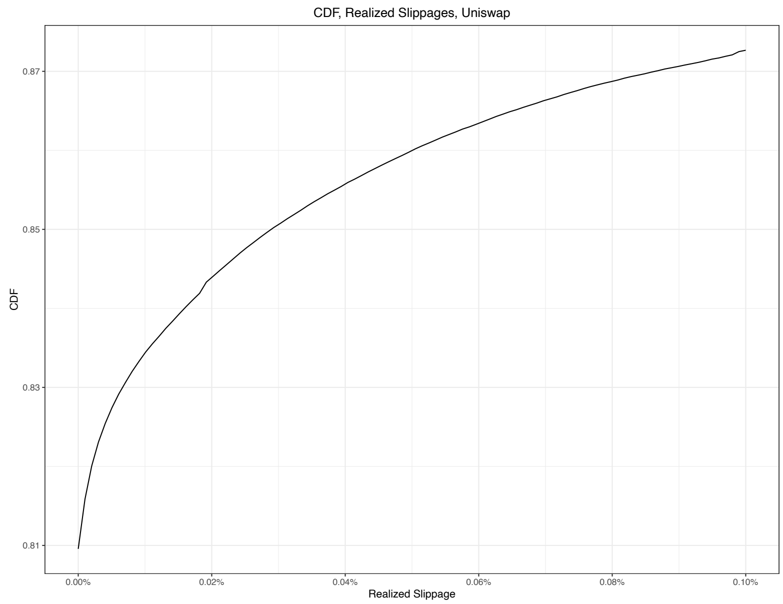


Figure 2.8: Realized slippage in successful Uniswap transactions.

2.6 Conclusions

In this study, we examined the impact of slippage tolerance settings on the health of the decentralized exchange (DEX) ecosystem, focusing on the default slippage tolerance offered by various platforms.

We observed that a significant proportion of transactions, 24% in Uniswap and 67% in Sushiswap, respectively, adheres to the default slippage settings provided by the protocols. This indicates the substantial impact of default options on users' choices and highlights the importance of selecting optimal defaults to promote greater social welfare.

Additionally, we evaluated the effectiveness of Uniswap's recent modification to its default slippage mechanism. By analyzing traders' data, we assessed the benefits of the new default setting in terms of traders' welfare. Our results indicated a substantial reduction in traders' losses, approximately 54.7% (and 90.0% for default transactions). This demonstrates the potential of dynamic slippage mechanisms to improve the overall trading experience and protect users from malicious activities. Considering that default users are still encountering losses due to sandwich attacks, without incurring significant additional costs from unnecessary transaction failures, this suggests that there is room for improvement in the default settings.

The significance of our research highlights the necessity for rigorous academic exploration of default settings within the DEX and decentralized finance (DeFi) domains. By comprehending the implications of default options and introducing innovative approaches to enhance security and traders' welfare, we can actively contribute to the development and long-term sustainability of the DeFi ecosystem.

Chapter 3

Motivating Academic Success: The Role of Leaderboards in Shaping Student Study Behaviors

with Anna Jaskiewicz, Ruth Morales, & Caroline J. Zhang

3.1 Introduction

Leaderboards, which visually display real-time rankings and offer instant performance feedback, have been widely implemented across a variety of contexts. These include athletic competitions (e.g., golf, tennis, chess, etc.), web-based educational platforms (e.g. Khan Academy, Coursera, Doulingo, LeetCode, etc.), ride-sharing mobile applications (Ai et al., 2023) and platforms for public contributions (Chen et al., 2017). These tools are intended to motivate individuals by gamifying experiences and to enhance engagement across activities. However, leaderboards have also been found to lead to a decline in performance in other contexts, such as high school classrooms (Bursztyn and Jensen, 2015; Bursztyn et al., 2019). This has been attributed to disclosure of real names and identities on the leaderboard, which subsequently discouraged some individuals from further competition. Nonetheless, these results do not necessarily imply that leaderboards as such are an unreliable incentive tool. It may simply be that certain settings are particularly sensitive to specific leaderboard features, such as lack of anonymity.

In this paper, we explore if an anonymized leaderboard can effectively incentivize students to enhance their study behaviors and achieve greater academic success. The field experiment is conducted in the context of an undergraduate Economics course at a large public university in California. We leverage an online autograding platform through which students submit their weekly self-paced assignments. Within the treatment group, students have access to a leaderboard that ranks them based on submission time, conditional on successfully completing their assignment. Within the the control group, students do not have such access to the leaderboard.

We find that the implemented leaderboard positively shapes students' study behaviors. First, we report that exposure to the leaderboard reduces assignment completion times. These effects are substantial in magnitude, with an over 15 hour and a nearly 20 hour reduction for the first two treated assignments. The effects persist–although smaller–throughout the quarter. Second, we also find that treated transfer students and treated male students experience an improvement in their overall course grade by a letter grade (e.g., from $B-$ to B). This is particularly relevant given that transfer students encounter unique challenges when adapting to a new academic environment, which could hinder their ability to achieve the grade they desire.

We contribute to several lines of work. First, we extend to the literature directly evaluating how leaderboards as well as other ranking systems affect effort provision. Past research in this space has largely relied on public (non-anonymous) leaderboards to generate sufficient social pressure to impact behavior (Bursztyn and Jensen, 2015;

Bursztyn et al., 2019; Gill et al., 2019; Hudja et al., 2022). We focus on a leaderboard that is *anonymous*, i.e., it does not reveal true identities of participants and instead relies on confidential pseudonyms selected by participants themselves.

Second, we contribute to the scholarship documenting the various ways of gamifying academic assessments and the effects of such gamification (Markopoulos et al., 2015; Chang and Wei, 2016; Subhash and Cudney, 2018; Dichev and Dicheva, 2017; Buckley and Doyle, 2014). While the leaderboard itself already introduces an element of gamification, we amplify this by enabling students to design their own playful pseudonyms that are displayed on the leaderboard instead of their real names.

Third, we extend the line of work on procrastination in the academic context Schiming (2012); Hen and Goroshit (2018); Dewitte and Schouwenburg (2002). We show that a leaderboard which ranks individuals based on not only submission accuracy but also submission speed can be an effective tool at reducing procrastination.

Fourth, we contribute to the literature exploring how symbolic rewards affect behaviors. Prior scholarship has documented that interventions such as thank you cards are successful at increasing effort provision and improving performance (Bradler et al., 2016; Kosfeld and Neckermann, 2011). Despite the fact that in our setting one's rank is not associated with any "real" rewards, such as extra course credit, the leaderboard still induces a significant behavioral change. In consequence, achieving a high rank on a leaderboard may also act as symbolic reward.

Finally, past research documented that interventions such as mentoring programs can help individuals from groups underrepresented in the field of Economics at relatively later stages of their career, i.e. post graduation (Ginther et al., 2020; Ginther and Na, 2021). We show that leaderboards have heterogeneous effects across student subgroups and may thus be used as tools that potentially help foster diversity within the profession at earlier career stages (pre-graduation).

The remainder of this paper is structured as follows: Section 3.2 introduces the background information and outlines the general setup of the field experiment. Section 3.3 offers an overview of the collected data and conducts a balance check. Section 3.4 delineates our hypotheses, while Section 3.5 describes our identification strategy. Section 3.6 presents the main findings of our analysis. Finally, Section 3.7 provides a conclusion and discusses the implications of our results.

3.2 The Experiment

3.2.1 Setting

We conducted the experiment within an undergraduate Economics course taught at a large public university in California. The university is designated as Hispanic serving; it also hosts a non-trivial number of transfer students from the local community college as well as international students.

The economics course is required for all prospective Economics majors and covers topics such as probability and probability distributions, sampling and sampling distributions, and hypothesis testing. The course is offered every quarter throughout the academic year but tends to achieve its highest enrollment levels in the Fall quarter, which is when we carried out the study. The course is usually taught by graduate student instructors, and students are split into sections capped at 50.

In Fall 2023, when we carried out the experiment, there were 642 students enrolled across 15 different sections. As this is an introductory pre-major course, enrolled students predominantly consist of sophomores and juniors. Majority of enrollments are typically finalized before the first day of instruction but a small number of students tend to drop the class or join from wait list during the first week of the quarter. To account for this, we recruited participants during the second week of the quarter.

3.2.2 Excel Assignments

The final course grade is calculated based on student's performance on a number of assessments: a midterm and a final exam, short lecture quizzes, group activities performed in class, self-paced homework modules, and online Excel assignments.

Online Excel assignments are short applied projects which students solve using Microsoft Excel. Excel assignments typically are due on Friday each week of class, with a penalty-free grace period extending until Sunday at midnight. Prompts for each Excel assignment are released to students periodically throughout the quarter. Data used in each assignment is unique to each student and is generated using a custom-built R script. The data is made available to students through a website administered by the instructors (see Appendix C.1 for a screenshot of the assignment download site). The release and due times for all assignments were made available to students in advance.

Each Excel assignment is graded out of 5 points and frequently involves both computational and graphing questions. Assignments are self paced in the sense that students can spend as much time as they wish working on them as long as the grace period has not elapsed. Additionally, students can submit assignments multiple times and only the score achieved on their last submission counts towards the course grade calculation. Usually, students have around 10-14 days to work on each Excel assignment, i.e. there is 10-14 day gap between the time when a prompt is made available to students and the time when grace period finishes.

Performance on Excel assignments can substantially impact a student's course grade as seven best scores across nine assignments in total contribute 28% of the overall grade. To assist students as they work on Excel assignments, experienced peer helpers employed by the department hold weekly office hours dedicated specifically to answering questions on each week's Excel assignment. Students enrolled in the course also receive general Excel instruction during weekly discussion sections led by graduate student teaching assistants.

3.2.3 Leaderboard

Excel assignments are submitted by students through an online autograding platform, which allows for immediate feedback on performance on the computational part of the assignment.[1] The platform allows instructors to include a leaderboard as a separate subpage feature of each assignment page (a leaderboard is by default hidden unless it is explicitly enabled).

We customize the leaderboard on the platform so that it ranks all submissions by the submission time among the class members, i.e., the earlier the submission time, the higher the rank on the leaderboard. At the same time, we constrain the leaderboard to only rank those who receive a full score on the autograded (computational) part of the assignment. For example, if Alice and Bob both receive a full score, and Alice submits earlier, then Alice will appear higher on the leaderboard than Bob. A student will appear at the bottom of the leaderboard if the student's submission scores less than a full score on the autograded part of the assignment.

In place of actual student names, the leaderboard displays student-selected leaderboard pseudonyms. This is done for two reasons: first, to preserve confidentiality of each student's performance of the assignments, and second, to further gamify the experience for students by enabling them to select playful names they enjoy using and identify with. Students get to input their leaderboard pseudonym upon submitting their Excel

[1] Students are required to submit their solutions to computational and graphing problems separately. While the autograding platform provides instant feedback on performance on the former, the latter are graded manually after the grace period elapses.

assignment. As leaving the leaderboard pseudonym blank would cause a submission not to be processed, students are informed that if at any point they no longer wish their leaderboard pseudonym to be uniquely identified, they can type Optout Otter as their leaderboard pseudonym instead.

It is important to note that after they submit their Excel assignment, students are not automatically redirected to the leaderboard page but instead need to navigate to it intentionally. Figure 3.1 presents an illustrative screenshot of the leaderboard as viewed from the student perspective.

3.2.4 Recruitment and Randomization

We conducted the study in the Fall quarter of academic year 2023/2024. The recruitment process took place during the second week of classes, to mitigate the effect of students joining the class from the wait list or dropping the class during the first week of classes. During in-person lectures, we distributed consent forms to students, along with a survey requesting basic demographic information. To encourage participation, students were informed that they would be entered into a lottery for a $25 gift card, with odds of winning estimated at approximately 1 in 50. It is important to note that apart from the lottery drawing, there were no other incentives provided to the participants, such as extra credit for class.

Of the 642 students enrolled in the course, 425 consented to participate in our study. Students were randomly divided into a treatment group and a control group. We decided to further split treated students into 50-student leaderboards to allow for relatively high mobility potential along the leaderboard and relatively high familiarity with others on the same leaderboard. Consequently, participants were randomly assigned to one of eight mutually exclusive and collectively exhaustive groups, each linked to a unique course page on the autograding platform. Of these, seven groups were designated as treatment groups and had access to the leaderboard, thereby differentiating them from the control group, which did not have access to the leaderboard.

For all Excel assignments, students assigned to the control group received standard prompts which outlined the requirements of each assignment and did not mention the leaderboard in any way. In addition to the standard prompts, students assigned to the treatment group received extra instructions which provided an overview of the leaderboard setup. In particular, the instructions explained how the leaderboard ranks individual student submissions, where to find it on the online submission platform, and how to select pseudonyms for the leaderboard.[2]

3.3 Data

Our data were sourced from four principal channels: the demographic survey completed by students immediately following their consent to participate in the experiment, the platform for Excel assignment submissions, and the course instructor's records. The demographic survey provided basic demographic information about the students, such as age, gender, international status, and transfer status. Information regarding the Excel assignment submissions was collected directly from the autograding platform. Lastly, the course instructor's records provided information on the overall grades students achieved in the course.

[2] Appendix C.2 contains the exact text delivered to the treatment group within the instructions for their Excel assignments.

3.3.1 Balance Check

Our sample consists of 425 students, with 236 assigned to the treatment group (i.e., with access to the leaderboard) and 189 students assigned to the control group (i.e., without access to the leaderboard). Table 3.1 provides a summary of the demographic characteristics of students in both the treatment group (Column 1) and control group (Column 2). Slightly over 40% of the students in our sample are male, about a half are transfer students, and over 90% are either sophomores or juniors. We report no statistical difference between the treatment group and the control group across nearly all demographic characteristics, except for Hispanic/Latino and International status (marginally significant).

3.3.2 Excel Assignment Submissions

For our outcome variables, we leverage information on various aspects of students' submissions for the nine Excel assignments, including the assignment release time and submission time, assignment scores, and leaderboard ranks.

Across the nine Excel assignments, we focus on student performance on assignments 2 through 7. This is because the prompt for Excel assignment 1 is released in the very first week of classes, prior to when students are introduced to the leaderboard. Moreover, the first assignment is designed to be straightforward, primarily to acquaint students with the autograding platform. For these two reasons, we anticipate no significant difference between the treatment and control groups in terms of completion times. As such, the first assignment also serves as a placebo test to verify the comparability of the treatment and control groups at the outset of the study.

Furthermore, recall that only the best seven scores on the nine Excel assignments count towards to the final course grade. Therefore, a student may choose to delay or altogether skip the submission for the last two assignments if they manage to achieve full marks on the initial seven assignments. Consequently, we omit assignments 8 and 9 from our primary analyses to account for this behavior.

3.4 Hypotheses

In this section, we develop our hypotheses within the context of the field experiment's framework.

Hypothesis 1 (Immediate Submission Effect). Following their initial exposure to the leaderboard, students in the treatment group will submit the subsequent Excel assignment more promptly than their counterparts in the control group.

After submitting their first Excel assignment, students in the treatment group will encounter the leaderboard for the first time. We anticipate that this initial engagement with the leaderboard–as well as the novelty of the leaderboard–will motivate students to improve their rank for the second assignment (if their original position is low) or to maintain their original rank (if their original position is high). Consequently, we hypothesize that students interacting with the leaderboard will, on average, finish Excel assignment 2 more quickly than their counterparts in the control group.

Hypothesis 2 (Persistent Submission Effect). Students in the treatment group will consistently complete their Excel assignments faster than students in the control group throughout the academic quarter.

A new leaderboard is created for each Excel assignment, a deliberate design choice to provide all students in the treatment group with a clean slate for each assignment. This mechanism serves to motivate students as everyone can rank highly on a new leaderboard, regardless of their prior rank. Accordingly, we hypothesize that the leaderboard will continue to affect assignment competition times throughout the quarter, i.e., from the second through the seventh Excel assignment. In other words, we anticipate that the leaderboard will have a lasting impact on students' study behaviors, beyond just the immediate effect posited in Hypothesis 1.

Hypothesis 3 (Academic Performance Effect). The leaderboard will not only positively affect assignment completion times but will also improve the overall performance in the class, leading to a higher final course grade.

The effects of leaderboard exposure may extend beyond just Excel assignments. Students may develop a habit of initiating their work on various tasks early on. For example, students may start studying for a test earlier than they otherwise would have. Similarly to the benefits associated with starting Excel assignments ahead of time, this may provide students with more opportunities to seek (and receive) more assistance from instructors and Teaching Assistants, leading to better performance overall. At the same time, because the treatment is not associated with any financial incentives or extra academic credits, we do not expect these effects to be very high in magnitude. Therefore, we expect the leaderboard intervention to exert a moderate indirect effect on students' final course grades.

Hypothesis 4 (Heterogeneous Effects on Different Demographic Groups). The leaderboard will have heterogeneous academic performance effects across different demographic groups.

Different demographic groups may respond to the leaderboard differently. We expect students who have prior experiences with various types of leaderboards, e.g. through gaming platforms or athletic events, to be more strongly affected than students who do not have such experiences. We observe in the data we collected through the demographic survey that male students more frequently report being engaged in video games and sports. Thus, we expect the effects of the leaderboard to be concentrated among male students.

In addition, our field experiment takes place at a university with a substantial population of transfer students from community colleges. Transfer students may be particularly responsive to interventions that encourage engagement. We hypothesize that the leaderboard will have stronger effects among transfer students than non-transfer students.

3.5 Identification

In this section, we specify our identification strategy.

To estimate the average treatment effect of exposure to a leaderboard on study habits, we first estimate the following OLS regression model:

$$Y_i^j = \alpha + \beta Treat_i + \gamma X_i + \epsilon_{ij} \tag{3.1}$$

where Y_i^j denotes the number of hours taken by student i to submit their Excel assignment j calculated as the difference between the time when the assignment prompt is released and the time when student i makes their last submission for assignment $j \in \{1, 2, \ldots, 7\}$. $Treat_i$ is a binary indicator equal to one if a student is assigned to the treatment group (i.e., exposed to the leaderboard) and equal to 0 if a student is assigned to the control group (i.e., not exposed to the leaderboard). Therefore, β, which is the main parameter of interest, indicates the effect of being assigned to treatment on assignment completion time. X_i is vector of demographic controls that include student's gender, age, race, ethnicity, major, transfer and international status. Wild-bootstrap standard errors are clustered at the level of enrollment section. The model is estimated separately for assignments one through seven, with assignment one being treated as a placebo as it was released prior to initial exposure to the leaderboard.

Furthermore, to estimate the average treatment effect of the leaderboard intervention on students' academic performance, i.e., the final course grade, we then estimate the following regression model:

$$Y_i = \alpha + \beta Treat_i + \gamma X_i + \epsilon_i \tag{3.2}$$

where Y_i denotes student i's final course grade operationalized either as a z-transformation of the overall number of points accumulated by student i throughout the duration of the class or as the raw number of points accumulated by student i throughout the duration of the class.

To allow for heterogeneous effects by transfer status and gender, we also estimate supplementary regression models in which we include interaction terms between a student's treatment assignment as well as their transfer status and gender.

3.6 Results

In this section, we present the results from our field experiment. We first examine the effect of the leaderboard intervention on students' assignment completion times. We then examine the impact of exposure to a leaderboard on students' academic performance.

Finally, we end with a discussion of heterogeneous effects on transfer and male students.

Figure 3.2 plots the average assignment completion times separately for the treated leaderboard group (purple line) and the non treated no leaderboard group (orange line) across assignments 1 through 7. It is worth noting here that Figure 3.2 merely shows the raw means across the treatment and control groups and does not include any demographic controls. The average completion times for students in the treatment group fall consistently below the average submission times of students in the control group. The only exception is Excel Project 1 which was released prior to the initial exposure to the leaderboard and can be therefore considered a placebo. This provides suggestive evidence in support of both Hypothesis 1 (Immediate Submission Effect) and Hypothesis 2 (Persistent Submission Effect).

Figure 3.3 provides the effects of being assigned to treatment on assignment completion time estimated using Equation 3.1. Recall that completion time is defined as the difference between the last submission time and assignment release time. The effects are estimated separately for assignments 1 through 7. For assignment 1, we observe no statistical difference in assignment completion times between the treatment and the control group. As explained before, this is intuitive as assignment 1 can be interpreted as a placebo. In the case of subsequent Excel assignments, exposure to the leaderboard is associated with a statistically significant reduction in assignment completion time for all assignments except for assignment 6. The prompt for assignment 6 was released soon before the midterm exam. Given that the midterm exam score has a larger impact on the overall course grade than an individual Excel assignment, students might have prioritized preparing for the midterm and simultaneously postponed completing the assignment. The effects on assignments 2, 3, 4, 5, and 7 are substantial in magnitude. Notably, in the case of Excel Assignment 2, being assigned to the treatment group resulted in a over 15-hour reduction in assignment completion time, which corresponds to over half a day. The strong immediate effect of the leaderboard on assignment 2 completion time provides support for Hypothesis 1. This effect, although decreasing in magnitude, persists throughout the quarter. Even in the case of assignment 7 (the last assignment we consider, which is due almost two months after the initial exposure to the leaderboard), the reduction in completion time is still significant (over 10 hours). The persisting effect of the leaderboard on completion time provides evidence for Hypothesis 2.

Is the leaderboard also associated with a better performance in the class overall? Table 3.2 presents the effects of the treatment (leaderboard) on z-transformed final course grades, estimated using Equation 3.2. We report no significant effects of the treatment on total course grades (Columns 1 and 2). Although going against Hypothesis 3, these results are not surprising. When the sample as a whole is considered, exposure to the leaderboard may be too light-touch of an intervention to result in a change in behavior extending beyond assignment completion times, which would subsequently improve the final course grades. It is unlikely that all students in the sample respond to the treatment in the same way, which could affect the overall treatment effect. However, certain demographic groups may be more sensitive to exposure to the leaderboard than others. Therefore, in columns 3 through 8, we explore whether the leaderboard has different effects on subgroups of students.

There appears to be substantial treatment heterogeneity for transfer/non-transfer students as well as male/female students, as put forward in Hypothesis 4. Columns 3 and 4 provide the coefficient estimates when the baseline model in Equation 3.2 is extended by adding an interaction term between a student's transfer status and treatment assignment. Transfer students assigned to the treatment group appear to achieve higher overall course grades relative to their peers assigned to the control group.[3] This finding is particularly

[3]P-values from the F-test for joint significance significant at the 10% level.

important given that transfer students are often considered as being at a higher risk of attrition from the major track than non-transfer students due to the additional challenges associated transitioning from another school. Similarly, columns 5 and 6 provide the coefficient estimates when the baseline model in Equation 3.2 is extended by adding an interaction term between a student's gender and treatment assignment. Male students assigned to the treatment group tend to achieve higher overall course grades as compared to their peers assigned to the control group.[4] These effects are non-trivial in magnitude. To put this is more context, we replicate the analyses presented in the previous table with raw final course grade as the outcome (as opposed to z-transformed course grades). Table 3.3 presents the effects of the treatment on the final course grade when the final course grade is operationalized as the raw number of points accumulated throughout the course across all assignments. As being assigned to the treatment increases the overall course grade by around 3 points for transfer students, the effect is large enough to move up a letter grade (e.g., from B- to B).

3.7 Conclusion

We conducted a field experiment in a large undergraduate Economics course to investigate whether gamifying online Excel assignments through implementation of leaderboards will affect assignment completion times and overall course performance. Our results indicate that students assigned to the treatment group exhibit faster assignment completion times, suggesting that the leaderboard positively alters students' study behaviors. At the same time, while we do not find significant effects of the leaderboard on performance in the class in the context of the entire sample, we report that transfer students as well as male students exposed to the leaderboard achieve higher overall course

[4]P-values from the F-test for joint significance marginally significant at the 10% level.

grades than their peers in the control group. These results are particularly interesting given that the intervention (leaderboard) is light-touch (i.e., students are not required to check the leaderboard) and costless (i.e., implementing the leaderboard on the assignment submission platform requires no additional fees as well as minimal effort upfront and virtually no effort associated with maintenance).

Our study, like all studies, has its limitations. First, we capped the leaderboards at 50 students which we hoped would create a sufficient potential for mobility along the leaderboard without alienating the low performers. However, future work should explore more closely the relationship between leaderboard size and leaderboard effects to arrive at a more precise optimum. Second, in our study we did not have the technical infrastructure to automatically redirect students to the leaderboard upon assignment submission, nor did we have access to the information on if and how frequently individual students chose to inspect the leaderboard. Future research may experiment with various leaderboard settings, while attempting to collect more data on how students interact with the leaderboard. Third, our study was limited to a single class. It would be interesting to explore if leaderboards implemented in several courses at once crowd each other out or amplify each other's effect.

Despite these limitations, our findings still provide important insights into how leaderboards can be used as incentives to shape behaviors. More narrowly, our results can be directly used to guide how instructors design their assignments with the goal of promoting engagement within digital learning spaces and fostering positive study habits among students.

Figures

[This page intentionally left blank]

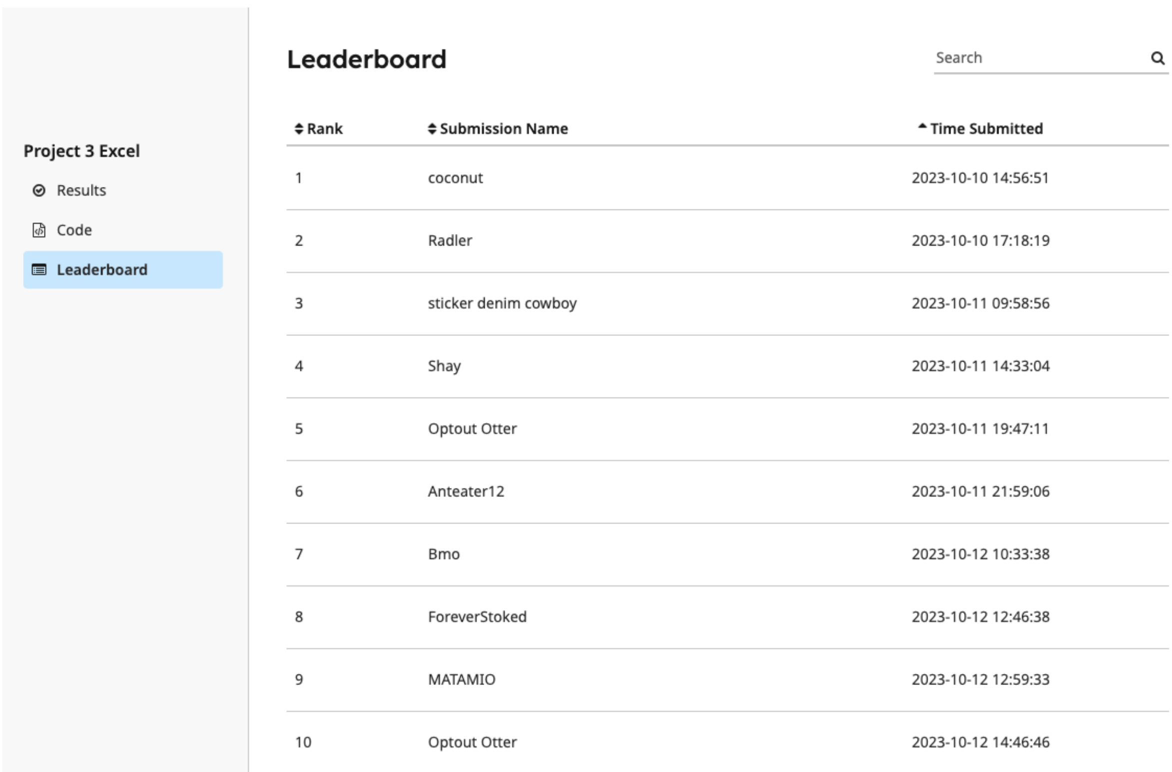


Figure 3.1: Screenshot of the leaderboard as viewed from the student perspective

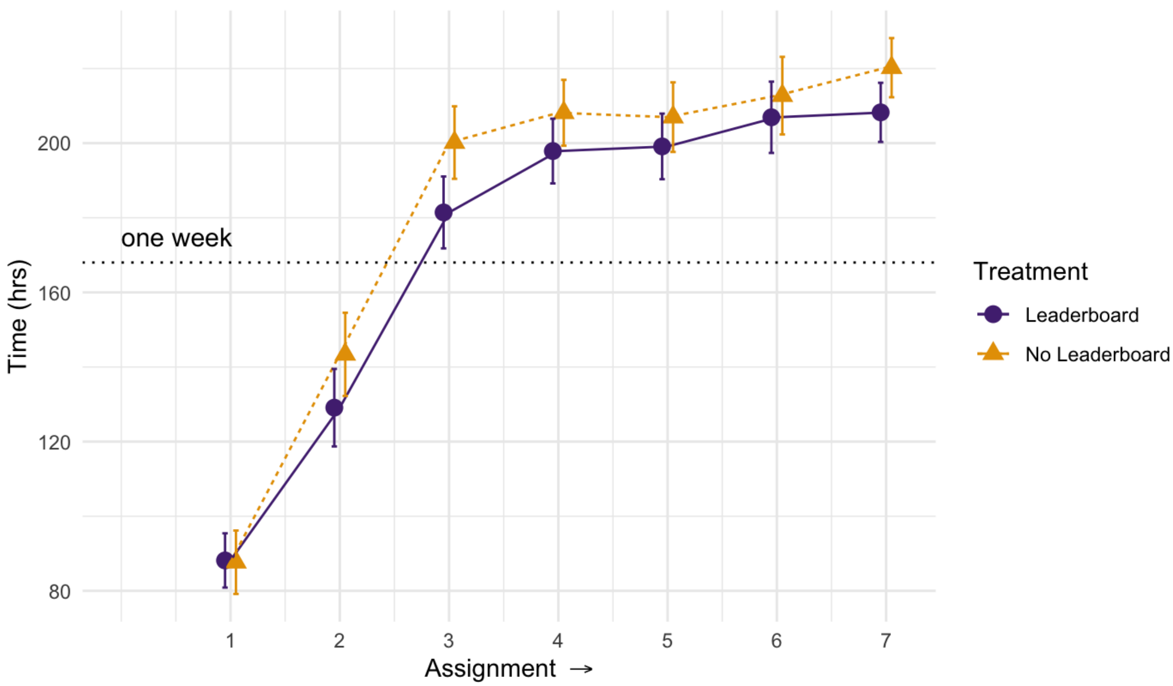


Figure 3.2: **Average Completion Times for Treatment (Leaderboard) Group and Control (No Leaderboard) Group Across Assignments 1 Through 7**

Notes: Purple circles denote raw mean completion times for students in the treatment group (no controls included). Orange triangles denote raw mean completion times for students in the control group (no controls included). Whiskers denote 90% confidence intervals. Assignment 1 provides a placebo test (no expected difference between treatment and control group). Completion time is defined as the difference between assignment release time and the last submission made by the student.

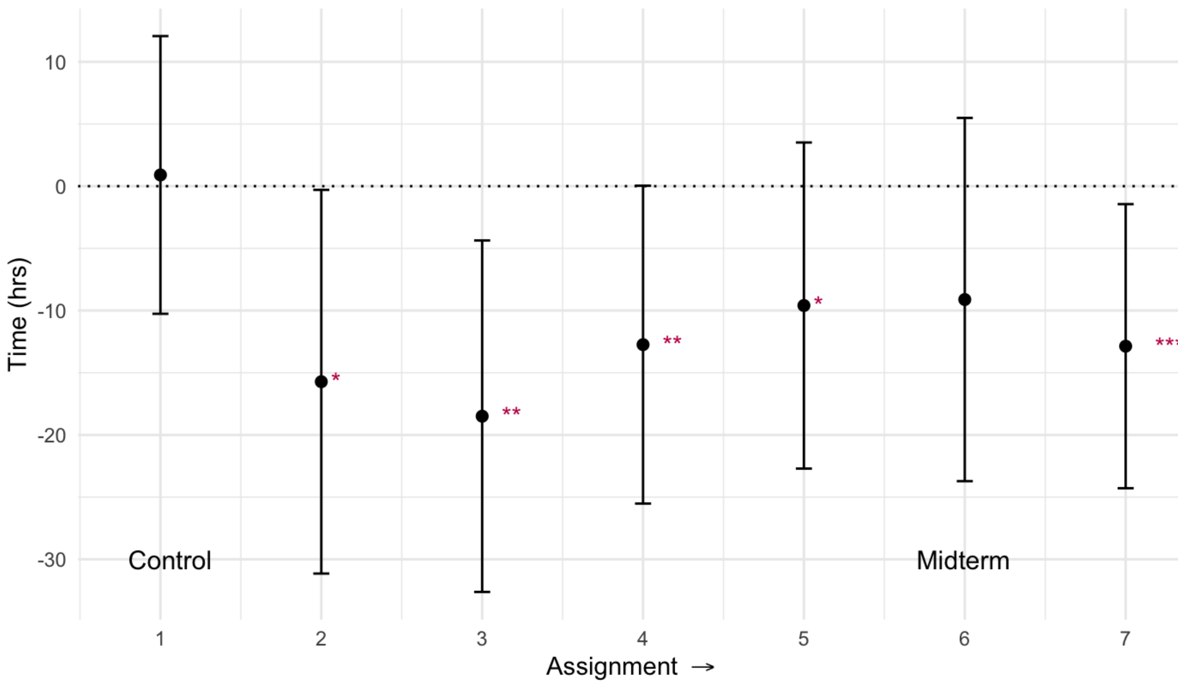


Figure 3.3: **Regression Estimates of the Treatment Effect (Leaderboard Exposure) on Completion Times Across Assignments 1 Through 7**

Notes: Circles denote coefficients estimated using equation (1). Whiskers denote 90% confidence intervals. * denotes significance at the 10% level, ** denotes significance at the 5% level, and *** denotes significance at the 1% level. Controls include student's gender, age, race, ethnicity, major, transfer and international status. Wild-bootstrap standard errors are clustered at the level of enrollment section. Assignment 1 provides a placebo test (no expected difference between treatment and control group).

Tables

[This page intentionally left blank]

Table 3.1: Balance Check Across Treatment (Leaderboard) and Control (No Leaderboard) Groups

	(1) Leaderboard Mean (Std.)	(2) No Leaderboard Mean (Std.)	(3) Mean Diff. (1) - (2)
Male	0.413 (0.496)	0.421 (0.495)	-0.008
Transfer	0.504 (0.518)	0.481 (0.501)	0.023
Age 18 to 20	0.778 (0.416)	0.760 (0.429)	0.019
Age 21 to 24	0.209 (0.407)	0.213 (0.411)	-0.004
Age 25 or more	0.013 (0.114)	0.027 (0.164)	-0.014
Major Econ	0.565 (0.497)	0.546 (0.499)	0.019
Major Econ/Acc	0.400 (0.491)	0.383 (0.487)	0.018
Major Other	0.022 (0.146)	0.049 (0.217)	-0.028
Major Undecided	0.013 (0.114)	0.022 (0.147)	-0.009
Hispanic/Latino	0.135 (0.342)	0.208 (0.407)	-0.073*
Race White	0.509 (0.501)	0.443 (0.498)	0.066
Race Black	0.057 (0.231)	0.049 (0.217)	0.007
Race Asian	0.344 (0.476)	0.372 (0.485)	-0.028
Race Other	0.091 (0.289)	0.137 (0.344)	-0.045
Year Freshman	0.039 (0.194)	0.044 (0.205)	-0.005
Year Sophomore	0.430 (0.496)	0.432 (0.497)	-0.001
Year Junior	0.509 (0.501)	0.497 (0.501)	0.011
Year Senior	0.022 (0.146)	0.027 (0.164)	-0.006
International	0.087 (0.282)	0.153 (0.361)	-0.066*
N (Students)	236	189	

Table 3.2: Effect of Treatment (Leaderboard) on z-Transformed Final Course Grades

	(1)	(2)	(3)	(4)	(5)	(6)
Leaderboard	0.032 (0.104)	-0.022 (0.096)	-0.132 (0.116)	-0.198 (0.116)	-0.114 (0.152)	-0.178 (0.145)
Transfer			-0.709*** (0.136)	-0.588*** (0.190)		-0.394** (0.180)
Leaderboard * Transfer			0.344** (0.154)	0.364** (0.157)		
Male				-0.051 (0.129)	-0.092 (0.158)	-0.250 (0.166)
Leaderboard * Male					0.350** (0.148)	0.373** (0.147)
Additional Controls	-	✓	-	✓	-	✓
Observations	413	413	413	413	413	413
Number of students (cluster section level)	31	31	31	31	31	31

Notes: Significant at the *10%, **5%, and ***1% levels. Robust standard errors for clustered data in parentheses. Constants not displayed. Additional Controls: Age, Hispanic/Latino, Race, Major, International Student.

Table 3.3: Effect of Treatment (Leaderboard) on Raw Final Course Grades (%)

	(1)	(2)	(3)	(4)	(5)	(6)
Leaderboard	0.438	-0.304	-1.820	-2.739	-1.570	-2.452
	(0.144)	(1.322)	(1.610)	(1.604)	(2.100)	(1.995)
Transfer		-5.300**	-9.780***	-8.119***		-5.432**
		(2.432)	(1.880)	(2.619)		(2.481)
Leaderboard * Transfer			4.740**	5.026**		
			(2.120)	(2.167)		
Male		-0.619		-0.704	-1.270	-3.457
		(1.782)		(1.774)	(2.180)	(2.287)
Leaderboard * Male					4.830**	5.150**
					(2.040)	(2.026)
Additional Controls	-	✓	-	✓	-	✓
Observations	413	413	413	413	413	413
Number of students (cluster section level)	31	31	31	31	31	31

Notes: Significant at the *10%, **5%, and ***1% levels. Robust standard errors for clustered data in parentheses. Constants not displayed. Additional Controls: Age, Hispanic/Latino, Race, Major, International Student.

Appendix A

Appendix for "Estimating Investor Preferences for Blockchain Security"

A.1 Linear Discriminant Analysis

A competing method of binary logistic regression is linear discriminant analysis, a linear method in classification. While the relative efficiency of linear discriminant analysis (LDA) is superior to binary logistic regression (BLR) if the LDA's assumptions are met (Efron, 1975), the assumption of normality is hard to meet with our data. In one predictor (W) case, the LDA assumes that $W|Y = k \sim N(\mu_k, \sigma^2)$, that is, the predictor given a different class, follows a normal distribution with different mean and variance. We test this assumption and found the predictor is far from normal distributed through Skewness-kurtosis graph (Cullen et al., 1999).[1]

[1] Using R package *fitdistrplus* (Venables and Ripley, 2002).

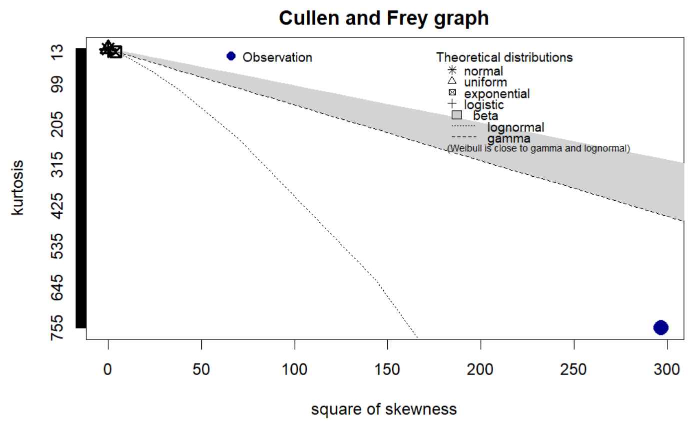


Figure A.1: Skewness-kurtosis graph, for Ethereum transactions

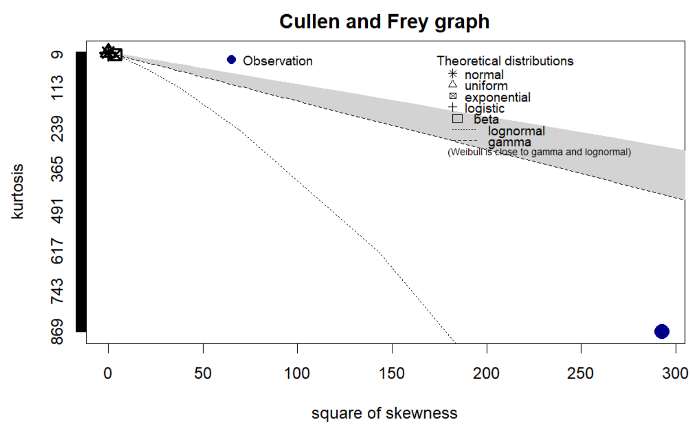


Figure A.2: Skewness-kurtosis graph, for Optimism transactions

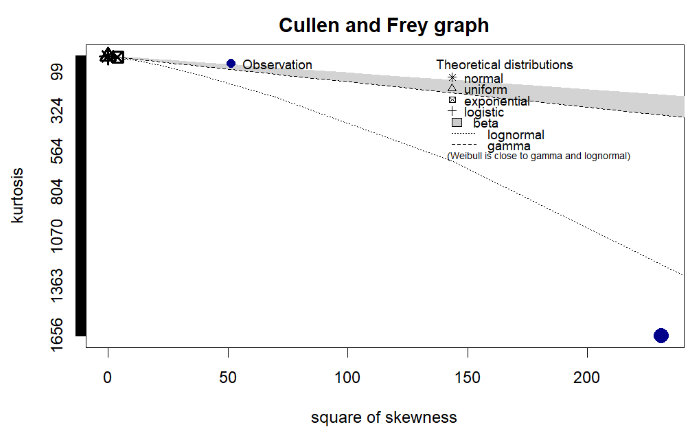


Figure A.3: Skewness-kurtosis graph, for Polygon transactions

A.2 Threshold the predicted probability

After obtaining the logit regression, we can predict the class (transaction is on L2 or Ethereum) by thresholding the predicted probability. For example, one might predict $Y = 1$ (on L2) for any transaction value whose predicted probability is greater than 0.5. Or, if we are being conservative in predicting transaction value to be in Ethereum, we could predict $Y = 1$ (on L2) for any transaction value whose predicted probability is greater than 0.1.

To evaluate the classification performance under different threshold probability, one can construct confusion matrix and pin down the threshold probability that obtain a low false positive rate (FPR, the fraction of negative examples that are classified as positive, which in our study is the portion of transaction on Ethereum that are classified as on L2) while also maintaining a low false negative rate (FNR, the portion of transaction on L2 that are classified as on Ethereum). We want to choose the probability threshold that is closest to (FPR, FNR) = $(0, 0)$. There are many ways to determine which threshold probability corresponds to the smallest distance, but we calculate the euclidean distance between each point of (FPR, FNR) and $(0, 0)$. Figure Figure A.4 and Figure A.5 show the ROC curve and optimal threshold selection for 2022/4/11 data.[2] The optimal threshold for that day is 77.91%, meaning when predicting platform, only when the $Pr(Y = 1|W) > 77.91\%$, we category the transaction to be on L2.

[2] Using R package *ROCR* (Sing et al., 2005).

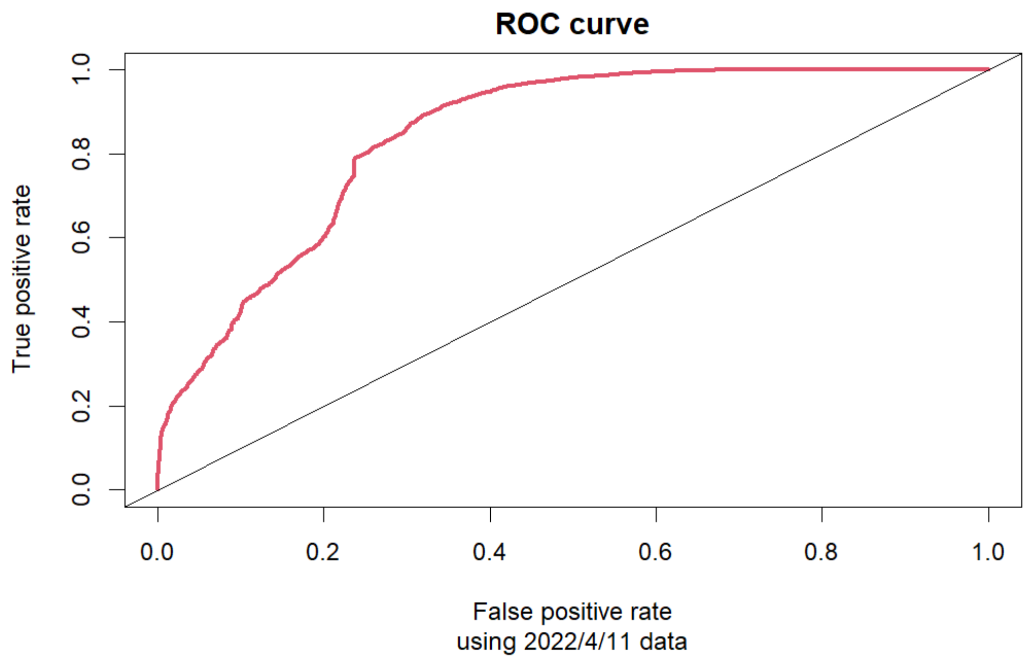


Figure A.4: Example ROC curve (2022/4/11 data)

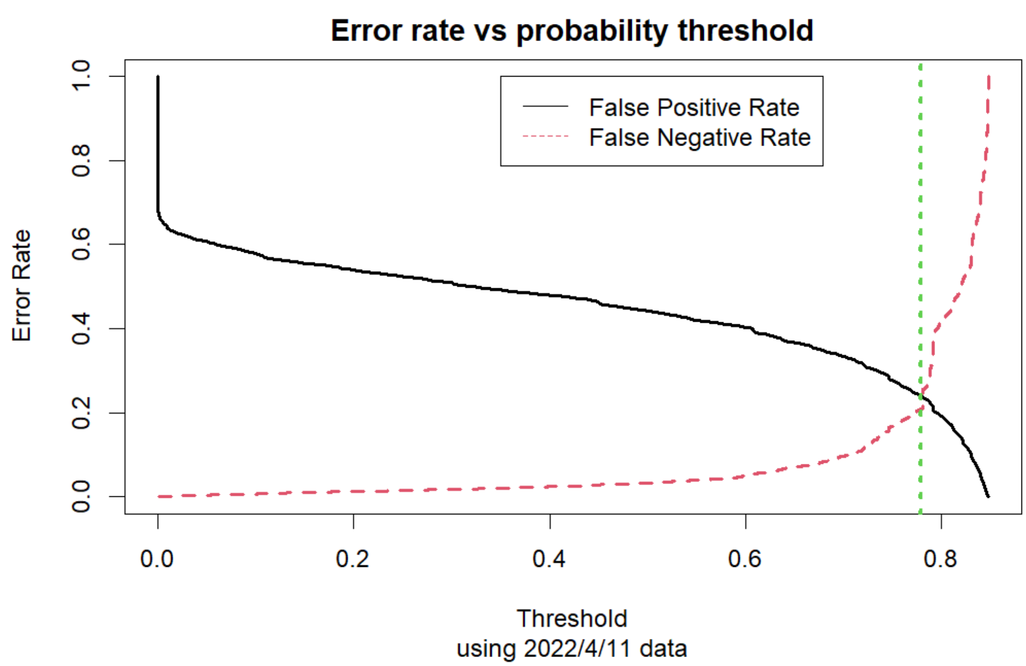


Figure A.5: Example FPR, FNR graph (2022/4/11 data)

A.3 Wallets that swap on both ETH and POLY

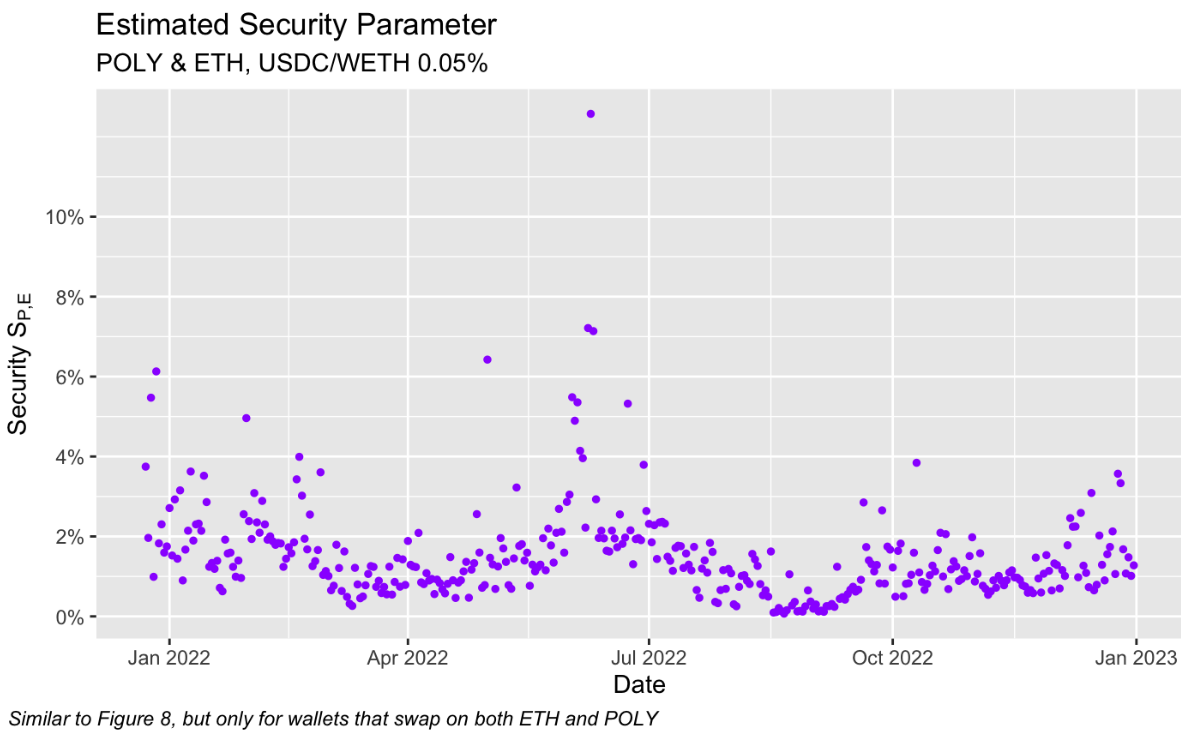


Figure A.6: Estimated Security Parameter for subset data

A.4 Generalization Optimism

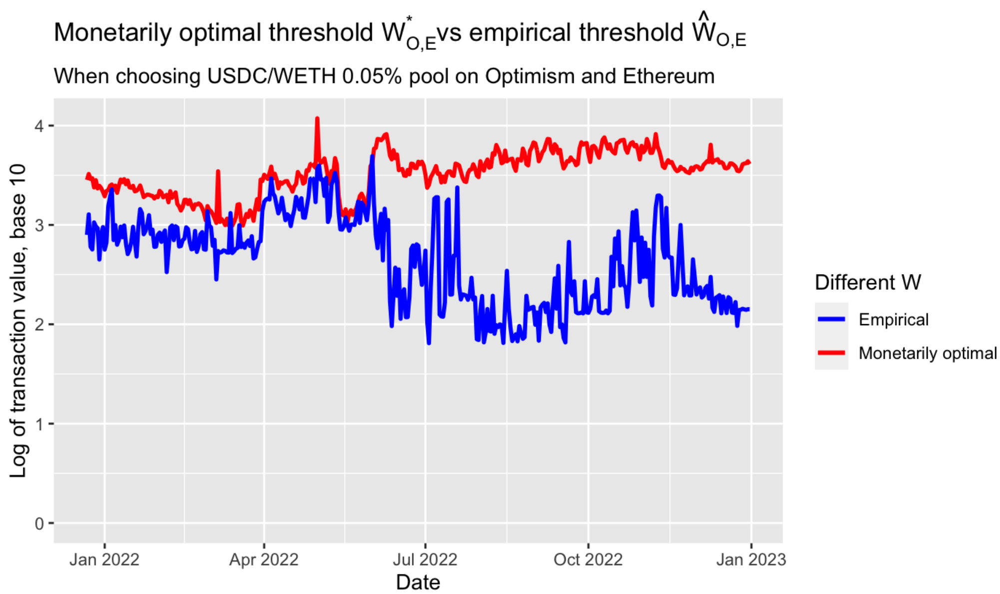


Figure A.7: Monetarily Optimal Threshold $W^*_{O,E}$ vs Empirical Threshold $\hat{W}_{O-E}$

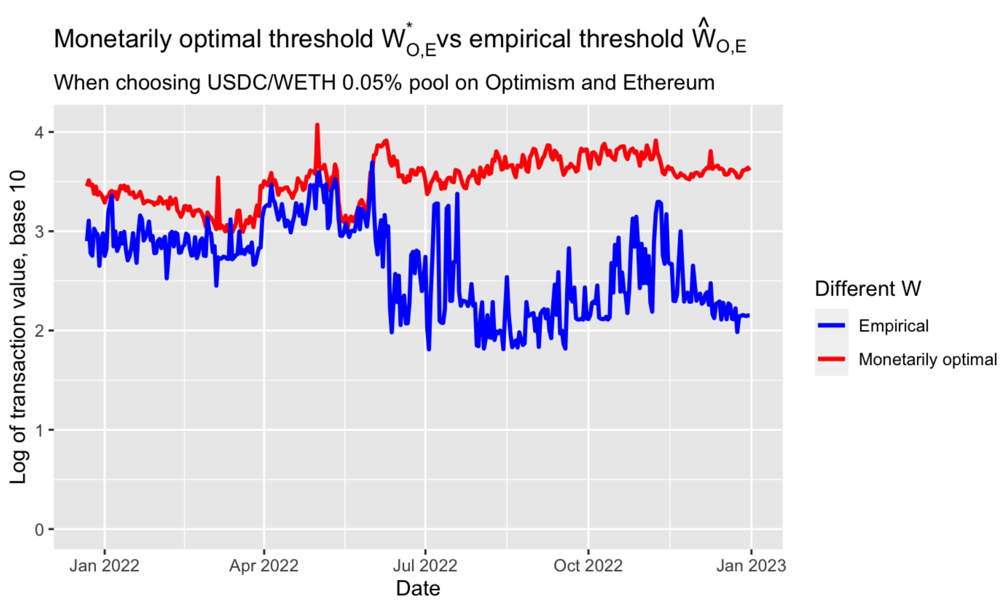


Figure A.8: Estimated Security Parameter, O,E

A.5 Generalization all other pairs

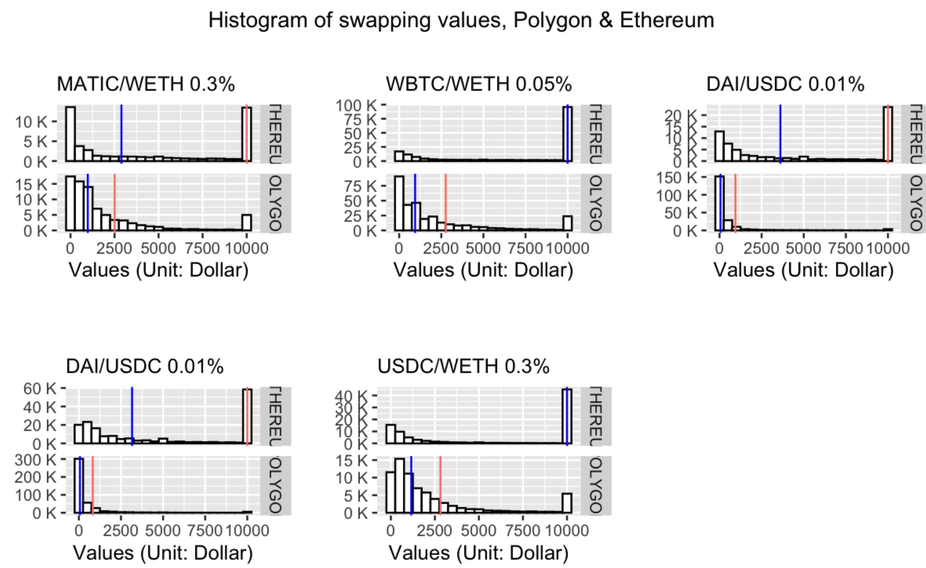


Figure A.9: Histogram of 5 other pairs from Polygon and Ethereum

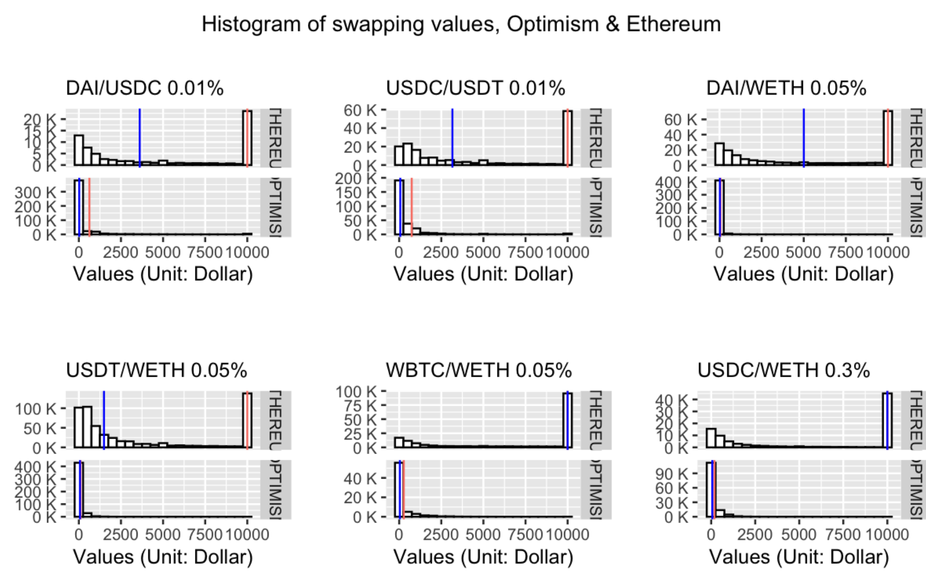


Figure A.10: Histogram of 6 other pairs from Optimism and Ethereum

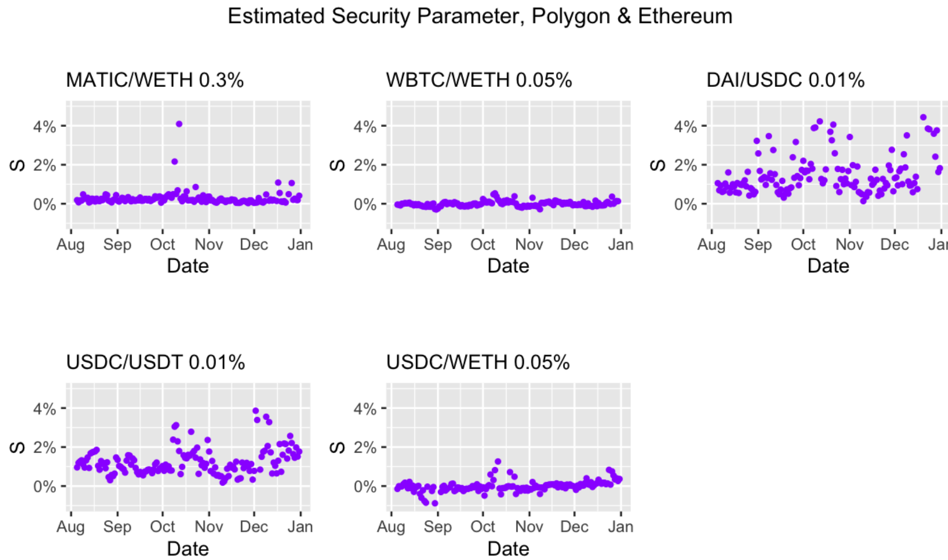


Figure A.11: Estimated Security parameter S for 5 other pairs from Polygon and Ethereum

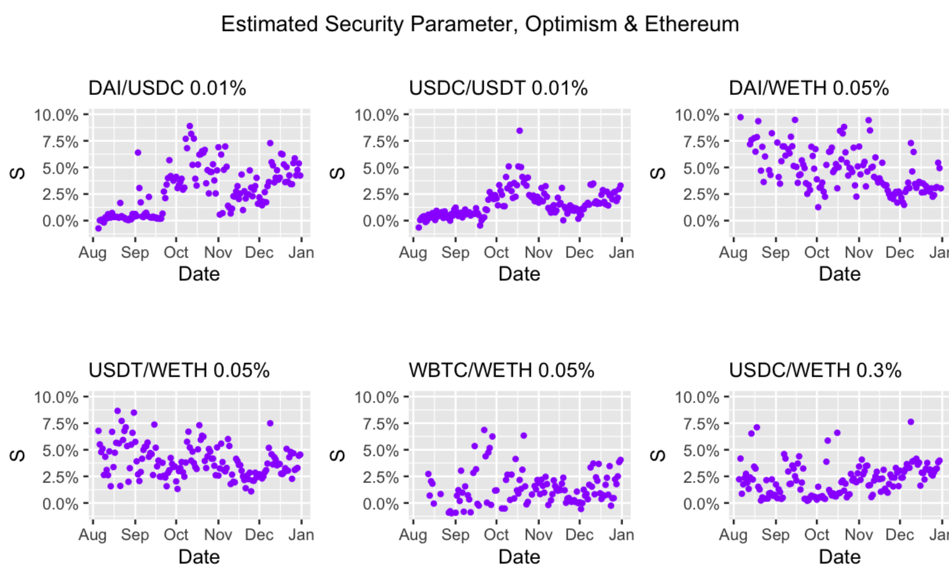


Figure A.12: Estimated Security parameter S for 6 other pairs from Optimism and Ethereum

A.6 Trade-washing

To address the potential concerns on trade-washing and price manipulation transactions, we conducted a robustness check. This involved analyzing a subset of wallets that, on average, conducted two or fewer transaction per day during the period of analysis. These wallets are less likely to be involved in acts of price manipulation.

Table A.1 provides an overview of the subset data. We conducted the same analysis as in Section 1.6.3 to determine the mean (median) estimated security parameter. The findings indicate that trade-washing and price manipulation transactions cannot explain the observed gap. If anything, this suggests that our representative agent result is only a lower bound of the security estimate parameter.

	DEX	# TXN (%)	Total Volume (%)	Mean Estimated Security Parameter (Median)
Full	ETH	2,789,976	$220,065,992,854	
	POLY	4,991,764	$12,401,731,851	0.751% (0.554%)
Subset	ETH	1,584,914 (56.8%)	$60,484,738,063 (27.5%)	
	POLY	1,255,020 (25.1%)	$ 2,401,452,530 (19.4%)	1.443% (1.136%)

Table A.1: Subset Data for users that average daily transaction ≤ 2

A.7 Adoption of the L2 networks

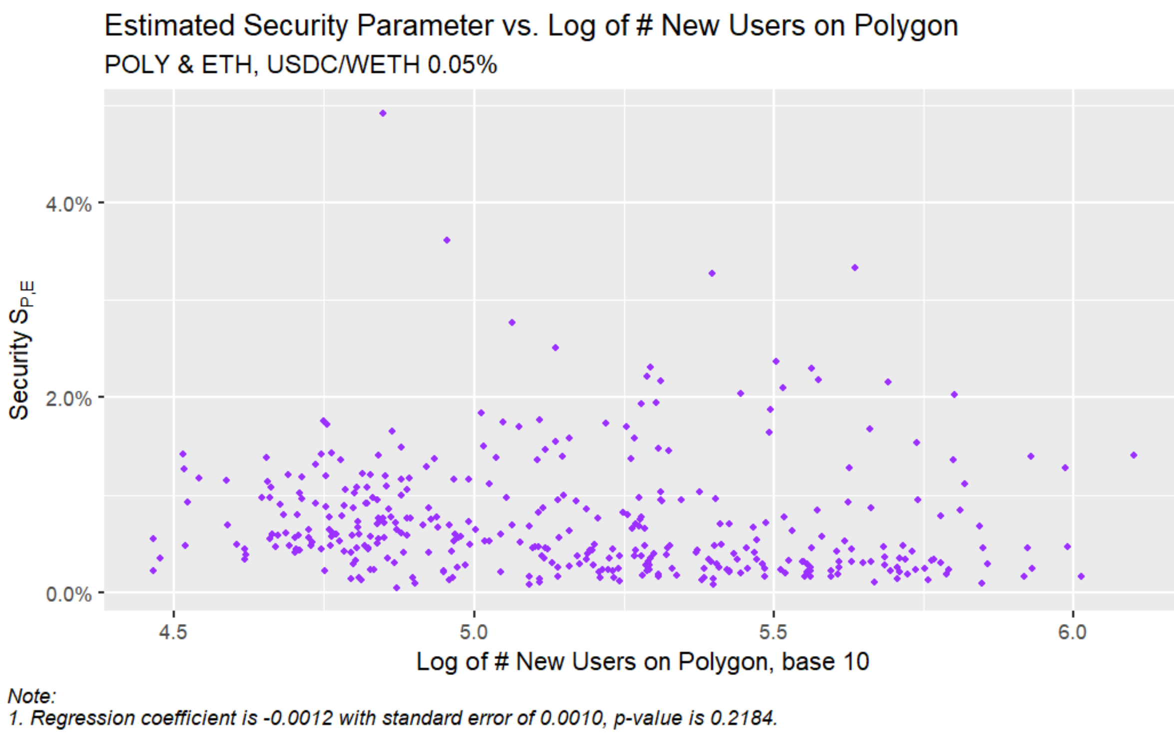


Figure A.13: Estimated Security Parameter vs. Log of # New Users on Polygon

Appendix B

Appendix for "The Power of Default: Measuring the Effect of Slippage Tolerance in Decentralized Exchanges"

B.1 Detection of Sandwich Attacks on Decentralized Exchanges

In this paper, we address the issue of sandwich attacks in liquidity pools by presenting a straightforward measurement approach to identify and detect such attacks using trading data from DEX platforms. Our approach builds upon previous research, including studies by Lehar and Parlour (2023); Züst (2021); Qin et al. (2022); Torres et al. (2021).

Specifically, we use the following rules for identifying attacks:

(A) At least two swaps T_{M1} and T_{M2} are included in the same block and swap assets in the same liquidity pool.

(B) T_{M1} and T_{M2} have different transaction hashes.

(C) T_{M1} and T_{M2} swap in different directions.

(D_1) T_{M1} and T_{M2} are initiated by the same wallet.

(D_2) Input amount of T_{M1} equals output amount of T_{M2}, or output amount of T_{M1} equals input amount of T_{M2}.

Additionally, we have the following rules for identifying successful attacks:

(E) At least one additional swap T_A is included in the same block that swapped assets in the same liquidity pool.

(F) T_A executes between T_{M1} and T_{M2}.

(G) T_A and T_{M1} swap in the same direction.

(H) If more than one swap executes between T_{M1} and T_{M2}, then T_{M1} should swap in the same direction as all additional swaps, T_A, T_B, etc.

Rule (A) considers only attacks that are fully executed within the same block. It should be noted that a sandwich attack can still be successful even if T_{M1} and T_{M2} are located in different blocks. However, attackers prefer to have T_{M1} and T_{M2} placed in the same block to minimize the risk of compromising their profits and allowing other users to benefit from them.

Some transactions perform two swaps on the same liquidity pool but in opposing directions. [1] We adopt rule (B) from Torres et al. (2021), which ensures that we do not identify such transactions as a sandwich attack.

The contributions of our method can be summarized into two primary aspects. Firstly, we expand the detection capabilities of sandwich attacks to encompass multi-meat attacks, wherein a single attack targets multiple victims simultaneously. Such attacks involve the inclusion of at least two victims (referred to as layers/meat slices in the sandwich) within a single sandwich attack. To illustrate, a sandwich attack involving three victims is commonly referred to as a multi-meat attack with three layers. Conversely, a sandwich attack involving only one layer is commonly referred to as a single-meat attack. Notably, we observed a total of more than 6,000 instances of multi-meat attacks (6.3% of the successful attacks) in Uniswap and Sushiswap in March (Table B.1).

Secondly, we introduce more flexible rules for analyzing attackers' decisions regarding "cash back," thereby facilitating enhanced detection of a wider range of attacks. The concept of "cash back" refers to the choice of tokens in which the attacker wishes to retain their profits (Token A or B). Previous studies have employed heuristics that restrict the attacker from retaining their profit exclusively in Token B or Token A.

For example, in Figure B.3, the malicious attacker decides to withdraw 0.2 Tokens B as

[1] Rule (B) removes the noise of misidentified attacks due to the use of aggregator apps which collect several users' swaps into the same transaction. For example, Tokenlon (2022) does such an aggregation. For these apps, it is possible for our rules erroneously classify transactions as attacks when two users intend to swap in opposite directions using the same pool.

Meat Layer	Attacks	Percentage
1	102,248	93.7%
2	5,116	4.7%
3	1,100	1.0%
≥ 4	156	0.6%

Table B.1: Layers of Sandwich Attack.

their profit. However, in real-world scenarios, attackers may have liquidity constraints or specific preferences for other tokens, which can influence their decision-making process. To address this, we have introduced more flexible rules for the "cash back" decision, allowing attackers to choose any combination of tokens.

In previous studies, only (D_2), or modified versions of (D_2), are used. In our study, (D_1) and (D_2) do not need to be satisfied at the same time: only one of the two needs to be satisfied. By adding (D_1), we are able to address the "cash back" issue as mentioned above.

Out of the total 109,120 successful attacks we detected, most of the attacks satisfied (D_1) and (D_2) at the same time. However, by adding Rule (D_1), we detect more attacks than by just using Rule (D_2) alone (see Table B.2).

Satisfy D_1	Satisfy D_2	Number of successful attacks	Percentage of successful attacks
Yes	Yes	72,624	66.6%
Yes	No	33,989	31.1%
No	Yes	2,507	2.3%

Table B.2: Number of attacks satisfy Rule D_1 and D_2.

Some other examples of common heuristics used in other works include:

(E)' Exactly one swap T_A is executed between T_{M1} and T_{M2}.

Block Size	Number of successful attacks
3 (Rule I)	78,885
4	18,552
≥ 5	11,683

Table B.3: Expansion from common heuristics.

(I) Consider blocks that contain exactly three transactions.

(J) T_{M1} is the first executed transaction for the traded liquidity pool in a block.

Rule (I) will omit almost 30k successful attacks (see Table B.3).

B.2 Swap Data

Uniswap	Number of Swaps	Total value traded ($)	Number of Unique Pools
All data (March 1st to March 31st)	4,612,958 (100%)		17,687 (100%)
Filtered data (Pools with top 1000 tokens)	2,253,712 (48.9%)		2153 (12.2%)
Filtered data (Pools with top 1000 tokens) (Also appear everyday)	2,096,498 (45.4%)	5.32×10^{10}	721 (4.1%)

Sushiswap	Number of Swaps	Total value traded ($)	Number of Unique Pools
All data (March 1st to March 31st)	219,412 (100%)		1,190 (100%)
Filtered data (Pools with top 1000 tokens)	184,686 (84.2%)		401 (33.7%)
Filtered data (Pools with top 1000 tokens) (Also appear everyday)	167,358 (76.3%)	6.64×10^{8}	103 (8.7%)

Table B.4: Swap Data Overview.

B.3 Additional plots

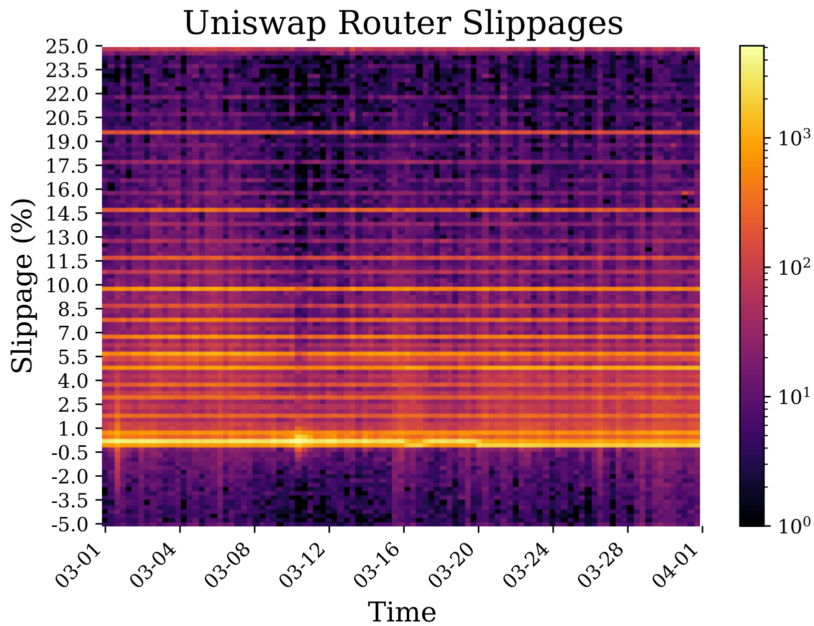


Figure B.1: Inferred router slippages over time, larger range. Color is drawn on a log scale to show detail. Each colored cell counts the number of swaps in a slippage range at a segment of time.

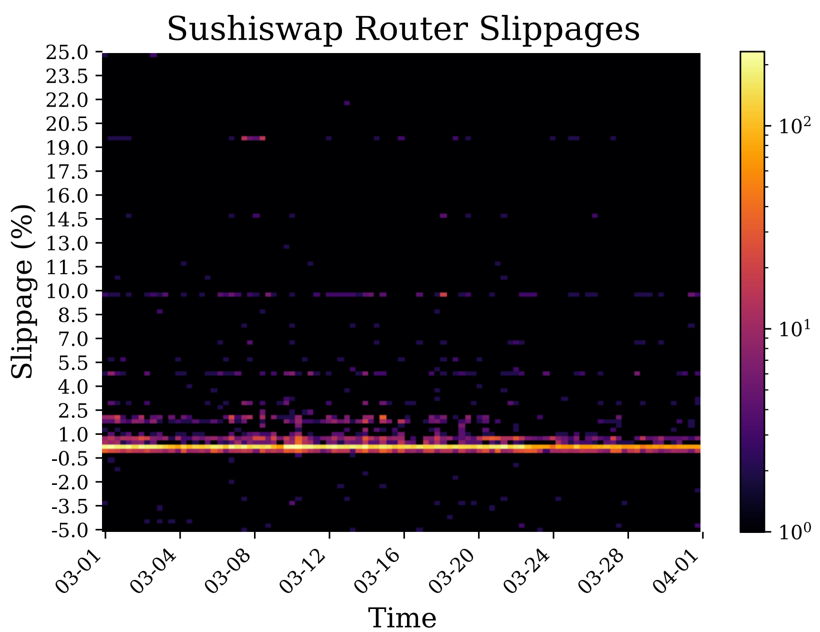
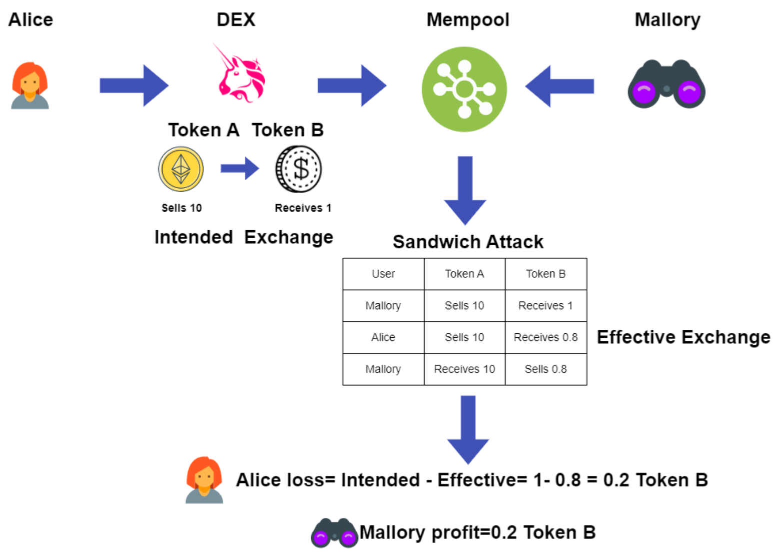


Figure B.2: Inferred router slippages over time, larger range. Color is drawn on a log scale to show detail. Each colored cell counts the number of swaps in a slippage range at a segment of time.

B.4 Sandwich attack

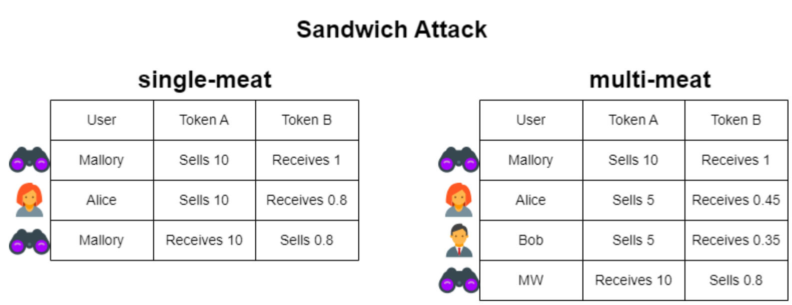
B.4.1 Sandwich attack



User	Token A	Token B
Mallory	Sells 10	Receives 1
Alice	Sells 10	Receives 0.8
Mallory	Receives 10	Sells 0.8

Figure B.3: Sandwich attack example: the transactions introduced by Mallory before and after Alice's transaction result in a worse exchange for Alice and a profit for Mallory.

Depicting a scenario where Mallory observes transactions of both Alice and Bob in the mempool, thereby enabling the execution of a multi-meat attack as shown in Figure B.4. Such attacks involve the inclusion of two or more victims (referred to as layers/meat slices in the sandwich) within a single sandwich attack.



User	Token A	Token B
Mallory	Sells 10	Receives 1
Alice	Sells 10	Receives 0.8
Mallory	Receives 10	Sells 0.8

User	Token A	Token B
Mallory	Sells 10	Receives 1
Alice	Sells 5	Receives 0.45
Bob	Sells 5	Receives 0.35
MW	Receives 10	Sells 0.8

Figure B.4: Single versus multi-meat attacks.

Appendix C

Appendix for "Motivating Academic Success: The Role of Leaderboards in Shaping Student Study Behaviors"

C.1 Excel Assignment download site

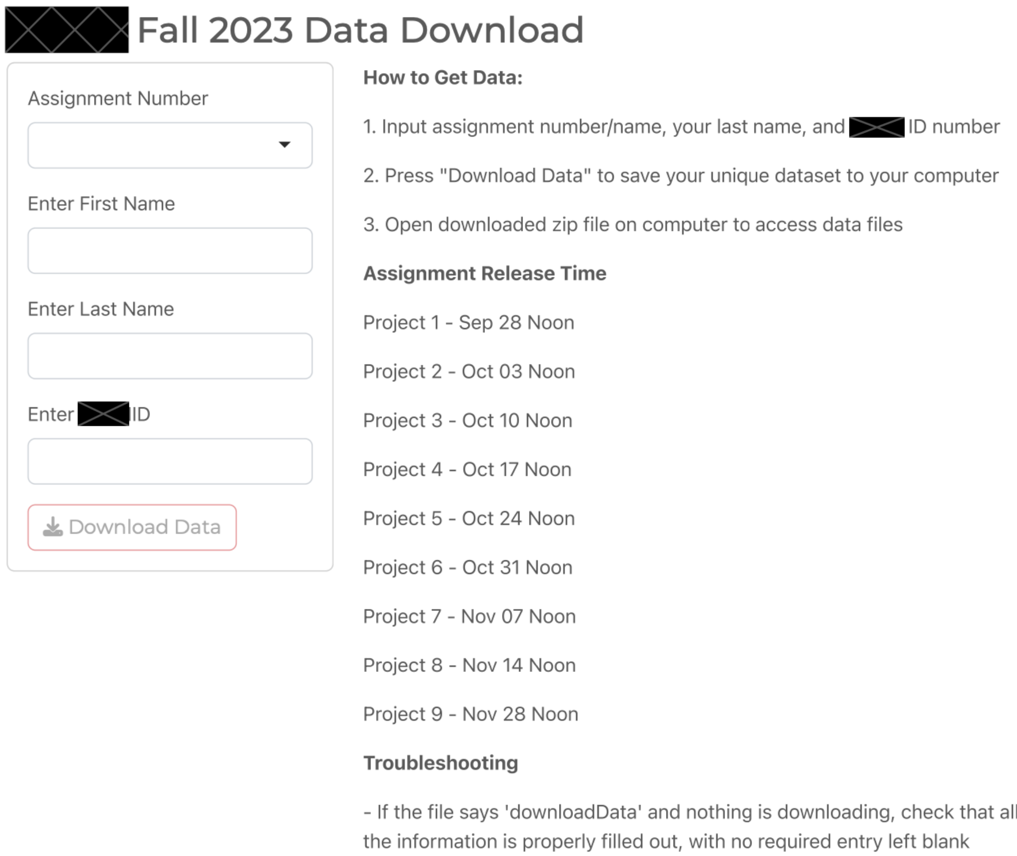


Figure C.1: Screenshot of the assignment download site

C.2 Excel Assignment extra infomation for treatment group

- When uploading the Excel spreadsheet to [the autograding platform], you will be prompted to provide a name for the leaderboard. A leaderboard is a tool that allows you to monitor accuracy and timing of your submission relative to others in the class.

- You can pick your own leaderboard name, please make sure it is appropriate and respectful, given that the leaderboard is visible to other students in the class.

- DO NOT use your real name as your leaderboard name. If you accidentally use your real name as your leaderboard name, submit again, the original leaderboard entry will be deleted.

- The leaderboard will NOT affect your course grade at all. If you don't want your leaderboard name to be uniquely identified, you can type Optout Otter as your leaderboard name.

- DO NOT leave the leaderboard name blank, your submission will not be processed.

- DO NOT share your leaderboard name with other students. A student's leaderboard name is known only to that particular student and course instructors.

- Leaderboard ranks all submissions by the SUBMISSION TIME among the class members. The earlier the submission, the higher the rank. Note though, the leaderboard ONLY ranks those who got FULL SCORE on the autograder part of the assignment. For example, if you and another student both got a full score, and you submitted earlier, your leaderboard name will appear higher on the leaderboard than the other student's. Please notice that your leaderboard name would appear at the bottom of the leaderboard if your submission scored less than FULL SCORE on the autograder part of the assignment.

- After your submission, click on "Leaderboard" on the "Autograder Results page" to check the ranking.

Bibliography

Abowd, J. M. and Schmutte, I. M. (2019). An economic analysis of privacy protection and statistical accuracy as social choices. *American Economic Review*, 109(1):171–202.

Adams, H., Zinsmeister, N., and Robinson, D. (2020). Uniswap v2 core.

Adams, H., Zinsmeister, N., Salem, M., Keefer, R., and Robinson, D. (2021). Uniswap v3 core. Technical report, Technical report.

Ai, W., Chen, Y., Mei, Q., Ye, J., and Zhang, L. (2023). Putting teams into the gig economy: A field experiment at a ride-sharing platform. *Management Science*, 69(9):5336–5353.

Amiram, D., Lyandres, E., and Rabetti, D. (2021). Competition and product quality: fake trading on crypto exchanges. *Available at SSRN 3745617*.

Barbon, A. and Ranaldo, A. (2021). On the quality of cryptocurrency markets: Centralized versus decentralized exchanges. *arXiv preprint arXiv:2112.07386*.

Bastos, M. L., Perlman-Arrow, S., Menzies, D., and Campbell, J. R. (2021). The Sensitivity and Costs of Testing for SARS-CoV-2 Infection With Saliva Versus Nasopharyngeal Swabs. *Annals of Internal Medicine*, 174(4):M20–6569.

BIS (2022). "the future monetary system". *Annual Economic Report 2022*.

Bradler, C., Dur, R., Neckermann, S., and Non, A. (2016). Employee recognition and performance: A field experiment. *Management Science*, 62(11):3085–3099.

Buckley, P. and Doyle, E. (2014). Gamification and student motivation. *Interactive Learning Environments*, 24(6):1162–1175.

Bursztyn, L., Egorov, G., and Jensen, R. (2019). Cool to be smart or smart to be cool? understanding peer pressure in education. *The Review of Economic Studies*, 86(4):1487–1526.

Bursztyn, L. and Jensen, R. (2015). How does peer pressure affect educational investments? *The quarterly journal of economics*, 130(3):1329–1367.

Caldarelli, G. (2021). Wrapping trust for interoperability: A preliminary study of wrapped tokens. *Information*, 13(1):6.

Caparros, B., Chaudhary, A., and Klein, O. (2023). Blockchain scaling and liquidity concentration on decentralized exchanges. *arXiv preprint arXiv:2306.17742*.

Capponi, A. and Jia, R. (2021). The adoption of blockchain-based decentralized exchanges. *arXiv preprint arXiv:2103.08842*.

Cecchetti, S. G., Lam, P.-s., and Mark, N. C. (2000). Asset pricing with distorted beliefs: are equity returns too good to be true? *American Economic Review*, 90(4):787–805.

Cernera, F., Morgia, M. L., Mei, A., and Sassi, F. (2023). Token spammers, rug pulls, and sniper bots: An analysis of the ecosystem of tokens in ethereum and in the binance smart chain (BNB). In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3349–3366, Anaheim, CA. USENIX Association.

Chang, J.-W. and Wei, H.-Y. (2016). Exploring engaging gamification mechanics in massive online open courses. *Journal of Educational Technology Society*, 19(2):177–203.

Chemaya, N. and Liu, D. (2023). Estimating investor preferences for blockchain security. In *Estimating Investor Preferences for Blockchain Security: Chemaya, Nir— uLiu, Dingyue*. [Sl]: SSRN.

Chemaya, N. and Liu, D. (2024). The suitability of using uniswap v2 model to analyze v3 data. *Finance Research Letters*, 59:104717.

Chen, R., Chen, Y., Liu, Y., and Mei, Q. (2017). Does team competition increase pro-social lending? evidence from online microfinance. *Games and Economic Behavior*, 101:311–333.

Coinbase (2023). Coinbase. https://www.coinbase.com/legal/trading_rules.

Cong, L. W., Hui, X., Tucker, C., and Zhou, L. (2023a). Scaling smart contracts via layer-2 technologies: Some empirical evidence. Technical report, National Bureau of Economic Research.

Cong, L. W., Li, X., Tang, K., and Yang, Y. (2023b). Crypto wash trading. *Management Science*.

Cullen, A. C., Frey, H. C., and Frey, C. H. (1999). *Probabilistic techniques in exposure assessment: a handbook for dealing with variability and uncertainty in models and inputs*. Springer Science & Business Media.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. (2020). Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge. ISBN 0-521-57391-2.

defillama (2023). Dexes tvl rankings. https://defillama.com/protocols/dexes/Ethereum.

Dewitte, S. and Schouwenburg, H. C. (2002). Procrastination, temptations, and incentives: The struggle between the present and the future in procrastinators and the punctual. *European Journal of Personality*, 16(6):469–489.

Dichev, C. and Dicheva, D. (2017). Gamifying education: What is known, what is believed and what remains uncertain: A critical review. *International Journal of Educational Technology in Higher Education*, 14(1).

Easley, D., O'Hara, M., and Basu, S. (2019). From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics*, 134(1):91–109.

Efron, B. (1975). The efficiency of logistic regression compared to normal discriminant analysis. *Journal of the American Statistical Association*, 70(352):892–898.

Evans, C. J. and Crawford, B. (2000). Data Collection Methods in Prospective Economic Evaluations: How Accurate Are the Results? *Value in Health*, 3(4):277–286.

Flashbots (2023). Flashbots protect. https://docs.flashbots.net/flashbots-protect/overview.

Foundation, E. (2023). go-ethereum. https://geth.ethereum.org/.

Garratt, R. J. and van Oordt, M. R. (2023). Why fixed costs matter for proof-of-work–based cryptocurrencies. *Management Science*, 69(11):6482–6507.

Gill, D., Kissová, Z., Lee, J., and Prowse, V. (2019). First-place loving and last-place loathing: How rank in the distribution of performance affects effort provision. *Management Science*, 65(2):494–507.

Ginther, D. K., Currie, J. M., Blau, F. D., and Croson, R. T. A. (2020). Can mentoring help female assistant professors in economics? an evaluation by randomized trial. *AEA Papers and Proceedings*, 110:205–09.

Ginther, D. K. and Na, R. (2021). Does mentoring increase the collaboration networks of female economists? an evaluation of the cement randomized trial. *AEA Papers and Proceedings*, 111:80–85.

Heimbach, L. and Wattenhofer, R. (2022). Eliminating sandwich attacks with the help of game theory. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 153–167.

Hen, M. and Goroshit, M. (2018). The effects of decisional and academic procrastination on students' feelings toward academic procrastination. *Current Psychology*, 39(2):556–563.

Hudja, S., Roberson, B., and Rosokha, Y. (2022). Public leaderboard feedback in sampling competition: An experimental investigation. *The Review of Economics and Statistics*, page 1–45.

Hummel, S., Hudak, A., Uebler, E., Falkowski, M., and Megown, K. (2011). A Comparison of Accuracy and Cost of LiDAR versus Stand Exam Data for Landscape Management on the Malheur National Forest. *Journal of Forestry*, 109(5):267–273.

Kosfeld, M. and Neckermann, S. (2011). Getting more work for nothing? symbolic awards and worker performance. *American Economic Journal: Microeconomics*, 3(3):86–99.

Lehar, A., Parlour, C., and Zoican, M. (2023). Liquidity fragmentation on decentralized exchanges. *arXiv preprint arXiv:2307.13772*.

Lehar, A. and Parlour, C. A. (2021). Decentralized exchanges. Technical report, working paper, University of Calgary and University of California, Berkeley.

Lehar, A. and Parlour, C. A. (2023). Battle of the bots: Flash loans, miner extractable value and efficient settlement. *Miner Extractable Value and Efficient Settlement (March 8, 2023)*.

Leifeld, P. (2013). texreg: Conversion of statistical model output in R to LaTeX and HTML tables. *Journal of Statistical Software*, 55(8):1–24.

Liu, X., Chen, W., and Zhu, K. (2022). Token incentives and platform competition: A tale of two swaps. *Available at SSRN 4176638*.

Lo, Y. and Medda, F. (2022). Uniswap and the rise of the decentralized exchange, in: Journal of financial market infrastructures, 2021, 10 (2): 1-25.

Makarov, I. and Schoar, A. (2022). Cryptocurrencies and decentralized finance (defi). Technical report, National Bureau of Economic Research.

Makridis, C. A., Fröwis, M., Sridhar, K., and Böhme, R. (2023). The rise of decentralized cryptocurrency exchanges: Evaluating the role of airdrops and governance tokens. *Journal of Corporate Finance*, 79:102358.

Malinova, K. and Park, A. (2023). Learning from defi: Would automated market makers improve equity trading?

Markopoulos, A. P., Fragkou, A., Kasidiaris, P. D., and Davim, J. P. (2015). Gamification in engineering education and professional training. *International Journal of Mechanical Engineering Education*, 43(2):118–131.

Metamask (2023). Metamask. https://metamask.io/.

Muradian, M. L., Branch, T. A., and Punt, A. E. (2019). A framework for assessing which sampling programmes provide the best trade-off between accuracy and cost of data in stock assessments. *ICES Journal of Marine Science*, 76(7):2102–2113.

Park, A. (2021). The conceptual flaws of constant product automated market making. *Available at SSRN 3805750*.

Park, A. (2023). The conceptual flaws of decentralized automated market making. *Management Science*.

Pedersen, T. L. (2021). *ggforce: Accelerating 'ggplot2'*. R package version 0.3.3.

Qin, K., Zhou, L., and Gervais, A. (2022). Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214. IEEE.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sayeed, S. and Marco-Gisbert, H. (2019). Assessing blockchain consensus and security mechanisms against the 51% attack. *Applied Sciences*, 9(9):1788.

Schär, F. (2021). Decentralized finance: On blockchain-and smart contract-based financial markets. *FRB of St. Louis Review*.

Schiming, R. C. (2012). Patterns of Homework Initiation for Web-based Activities in Economics: A Study of Academic Procrastination. *Journal for Economic Educators*, 12(1):13–25.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). Rocr: visualizing classifier performance in r. *Bioinformatics*, 21(20):7881.

Sokolov, K. (2021). Ransomware activity and blockchain congestion. *Journal of Financial Economics*, 141(2):771–782.

Subhash, S. and Cudney, E. A. (2018). Gamified learning in higher education: A systematic review of the literature. *Computers in Human Behavior*, 87:192–206.

Sushiswap (2021). Sushiswap. https://www.sushi.com/.

Thaler, R. H. and Benartzi, S. (2004). Save more tomorrow™: Using behavioral economics to increase employee saving. *Journal of political Economy*, 112(S1):S164–S187.

Thaler, R. H. and Sunstein, C. R. (2008). Nudge: improving decisions about health. *Wealth, and Happiness*, 6:14–38.

Thibault, L. T., Sarry, T., and Hafid, A. S. (2022). Blockchain scaling using rollups: A comprehensive survey. *IEEE Access*.

Tokenlon (2022). Tokenlon library. https://tokenlon.gitbook.io/docs/v/docs.en/.

Torres, C. F., Camino, R., and State, R. (2021). Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1343–1359. USENIX Association.

Tsankov, P., Dan, A., Drachsler-Cohen, D., Gervais, A., Buenzli, F., and Vechev, M. (2018). Securify: Practical security analysis of smart contracts. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 67–82.

Uniswap (2021). Uniswap. https://uniswap.org.

Uniswap (2023a). Uniswap interface. https://github.com/Uniswap/interface/blob/main/src/hooks/useAutoSlippageTolerance.ts#L75.

Uniswap (2023b). Uniswap interface - dynamic slippage. https://github.com/Uniswap/interface/commit/0923cf4ac977f321b4cf11295f7d95b01beae91a.

Venables, W. and Ripley, B. (2002). Random and mixed effects. In *Modern applied statistics with S*, pages 271–300. Springer.

Victor, F. and Weintraud, A. M. (2021). Detecting and quantifying wash trading on decentralized cryptocurrency exchanges. In *Proceedings of the Web Conference 2021*, pages 23–32.

Viscusi, W. K. and Aldy, J. E. (2003). The value of a statistical life: a critical review of market estimates throughout the world. *Journal of risk and uncertainty*, 27:5–76.

Wang, T., Zhao, C., Yang, Q., Zhang, S., and Liew, S. C. (2021). Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain. *IEEE Transactions on Network Science and Engineering*, 8(3):2131–2146.

Wolfers, J. and Zitzewitz, E. (2006). Interpreting prediction market prices as probabilities.

Xia, P., Wang, H., Gao, B., Su, W., Yu, Z., Luo, X., Zhang, C., Xiao, X., and Xu, G. (2021). Trade or trick? detecting and characterizing scam tokens on uniswap decentralized exchange. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(3):1–26.

Xu, J., Paruch, K., Cousaert, S., and Feng, Y. (2023). SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) Protocols. *arXiv preprint arXiv:2103.12732*.

Yaish, A., Dotan, M., Qin, K., Zohar, A., and Gervais, A. (2023). Suboptimality in defi. *Cryptology ePrint Archive*.

Zhou, L., Qin, K., Torres, C. F., Le, D. V., and Gervais, A. (2021). High-frequency trading on decentralized on-chain exchanges. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 428–445. IEEE.

Züst, P. (2021). Analyzing and preventing sandwich attacks in ethereum.