THE XFAI V0 AUTOMATED MARKET MAKER MODEL WITH ENTANGLED LIQUIDITY POOLS

info@xfai.com

May 5, 2023

ABSTRACT

While automated market makers have found widespread adoption in decentralized finance, slippage still remains one of the major hurdles in the space. We show that trade performances can drastically improve, by using a weighted pool model. Unlike token-pair-based constant product market makers, the Xfai model does not suffer from fragmented liquidity. We show that for the majority of token trades, the Xfai model outperforms many of the other automated market maker models.

1 Introduction

Xfai is a new type of decentralized exchange (DEX), that is based on a system of on-chain constant function market maker (CFMM) smart contracts. It supports instant trades, as well as two forms of liquidity provisioning for ERC20-compatible tokens and blockchain-native coins (e.g. ETH in the case of Ethereum). With the Xfai CFMM model design, the Xfai team aims to introduce a new standard for on-chain CFMM models. Due to the model's better slippage performance through its liquidity concentration, we expect other AMMs to adopt the weighted model design of Xfai into future versions. Unlike centralized exchanges (CEXs) where funds are controlled by a trusted party, DEXs make use of the self-executing nature of smart contracts to perform trades and provide liquidity in a non-custodial and censorship-resistant way. The users have full control of their funds during every step when interacting with a DEX and no other third party is able to interfere.

The purpose of the next subsections is to become familiar with the key concepts required to understand Xfai's CFMM model. Section 2 explains how trades on the Xfai CFMM work. Section 3 describes how liquidity is provided in the XFai CFMM. Section 4 describes the mechanism for liquidity redemption. Section 5 introduces the INFT cryptoeconomic primitive used within the Xfai CFMM. Section 7 lays down the architectural organization of the Xfai CFMM. Section 6 briefly covers the concept of flash loans. In the experiments section (8) we explore how the Xfai model performs against Uniswap V2 and Pancakeswap. Finally, section 11 covers our concluding thoughts on the Xfai model.

2 Entangled Swaps

The use of DEXs, and more concretely the use of CFMMs, has grown dramatically during the last years Angeris et al. [2021], with many new automated market maker (AMM) model designs popping-up every year Zinsmeister et al. [2020], Adams et al. [2021], Martinelli and Mushegian [2019], Egorov [2019], Hertzog et al. [2017], Finance [2020]. CFMMs are a type of AMM that were originally designed as an alternative to order-book-based exchanges for the blockchain space. Unlike order-books, trades on CFMMs do not have to wait for buy and sell orders and are usually instant in nature. Instead of trading with a specific buyer or seller, traders in CFMMs interact directly with a smart contract that has access to the required liquidity. The liquidity of the smart contracts is provided by liquidity providers (LPs). In exchange for providing liquidity, LPs receive fees for trades performed within the CFMM. This approach allows LPs to make a return on their assets passively, without having to actively manage a position.

The exchange value for a trade (aka swap) is determined using a deterministic "exchange function". In the case of Xfai's CFMM implementation, a constant product market maker (CPMM) model is used:

$$(r_i + \gamma\Delta_i)(w_i - \Delta_w)(r_j - \Delta_j)(w_j + \Delta_w) = k \tag{1}$$

Where r_i and r_j are the smart contract's reserves of token i and token j within pool i and pool j, w_i and w_j are the exchange value weights of pool i and pool j, Δ_i is the amount of tokens i that are sent to pool i, Δ_j is the amount of tokens j that get removed from pool j and sent to a recipient, and Δ_w is the amount of weights that need to be added to w_j and subtracted from w_i.

In Xfai, as in any CPMM model, the trade between a pair of assets has to happen in a way that keeps the product of the two asset reserves unchanged after the trade. In other words, the constant k has to remain equal before and after a swap. In practice, because γ is usually set to a none-zero value, each trade slightly increases k. If we assume that a trade has no fees, we can rewrite equation 1 as:

$$(r_i + \Delta_i)(w_i - \Delta_w)(r_j - \Delta_j)(w_j + \Delta_w) = r_i w_i r_j w_j \tag{2}$$

To know therefore how many tokens j we receives for inserting a given amount of i tokens, one would rewrite equation 2 as:

$$\Delta_j = \left(\frac{\frac{\Delta_i w_i}{r_i + \Delta_i} r_j}{w_j + \frac{\Delta_i w_i}{r_i + \Delta_i}} \right) \tag{3}$$

It is worth noting that the Xfai CFMM model design does not use ERC20 token pairs, that is, it does not rely on a sparse matrix of exchange values, but instead uses unique ERC20 token pools. In other words, it relies on a vector of exchange values. Xfai uses an ETH weighted pool system instead. Each pool is made of an ERC20 token reserve r and a dynamic weight w to determine the exchange values of the token. One key advantage of Xfai's weighted pool model, is that i removes the need for liquidity fragmentation. This allows the Xfai CPMM model to form *deep liquidity pools* that can perform swaps with an overall lower slippage than in token-pair-based CFMM models.

Figure 1: A depiction of the Xfai swap process.

3 Pool Creation and Liquidity Provisioning

Unlike Bancor V3 Richardson and Hindman [2021], the Xfai CPMM model does not require a whitelisting procedure for the on-boarding of new pools. As in Uniswap's models, anyone on Xfai can create new pools in a permissionless way, with deterministic addresses that can be computed offline.

Xfai uses a two-sided liquidity provisioning approach to provide liquidity to a pool. As an example, let there be a pool p_i. To mint a given amount of liquidity tokens for p_i, we send tokens i to p_i as well as ether (Ethereum's native coin). After the states are updated, a given amount of liquidity tokens are minted and sent to a recipient:

$$l_i = min\left(\frac{\Delta_i T_i}{r_i}, \frac{\Delta_w T_i}{w_i} \right) \tag{4}$$

Where l_i represents the amount of liquidity tokens minted, r_i represents the reserve of tokens i within the pool, w_i represents the ETH denominated weights of the pool, Δ_i represents he amount of tokens i inserted into p_i, Δ_w represents the amount of ether provided to p_i, and T_i represents the total amount of already minted liquidity tokens for p_i. Notice that formula 4 for minting liquidity tokens is the same one as used by Uniswap V2 Zinsmeister et al. [2020].

4 LP Fees and Liquidity Redemption

Liquidity providers receive fees whenever a trade occurs within the pool of their liquidity token. Swap fees occur at the input pool, that is, if someone initiates a swap from p_i to p_j, the LP fee is applied at p_i. Besides swap fees, the same fee is applied to burning liquidity tokens as well. The burn fee exists to prevent external contracts from using Xfai's liquidity for fee-less swaps (by adding and removing liquidity across pools within the same transaction). In the case of trades, fees are applied to the input pool (i.e. the primary pool). In the case of liquidity redemption, when the liquidity tokens of a pool p_i are burned to redeem the underlying liquidity and any accrued fees, the fee is applied to the secondary pool (p_j).

Liquidity redemption on Xfai requires selecting a primary pool and a secondary pool. By burning one's primary pool liquidity tokens, one can redeem a given amount Δ_i and Δ_j:

$$\Delta_i = \frac{l_i r_i}{T_i} \qquad \Delta_w = \frac{l_i w_i}{T_i} \qquad \Delta_j = \frac{\Delta_w r_j}{w_j + \Delta_w} \tag{5}$$

Where l_i represents the amount of liquidity token that get burned from the selected primary pool. The weights of the primary and secondary pool get updated after burning liquidity tokens:

$$w_i = w_i - \Delta_w \qquad w_j = w_j + \Delta_w \tag{6}$$

To minimize therefore one's divergence loss (also known as impermanent loss) Loesch et al. [2021], it is in one's own interest to add and redeem liquidity from correlated pools, i.e. pools whose exchange values tend to move together. If a user chooses to exit in ETH, no redeeming fee is applied.

5 INFTs and INFT fees

The Xfai CPMM model makes use of a novel cryptoeconomic primitive referred to as Infinity Non-Fungible Tokens (INFTs). INFTs enable an additional way to passively earn fees on Xfai, by permanently locking XFIT, Xfai's token. In the traditional liquidity provisioning approach in Xfai (section 3), a user provides two-sided liquidity in exchange for a liquidity token. Whenever a swap (or a redemption) occurs, liquidity providers receive a fee. By burning their liquidity token, they are able to get their provided liquidity back, plus the accumulated fees. This process is however also susceptible to impermanent loss. In the case of Infinity Staking, liquidity providers are able to lock their liquidity permanently into the DEX in exchange for an INFT. An INFT has a share, the value of which gets determined by the amount of underlying tokens locked:

$$s(\Delta_i) = \frac{u\Delta_j}{r_{INFT} + \Delta_j} \tag{7}$$

Where $s(\Delta_i)$ represents the share value of an INFT for a given amount of Δ_i tokens, u represents a system-wide constant, Δ_j represents the amount of underlying tokens j (In Xfai's case the XFIT token) that one permanently locks for a given amount of Δ_i ERC20 tokens, and r_{INFT} represents the contract's reserve of locked XFIT tokens. Unlike liquidity token holders that collect fees for only a given pool within the DEX, INFT token holders collect fees from every pool and future pool within the DEX.

If we look at equation 7, we can see that the amount of shares that one can receive for a given amount of staked tokens decreases non-linearly, the more underlying tokens have already been staked. This design choice incentivizes users to stake earlier. Once an INFT holder harvests their fees from a given pool, their share within that pool decreases, and the shares of the other INFT holders increases. Because of the non-linear function used to issue shares, and the share reducing mechanism used when harvesting fees from a pool, we can imagine fee allocation to behave very much like a queue. earlier INFT holders get on average more fees. Once they "exit", i.e. harvest from a given pool, the share value of the other INFT holders for that pool increases. Harvesting fees from one pool does not influence an INFT holder's share within another pool.

6 Flash Loans

A flash loan is a cryptoeconomic primitive usually used by arbitrageurs in decentralized finance (DeFi) applications. In conventional loans, lenders require up front collateral before giving money to a borrower. Flash loans are different

in that regard - Flash loans allow borrowers to take out as much liquidity as desired out of a DeFi service, without requiring any collateral at all. The borrower has to return however the borrowed liquidity (as well as a slight interest on top of it) at the end of the same transaction back. The Xfai CPMM model allows for flash loan requests from every pool within the DEX. Since Xfai has token pools instead of token pairs, the liquidity that arbitragers can borrow can be significantly higher than in other token-pair-based DEXs.

7 DEX Architecture

The Xfai CPMM model consists of several smart contracts (see Figure 2):

1. **Pools** - Every hosted ERC20 token on Xfai has its own pool contract. A pool contract keeps track of state, i.e. it keeps track of its reserve, weights, and liquidity token. A pool's reserves and weights can only be modified by Xfai's Core contract.
2. **Core** - The core contract is where the AMM logic resides. It contains the functions to perform swaps, liquidity provisioning, and liquidity redemption. The core contract is responsible to change the states of pools e.g. when a swap is initialized.
3. **Factory** - The Factory contract's role is to create new Xfai Pools and to inform Xfai pools on the current Core address. If the pools are contacted by a contract, that according to the Factory is not the current Core contract, then access to state modifying functions is restricted. This decoupling of state and execution logic allows Xfai to easily swap Core contracts with upgraded versions, without having to migrate liquidity to new pools.
4. **Periphery** - When executing functions such as swap, mint, and burn, it is not advised to interact with the core contract directly, as it does not perform important safety checks. Most interactions with the Xfai CPMM go through the periphery contract, which then contacts the core contract. Developers interacting with Xfai should always try to interact with the periphery contract.
5. **INFT** - INFT is the Infinity NFT contract responsible for INFT issuance, INFT fee collection, staking, harvesting, etc.

The **Dapp** box in figure 2 simply represents the Web3 interface of Xfai and does not represent a smart contract.

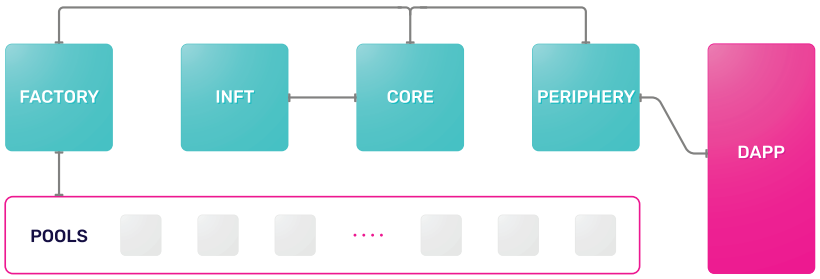

Figure 2: A depiction of the Xfai smart contract architecture

8 Experiments

The Xfai CPMM model's trade performance is compared to to that of Pancakeswap and Uniswap V2. The purpose of this experiment is to see how the Xfai CPMM model's slippage compares to that of other DEXs, given the same amount of liquidity and trade volume. All data-sets and simulations for these experiments are publicly accessible via Xfai's Github repository.

9 Setup

Two data-sets with the top 100 tokens with the highest market cap (as of November, 2022) for the Ethereum and Binance chain were first created. In the Ethereum data-set, the tokens that had no token pair with ETH on Uniswap V2, were dropped and substituted with the next highest market cap tokens that had a token pair with ETH. In the Binance data-set, the tokens that had no token pair with BNB on Pancakeswap, were dropped and substituted with the next highest market cap tokens that had a token pair with BNB. In both cases, more than 90% of the top market cap tokens had token pairs with ETH or BNB. The reasons for this filtration step will become apparent in section 10.

After the top 100 tokens with the highest market cap for both Ethereum and Binance were identified and stored, two additional data-sets were created: For Ethereum, every existing token pair (as well as their reserves) for the top 100 tokens on Uniswap V2 were saved. For Binance, every existing token pair (as well as their reserves) for the top 100 tokens on Pancakeswap were saved. Figure 3 shows the token pair distribution for the top 100 tokens on Ethereum and Binance, for Uniswap V2 and Pancakeswap respectively. Every pink dot is a token pair. As we can see, these matrices are very sparse. This means that most combinations for token trades inevitably experience double slippage during swaps.

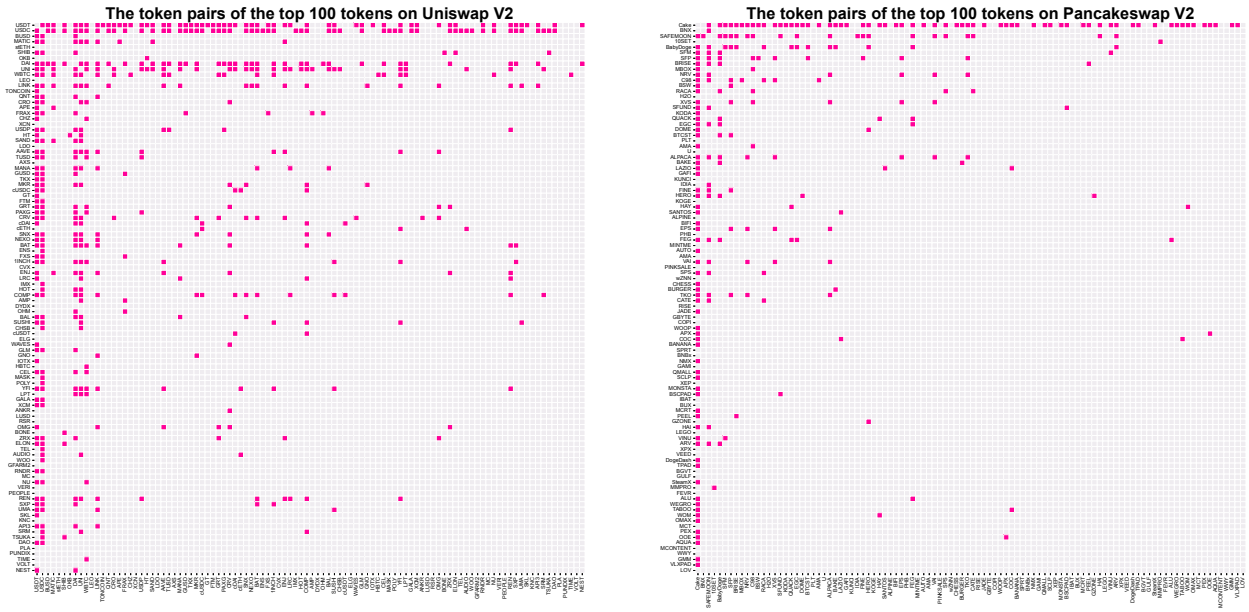

Figure 3: Token pair distribution for the top 100 tokens on Uniswap V2 (on the Ethereum chain) and Pancakeswap (on the Binance chain).

10 Simulations

Our goal was to design the experiments for the simulations as fairly as possible. Both Uniswap and Pancakeswap have token pairs, i.e. every token pair has a given reserve of a token A and a takon B. In the Xfai CPMM model on the other hand, there are no token pairs, only unique token pools, i.e. instead of having many reserves for a given token fragmented across many token pairs, Xfai has one single reserve per token. During the simulations, the reserve of every pool in the Xfai CPMM was set to be equivalent to the sum of the reserves across the token pairs in the corresponding DEX (Uniswap V2 for Ethereum, and Pancakeswap for Binance). This way, the Xfai CPMM simulations have identical liquidity to their corresponding DEX comparison.

Let us recall that every pool in the Xfai CPMM model also has a weight w. w determines a pool's exchange value. To determine what the weighs of the simulation for the Xfai CPMM should be, the Uniswap / Panckakeswap reserves of the ETH / BNB were used:

$$w_i = \frac{P_{r_i} r_{e,i}}{r_{i,e}} \tag{8}$$

Where P_{r_i} represents the pool reserve of a given token i in Xfai, $r_{e,i}$ represents the reserve of ETH, or BNB within the token pair (e, i) on Uniswap or Pancakeswap, and $r_{i,e}$ represents the reserve of a token i within the token pair (e, i) on Uniswap or Pancakeswap. This step ensures that every token in the Xfai simulation has the same exchange value as every token in the corresponding DEX comparison. That is why the top market cap tokens were inspected to have a token pair with ETH or BNB in section 9.

To best demonstrate the performance capabilities of the Xfai CPMM model, every trade is performed using the pool's reserve as input amount:

$$\Delta_i = r_i \tag{9}$$

For the simulations, a swap is performed for every possible token pair combination using the corresponding Δ_i for every pool. In case of the Uniswap and Pancakeswap simulations, optimal routing is also performed. To illustrate the performance differences between the simulated DEXs, we normalized the output amounts shown in Figure 4 and Figure5 as follows:

$$\hat{Y} = \frac{Y}{X+Y} \qquad \hat{X} = \frac{X}{X+Y} \tag{10}$$

Where Y represents The output amounts matrix of Uniswap or Pancakeswap, and X represents the output amounts matrix of Xfai.

10.1 Results

Figure 4 shows the trade performances of Uniswap V2 and Xfai via two heat maps. Every row and column of the heat maps represents a token pair combination of the top 100 tokens selected from section 9. The intensity of the color (pink in the case of the Uniswap comparison), shows which DEX has a higher output amount for the given token pair trade, given the same input amount. Note, both Uniswap and Xfai simulations have the same amount of total liquidity within the system. The same simulation is performed in Figure 5 using the liquidity on Pancakeswap. For the majority of trades, the Xfai CPMM model outperforms both Uniswap V2 and Pancakeswap.

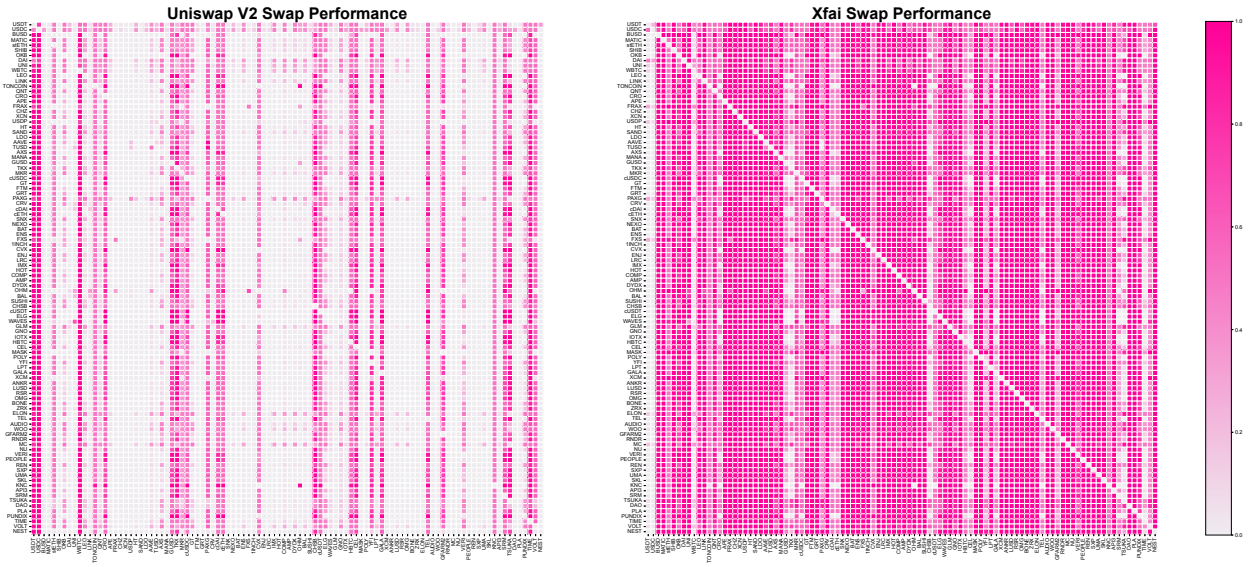

Figure 4: Uniswap V2's trade performance compared to Xfai's trade performance for every possible token combination.

10.2 Comments

The reason why the diagonals in figure 4 and 5 are empty for both heat maps, is because because one cannot trade a digital asset with itself. The reason why the lower and upper part of the diagonal for the Uniswap and Pancakeswap heat map are not symmetric, is because both DEX simulations make use of optimal routing. The reason why Xfai's heat maps are not symmetric, is because the input amount from e.g. pool i to pool j, is not the same as from pool j to pool i, as the weights and reserves differ between pools.

A performance comparisons with Sushiswap was not performed, because experiments showed that in the case of Sushiswap, too few high market cap tokens have a token pair with ETH, i.e. Sushiswap is highly fragmented. A performance comparison with Uniswap V3 was not performed, as the differences in the model designs made it difficult to make a fair comparison. The token pair matrix of Uniswap V3 is highly sparse, just as for Uniswap V2 and Pancakeswap (figure 3), therefore we expect Xfai to perform significantly better than Uniswap V3 for the absolute majority of token combinations. One factor is however important to keep in mind: The process of liquidity provisioning

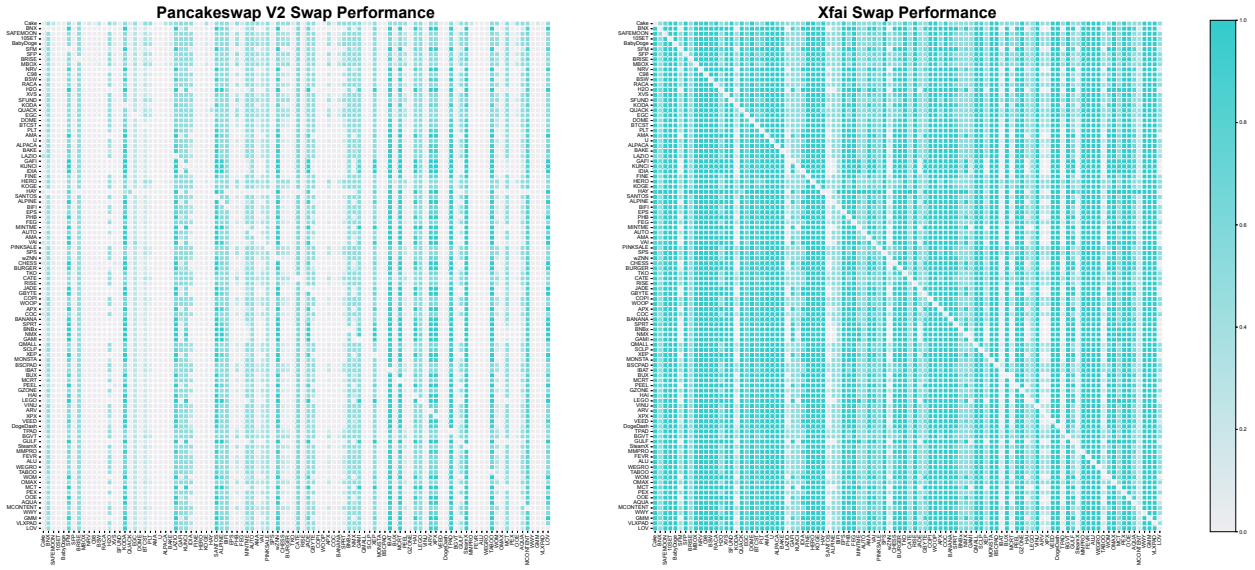

Figure 5: Pancakeswap's trade performance compared to Xfai's trade performance for every possible token combination.

fundamentally differs between the two models. Xfai's liquidity provisioning approach is closer to Uniswap V2, Pancakeswap, Sushiswap, etc., i.e. liquidity provisioning is "passive" on Xfai. That is, once the liquidity is provided, it does not have to be managed. Because of Uniswap V3's concentrated liquidity feature, liquidity provisioning on V3 is an active process, i.e. To prevent massive impermanent loss, it has to be actively managed by market makers. The types of liquidity providers between Xfai and Uniswap V3 therefore differ.

11 Conclusion

In this work, we have presented a new type of weighted CPMM model. It provides a form of "concentrated liquidity", by removing the need for token pairs all together. Unlike most AMMs, Xfai does not have to fragment its liquidity across token pairs, which enables it to perform lower slippage swaps. Furthermore, since users can decide from which tokens they want to exit when burning their liquidity tokens, unlike in other DEXs, liquidity providers on Xfai can optimize for impermanent loss minimization, by entering and exiting in correlated tokens.

Acknowledgement

I would like to thank Taulant Ramabaja for his invaluable feedback, brainstorming and stimulating discussions around the Xfai CPMM model architecture. A debt of gratitude is also owed to Dian Fishekqi, who provided critical feedback on the structure of the Xfai CPMM.

References

Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of Uniswap markets, February 2021. URL `http://arxiv.org/abs/1911.03380`. arXiv:1911.03380 [cs, math, q-fin].

Noah Zinsmeister, Dan Robinson, and Hayden Adams. Uniswap V2 Core. page 10, March 2020. URL `https://uniswap.org/whitepaper.pdf`.

Hayden Adams, Noah Zinsmeister, River Keefer, Moody Salem, and Dan Robinson. Uniswap V3 Core. page 9, March 2021. URL `https://uniswap.org/whitepaper-v3.pdf`.

Fernando Martinelli and Nikolai Mushegian. A non-custodial portfolio manager, liquidity provider, and price sensor. September 2019. URL `https://balancer.fi/whitepaper.pdf`.

Michael Egorov. StableSwap - efficient mechanism for Stablecoin liquidity. page 6, November 2019. URL `https://classic.curve.fi/files/stableswap-paper.pdf`.

Eyal Hertzog, Guy Benartzi, and Galia Benartzi. Continuous Liquidity and Asynchronous Price Discovery for Tokens through their Smart Contracts; aka "Smart Tokens". page 13, May 2017. URL https://cryptorating.eu/whitepapers/Bancor/bancor_protocol_whitepaper_en.pdf.

Hakka Finance. Decentralized Stablecoin Exchange with Unlimited Liquidity, July 2020. URL https://blackholeswap.com/documents/en.pdf.

Mark B Richardson and Nate Hindman. Bancor3 Primer, 2021. URL https://drive.google.com/drive/folders/1TUNF7gOFitTkl52-PGqS4m28edp-eyst.

Stefan Loesch, Nate Hindman, Mark B. Richardson, and Nicholas Welch. Impermanent Loss in Uniswap v3, November 2021. URL http://arxiv.org/abs/2111.09192. arXiv:2111.09192 [q-fin].

Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations.