Digital Finance (2024) 6:113–143
https://doi.org/10.1007/s42521-024-00105-4

ORIGINAL ARTICLE

Clustering Uniswap v3 traders from their activity on multiple liquidity pools, via novel graph embeddings

Deborah Miori[1,4] · **Mihai Cucuringu**[1,2,3,4]

Received: 15 January 2023 / Accepted: 2 January 2024 / Published online: 30 January 2024

Abstract

Uniswap is a Constant Product Market Maker built around liquidity pools, where pairs of tokens are exchanged subject to a fee that is proportional to the size of transactions. At the time of writing, there exist more than 6000 pools associated with Uniswap v3, implying that empirical investigations on the full ecosystem can easily become computationally expensive. We propose a systematic workflow to extract a tractable sub-universe of liquidity pools, where the interconnection among such pools is maximised to capture broader dynamics within the ecosystem. The resultant set of 34 pools is then used to cluster market participants according to their liquidity consumption behaviour over such environments, for the time window January–June 2022. Introducing a novel approach, we proceed to represent each liquidity taker by a suitably constructed *transaction graph*. The graph is a fully connected network where nodes are the liquidity taker's executed transactions on the 34 pools of reference, and edges contain weights encoding the time elapsed between any two transactions. We then extend the NLP-inspired *graph2vec* algorithm to the weighted undirected setting, and employ it to obtain an embedding of the set of graphs representing market participants. This embedding allows us to extract seven clusters of liquidity takers, with equivalent behavioural patterns that can be interpreted in terms of trading attributes, i.e. preference for exotic assets over stablecoins, frequency of activity, tolerance for higher trading fees.

Keywords Clustering · Decentralised finance · Network analysis · NLP · Uniswap v3

JEL Classification C0 · C6 · C8

✉ Deborah Miori
deborah.miori@maths.ox.ac.uk

[1] Mathematical Institute, University of Oxford, Oxford, UK

[2] Department of Statistics, University of Oxford, Oxford, UK

[3] The Alan Turing Institute, London, UK

[4] Oxford-Man Institute of Quantitative Finance, Oxford, UK

1 Introduction

A *blockchain* is a type of Distributed Ledger Technology (DLT) that stores users' transactions on an increasingly long sequence of blocks of data. The ledger is replicated across a network of servers to allow the validation of new transactions by the peer-to-peer (P2P) computer network and consequent addition of blocks, thus increasing trust, security, transparency, and traceability of data. Bitcoin was the first blockchain to acquire worldwide notoriety. It was designed by the person(s) known via the pseudonymous Satoshi Nakamoto during 2007 and 2008, and subsequently described in the whitepaper (Nakamoto, 2009) in 2009. The project was released as an open source software in 2009, at which point Bitcoin started slowly acquiring increasing value and seeing higher trading volumes. However, there are important limitations preventing Bitcoin from hosting complex applications and general smart contracts, such as lack of Turing-completeness, value-blindness, lack of state, and blockchain-blindness (Buterin, 2013). These limitations fueled the rise of the Ethereum blockchain, which was first described in the 2013 whitepaper (Buterin, 2013).

Ethereum supports smart contract functionality and, due to this, it is able to offer financial services that do not rely on intermediaries such as brokerages, exchanges or banks. Thus, Ethereum is commonly considered as the protocol that first allowed the formulation of foundations for Decentralised Finance (DeFi). Within DeFi, individuals can lend, trade, and borrow using software that automatically broadcasts their intentions for P2P verification. Valid financial actions are then recorded on the blockchain. Decentralised Exchanges (DEXs) are a direct result of this setup, and started being designed and implemented mainly from 2017. They differ from the usual centralised exchanges, since they are non-custodial and leverage the self-execution of smart contracts for P2P trading, allowing users to retain control of their private keys and funds. One of the first and most established DEXs at the time of writing is Uniswap, built on Ethereum and launched in November 2018. There exist three versions of Uniswap (namely v1, v2, v3, see the whitepapers Adams (2020), Adams et al. 2020; Adams 2021 respectively) that update its design and evolve its functionalities.

In this research, we focus on an analysis of Uniswap v3 data and investigate the related ecosystem. In particular, we achieve a clustering of the behaviour of active agents that can be mapped to interpretable classes. Thus, we now introduce Uniswap's core aspects as deployed in its first two versions v1 and v2. These details serve indeed to define the main mechanisms governing Uniswap liquidity pools (i.e. exchange venues), and thus provide the relevant background on both the actions that market participants can take, and the attributes that they might assess before finalising their trading decisions. Similarly, attractiveness and health of liquidity pools can be then derived. Uniswap is an Automated Market Maker (AMM), and in particular, a Constant Function Market Maker (CFMM). This implies that digital assets are traded without centralised permission and the pricing occurs following a mathematical formula, rather than relying on an order book as in traditional exchanges. Uniswap smart contracts hold liquidity reserves of various token pairs and trades initiated by liquidity takers (LTs) are executed directly against these reserves. Reserves are pooled from a network of liquidity providers (LPs) who supply the system with tokens in exchange for a proportional share of transaction fees. A standard Uniswap liquidity pool allows the exchange, or *swap*, of two assets via the constant product market maker mechanism

$$(x - \Delta x) \times \Big[y + \big(1 - \frac{\gamma}{10^6}\big)\Delta y\Big] = x \cdot y = k, \tag{1}$$

where $x, y \in \mathbb{Q}^+$ are the current pool reserves of tokens X, Y respectively. Similarly, Δx (resp. Δy) denotes the quantity of token X (resp. Y) exchanged in an individual swap transaction. The signs in Eq. (1) imply that a LT sells an amount Δy of token Y to the pool, while receiving Δx of token X back. The instantaneous exchange rate Z between the two digital assets is given by the proportion of respective reserves in the pool, i.e.

$$Z = \frac{x}{y}, \tag{2}$$

and changes following the trades of LTs. Parameter $k \in \mathbb{Q}^+$ tracks the evolution of liquidity of the pool, and indeed it encapsulates the size of token reserves currently available. Swaps do not change k, while this invariant does vary if new liquidity is *minted* (added) or *burned* (withdrawn) in the pool by LPs. Finally, $\gamma \in \{100, 500, 3000, 10{,}000\}$ (i.e. $\{1, 5, 30, 100\}$ basis points) denotes the *feeTier* characteristic of the pool. A lower feeTier has the immediate effect to attract more LTs, since they need to pay a smaller percentage fee (i.e. $\frac{\gamma}{10^6}$) on their orders. On the other hand, LPs are motivated to target pools with higher feeTier, since they share the above fees proportionally to their involvement into the whole liquidity of the pool. Of course, higher liquidity assures less price slippage for LTs and is thus also gauged before trading. In Uniswap v3, the main difference from the previous versions is that *concentrated liquidity* is implemented. The continuous space of exchange rates of pools is discretised into intervals whose boundaries are called *ticks*, and every minting action of LPs specifies two ticks between which the liquidity will specifically be provided. This means that LPs can choose the range of prices and proportions over which they lock their tokens. However, concentrated liquidity also implies that LPs collect LTs' fees only while the exchange rate of executed trades lies between two ticks over which they are indeed providing liquidity. For a full characterisation of the wealth of LTs and LPs in Uniswap v3, we point to the mathematical analyses pursued in Cartea et al. (2022), Cartea and Drissi (2022).

For completeness, it is also worth mentioning that every action (i.e. creation of a pool, swap, mint or burn operation...) that occurs on Uniswap, or in the general DeFi universe, must be validated and registered on the blockchain to be considered executed. This introduces a further cost for the initiator of the action, who needs to pay non-negligible *gas fees* (Pierro & Rocha, 2019) to miners to compensate them for the computational power they devote. This mechanism directly applies to blockchains based on the Proof-of-Work consensus protocol (such as Ethereum until September 2022, when it transitioned to Proof-of-Stake via the upgrade named "The

Merge”). However, gas fees are similarly requested for general transaction verification also on blockchains with Proof-of-Stake consensus protocol.

Many further interesting new and old finance concepts live within DeFi and beyond DEXs. One such concept worth mentioning first is that of *stablecoins*. Stablecoins are digital assets that are pegged to the value of a fiat currency, and can be useful to exit risky positions while remaining inside the crypto ecosystem. Some stablecoins are fiat-backed (e.g. USDC, Tether), while others are backed by an over-collateralised pool of cryptocurrencies (e.g. DAI). There also exist algorithmic coins (e.g. UST), which closely resemble traditional pegged exchange rates and are indeed also vulnerable to speculative attacks, i.e. as it happened with the Terra-Luna crash in May 2022. Apart from stablecoins, DeFi provides several lending protocols (e.g. Aave, Compound, Instadapp, Maker), protocols for derivatives trading (e.g. dYdX, Futureswap, Nexus), and DEX aggregators (e.g. 1inch) that optimise routing to take advantage of the best exchange rates across multiple other exchanges. In Kitzler et al. (2021), there is an interesting study of the interactions between different blockchain protocols.

While DeFi is fascinating, it is also the stage of many scams, speculative high-risk investments, direct blockchain attacks, and money laundering events. On top of that, its complexity and atomicity might disadvantage small users, whose transactions can e.g. be re-ordered before execution by the validators for their own profit, known here as *miner extractable value* (MEV). Despite the current effort of regulators to penetrate the crypto world and establish some equilibrium between centralisation and decentralisation of power, the current situation and possible upcoming developments are still highly confusing, especially for outsiders or newcomers. Interesting overviews and critical thoughts are presented in Schär (2020) and Makarov and Schoar (2022), where the latter work especially discusses enforcing tax compliance, anti-money laundering laws and how to possibly prevent financial malfeasance. The different layers of DeFi are studied in Inzirillo and de Quenetain (2022), where the related specific risks, i.e. at the blockchain, protocol, pool, and token level, are also analysed and a risk parity approach for portfolio construction in Uniswap is proposed.

The current academic research has also become significantly active in its efforts to understand the inner dynamics of DEXs and external relationships with the well-known traditional stock market, especially from an empirical and data-driven point of view. Among interesting recent studies there is Fan et al. (2022), where the authors investigate how promoting a greater diversity of price-space partitions in Uniswap v3 can simultaneously benefit both liquidity providers and takers. Then, Fritsch et al. (2022) studies whether AMM protocols, such as Uniswap, can sustainably retain a portion of their trading fees for the protocol and the expected outflow of traders to competitor venues. Inefficiencies between Uniswap and SushiSwap are investigated in Berg et al. (2022), where sub-optimal trade routing is addressed. However, Angeris et al. (2019) shows that constant product markets should closely track the reference market price. Flows of liquidity between Uniswap v2 pools are studied in Heimbach et al. (2021), while Heimbach et al. (2022) shows the difficulty of earning significant returns by providing liquidity in Uniswap v3. Interestingly, Cartea et al. (2022) fully characterises the wealth of LTs and LPs in Uniswap v3, and shows that LPs suffer a “Predictable Loss”. In another recent line of work, Miori and Cucuringu (2023) considers the problem of modeling and forecasting trading volume on Uniswap v3 liquidity pools via a multivariate regression model, using features that relate to the most recent activity on such pools, and including spillover effects from neighbouring pools.

We nevertheless believe that there does not exist any study yet, which is able to clusters DEX traders in interpretable classes as we do, by simply considering the temporal component of executed transactions and the generic related pools of reference. Furthermore, only few research efforts have investigated similar objectives on Limit Order Book (LOB) data, since the confidential nature of such centralised data oftentimes prevents accessibility. A relevant study is the recent work in Cont et al. (2023), in which the authors extract and analyse four distinct clusters of client order flow from a large broker in US equity markets. This is achieved by assessing the similarity of agents’ trading behaviour from e.g. the side usually taken (buy/sell orders), number of orders, time of submission, and size. The uncovered heterogeneity of order flow is modelled by partitioning clients into different clusters, for which representative prototypes are also identified. Pursuing a complementary analysis, the authors of Cartea et al. (2023) build statistical models to describe how individual market participants choose the direction, price, and volume of orders. Such models are endorsed with features representative of both the recent and current state of the LOB, and also the agent’s past actions, cash levels, and inventory levels. Then, the coefficients from such fitted models are used to cluster trading behaviour, suggesting the existence of three separate groups of agents, namely directional traders, opportunistic traders, and market makers. Relatedly, the authors of Mankad et al. (2011) analyse audit trail data at transaction level from the S &P 500 eMini Future during four days, including the flash crash in May 2010. They parse through the footprints of all traders with a proposed machine-learning method, and identify five related categories of agents (i.e. high frequency traders, market makers, opportunistic traders, fundamental buyers and sellers, and small traders). These classes mainly differ from the dynamics of intraday trading volume and end of day holdings, as similarly considered in Kirilenko et al. (2017). The seminal work in Kirilenko et al. (2017) further argues for the importance to study the heterogeneity of order flow, as for its implications for both risk managers and market regulators. And this is indeed one of the reasons why we study such topic also in the context of decentralised markets. The work in Wright (2022) also introduces machine learning methods to cluster active traders, by leveraging on a custom discrimination metric that analyses times, volumes, and prices at which traders are willing to transact. In particular, the authors compare traders’ individual order book participation over defined windows of time.

For completeness, we also want to highlight how the above-mentioned clustering analyses can allow more specific downstream characterisation of the different types of agents on the market. So far, we mainly find in-depth studies that focus on the impact of the recently born classes of high-frequency traders (HFTs) or market makers (MM), due to related concerns on their behaviour by regulators. As a first example, Megarbane et al. (2017) investigates how HFTs affect the markets, questioning whether their activity improves or damages liquidity, and if their involvement increases or decreases during periods of market stress. Overall, the authors claim that HFTs are important contributors of liquidity. Relatedly, Ho and Stoll (1983) suggests that competition between HFTs decreases their profits, and leads to lower trading costs for other market participants. Furthermore, Brogaard (2010) finds evidence that HFTs do not withdraw from market conditions of stress, reduce price volatility, and strongly contribute to price discovery. On the other hand, several theoretical models (e.g. Foucault et al. 2016a, 2016b) suggest that when speed differentials between traders exist, adverse selection may increase and liquidity become more expensive.

Main contributions We systemically investigate how to subset the extremely large set of Uniswap v3 pools, most of which are highly illiquid, to still retain an highly accurate view of Uniswap v3 dynamics. Conducting research over the totality of such universe would indeed lead to less insightful results, due to the large noise and sparsity of the data. We assess both the pools' inner features and their interconnectedness, and present a workflow for extracting significant sub-universes of pools in time. Our proposed filtration mechanism thus facilitates analysing the behaviour of multiple liquidity takers across different pools, and strikes a balance between maintaining a sufficiently large universe of pools in order to capture the broader dynamics of the ecosystem, and large enough activity of the LT to ensure their presence across multiple pools. We then leverage on the results of such approach, to indeed cluster LTs and characterise their broad behaviour. This clusterisation highlights the main typologies of market participants, and enables further downstream tasks customised for each cluster individually.

Structure of the paper In Sect. 2, we identify the most important and interconnected liquidity pools for different time windows within 2022. Next, we cluster LTs according to their behaviour on the relevant sub-universes in Sect. 3. Due to the complexity of this ecosystem, we draw on intuition from Natural Language Processing (NLP) and graph embedding techniques to assess structural equivalence of trading behaviour in a novel way. Finally, we summarise our thoughts and discuss future research directions in Sect. 4.

2 Systematic selection of Uniswap v3 pools of interest

2.1 Empirical introduction to the ecosystem

Our first goal is to extract a tractable sub-universe of Uniswap v3 liquidity pools, in which the interconnectedness among such pools is maximised to capture broader dynamics within the ecosystem. This sub-universe will then be used to indeed investigate the broader trading behaviour of market participants, and experiment in the objective of clustering these agents into interpretable classes.

At the time of writing, Uniswap v3 is the latest implementation of the Uniswap DEX. Uniswap v3 launched in May 2021, introduced the concept of concentrated liquidity and allowed multiple feeTiers (i.e. $\gamma \in \{100, 500, 3000, 10{,}000\}$). For each

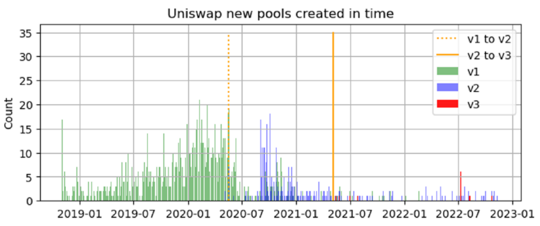


Fig. 1 Daily count of new pools created via UniswapV1Factory, UniswapV2Factory and UniswapV3Factory smart contracts. The two vertical orange lines depict the dates of official transition from Uniswap v1 to v2, and from v2 to v3

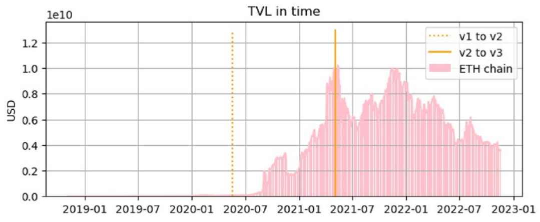


Fig. 2 Evolution in time of the TVL in USD on Uniswap main Ethereum chain, where the data are downloaded from Defi Llama API. The higher the TVL, the more liquid the ecosystem is considered to be. In orange, we show the dates of transition from Uniswap v1 to v2, and from v2 to v3

Uniswap version $N = 1, 2, 3$, we access the related liquidity pool smart contracts via Etherscan,[1] and quantify the daily creation of new pools in Fig. 1. The dates of transition from Uniswap v1 to v2, and v2 to v3, are also depicted. Although there is a total of 4889 pools directly created with UniswapV*N*Factory contracts, the majority of them are the result of the 2020–2021 cryptomania and inflated creation of new tokens (see Fig. 2 for a proxy of Total Value Locked, TVL, in the protocol over time). This translates to many pools not containing any relevant amount of liquidity locked, but which do not disappear due to the immutability of the blockchain. On the other hand, we also expect wrapped calls to the Factory contracts and thus refer to the above as a lower bound to the number of pools created.

2.2 Data download and coarse refinement of pools

"The Graph"[2] is a decentralised protocol for indexing and querying blockchain data. It allows to query data that are otherwise difficult to access, such as information from smart contract computations that is stored on the Ethereum blockchain. Thus, we construct Python scripts that connect to The Graph Uniswap v3 pointer (i.e. https://api.thegraph.com/subgraphs/name/uniswap/uniswap-v3) and download data on both the latest states of pools, and the full history of related swap, mint, and burn events. This is achieved via GraphQL queries that follow the Uniswap v3 own subgraph schema,[3] meaning that we directly access "Entity" GraphQL objects with

1 https://etherscan.io/.
2 https://thegraph.com/docs/en/about/.
3 https://github.com/Uniswap/v3-subgraph/blob/main/schema.graphql#L1.

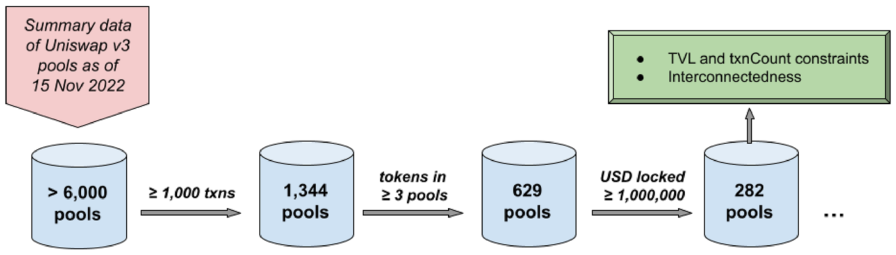
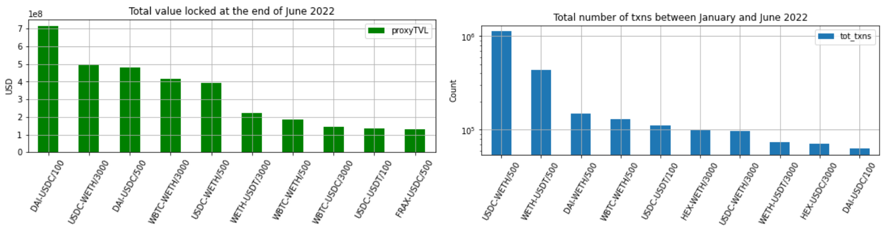


Fig. 3 Summary diagram of the filtration steps pursued during our coarse refinement of pools. Stronger constraints on the TVL and txnCount of pools will follow, such as an attention to maximise the interconnectedness of the final sub-universe of pools

full features for the operations of interest. Our first aim is to identify the pools most representative of the Uniswap ecosystem, which we interpret as having significant liquidity consumption and provision events, but also showing high interconnectedness. To this end, we extract the latest summary data of all possible pools (from "Pool" GraphQL entities), full historical record of liquidity consumption operations (from "Swap" GraphQL entities), and full record of liquidity provision actions (from "Mint" and "Burn" GraphQL entities). Next, we develop a systematic approach that aims at increasingly discarding layers of pools with weakest features and dynamics. The filtered data set represents the starting point of our subsequent analyses, since it encompasses the major part of the behaviour of liquidity takers, and can be consequently used to characterise their trading structure.

We consider the data from Uniswap v3 liquidity pools as of 15 November 2022. We begin by applying a threshold of minimum 1000 transactions already happened on each pool, and recover an initial number of 1344 pools to consider. Then, we also restrict ourselves to pools where both exchanged tokens are traded in at least 3 pools (e.g. token T is traded against a stablecoin, against ETH and against ETH with different feeTier), in order to focus on interesting dynamics of the full ecosystem. The result is a subset of 696 Uniswap liquidity pools to further consider in our study. To lower the noise in the data considered, we limit them to the 6-months period from January 2022 to the end of June 2022. In this way, we assure to have all four Uniswap v3 feeTiers active (i.e. after November 2021), but avoid turbulences from The Merge upgrade (i.e. in September 2022).

LP data filters We extract liquidity provision data for the above 696 pools, of which only 629 have non-empty entries due to issues on Uniswap end. Liquidity provision provides a historical record of all liquidity mint and burn operations on each pool, with related USD value. By computing the total cumulative sum of LPs activity, we proxy the TVL in USD that each pool contains at each point in time and denote it as "proxyTVL". Unfortunately, we cannot simply rely on the "PoolDayData" values provided by the subgraph due to incoherences found when cross-checking with Ethereum blochckain data on Etherscan. We then proceed to filter pools by our proxyTVL being larger than 1,000,000 USD (one million dollars) at any point before the



(a) Pools with highest liquidity at the end of June 2022. (b) Pools with highest total number of transactions during case A.

Fig. 4 The 10 most liquid and active pools for case A, i.e. over the time window between January and June 2022

end of June 2022. This is motivated by the aim to find pools that were liquid enough at some point in our time window to capture interesting behaviour of LTs and LPs. We find 282 pools that satisfy this further requirement, as highlighted in the filtration summary diagram of Fig. 3.

LT data filters To further lower the noise-to-signal ratio in the data, we can subset our universe of interest to the pools with enhanced activity of LTs. As already motivated, we wish to focus on the 6-months window [*January*, *July*) 2022, which we denote as our case A. We also consider five sub-ranges, namely the two 3-months windows [*January*, *April*), [*April*, *July*) that we denote as cases $B1/B2$, and the three two-months windows [*January*, *March*), [*March*, *May*), [*May*, *July*), that we call cases $C1/C2/C3$. For each case and related time window [*start*, *end*), we extract the pools with at least 1000 transactions before *start* (where the number of transactions in time is calculated via the cumulative sum of both swap events and mint or burn operations) and that also had at least 1,000,000 USD in proxyTVL both at the *start* and *end* of the interval. Considering sub-ranges allows us to further account for the appearance of new pools that became significantly liquid or active after January 2022, or pools that lost the majority of their liquidity before July 2022, thus lowering survivorship/loser bias. For cases $A/B1/B2/C1/C2/C3$ in order, we find respectively 113/126/148/131/146/155 pools that satisfy the above requirements, for which we save the related addresses and information. Taking the union of these sets of pools, we notice that we are considering 177 different pools overall. Of these, five pools belong to the 100 feeTier, 28 pools to the 500 feeTier, 84 pools to the 3000 feeTier and 60 pools to the 10,000 feeTier.

To gain a brief insight into the most liquid and active venues, we consider the pools extracted for case A and plot in Fig. 4a the 10 pools with highest proxyTVL at the end of June 2022, and in Fig. 4b the 10 pools with highest total number of transactions over the six months of relevance. As a convention, we refer to pools with the format "SYMBOL1-SYMBOL2/feeTier", where we use the trading symbols of the two tokens exchanged by the pool. Stablecoins, wrapped Ether (WETH) and wrapped Bitcoin (WBTC) dominate the landscape of tokens

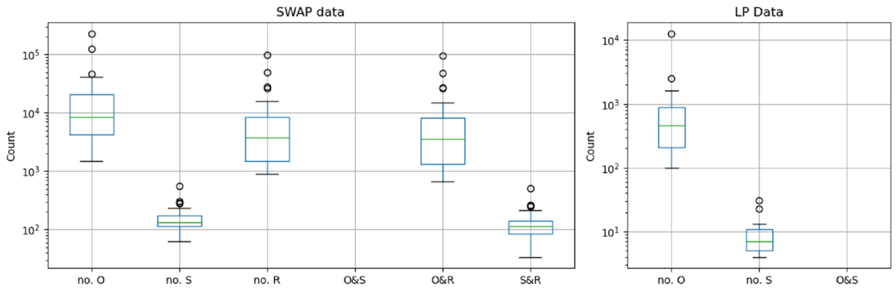


Fig. 5 The intersection of origins and senders is always zero since the former are wallets of users and the latter smart contracts. Recipients can instead be both, hinting to more complex patterns in the execution of transactions. To be precise, here we are specifically considering the sub-universe of pools relevant for case *A* at the end of all our refinements

swapped in the most liquid and active venues, which is expected since they are the oldest, most established, or safest cryptocurrencies that agents can trade and develop strategies onto.

2.3 Network-driven filters

We provided above the systemic steps needed to complete an initial filtration of the extremely large set of Uniswap v3 pools. Now, we further apply network-based measures to extract the most interconnected subset of such pools, based on the distributed behaviour of liquidity takers. The resultant pools will be highly representative of the trading structures of such market participants, allowing us to cluster them with confidence in the significance of results.

Filtering of pools by basic interconnectedness measures For each one of our cases $A/B1/B2/C1/C2/C3$, we build a weighted graph $G = (P, E)$. The set of nodes P denotes relevant pools, and edges $(p, q) \in E$ with $p, q \in P$ have weights w_{pq} that encode a measure of similarity defined below. We start by considering two possible different measures of connection between pools:

1. Number of common LTs active on both pools, which are identified by the entry "origin" in the Uniswap data.
2. Number of common smart contracts, i.e. "senders" in the Uniswap data, called by origins to execute swap transactions.

To clarify the Uniswap terminology adopted above, we remind that every "swap action" is initiated by an *origin O*, it then calls a smart contract referred to as *sender S*, and ends to the *recipient R*. Figure 5 shows the distribution of number of origins, senders and recipients in each pool for both LT and LP data over the time window of case *A*. We also show the distribution of the intersection between origins, senders and recipients' addresses.

We now proceed to studying the relationship between the size of each graph's giant component and a minimum threshold on the value of the measure used to

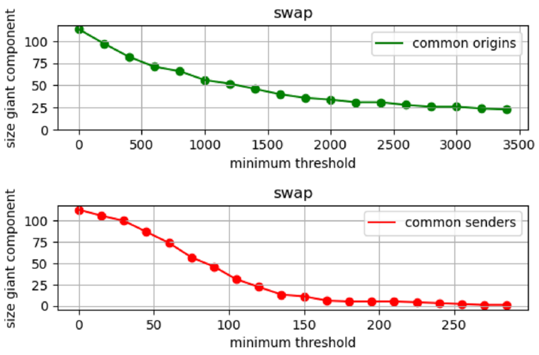


Fig. 6 Evolution of the size of the giant component for graphs of pools in case A, when varying the threshold of common origins and senders for swap transactions

create the link between each pair of pools. After fixing a threshold, we consider the pools in the related giant component as our relevant interconnected sub-universe. Figure 6 shows the variation in size of the giant component for case A, when modifying the minimum number of common origins or senders for LT data. We aim at considering the tails of the distributions for each case (i.e. time interval), which amount to $\sim$ 20–30 pools in each instance, to retain the most significant connections and possible dynamics of the ecosystem. For case A, this results in the choice of thresholds 2000 and 100 for minimum common origins and common senders respectively. Finally, we consider the intersection of survival pools for the two graphs generated by LT data, and find 27 common pools (out of the 34 and 36 pools, respectively in each graph). The full pipeline is repeated for cases $B1/B2/C1/C2/C3$.

Enhancement of pools for liquidity consumption analysis Ideally, we should also consider the flow of funds across pools and find the related most interconnected graph. However, this is intractable if using only Uniswap data and not the full list of Ethereum blockchain transactions. Indeed, LTs are active across different DeFi protocols and can easily move liquidity from one venue to another and back. We propose an approximation to the problem taking advantage of the fact that each trader's transaction can include more actions, which happen "instantaneously but in order" when the full transaction is validated. Thus, if a LT executes two swaps of the form $X \rightarrow Y, Y \rightarrow Z$ for tokens X, Y, Z in one same transaction, then we interpret Y as a *bridge* between the action of selling X to buy Z. We view this as an indication of the flow of (smart) money between pools and of possible arbitrage opportunities, relevant to the LT sub-universe. In summary, we consider the following steps:

1. Merge all LT data before the interconnectedness analyses, e.g. data for the 113 pools of case A.
2. Keep all the transactions for which there are at least two inner actions, i.e. same "transaction id" but different "logIndex" in Uniswap terminology.
3. For each resulting transaction:

   (a) For each token that appears in the transaction actions, keep a *flow list* of related buying (-1) or selling ($+1$) trades in all the related pools by looking at the sign of the amount swapped by the pool.

(b) For each token, consider its flow list and find all the occasions when a -1 is immediately followed by a $+1$ (i.e. the token was first bought in a pool and then sold in another pool, acting as one of our bridges).
(c) Save this occurrence of a flow between pools as a *bridge transaction*,

where we are approximating only jumps of length one. As an example, for a flow list of the form $[-1, +1, +1]$, we only consider the flow as one from the first pool to the second one. For more specific analyses, one could consider the specific amounts traded and check the relative proportions exchanged from the first pool to the second and third ones, but this is outside the scope of our current investigation.

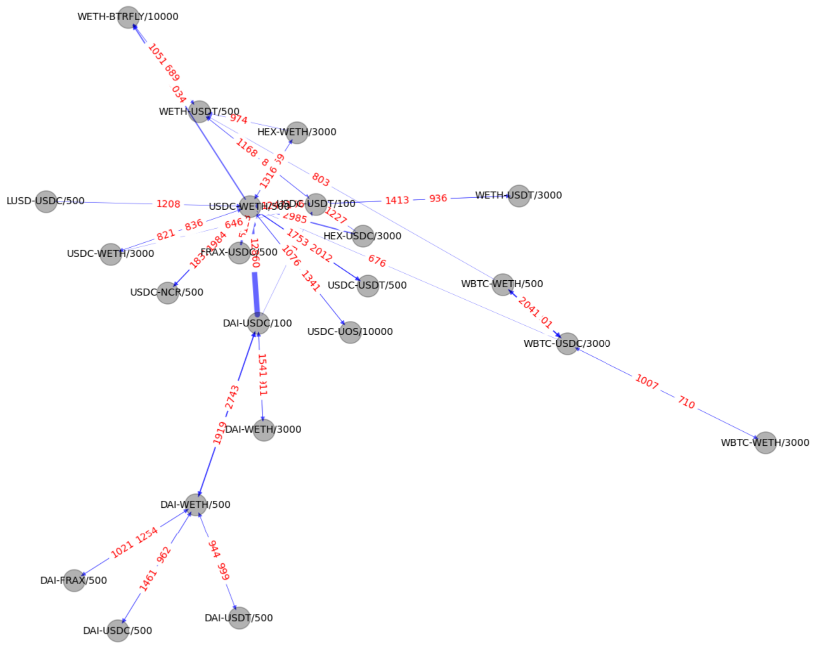
We extract all bridge transactions between pools and create a directed graph for each one of our temporal cases. Nodes are pools as usual, and edges are built for each pair of pools that have at least some number of bridge transactions between them. Of course, each pair of pools can have up to two edges between them according to the direction of related bridge transactions. Then, we keep the largest connected component from the undirected version of the graph and add the resultant set of nodes to the LTs pools saved from the previous interconnectedness analyses. For case A, we require at least 800 bridge transactions between two pools to create the related edges. The resultant giant component (see Fig. 7a for a visualisation) has 22 nodes, seven of which were not included in our LT set of pools from the previous analyses and are thus added. Figure 7b further highlights the nodes with highest eigenvector centrality in the graph, where we can especially notice how several pools of WETH against a stablecoin are proposed. This is intuitively sensible, since LTs can take advantage of routing to complete specific re-balancing of tokens via more liquid and favourable pools, which tend to have stablecoins, WETH and WBTC as their tokens, as shown in the earlier analyses.

Thus, our framework finally proposes a set of 34 pools to consider for LTs analyses for case A, which encompass the most significant part of their transactions and will be used to cluster their trading behaviour.. The full lists of pools are reported in Appendix A, where we also propose our filtering results for cases $B1/B2/C1/C2/C3$.

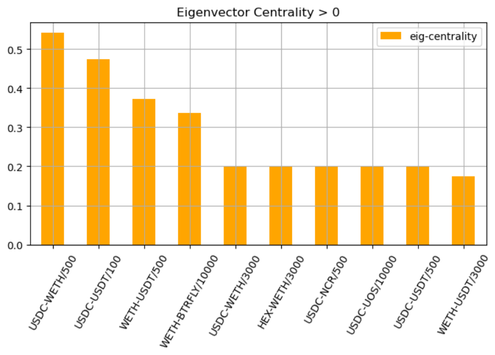
3 Structural investigation of the Uniswap v3 ecosystem

3.1 Clustering of liquidity takers

The DeFi ecosystem has grown increasingly complex in the recent years. The first step to shed more light on its intrinsic features and dynamics is to better understand its own components, which is what motivates the following empirical investigation of LTs trading behaviour on Uniswap v3. This is a non-trivial task, for a number of reasons. First of all, agents can easily generate numerous crypto wallets, and hence in some sense, “multiply” their identities to hide or obfuscate their full behaviour. Their actions are then generally spread over a broad set of possible pools, vary significantly in size both within and across different types of pools, and also happen with evolving frequencies over time. Due to the high interconnectedness of the ecosystem and noticeable volatility across cryptocurrencies (which is of interest to



(a) The resulting giant component, with edge weights reported in both directions.



(b) Pools with highest eigenvector centralities.

Fig. 7 Results from our bridges investigation for case A, which covers the six-months window from January to June 2022. If a LT executes two swaps $X \rightarrow Y, Y \rightarrow Z$ one after the other (for tokens X, Y, Z), then we interpret Y as a bridge between the action of selling X to buy Z. We save all pairs of pools for which there is a common token that acts as a bridge, with the related number of occurrences of bridge transactions. Then, we create a directed graph where nodes are pools and edges are built for each pair of pools that have at least 800 bridge transactions between them

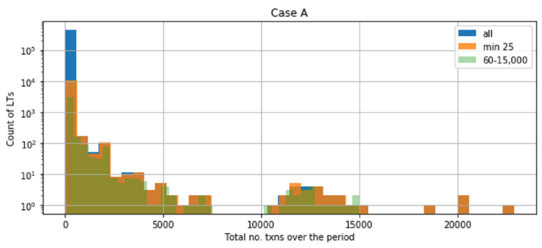


Fig. 8 Distribution of total number of transactions (txns) performed by LTs during case *A*. We show the full distribution, the result after requiring a minimum of at least 25 transactions, and the distribution after applying thresholds of minimum 60 and maximum 15,000 transactions. The latter scenario results in our final set for case *A*, which comprises 3415 LTs. A small cluster of LTs much more active than others is already discernible

speculators), we believe that it is essential to consider the broad trading behaviour of agents across pools, which we can easily do thanks to the transparency of our Uniswap data. The filtering of pools just completed thus becomes of immediate use, since it points to the most important venues that characterise agents' trading behaviour. However, this choice forces us to diverge from the literature on clustering order flow discussed in Sect. 1, because such studies do not account for the contemporaneous exposure to multiple different assets. Thus, we will soon propose a novel method to express and cluster the structural trading equivalence of agents active on multiple environments, by leveraging on both network analysis and NLP techniques. Nevertheless, we will also gauge whether the unravelled clusters can be interpreted with proxies of some of the mentioned main discerning factors of LOB order flow (i.e. volume and time of submission, similarly to Cont et al. 2023; Cartea et al. 2023; Wright 2022). We will also build features from the decentralised ecosystem of reference, to better characterise the groups identified and extract insights on the main types of agents present. Importantly, we are not able to compute the inventory of agents as done in LOB models, since traders tend to be active on multiple DeFi protocols for the same assets, but we only see transactions specific to Uniswap.

3.1.1 Overview and pre-processing

We focus on LTs transactions happening during our three longest periods $A/B1/B2$, for the related pools of interest as identified in the previous section. For each case, we first look at the distribution of the total number of transactions performed by the different LTs over each full time window. As an example, the distribution for case *A* is represented by the blue bars in Fig. 8. We then require a minimum number of transactions completed by each LT, since considering only a very small sample of trades per agent does not provide meaningful structural information on their behaviour. Thus, we impose a lower bound of a minimum average of 10 transactions per month that each LT must have completed. On the other hand, we manually define maximum thresholds to remove only extreme singular outliers from each distribution for computational purposes. The initial total distribution for case *A* is shown in Fig. 8, where we also highlight how it changes when requiring a minimum number of transactions equal to 25 (orange bars), and when we require our final minimum and maximum thresholds of 60 and 15,000 total number of transactions, respectively (green bars). For cases $B1/B2$, we require the range 30–5000 transactions for the former, and 30–11,000 transactions for the latter. Overall, we find a number of LTs approximately between 3500 and 5000 for all our periods $A/B1/B2$. This altogether defines the final sets of LTs along with their transactions. Next, we proceed to define and compute their embeddings, which are subsequently used for the final clustering stage.

3.1.2 Methodology

NLP background and *graph2vec* The field of Natural Language Processing (NLP) studies the development of algorithms for processing, analysing, and extracting meaningful insights from large amounts of natural language data. Examples of its myriad applications include sentiment analysis of news articles, text summary generation, topic extraction and speech recognition. One of the turning points in NLP was the development of the *word2vec* word embedding technique (Mikolov et al., 2013), which considers sentences as directed subgraphs with nodes as words, and uses a shallow two-layer neural network to map each word to a unique vector. The learned word representations capture meaningful syntactic and semantic regularities, and if pairs of words share a particular relation then they are related by the same constant offset in the embedding space. As an example, the authors observe that the singular/plural relation is captured, e.g. $x_{\text{apple}} - x_{\text{apples}} \approx x_{\text{car}} - x_{\text{cars}}$, where we denote the vector for word i as x_i. Words sharing a common context in the corpus of sentences also lie closer to each other, and therefore, relationships such as $x_{\text{king}} - x_{\text{man}} + x_{\text{woman}} \approx x_{\text{queen}}$ are satisfied with the analogies indeed predicted by the model.

Taking inspiration from this idea of preserving knowledge of the context window of a word in its embedding, the *node2vec* algorithm (Grover & Leskovec, 2016) learns a mapping of nodes in a graph to a low-dimensional space of features by maximising the likelihood of preserving network neighbourhoods of nodes. The optimisation problem is given by

$$\max_f \sum_{s \in S} \log Pr(N_L(s)|f(s)), \tag{3}$$

where $G = (S, T)$ is a graph with nodes S and edges T, f is the mapping function for nodes to n-dimensional vectors that we aim to learn, and $N_L(s) \subset S$ is the network neighbourhood of node s generated with sampling strategy L. The latter is designed by the authors of *node2vec* as a biased random walk procedure, which can be tuned to either focus on sampling a broader set of immediate neighbours, or a sequence of deeper nodes at increasing distances. Then, Problem (3) is solved for f by simulating several random walks from each node and using stochastic gradient descent (SGD) and backpropagation.

By taking a further step towards general language representations, Le and Mikolov (2014) proposes the unsupervised algorithm *Paragraph Vector* (also known as *doc2vec*), which learns continuous fixed-length vector embeddings from variable-length pieces of text, i.e. sentences, paragraphs and documents. The vector representation is trained to predict the next word of a paragraph from a sample of the previous couple of sentences. Both word vectors and paragraph vectors need to be trained, which is again performed via SGD and backpropagation.

As *doc2vec* extends *word2vec*, *graph2vec* (Narayanan et al., 2017) is a neural embedding framework that aims to learn data-driven distributed representations of an ensemble of arbitrary sized graphs. The authors propose to view an entire graph as a document, and to consider the rooted subgraphs around every node in the graph as words that compose the document, in order to finally apply *doc2vec*. This approach is able to consider non-linear substructures and has thus the advantage to preserve and capture structural equivalences. One necessary requirement to pursue this analogy is for nodes to have labels, since differently labelled nodes can be then considered as different words. These labels can be decided by the user, or can be simply initiated with the degree of each node. Thus, *doc2vec* considers a set of graphs $\mathcal{G} = \{G_1, G_2 ...\}$, where the nodes S of each graph $G = (S, T, \lambda)$ can be labelled via the mapping function $\lambda : S \rightarrow \mathcal{L}$ to the alphabet $\mathcal{L}$. The algorithm begins by randomly initialising the embeddings for all graphs in the set $\mathcal{G}$, then proceeds with extracting rooted subgraphs around every node in each one of the graphs, and finally iteratively refines the corresponding graph embedding in several epochs via SGD and backpropagation, in the spirit of *doc2vec*. The rooted subgraphs act as the context words, which are used to train the paragraph (i.e. graph) vector representations. Subgraphs are extracted following the Weisfeiler–Lehman (WL) relabeling process (Shervashidze et al., 2011). The intuition is that, for each node in a graph, all its (breadth-first) neighbours are extracted up to some depth d. Labels are then propagated from the furthest nodes to the root one, and concatenated at each step. In this way, a unique identifier for each node is identified from its "context" and the full set can be used to train an embedding for the graph. The optimisation problem thus becomes

$$\max_{f'} \sum_{G \in \mathcal{G}} \sum_{s \in S} \log Pr(g^d_{\mathrm{WL}}(s) | f'(G)), \tag{4}$$

where the aim is to maximise the probability of the WL subgraphs given the current vector representation of the graph. Here, f' is a mapping function of graphs to n-dimensional representations, and g^d_{WL} are WL subgraphs with depth d.

A modification of *graph2vec* for LTs embedding For each one of our cases $A/B1/B2$, we consider all the related LTs and their full set of transactions on the sub-universe of LTs' pools of relevance. We then introduce the concept of a *transaction graph* G_{txn}, which we use to represent the behaviour of each active agent.

Definition 3.1 (Transaction graph) A *transaction graph* $G_{\mathrm{txn}} = (S, T, W)$ is the complete weighted graph where nodes S are the swap actions that the LT under consideration has executed, and edges $(s, r) \in T$ with $s, r \in S$ are built between every pair of nodes. Each edge has a weight $w_{sr} \in W$, which encodes the amount of time

Δt (in seconds) elapsed between the two transactions s, r. Each node $s \in S$ has a label l_s from the alphabet $\mathcal{L}$, which uniquely identifies the pool that the swap was executed into. Importantly, $\mathcal{L}$ is shared among the full set of LTs and related transaction graphs.

Labels in the alphabet $\mathcal{L}$ differentiate between swaps executed on different pools, i.e. pools with unique combination of tokens exchanged and feeTier implemented. This implies that the algorithm receives as input only general identifiers of pools. Thus, we can consider intuitive differences (e.g. expected volatility of the exchange rate on pools of stablecoins versus on pools of more exotic tokens) only afterwards, when assessing and investigating the meaningfulness and interpretability of the extracted clusters.

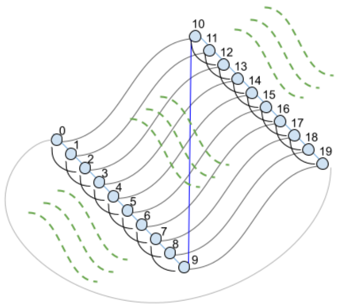
We now have a set of graphs representing LTs, and our aim is to find a n-dimensional vector representation of each one of its elements. We cannot plainly apply the *graph2vec* algorithm, since the concept of neighbours of a node is irrelevant in a complete graph. Thus, we modify its mechanism to take advantage of the weight that the different links between nodes have, while maintaining the overall intuition. For each node $s \in S$ of a graph G_{txn}, we sample a set of neighbours $N_{\text{txn}}(s)$ by generating random numbers from a uniform distribution between [0, 1] and comparing them to the *cut-value* of the edges between the node and possible neighbours. If the value is below the cut-value, then the link is kept and the associated node added to $N_{\text{txn}}(s)$. In this way, the probability of an edge to be chosen is inversely proportional to its weight Δt, and the sub-structures kept represent clustered activity in time.

Definition 3.2 (Cut-value) The *cut-value* $C(w_{sr})$ of an edge $(s, r) \in T$ with weight $w_{sr} \in W$ in graph $G_{\text{txn}} = (S, T, W)$ is computed as

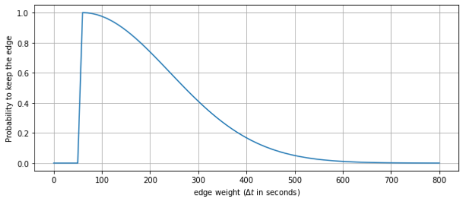
$$\begin{aligned} C(w_{sr}) &= \frac{H(f^{scal}(w_{sr}))}{H(f^{scal}(\min W))}, \\ \text{with} \qquad H(w_{sr}) &= \sqrt{\frac{2}{\pi}} \exp \frac{-w_{sr}^2}{2}, w_{sr} \geq 0, \\ f^{scal}(w_{sr}) &= \frac{w_{sr} - \min W}{(\max W)/|S|}, \end{aligned} \tag{5}$$

where we are using a half-norm that is shifted and scaled to adapt to each LT's extreme features, i.e. $\min W$ and $\max W$. The final cut-value is also normalised to impose a value of $C(\min W) = 1$, meaning that the shortest link(s) in the graph is chosen with probability 1 (of course, only if it is involved in the current node under consideration).

After having generated the set of $N_{\text{txn}}(s), \forall s \in S$, we perform WL relabeling and proceed as in the vanilla version of the *graph2vec* algorithm. We set all the hyperparameters to their default values, i.e. number of workers = 4, number of epochs = 10, minimal structural feature count = 5, initial learning rate = 0.025, and down sampling rate of features = 0.0001. The only exception is the number



(a) Transaction graph $G_{txn} = (S, T, W)$.



(b) Cut-value $C(w)$.

Fig. 9 For our illustrative example, we show in **a** a simplified representation of the LT's transaction graph, and in **b** the cut-value that defines probabilities of keeping edges as neighbours

of WL iterations, which in our case must be set to 1 instead of 2. The result is an embedding for each graph in our set of transaction graphs, which becomes a set that we can subsequently cluster via the popular k-means++ methodology. Importantly, we want to underline that our embeddings and clusters do not depend on the real magnitudes of weights Δt, since the sampling is adjusted on that. In addition, they also have no notion of the amount of USD traded, thus being agnostic to the transaction value. As a final note, we refer the reader to Chen and Koga (2019) for a version of *graph2vec* that uses edge labels. However, the algorithm creates the dual version of the graph and would not be effective in our case, thus providing ground for our proposed extension.

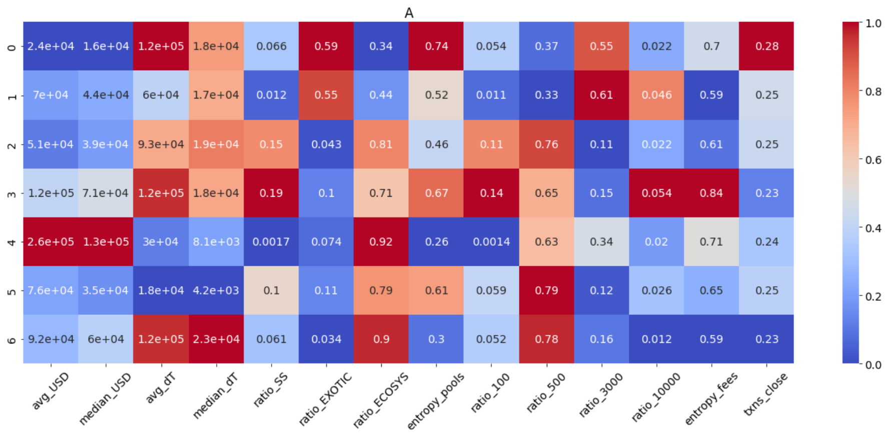
An illustrative example To clarify our approach, we describe a simple example. Consider an agent that executes 20 transactions. She executes the first 10 transactions shortly clustered in time, waiting only 60 s one after the other. Then, she waits 42 min to action on the final 10 transactions with, again, a frequency of one minute. Her behaviour is plotted in the transaction graph G_{txn} of Fig. 9a, where we assume for simplicity that each transaction is performed on the same pool and thus colour-code nodes all the same. We also number nodes to show the order in which the related transactions are executed. We do not draw all the edges of this complete graph for clarity and ease of visualisation, but hint with the green dashed lines that indeed there are more connections to be remembered. In this example, the minimum time between transactions is 60 s (light blue edges) and the maximum one is one hour, i.e. 3600 s (light grey edge). Some intermediate times are depicted as edges with the same colour for the same weight. The resultant cut-value function $C(w)$ that defines our sampling probabilities to choose edges is shown in Fig. 9b. As intuitively desired, we aim at always keeping the shortest edges and indeed these have probability 1. Then, we also aim to keep the most clustered “communities”, and indeed, we observe from the plot that transactions five minutes away are still chosen with 40% probability, but longer times are very easily dropped.

3.1.3 Analysis of results

For each case $A/B1/B2$, we study the structural equivalence of LTs' trading activity by clustering the representations generated via our modified *graph2vec* algorithm. Focusing first on case A, we compute embeddings for dimensions $n \in \{8, 16, 32, 64\}$, and confirm with Principal Component Analysis (PCA) that the proportions of data's variance captured by different dimensions are well-distributed. For each n-dimensional set of vectors, we then group LTs by performing a series of k-means++ clusterings with different number of desired groups. We compute the inertia of each partition found, where inertia is defined as the sum of squared distances of samples to their closest cluster center. Then, we choose the optimal clustering via the *elbow method*, i.e. by picking the point where the marginal decrease in the loss function (inertia) is not worth the additional cost of creating another cluster. The similarity between optimal clusterings for different dimensions is then computed, in order to investigate the stability of results across representations of increasing dimensionality. We achieve this by computing the Adjusted Rank Index (ARI) (Hubert & Arabie, 1985), which is a measure of similarity between two data clusterings in the range $[-1, 1]$, adjusted for the chance of grouping of elements. We find ARIs for clusterings on 8-vs-$\{16, 32, 64\}$ dimensional data around 0.75, while clusterings on 16-vs-32, 16-vs-64 and 32-vs-64 dimensional data reach approximately the value of 0.90. Therefore, we conclude that there is a high stability of results when our data are embedded at least in 16 dimensions, and use the related 16-dimensional vector representations for our final analyses. The related optimal number of clusters of LTs for case A is seven. Similar results arise for cases $B1/B2$ too, and the related optimal numbers of clusters of LTs are six and seven, respectively.

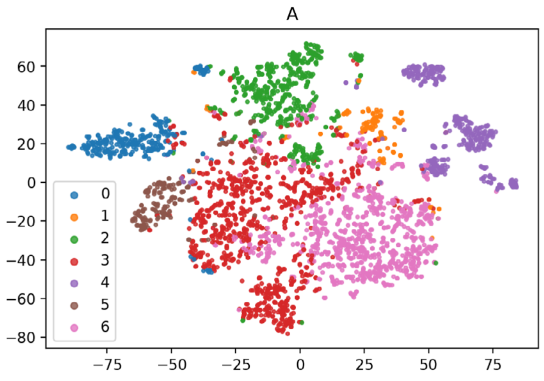
Each extracted clustering is based on the structural similarity of LTs' trading behaviour. To judge the goodness of our modified algorithm and assess the results, we investigate whether there are specific features or trends that are highly representative of only some of the groups. Thus, we proceed to computing a set of summary statistics for each LT, and calculate the average of these results over the LTs belonging to each different group. The features that we consider are:

- average and median USD traded,
- average and median time Δt in seconds between transactions,
- proportion of transactions done in "SS", "EXOTIC" or "ECOSYS" pools, and related entropy,
- proportion of transactions done in pools with a specific feeTier, and related entropy,



	avg_USD	median_USD	avg_dT	median_dT	ratio_SS	ratio_EXOTIC	ratio_ECOSYS	entropy_pools	ratio_100	ratio_500	ratio_3000	ratio_10000	entropy_fees	txns_close
0	2.4e+04	1.6e+04	1.2e+05	1.8e+04	0.066	0.59	0.34	0.74	0.054	0.37	0.55	0.022	0.7	0.28
1	7e+04	4.4e+04	6e+04	1.7e+04	0.012	0.55	0.44	0.52	0.011	0.33	0.61	0.046	0.59	0.25
2	5.1e+04	3.9e+04	9.3e+04	1.9e+04	0.15	0.043	0.81	0.46	0.11	0.76	0.11	0.022	0.61	0.25
3	1.2e+05	7.1e+04	1.2e+05	1.8e+04	0.19	0.1	0.71	0.67	0.14	0.65	0.15	0.054	0.84	0.23
4	2.6e+05	1.3e+05	3e+04	8.1e+03	0.0017	0.074	0.92	0.26	0.0014	0.63	0.34	0.02	0.71	0.24
5	7.6e+04	3.5e+04	1.8e+04	4.2e+03	0.1	0.11	0.79	0.61	0.059	0.79	0.12	0.026	0.65	0.25
6	9.2e+04	6e+04	1.2e+05	2.3e+04	0.061	0.034	0.9	0.3	0.052	0.78	0.16	0.012	0.59	0.23

(a) Average features for LTs in case A.



(b) t-SNE embedding visualisation of case A.

Fig. 10 Clustering of LTs for case A, i.e. over the six-months time window between January and June 2022. In **a**, each row represents one of the recovered clusters and columns are the different features computed to characterise species of LTs. The color-code employed applies to each column separately to be able to quickly identify the related smallest and biggest values in magnitude, and judge the general distribution. It is essential to always check the magnitudes of cells per se too, due to highly variable variance between columns. In **b**, the t-SNE plot of embeddings of LTs is reported with perplexity = 15 and points are color-coded according to their cluster of membership

- proportions of trades on days when the SP LargeCap Crypto Index[4] increased or decreased in value, or when the market was closed, due to weekends and bank holidays.

[4] https://www.spglobal.com/spdji/en/indices/digital-assets/sp-cryptocurrency-largecap-index/#overview.

The first two points above are inspired from the literature on clustering of LOB order flow, i.e. Cont et al. (2023), Cartea et al. (2023), and Wright (2022), which highlights the importance of volume and time of submissions to characterise types of agents active on centralised exchanges. The distinction between "SS", "EXOTIC" or "ECOSYS" pools is then inspired by the classification in Heimbach et al. (2021), where the authors introduce a notion of normal pools, stable pools and exotic pools. For them, stable pools exchange tokens that are both stablecoins. Normal pools trade instead tokens that are both recognised in the crypto ecosystem, while exotic pools deal with at least one token that is extremely volatile in price (e.g. YAM, MOON and KIMCHI). We slightly divert from this classification and define "SS" pools as pools whose tokens are both stablecoins, "ECOSYS" pools as pools that exchange only tokens that are either stablecoins or pegged to the most established BTC and ETH coins, and "EXOTIC" pools as the remaining ones. ECOSYS pools can be seen as the venues carrying the "safest" opportunity for profit for a novice crypto investor, since they trade volatile tokens though directly related to the most established blockchains that are the true foundations of the whole DeFi environment.

The average magnitude of features computed over the LTs belonging to each different cluster for case A, i.e. over January–June 2022, is reported in Fig. 10a. We focus on the groups found specifically for this period because it is the longest one and thus, it provides us the most general results and insights. Cases $B1/B2$ will be later described too, in order to assess the overall stability of recovered *species* of LTs and highlight any specific variations due to different sub-periods in time and related pools of relevance considered. The seven clusters of LTs found have sizes of 304/142/512/978/379/186/914 agents respectively, which means that we are able to find a well-balanced distribution of cluster sizes without any dominant group. Thanks to the heatmap in Fig. 10a, we also easily confirm that our methodology is able to extract different groups of LTs that have significant variation of behaviour with respect to the outer features defined. However, a few columns had to be dropped due to non-significance of their results. Importantly, we also recall that inner biases on ratios are present (e.g. when considering that our sub-universe does not have a uniform distribution of numbers of pools with specific feeTier), and thus we can expect more/less transactions of some type on average. For visualisation purposes, we also embed the 16-dimensional representations of LTs into a 2-dimensional view via t-SNE, and plot them with perplexity = 15 in Fig. 10b. LTs are colour-coded according to the cluster they belong to, and we indeed observe that different groups lie on different parts of the plane.

Focusing on Fig. 10a, one can draw the following high-level remarks.

- **Groups 0 and 1** have a strong focus on trading exotic cryptocurrencies. The former set of LTs mainly uses feeTier 3000 for the purpose, and shows slightly higher than average tendency to trade when the market is closed. The latter group uses significantly both the 3000 and 10,000 feeTiers, meaning that the related LTs are willing to accept also extremely high transaction costs. This behaviour could indicate that they have high confidence on their intentions and possibly urgency.
- On the other hand, **groups 2 and 3** trade stablecoins more than usual. The former cluster could point to an enhanced use of SS pools to take advantage of opti-

mised routing, while the latter has a non-negligible proportion of trades in exotic pools with feeTier 10,000. Likely, group 3 isolates a set of LTs that are interested in niche exotic tokens, which are only proposed in pools against stablecoins that do not overlap. Diverting funds between two of these exotic tokens requires an exchange between the two related stablecoins too, which motivates the recovered statistics. We also witness strong usage of the feeTier 100, which hints to traders trying to compensate the high costs suffered in pools with feeTier 10,000 by paying the lowest possible fees on the SS pools.

- **Groups 4 and 6** are more active than average on ECOSYS pools. The two groups differ noticeably from their opposite relative strength of USD traded and time between operations. Overall, group 6 trades less money and waits longer, mainly using pools with low feeTier 500. These features can be interpreted as characteristics of cautious retail traders that invest in less risky and highly well-known crypto possibilities. And indeed, we also find that this group is one of the largest in size. Then, group 4 also relates to ECOSYS pools. However, these users tend to trade more USD with higher frequency, and this is also the cluster with much higher than average proportion of LTs that also act as LPs ($\sim$ 16%). Therefore, we identify here a group of more professional investors.
- Finally, **group 5** shows a significant usage of all the three types of liquidity pools, but trades are concentrated in pools with cheap feeTier 500. These agents trade often, and indeed show the smallest median time between transactions. These eclectic, active and thrifty LTs are probably our group of smartest investors.

Our results confirm that the proposed algorithm is able to recognise variance in the data, and allow us to extract interesting insights into the behaviour of different types of species of LTs. In particular, we observe how the type of pools on which LTs are active plays a primary role in the definition of their trading behaviours. This is especially interesting since no full notion of tokens and feeTier is used in the generation of the embeddings. Indeed, only a unique label per pool is provided as input to our algorithm, e.g. USDC-WETH/500 could be pool “P1”, USDC-WETH/3000 pool “P2” and FXS-WETH/10000 pool “P3”. Thus, these pools would be considered equally different if no structural discernible pattern was recognised by the methodology, providing some further evidence of the strength of our proposed extension to *graph2vec*.

Stability analyses As already motivated, we now pursue the same analyses described above but for cases $B1/B2$. We cluster the n-dimensional embeddings for $n \in \{8, 16, 32, 64\}$ and compute the ARIs between each pair of resultant sets of LT groups. We confirm that at least a 16-dimensional embedding is required in order to have a stability of clusters in case $B1$, while only eight dimensions suffice for the case $B2$. For simplicity, we use the 16-dimensional representations consistently in all cases. We recover six groups of LTs in case $B1$, and seven in case $B2$. In both cases, we find two clusters with same characteristics as groups 4 and 6 of case A, i.e. traders mainly active on ECOSYS pools. We also recover the eclectic traders of group 5. Therefore, we observe several stable and persistent types of LTs. Small perturbations happen instead on the groups trading on SS or EXOTIC pools, as one could expect from the mere evolution of time and external market conditions, and consequently generation of different behaviours. In particular, all case A species, except group 1, are also found in case $B1$. On the other hand, case $B2$ shows less intensity on group 3, probably due to investors diversifying more during the crypto turmoils of the second quarter of 2022. Overall, we observe general agreement on the groups and main features recovered during cases $A/B1/B2$, and we can thus rely on our species of LTs found for the longest duration case A as descriptors of the ecosystem.

The above stability-related findings are of interest in themselves, first of all, since central banks started hiking interest rates in March 2022. This consequently stopped a strong influx of liquidity into the crypto ecosystem and accentuated a period of significant underperformance, that could have indeed weakened the stability of results. On top of that, the Terra-Luna crash happened in May 2022 and it could have in theory enhanced noise and instabilities especially in the structural clustering on case $B2$. As a very last remark, we notice that only $\sim 20\%$ addresses are present in all cases $A/B1/B2$. Therefore, we are either recovering similar behaviour but for different people, or in some cases it could be the same person simply employing a new wallet to better hide their trading behaviour.

We have mainly focused on the liquidity consumption component of the crypto ecosystem thus far. In the next step of our investigation, we shift the focus from LTs to pools. We first aim to perform a clustering of pools based on features built from simple statistics that consider both liquidity consumption and liquidity provision. This will allow us to assess whether the SS, ECOSYS and EXOTIC classification is beneficial for describing the crypto ecosystem or is only useful for LTs characterisation.

3.2 Clustering of pools

The above analyses revealed a characterisation of the main types of LTs structural trading behaviour. While the importance of different types of pools in the ecosystem seems to be also clear, we stress that a full understanding of liquidity pools goes beyond the mere liquidity consumption mechanism (i.e. it needs to further account for both liquidity provision and price evolution). Thus, we now pursue an intuitive initial investigation of the similarity of pools themselves, in order to gain additional insights on the entire ecosystem.

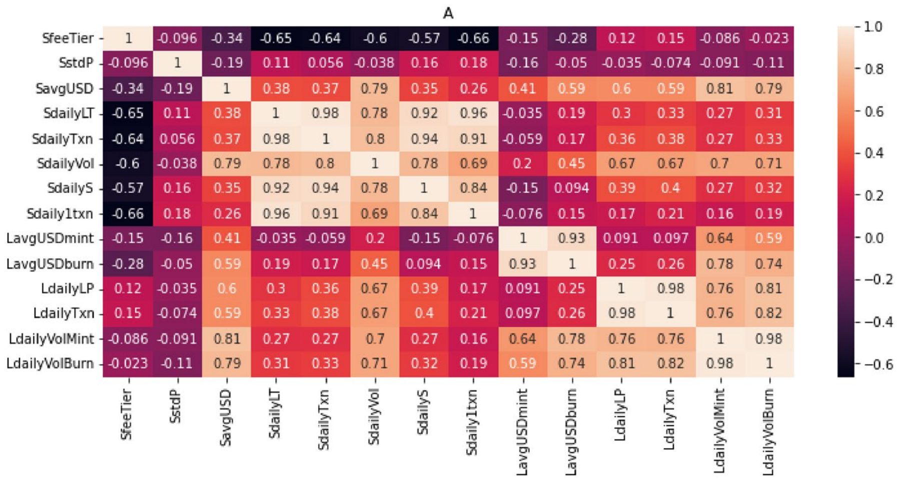
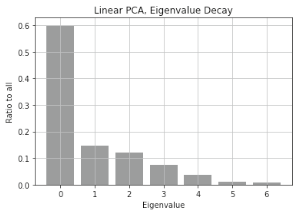
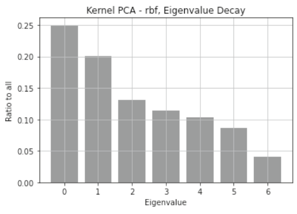


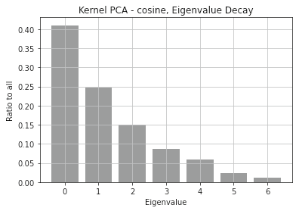
Fig. 11 Spearman correlation between the computed features for pools, with the addition of feeTier, for our case *A*



(a) Linear PCA.

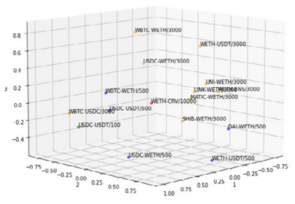


(b) rbf kernel PCA.

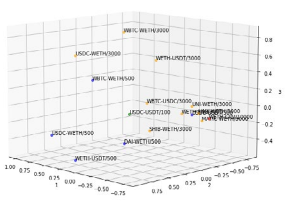


(c) Cosine kernel PCA.

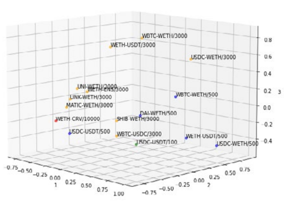
Fig. 12 Decay of eigenvalues for the first seven out of 13 eigenvalues for different PCA kernels



(a) Angle = 45 degrees.



(b) Angle = 135 degrees.



(c) Angle = 315 degrees.

Fig. 13 Projection on a 3D space of the vectors encoding different features of pools, from the application of PCA with cosine kernel. Views from different angles are reported for a better judgment of the results, and pools are color-coded according to their feeTier (green relates to feeTier 100, blue to 500, orange to 3000 and red to 10,000)

Methodology We focus on case *A*, as it covers the longest period in time. We consider the intersection of pools relevant for both LTs and LPs to properly account for both mechanisms, and find a resulting set of 16 pools. For each pool, we compute the following 13 features:

- average daily number of active LTs/LPs - “SdailyLT” and “LdailyLP” respectively,
- volatility of the execution price of the pool - “SstdP”,
- average size of swap/mint/burn operations in dollars - “SavgUSD”, “LavgUSDmint” and “LavgUSDburn”,
- average daily amount of dollars used in swap/mint/burn operations, i.e. volume - “SdailyVol”, “LdailyVolMint” and “LdailyVolBurn”,
- average daily number of LTs/LPs transactions - “SdailyTxn” and “LdailyTxn”,
- average daily number of different senders, i.e. smart contracts, called within swap transactions - “SdailyS”,
- number of agents with only one transaction normalised by the number of days considered - “Sdaily1txn”. This measure is computed to gauge the tendency of external smart investors to hide their behavior by creating several different wallets on the pool.

For the above features, we create related labels for ease of reference, which start with letter “S” if the quantity is computed from swap operations, or letter “L” if the quantity is computed from liquidity provision operations. In Fig. 11, we show the heatmap of Spearman correlations between the above attributes plus feeTier (“SfeeTier”) for our pools. There are significant positive correlations, especially among features developed from LT data and LP data, respectively. Thus, we standardise entries and employ linear PCA and kernel PCA (with both “rbf” and “cosine” kernel in the latter) to reduce the dimensionality of our data. The eigenvalue decay for all three mentioned cases is shown in Fig. 12, where only the first seven eigenvalues are depicted for clarity of visualisation. The cosine kernel PCA is seen to capture more variance in fewer dimensions, and thus we embed the data by projecting on its related first three components. The resulting 3D embedding is shown in Fig. 13 from three different angles, where we color-code pools according to their feeTier. In particular, green relates to feeTier 100, blue to 500, orange to 3000, and red to 10000.

Discussion From the projections shown in Fig. 13 and initial trials of clustering, it is clear that the division between SS, ECOSYS and EXOTIC pools does not hold when considering the full set of dynamics on pools (while it is indeed suitable in connection to LTs' behaviour specifically). Similarly, we do not witness strong proximity of pools with same feeTier. Liquidity consumption, provision and price evolution are all essential mechanisms to consider for a full description of the Uniswap ecosystem, and our intuition is that certain combinations of tokens and feeTiers are more similar and suitable for trading at different moments in time. LPs are more incentivised to enhance liquidity on pools with strong LTs activity, low volatility of the exchange rate to avoid predictable loss, and possibly high feeTier from which they indeed mainly profit. In parallel, LTs are more interested in pools with low fees but high volatility of the price of tokens in order to extract gains from trading opportunities, and high liquidity to diminish the market impact of their trades. Thus, different adjustments of these mechanisms can result in the proximity or not of our projections of pools, but a related systematic study is left for future work.

4 Conclusions

Despite blockchains and Decentralised Finance are quite recent concepts, they have quickly established themselves among the main topics of interest of both practitioners and academics. Nevertheless, a real comprehension of the characteristic dynamics within the related protocols is still far away. Our investigations aim at being a stepping stone towards a deeper understanding of the crypto ecosystem, and we achieve this task by empirically studying and characterising the Uniswap v3 DEX. We build a workflow to define the most relevant liquidity pools over time by assessing the inner features of pools along with their interconnectedness, and provide related lists of liquidity pools significant for six different windows in time, i.e. cases $A/B1/B2/C1/C2/C3$, that can be directly used for future research studies. We then focus on LTs and show the existence of seven "species" of traders with interpretable features. These clusters are recovered by assessing the equivalence of LTs structural trading behaviour on a set of highly interconnected pools, via a novel graph embedding approach developed for the purpose. These results suggest a connection between patterns in the features of traders' swap transactions in the broad Uniswap ecosystem, and specific types of pools on which these operations are indeed executed.

Future work Regarding future directions of research, there are two main threads we aim to pursue. The first one is a detailed investigation into the behaviour of LPs, along with the corresponding clustering of species, also leveraging on a strongly data-driven approach. Then, we plan to leverage the entire data on the Ethereum blockchain to track the flow of funds over multiple DEXs and active protocols. This will enable us to gain a better understanding of LTs, as we will then be able to approximate their profit and loss (PnL). Indeed, the crypto ecosystem is highly interconnected, and agents easily trade between different exchanges on the same blockchain, also with the possibility to enhance their positions via borrowing. In parallel, one could also investigate the "optimal routing problem" (Angeris et al., 2022) on the Ethereum blockchain, which is formulated as the problem of optimally executing an order involving multiple crypto assets on a network of multiple constant function market makers.

Appendix A: Sub-universes of pools for cases A/B1/B2/C1/C2/C3

The pools found most significant by the proposed methodology are here reported, for cases *A*/*B*1/*B*2/*C*1/*C*2/*C*3. We provide three different tables, i.e. Table 1 for case *A* first, followed by Table 2 for cases *B*1, *B*2, and concluding with Table 3 for cases *C*1, *C*2, *C*3. Each table lists pools relevant for liquidity consumption (LT data) and it further highlights the thresholds chosen to define the giant components as described in Sect. 2.

Table 1 Final set of pools to consider for case *A*, i.e. for analyses spanning the 6 months from January to June 2022

Case	Agent type	Thresholds	Final pools
A	LT	(2,000, 100, 800)	DAI-WETH/3000, CEL-WETH/3000, USDC-UOS/10000, DAI-USDC/100, SPELL-WETH/3000, WETH-CRV/10000, USDC-USDT/500, DAI-FRAX/500, WETH-BTRFLY/10000, GALA-WETH/3000, WETH-USDT/3000, WBTC-USDC/3000, DAI-USDT/500, UNI-WETH/3000, WETH-ENS/3000, DAI-USDC/500, WBTC-WETH/500, MATIC-WETH/3000, DAI-WETH/500, WETH-USDT/500, USDC-WETH/500, LINK-WETH/3000, WBTC-WETH/3000, FXS-WETH/10000, FRAX-USDC/500, USDC-WETH/3000, USDC-WETH/10000, LUSD-USDC/500, HEX-USDC/3000, USDC-NCR/500, SHIB-WETH/3000, DYDX-WETH/3000, USDC-USDT/100, HEX-WETH/3000
A	LP	(30, 3, NA)	

Thresholds are listed as (threshold for common origins, for common senders, and for bridges)

Table 2 Final set of pools to consider for cases $B1$, $B2$

Case	Agent type	Thresholds	Final pools
B1	LT	(1500, 60, 600)	DAI-WETH/500, FRAX-USDC/500, SPELL-WETH/3000, agEUR-USDC/500, USDC-USDT/100, LUSD-USDC/500, USDC-UST/100, WBTC-WETH/3000, DAI-USDC/100, USDC-NCR/500, WETH-USDT/500, XSGD-USDC/500, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, WETH-BTRFLY/10000, FEI-USDC/500, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, USDC-USDT/500, XSGD-WETH/500, WBTC-WETH/500, FXS-WETH/10000, GALA-WETH/3000, WBTC-USDC/3000, WETH-CRV/10000, WETH-USDT/3000, USDC-UOS/10000, HEX-WETH/3000, USDC-WETH/500, USDC-GF/3000
B1	LP	(20, 3, NA)	
B2	LT	(1500, 80, 600)	DAI-WETH/500, FRAX-USDC/500, FEI-USDC/100, USDC-USDT/100, UST-WETH/3000, HDRN-USDC/10000, USDC-STG/3000, CEL-WETH/3000, DAI-USDC/500, WBTC-WETH/3000, DAI-USDC/100, WETH-USDT/500, APE-WETH/3000, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, USDC-WETH/10000, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, USDC-USDT/500, APE-USDC/3000, WBTC-WETH/500, WBTC-USDC/3000, WETH-USDT/3000, WETH-LUNA/10000, USDC-UOS/10000, BUSD-USDC/500, HEX-WETH/3000, WETH-LOOKS/3000, SHIB-WETH/3000, USDC-WETH/500, DAI-FRAX/500
B2	LP	(15, 3, NA)	

Thresholds are listed as (threshold for common origins, for common senders, and for bridges)

Table 3 Final set of pools to consider for cases $C1$, $C2$, $C3$

Case	Agent type	Thresholds	Final pools
C1	LT	(1,000, 50, 400)	DAI-WETH/500, FRAX-USDC/500, SPELL-WETH/3000, agEUR-USDC/500, USDC-USDT/100, USDC-UST/100, WBTC-WETH/3000, DAI-USDC/100, USDC-NCR/500, WETH-USDT/500, XSGD-USDC/500, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, WETH-BTRFLY/10000, FEI-USDC/500, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, SOS-WETH/10000, XSGD-WETH/500, WBTC-WETH/500, FXS-WETH/10000, GALA-WETH/3000, WBTC-USDC/3000, WETH-CRV/10000, WETH-USDT/3000, HEX-WETH/3000, USDC-WETH/500
C1	LP	(10, 3, NA)	
C2	LT	(1,000, 50, 500)	DAI-WETH/500, FRAX-USDC/500, WETH-WRLD/10000, USDC-USDT/100, DAI-USDC/500, USDC-UST/100, WBTC-WETH/3000, DAI-USDC/100, USDC-NCR/500, WETH-USDT/500, XSGD-USDC/500, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, WETH-BTRFLY/10000, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, XSGD-WETH/500, WBTC-WETH/500, FXS-WETH/10000, GALA-WETH/3000, WBTC-USDC/3000, WETH-USDT/3000, HEX-WETH/3000, USDC-RSS3/3000, WETH-LOOKS/3000, SHIB-WETH/3000, USDC-WETH/500, DAI-FRAX/500
C2	LP	(10, 3, NA)	
C3	LT	(1,000, 75, 500)	DAI-WETH/500, FRAX-USDC/500, FEI-USDC/100, USDC-USDT/100, UST-WETH/3000, HDRN-USDC/10000, CEL-WETH/3000, WBTC-WETH/3000, DAI-USDC/100, WETH-USDT/500, UNI-WETH/3000, APE-WETH/3000, LINK-WETH/3000, USDC-WETH/3000, DAI-WETH/3000, HEX-USDC/3000, WETH-ENS/3000, MATIC-WETH/3000, USDC-USDT/500, APE-USDC/3000, WBTC-WETH/500, WBTC-USDC/3000, WETH-CRV/10000, WETH-USDT/3000, WETH-LUNA/10000, BUSD-USDC/500, HEX-WETH/3000, WETH-LOOKS/3000, SHIB-WETH/3000, USDC-WETH/500
C3	LP	(10, 3, NA)	

Thresholds are listed as (threshold for common origins, for common senders, and for bridges)

Acknowledgements We are grateful to Álvaro Cartea, Fayçal Drissi, Marcello Monga and Andrea Pizzoferrato for insightful discussions. Deborah Miori acknowledges financial support from the *EPSRC CDT in Mathematics of Random Systems* (EPSRC Grant EP/S023925/1).

Author contributions DM wrote the main manuscript text and run the analyses. All authors reviewed the manuscript.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Adams, H. (2020) Uniswap whitepaper. https://hackmd.io/@HaydenAdams/HJ9jLsfTz.

Adams, H., Zinsmeister, N., Salem, M., Keefer, R., & Robinson, D. (2021). Uniswap v3 core. Tech.rep., Uniswap, Tech. Rep.

Adams, H., Zinsmeister, N., & Robinson, D. (2020). Uniswap v2 core. https://uniswap.org/whitepaper.pdf.

Angeris, G., Evans, A., Chitra, T., & Boyd, S. (2022). Optimal routing for constant function market makers. In: *Proceedings of the 23rd ACM conference on economics and computation, EC'22* (pp. 115–128). New York, NY, USA, Association for Computing Machinery.

Angeris, G., Kao, H.-T., Chiang, R., Noyes, C., & Chitra, T. (2019). An analysis of Uniswap markets.

Berg, J. A., Fritsch, R., Heimbach, L., & Wattenhofer, R. (2022). An empirical study of market inefficiencies in Uniswap and SushiSwap. arXiv preprint arXiv:2203.07774.

Brogaard, J. (2010). High frequency trading and its impact on market quality. Northwestern University Kellogg School of Management Working Paper, 66.

Buterin, V. (2013). Ethereum white paper. *GitHub Repository, 1*, 22–23.

Cartea, Á., Drissi, F., Monga, M. (November 10, 2022). Decentralised Finance and Automated Market Making: Predictable Loss and Optimal Liquidity Provision. Available at SSRN: https://ssrn.com/abstract=4273989 or https://doi.org/10.2139/ssrn.4273989.

Cartea, Á., Drissi, F., & Monga, M. (2023). Decentralised finance and automated market making: Execution and speculation. arXiv preprint arXiv:2307.03499.

Cartea, Á., Cohen, S., Graumans, R., Labyad, S., Sánchez-Betancourt, L., & Veldhuijzen, L. (2023). Statistical predictions of trading strategies in electronic markets. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4442770.

Chen, H., & Koga, H. (2019). GL2vec: Graph embedding enriched by line graphs with edge features. In *International conference on neural information processing*.

Cont, R., Cucuringu, M., Glukhov, V., & Prenzel, F. (2023). Analysis and modeling of client order flow in limit order markets. *Quantitative Finance, 23*(2), 187–205.

Fan, Z., Marmolejo-Cossío, F. J., Altschuler, B., Sun, H., Wang, X., & Parkes, D. (2022, November). Differential Liquidity Provision in Uniswap v3 and Implications for Contract Design✱. In *Proceedings of the Third ACM International Conference on AI in Finance* (pp. 9–17). ISO 690.

Foucault, T., Hombert, J., & Ioanid, R. (2016). News trading and speed. *The Journal of Finance, 71*(1), 335–381.

Foucault, T., Kozhan, R., & Tham, W. W. (2016). Toxic Arbitrage. *The Review of Financial Studies, 30*(4), 1053–1094, 12.

Frtisch, R., Käser, S., & Wattenhofer, R. (2022, September). The economics of automated market makers. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies* (pp. 102–110).

Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 855–864).

Heimbach, L., Schertenleib, E., & Wattenhofer, R. (2022). Risks and returns of Uniswap V3 liquidity providers. arXiv preprint arXiv:2205.08904.

Heimbach, L., Wang, Y., & Wattenhofer, R. (2021). Behavior of liquidity providers in decentralized exchanges. arXiv preprint arXiv:2105.13822.

Ho, T. S. Y., & Stoll, H. R. (1983). The dynamics of dealer markets under competition. *The Journal of Finance, 38*(4), 1053–1074.

Hubert, L. J., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

Inzirillo, H., & De Quenetain, S. (2022). Managing Risk in DeFi Portfolios Available at SSRN 4228899.

Kirilenko, A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance, 72*(3), 967–998.

Kitzler, S., Victor, F., Saggese, P., & Haslhofer, B. (2023). Disentangling decentralized finance (DeFi) compositions. *ACM Transactions on the Web, 17*(2), 1–26.

Le, Q., & Mikolov, T. (2014, June). Distributed representations of sentences and documents. In: *International conference on machine learning* (pp. 1188–1196). PMLR.

Makarov, I., & Schoar, A. (2022). Cryptocurrencies and decentralized finance (DeFi). Working paper 30006, National Bureau of Economic Research, April 2022.

Mankad, S., Michailidis, G., & Kirilenko, A. (2011). Discovering the ecosystem of an electronic financial market with a dynamic machine-learning method. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1787577

Megarbane, N., Saliba, P., Lehalle, C.-A., & Rosenbaum, M. (2017). The behaviour of high-frequency traders under different market stress scenarios. *Mathematics eJournal, 3*, 1850005.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Miori, D., & Cucuringu, M. (2023). DeFi: Modeling and forecasting trading volume on Uniswap v3 liquidity pools. *SSRN*. https://doi.org/10.2139/ssrn.4445351

Nakamoto, S. (2009). Bitcoin. A peer-to-peer electronic cash system, 21260.

Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., & Jaiswal, S. (2017). graph2vec: Learning distributed representations of graphs. arXiv preprint arXiv:1707.05005.

Pierro, G. A., & Rocha, H. (2019). The influence factors on Ethereum transaction fees. In: *2019 IEEE/ACM 2nd international workshop on emerging trends in software engineering for blockchain (WETSEB)* (pp. 24–31).

Schär, F. (2020). Decentralized finance: On blockchain- and smart contract-based financial markets. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3571335

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., & Borgwardt, K. M. (2011). Weisfeiler–Lehman graph kernels. *Journal of Machine Learning Research, 12*(77), 2539–2561.

Wright, I. D., Reimherr, M., & Liechty, J. (2022). A machine learning approach to classification for traders in financial markets. *Stat*. https://doi.org/10.1002/sta4.465