Meme Coin Factories: Uncovering Large-Scale Manipulations on pump.fun

Nicolas Szwajcok*
nicolas@szwajcok.com
Carnegie Mellon University,
EPFL, ETH Zürich
Lausanne, Switzerland

Taro Tsuchiya*
ttsuchiy@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Enze Liu
hsyalexliu0809@gmail.com
Carnegie Mellon University
Pittsburgh, PA, USA

Kyle Soska
ksoska@alumni.cmu.edu
Fullstack
New York City, NY, USA

Mathias Payer
mathias.payer@nebelwelt.net
EPFL
Lausanne, Switzerland

Nicolas Christin
nicolasc@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

Abstract

Once complex, creating and deploying a new cryptocurrency has become trivial. *Coin launchpads* now allow users to generate a new coin with merely a few clicks, at a minimal cost. Launchpad popularity has grown in tandem with the rise of "meme coins," which usually do not offer any novel technological properties and are purely created for fun. The most prominent coin launchpad, pump.fun, has gained significant traction, grossing over 100 million USD in daily trading volume. The mass adoption of coin launchpads, however, also enables strategic actors to easily manipulate trading signals, unbeknownst to inexperienced traders who then buy certain coins, and enable these strategic actors to profit from rapid and unsustainable price increases ("pumps"). To identify such manipulations at scale, we conduct a large-scale study of pump.fun, collecting information on all 15 million coins launched in the last two years, and performing analysis on large, random samples of transaction data. We identify five classes of manipulation strategies: 1) wash trading, 2) creator address obfuscation, 3) coordinated sell, 4) copycat coins, and 5) social media manipulation. We find that strategic actors often bypass the platform interface and implement these strategies in a highly automated and low-latency fashion, by interacting directly with the blockchain. We further uncover the existence of "Market-Manipulation-as-a-service (MMaaS)," third-party tools that enable users to perform these manipulations without any technical expertise. We conclude by devising mitigations and proposing recommendations for traders, pump.fun, wallets or chain scanners, software development platforms, and regulators.

CCS Concepts

• **Security and privacy → Social aspects of security and privacy**; • **General and reference → Measurement**.

*Joint first authors with equal contribution, ordered alphabetically.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
CCS '26, The Hague, Netherlands
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846641

Keywords

Market Manipulation; Meme coin; Blockchain; Social Media

ACM Reference Format:
Nicolas Szwajcok, Taro Tsuchiya, Enze Liu, Kyle Soska, Mathias Payer, and Nicolas Christin. 2026. Meme Coin Factories: Uncovering Large-Scale Manipulations on pump.fun. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3830454.3846641

1 Introduction

When Bitcoin rose to prominence in the early 2010s, competitors, or "alt-coins," started to appear. At the time, creating an alt-coin required technical expertise to develop and deploy a new blockchain, or "forking" an existing one. Subsequently, the rise of programmable money, in particular through Ethereum's smart contracts, made alt-coin creation significantly less onerous. Ethereum specifies a well-defined ERC-20 standard to allow users to create "tokens" on an existing blockchain as a substrate. The mid-2020s made token creation even easier, with the introduction of online platforms and mobile apps—called *token launchpads*—making the generation of a new token literally as simple as pushing a button on a website. Token launchpads often operate on the Solana blockchain, which supports low transaction fees and high throughput, making coin creation nearly zero-cost. These tokens are often referred to as "meme coins.[1]" They generally do not offer any novel technological properties, and are instead mostly designed to be humorous [49].

pump.fun has been the most prominent token launchpad, attracting over 15 million new coins and totaling nearly 90 billion USD in trading volume [4, 24]. For instance, the most popular coin, "Fartcoin," reached a market capitalization of 2.5 billion USD at its peak. Importantly, pump.fun is extremely user-friendly. To create a coin, users only need to provide a coin name, symbol, coin image, and pump.fun automatically handles the technical details such as instantiating tokens on the Solana blockchain and generating a trading pool. This ease of use, combined with potential financial gains, attracts users with no technical expertise, potentially including teenagers [6]. Additionally, users only need to connect their Solana

[1]By a slight abuse of terminology, we will indistinctly refer to these as *tokens* or as *coins*. Technically, a coin requires to operate its own blockchain, and we should denote them as "meme tokens," but "meme coin" is the more common turn of phrase.

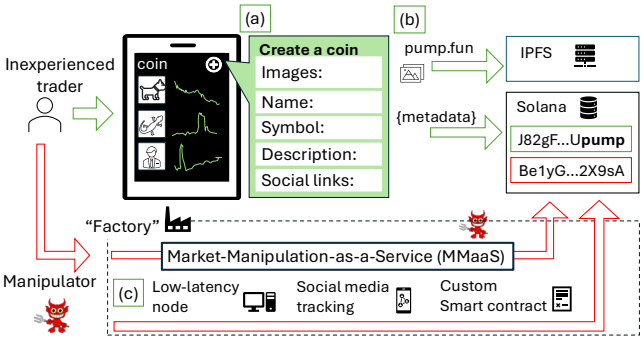


Figure 1: How to create a coin on `pump.fun` (both for inexperienced traders and strategic actors)

address and can remain pseudonymous. No identity verification or any other information—not even an email address—is required.

The user-friendliness and openness, however, create opportunities for *strategic actors* to take advantage of inexperienced traders. In particular, strategic actors often manipulate trading signals (e.g., trading volume, ownership, coin names, social media comments, and posts) to trick inexperienced traders into buying certain coins and profit from the resulting price increase. Inexperienced traders often fail to detect such highly automated, low-latency manipulations, which are buried in massive trading volume.

This study aims to detect market manipulation on `pump.fun` at scale and uncover manipulation strategies and infrastructure. We focus on `pump.fun`, because its trading volume far surpasses that of all its competitors (e.g., BONKfun, Raydium LaunchLab), which together feature only about 1 million coins and a combined market cap of 7 billion USD [5], compared to 15.2 million coins on `pump.fun` alone. We track *all* 15.2 million `pump.fun` coins on the Solana blockchain and collect their metadata (e.g., coin description, image URLs, social media links) during the first two years of `pump.fun`'s existence (Jan. 14, 2024 – Jan. 14, 2026). For §4-6, we collect the full transaction history of a random **1% coin sample** (as the primary sample) and all coins launched on a **5-day sample** (for consistency checks), totaling over 87 million transactions.

We identify five classes of manipulation strategies:

1. Wash trading. Manipulators can perform buy-sell transactions to inflate trading volumes ("wash trading"). We identify a lower-bound of 4 million wash trading transactions (2.2 million for the 1% coin sample, 1.8 million for the 5-day sample), representing 17% of all *trading* transactions. We show that wash trading can potentially increase a coin success probability (i.e., "graduation" defined in §2.1) up to 10 times.

2. Creator address obfuscation. Manipulators can use a different blockchain address to create coins, potentially hiding their coin ownership. We develop a method to group coin creator addresses based on funding patterns (e.g., which address funds a coin creator address). We show that the top-1% groups create 58.6% of all coins.

3. Coordinated sell. Manipulators can use multiple blockchain addresses to buy tokens and consolidate them into a single address, which then sells them all at once ("dump"). Using conservative criteria, we identify nearly 8,000 instances (4,402 for the 1% coin sample and 3,510 for the five-day sample) of such manipulations.

4. Copycat coins. Manipulators often deploy "copycat" coins, replicas of the existing coins, to lure uninformed traders. We detect at least 1.5 million (over 10% of all) coins that duplicate the name, symbol, description, and images of original coins. We find that manipulators often automate copycat creation (e.g., through a custom smart contract), while the original coins are much more successful.

5. Social media manipulation. We identify and analyze social media manipulation, including bots posting promotional comments on a coin profile pages, in Telegram groups, and in Twitter communities that consistently generate hundreds of coins. We also show that 3.5 million (23.5% of all) coins are created after Twitter or Truth Social posts. Among these, we observe that 31 posts *each* helped their respective coin creators earn *at least* 1 million USD.

Across all these strategies, we find evidence of sophistication. As shown in Figure 1, strategic actors often do not rely on the `pump.fun` web interface, but instead directly interact with the Solana blockchain, most likely using the custom programs (e.g., smart contracts) and low-latency blockchain nodes: "Meme Coin Factories." We also show that some of these strategies are effective at raising prices and extracting financial value.

As conducting such manipulations requires technical expertise and infrastructure, we further identify the existence of "Market-Manipulation-as-a-service (MMaaS)," websites or software tools that help commoditize such manipulations at scale. Our qualitative analysis reveals that MMaaS tools advertise the ability to create multiple blockchain addresses, copy tokens with low latency, generate comments using custom templates or AI, and deploy anti-detection tools (e.g., CAPTCHA solvers or proxy-rotation).

Compared to contemporary studies, our core contribution is twofold (see §2.4 for more details). First, we analyze known manipulation strategies (e.g., wash trading, dump) on a substantially larger scale (and using uniform samples) than has been studied before, thereby illuminating the pervasiveness of these activities. Second, we identify novel manipulative strategies (e.g., social media *post* manipulation, MMaaS). Based on these additional insights, we propose specific 1) improvements to trade signals (e.g., for users, `pump.fun`, wallets, and chain scanners) and 2) interventions (e.g., for financial regulators and software development platforms). Due to space constraints, the extended version of this paper is available at https://arxiv.org/abs/2609.10246 [48].

2 Background

We introduce `pump.fun`, describe our threat model, and highlight our contributions compared to related work.

2.1 `pump.fun`

Figure 1 illustrates the coin creation process on `pump.fun`. When users create a coin on `pump.fun`, they provide a coin name, symbol, and image. They can optionally provide a coin description and links to external platforms, e.g., Twitter or Telegram (Figure 1 (a)). After `pump.fun` receives this information, it uploads a coin image to the IPFS distributed file sharing service, records this information on the Solana blockchain (in "programs" similar to smart contracts in Ethereum), and generates a coin address ending with "pump" (Figure 1 (b)). `pump.fun` keeps track of the coin information through

the pump.fun's Token Mint Authority address.[2] Simultaneously, pump.fun creates a trading pool called "bonding curve" and inserts 1 billion coin tokens into it. Users send SOL (i.e., the native Solana cryptocurrency) to the pool to buy tokens, and can, conversely, redeem SOL from the pool to sell tokens. After the first trade, the trading pool uses an automated market maker (AMM) algorithm to determine the price based on the amount of SOL and the coin tokens in the pool. Sophisticated users can bypass the pump.fun web interface and perform the above actions directly on Solana through custom smart contracts (Figure 1 (c)).

When a bonding curve retains about 85 SOL, pump.fun moves the coin to an external decentralized exchange ("DEX," such as Raydium and PumpSwap) and stops trading on the launchpad [3, 43]. pump.fun calls this process "*graduation*" and considers it a success indicator that attracts greater public visibility. Unlike the bonding curve, a DEX has liquidity providers (who deposit both SOL and tokens to earn rewards). In our sample, 1.02% of coins graduate.

2.2 Threat model

We define a manipulator as a strategic actor who engages in the following process. Manipulators first fabricate signals of coin demand, such as trading signals (e.g., volume) or social signals (e.g., comments), or impersonate existing coins. The goal is to get a (set of) target coin(s) to the top of the pump.fun webpage to advertise it [26, 35]. This bump leads inexperienced traders to buy the target coin(s). The manipulator can then profit from this demand by selling previously accumulated target coin(s) at higher prices.

Manipulators can interact directly with the Solana blockchain via custom smart contracts, rather than manually using the pump.fun web interface. For instance, they can bundle many trades *within* a single transaction. We also assume that manipulators can create many Solana addresses, access low-latency Solana nodes to execute trades faster, and track social media posts in real time using platform APIs. We do not assume that manipulators control the blockchain network, compromise users' private keys, or control the pump.fun platform itself. Manipulators follow existing protocol rules and perform standard Solana transactions.

On the other hand, we define an "inexperienced" trader as a user who primarily uses the pump.fun's web interface to create or trade coins. They can use wallet services and blockchain explorers to gather coin information (e.g., owner holdings) or browse social media for related posts. However, we do not assume that inexperienced traders possess sufficient technical expertise to interact with or analyze data on the Solana blockchain, or to programmatically retrieve information from social media (e.g., via APIs).

2.3 Manipulation strategies

This section briefly introduces the five manipulative strategies and MMaaS, and surveys the existing literature on each strategy.

1. Wash trading. Wash trading is a manipulation to generate artificial demand by selling and buying assets to oneself, while leaving the asset balance unchanged [8, 31, 42]. Manipulators can also use multiple accounts to obfuscate their activity while keeping their overall balance unchanged.

[2]TSLvdd1pWpHVjahSpsvCXUbgwsL3JAcvokwaKt1eokM

2. Creator address obfuscation. Traders often consider a coin unsustainable if a coin creator or a single user owns a disproportionately large share of the circulating supply. Manipulators can conceal this by using a different blockchain address to create a coin.

3. Coordinated sell. Pump-and-dump is a trading scheme in which a group of traders coordinates to buy a certain asset, often advertising on social media, and then sells it at a higher price for profit [32]. In particular, manipulators can create multiple accounts to buy tokens, transfer them to a single account, and sell them at once.

4. Copycat coins. Manipulators often impersonate an authentic asset and create counterfeit ones (e.g., a slightly different symbol) and trick victims into buying them for profit [7, 18, 60]. Most blockchains (including Solana) have no restrictions on naming coins; users can copy-paste the existing coin information (e.g., name, symbol, description, image) to create a replica, which we call "copycat" coins. On pump.fun, a manipulator, as a coin creator, receives shares from trades victims make (i.e., a coin creator fee) and profits from price increases in copycat coins, offset by coin-creation fees.

5. Social media manipulation. Users often seek social signals (e.g., comments about coins) to look for a potential price increase [38, 41, 51, 61]. Manipulators can deploy automated bots to post comments on the coin's profile page on pump.fun, organize social media groups and posts to create a false sense of market excitement. Furthermore, social media influencers can create a Twitter post and a coin simultaneously to extract profits from the price increase.

MMaaS. Performing market manipulation is daunting for non-tech-savvy users because it requires technical expertise (e.g., creating blockchain addresses and smart contracts) and access to lower-latency blockchain nodes to achieve better profits. We observe a surge in trading apps, websites, and software tools that offer a user-friendly interface for executing trades on behalf of users, typically charging per-trade fees (e.g., 1%). Alarmingly, some apps could facilitate various forms of market manipulations. The US SEC describes such services as "Market-Manipulation-as-a-Service" (MMaaS, [45]); one court case already exists [55]. This phenomenon mirrors broader cybercrime-as-a-service, including spam-as-a-service [50], ransomware-as-a-service [10], DDoS-for-hire services [59], and front-running-as-a-service [15].

2.4 Related work

There exists an extensive line of work on pump-and-dumps [12, 20, 25, 28, 30, 32, 57], wash trading [9, 14, 39], and counterfeit tokens [7, 18, 19, 53, 60, 62], both in traditional and cryptocurrency markets. Most of these studies focus on centralized exchanges and rely on aggregated market data, such as coin prices and volumes, to infer manipulations. With the rise of high-throughput, low-fee blockchains, strategic actors began to execute their manipulations on the blockchain directly. A handful of studies started analyzing manipulation at account-level [14, 17, 58]. We further identify the use of *multiple* trading accounts within a single manipulation (in §6), which is challenging due to data availability and complexity.

While there also exist a few studies of meme coins [33–35, 39], the work closest to ours is Ding et al. [13], who study wash trading, owner obfuscation, and comment-generating bots on pump.fun and investigate their impact on coin outcome (e.g., price change).

Our work differs from previous efforts as follows. First, our paper draws conclusions from uniformly sampled and significantly larger datasets. We analyze metadata for *all* 15.2 million coins and the full transaction data for over 300,000 coins, whereas Ding et al. [13] analyze only about 6,000 coins (i.e., 0.04% of all coins) before and after the TRUMP coin. Our copycat coin analysis (1.5 million) is also on a much larger scale compared to other counterfeit token studies mentioned above. Second, we introduce more granular detection methods and analyses. For instance, Ding et al. [13] identified wash trading *accounts* that repeatedly buy and sell tokens (by setting thresholds). Instead, our work directly identifies wash trading *events*. This approach allows us to infer manipulators' intent and capability better and produces fewer false positives (i.e., WT1, §4). Last, our paper is the first to analyze creator obfuscation (using different addresses to create coins), social media *post* manipulation (making a post that creates new coins), and MMaaS.

3 Datasets

We collect both on-chain and external data related to coins created on pump.fun. We first describe our on-chain data collection and then outline the external data sources we use to enrich the analysis.

3.1 On-chain Data

Our on-chain dataset captures the creation and trading of coins launched on pump.fun. We obtain this data from three third-party providers: Chainstack, Syndica, and Arkham.

3.1.1 Coin addresses and creators. We begin by identifying all coins created on pump.fun in its first two years of existence (Jan. 14, 2024–Jan. 14, 2026). Conveniently, pump.fun uses a single address, the pump.fun Token Mint Authority, to track coins (see §2.1). We retrieve all transactions involving this address and filter for transactions that are parsable and successfully create new coins.

In total, we obtain 15,245,966 coins. This dataset spans 730 days and 151,780,482 Solana blockchain slots (241,768,575 to 393,549,057). Figure 2 illustrates the distribution of coins created over time. pump.fun has remained active during the data collection period, with a median of 17,926 new coins per day and a peak of 71,735 new coins on a single day.

We extract the *creator address* for each coin, defined as the address that pays the transaction fees for the first transaction. We also capture the *funding addresses* of each creator address. This set of addresses consists of the *funder address*, that is, the first address to send any SOL or any tokens to the creator; the parent of the funder address (the first address to have sent anything to the funder); and its own parent. In other words, we trace the funding addresses up to three hops (addresses) backward. We empirically decided on a three-hop threshold, based on saturation in new addresses at that level (see details in §5). In total, we obtain 5,826,346 distinct creator addresses and their associated funding addresses.

3.1.2 Sampled coin transactions. Due to the extremely large volume of coins and transactions, it is cost-prohibitive to retrieve and analyze all transactions for all coins. We therefore randomly sample 1% of coins out of all coins created (**1% coin sample**) to reduce sampling bias. We also create a second sample that includes coins created on five selected days (**5-day sample**) to demonstrate that

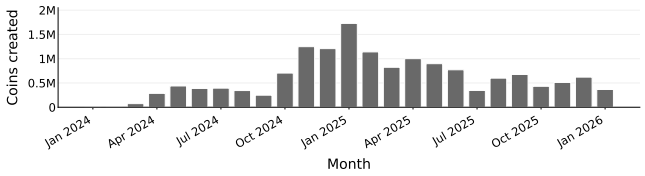


Figure 2: Nr. of pump.fun coins for each month.

trends are consistent across samples. We exclude any coin, in either sample, that features more than 5 million transactions, as including those would triple the size of our database. For the remaining coins, we retrieve all transactions involving these coins and conduct a detailed analysis in §4 and §6.

1% coin sample. The first sample is built from a 1% random coin sample. After excluding 5 coins with more than 5 million transactions, the sample contains 152,171 coins and 49,970,921 transactions.

5-day sample. The second dataset includes coins created on five days: Jan. 19, 2025, Aug. 1, 2025, Oct. 22, 2024, Mar. 13, 2025, and Sep. 3, 2025. Jan. 19, 2025 and Aug. 1, 2025 had exceptionally high and low coin creation activity, while Oct. 22, 2024, Mar. 13, 2025, and Sep. 3, 2025 were chosen at random. After excluding 10 coins, the sample contains 149,028 coins and 37,350,061 transactions.

3.2 External Data

We augment the on-chain data with external data sources, including address labels using the Arkham API [1], and data from the pump.fun API [2]. We followed the Terms of Use of pump.fun and maintained 4–5 second intervals between requests.

3.2.1 Address labels. Arkham labels—applied to creator and funding addresses—contain useful information, such as whether the address belongs to an exchange. These labels allow us to identify the types of entities funding the coin creators, aid our analysis on funding relationships, and address obfuscation (§5).

3.2.2 Data from pump.fun. Data from pump.fun consists of coin metadata, user comments, and historical price data.

Coin metadata. We collect the following metadata for all 15,245,966 coins in our on-chain dataset: 1) basic information, such as coin name, symbol, description, and creation timestamp, 2) metrics maintained by pump.fun, such as whether the coin has graduated and the number of comments in the coin profile page, and 3) social media links (e.g., Twitter), and links to the coin's profile images. In total, we retrieve metadata for 15,183,009 coins (99.59%).

User comments for selected coins. We collect comments that users post on each coin's profile page on pump.fun to analyze potential social signal manipulations. Due to rate limits, we only collect comments for coins in the **1% coin sample** and filter out coins with no comments. 58,715 (out of 152,171) coins have at least one comment, totaling 694,483 comments.

Historical price data for graduated coins. We retrieve historical daily price data (open, close, high, low) for all graduated coins to estimate potential profits from manipulations. We only gather price data for graduated coins because non-graduated coins are economically less significant. In total, we have price data for 154,513 graduated coins, with 57,623,422 daily price data points.

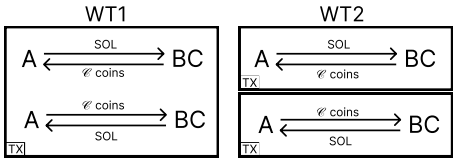


Figure 3: Wash trading heuristics representation.

4 Wash trading

In this section, we describe two heuristics for wash trading detection: one provides a conservative lower bound, while the other is more inclusive. We show that even under the conservative heuristic, a considerable share of coins exhibit wash-trading activity. Moreover, we find that, on average, for coins with more than 10,000 transactions, wash trading accounts for the majority of their transactions. Last, we evidence a strong correlation between wash trading activities and coin graduation.

4.1 Method

We define two heuristics for wash trading detection, denoted as **WT1** and **WT2**. WT1 provides a conservative lower bound on wash trading prevalence, while WT2 captures a broader set of transactions that may be consistent with wash trading.

WT1: As shown in Figure 3, we flag transactions in which address A buys and sells the same amount C **within a single transaction** against the same bonding curve BC (i.e., the same coin). This heuristic captures transactions that meet the definition of wash trading (§2.3), namely those in which the same entity buys and sells atomically. We believe WT1 introduces an extremely small number of false positives since WT1 does not label MEV transactions [11, 44] as wash trades; arbitrage or trading for another token is not possible within pump.fun's bonding curve. Notably, WT1 only captures transactions executed outside the pump.fun user interface (e.g., via custom smart contracts), since the pump.fun website does not allow users to buy and sell within the same transaction, reflecting the manipulators' technical sophistication.

WT2: As shown in Figure 3, we flag transactions in which the same address buys and sells the same coin amount at *approximately* the same price within a finite time window (whose length we will vary), regardless of whether these trades happen within a single transaction. As detailed in the extended version of this paper [48, Appendix D], "approximate" accounts for the trading fees. This heuristic relaxes the atomicity requirement of WT1. It captures the case where manipulators split the buy and sell trades across multiple consecutive transactions, with no other trade in between. However, it can also include false positives, such as when an address buys a coin with the intent to profit, but (almost) no one else buys, so the address sells the coin back.

4.2 Results

Figure 4 represents the number of wash-trading transactions (left) and the wash trading volume (right) for the **1% coin sample**. The x-axis represents the time lag between a manipulator's buy and sell. The green bars show transactions flagged by WT1 (i.e., no change across the x-axis), while the blue bars show those flagged by WT2 but not by WT1. We present only the results for the **1% coin sample**, as they are similar to those for the **5-day sample**. We report the full results for the **5-day sample** in the full version of this paper [48, Appendix E]. We see that increasing the buy-to-sell separation window (the x-axis) does not significantly capture more transactions.

Table 1: Nr. of total transactions and wash-trading ratio.

No. of txs	Wash trading ratio (%)	Nr. of coins
1–10	0.41%	58,084
11–100	2.69%	75,023
101–1,000	13.91%	15,905
1,001–10,000	21.65%	2,295
10,001+	50.31%	27

Under WT1, we find 2,221,734 wash-trading transactions across 8.26% of coins (i.e., coins with at least one wash-trading transaction). Under WT2 and a buy-to-sell time window of 1 second, we find 18,730 coins with wash trading instances, i.e., 12.38% of coins. Under WT2 and a 5-second time window, we find 2,356,189 wash-trading instances, corresponding to 44.05% of coins. If we further increase the window size, the percentage of coins flagged by WT2 increases dramatically, reaching 88.5% with a 1-day window. With WT2, while the number of wash trading candidate transactions does not grow significantly with the buy-to-sell time window, the number of affected coins, on the other hand, ramps up significantly, suggesting the presence of false positives.

We also quantify wash trading volume (in SOL). WT1 accounts for 66,636 SOL (1.32% of total trading volume and a median of 1.00% for each coin's volume). The wash trading volume is significantly smaller than the wash trading count. Manipulators appear to prioritize transaction count over volume to reduce the transaction fees since pump.fun charges fees based on the trading volume.

We next analyze wash-traded coins. We present two major results: 1) the proportion of wash-trading transactions is strongly correlated with the total number of transactions involving a given coin, and 2) the number of wash-trading transactions is strongly correlated with higher graduation rates. For the rest of this analysis, we use the (conservative) results from WT1.

Wash trading ratio per coin. In Table 1, we report the "wash-trading ratio," the share of wash-trading transactions among all transactions. While the majority of wash-traded coins feature 100 or fewer transactions, we find that as the number of transactions for a given coin increases, the proportion of wash-trading transactions also increases strongly. For instance, the wash-trading ratio is 21.65% for coins with 1,000 to 10,000 transactions (n =2,295) and 50.31% for coins with more than 10,000 transactions (n =27).

Graduation outcomes. We investigate how wash-trading transactions affect coin graduation. We hypothesize that coins with many wash-trading transactions are more likely to graduate because they may appear more often on the top page of the pump.fun web interface [26, 35]. The graduation rate for the wash-traded coins (by WT1) is 2.0% while that for the non-wash traded coins is 0.90%. Table 2 further illustrates that, when the number of wash-trading transactions is large, the graduation rates get higher. To statistically

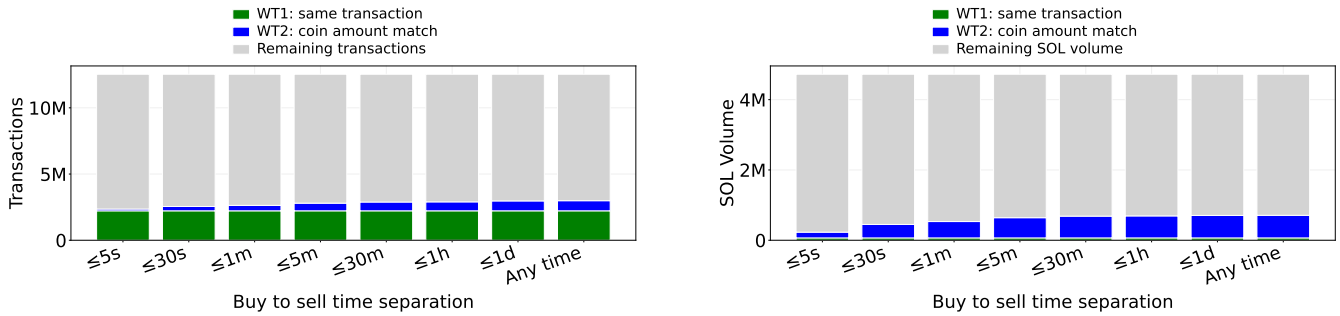


Figure 4: Wash trading transaction count (left) and volume in SOL (right) under WT1 and WT2, across time thresholds (the x-axis), for the 1% coin sample.

Table 2: Graduation rates by the number of wash-trading transactions.

Nr. of wash trading txs	Grad rate (%)	Nr. of coins
0	0.90%	138,839
1–5	0.50%	2,618
6–25	0.65%	3,873
26–100	1.12%	2,582
101–500	3.61%	2,437
500+	9.64%	985

examine the relationship between these two variables, we perform a logistic regression. For coins with substantial wash-trading activity, when the number of wash-trading transactions doubles, the graduation odds increase by around 19%. This relation is statistically significant (p-value: 3×10^{-59}).

Advanced atomic wash trading. We looked for more advanced wash-trading transactions to identify potential false negatives in WT1. Following Victor and Weintraud [58], we searched for transactions in which an address A buys coins, sends them to an address B, and B sells them. We did not find such transactions. While we do not capture all advanced forms of wash trading (i.e., buying coins, splitting them, and selling them separately), manipulators appear to wash trade often naively (WT1).

5 Creator Address Obfuscation

This section investigates creator address concentration and obfuscation.

5.1 Method

We cluster coin creator addresses using a "common funder" heuristic, following prior work [37, 56]. Intuitively, addresses controlled by the same entity are often funded by a shared upstream address.

We form a directed funding graph in which each coin-creating address A has a directed edge from its funder $F(A)$—the address that sends the first SOL or token transfer to A. Our approach yields a graph where each node has at most one incoming edge. To avoid spurious aggregation, we remove any node and its associated edges from this graph if Arkham's API labels the node as a third-party service (e.g., exchanges, bridges, smart contracts).

We then construct clusters by traversing the funding graph starting from coin-creator addresses up. For each such address A, we recursively follow the funder relation $F(A)$ to identify a chain of funders. We denote a cluster linking coin creators A to their funders $F(A)$ a 1-hop cluster. Similarly, we define 2-hop clusters as clusters linking coin creators to their funders' funders, i.e., $F(F(A))$, and 3-hop clusters by $F(F(F(A)))$. We stop at 3-hop as we observe minimal coverage increase when going from 2-hop to 3-hop, in this paper's extended version [48, Appendix F].

While Arkham claims that their verified labels have 98% accuracy [22], our approach erroneously links two unrelated addresses if Arkham misses labeling an address belonging to third-party services. This problem tends to exacerbate as we increase the funding hops [56]. We therefore check whether the existence of a single address that would dominantly fund the rest of the addresses skews our results. We perform an "adversarial leave-one-address-out" analysis, where we remove each address in the cluster one by one and check how much of the cluster's connectivity remains after the removal. We then identify the address that breaks the cluster the most (i.e., the worst-case deletion). We use *cluster connectivity retained* to assess the cluster connectivity after the worst-case deletion. It is defined as the share of addresses that belong to the largest component (i.e., the connected part).

5.2 Results

We apply the approach described in §5.1 to cluster 5,826,346 unique coin creator addresses.

Address concentration and obfuscation. Out of 5,826,346 unique coin creator addresses, over half of them are assigned to a multi-address cluster (i.e., we manage to find the common funder). Address-level counts may substantially overestimate the number of distinct coin creators. We find 322,620 multi-address clusters for 1-hop clustering, 295,978 for 2-hop clustering, and 290,319 for 3-hop clustering. Among addresses assigned to multi-address clusters (i.e., 3,203,502 addresses), our clustering (3-hop) reduces the apparent number of distinct creators by roughly 11 times.

Even under the conservative 1-hop clustering setting, multi-address clusters are nontrivial in size: the median multi-address cluster contains 3 addresses, and the largest cluster contains 10,531

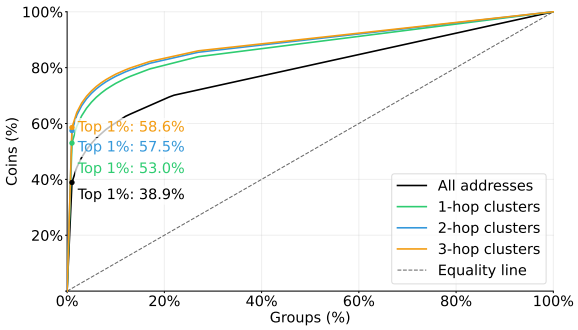


Figure 5: Fractions of coins (the y-axis) created by the top x% of clusters (including singletons).

addresses. These results indicate that a substantial fraction of coin creation activity is spread across multiple addresses, suggesting frequent address reuse and possible deliberate obfuscation.

Coin creation concentration. We observe that multi-address clusters account for a disproportionate share of coin creation. Under 1-hop clustering, addresses in multi-address clusters account for 62.99% of all coins, even though they represent only 48.02% of all coin creator addresses.

We note similar patterns when considering the overall concentration of coin creation. Figure 5 shows the fraction of coins created by the most active clusters. Without clustering, the top 1% of coin creator addresses account for 38.89% of all coins. After clustering, this concentration increases substantially. Under 1-hop clustering, the top 1% of clusters account for 52.99% of all coins. The concentration further increases under 2-hop and 3-hop clustering: the top 1% of clusters account for 57.47% and 58.57% of all coins, respectively.

Some of the largest clusters are highly active from late Feb. to May 2024 onwards, while others exist for only a certain period of time (e.g., 240 days for the 11th-largest cluster, just 9 days for the 38th-largest cluster). On average, the top 100 clusters (i.e., clusters with at least 2,464 creators) are active for 434 days.

Cluster validation. We perform validation analyses to assess whether the observed clusters genuinely reflect multi-address activity (e.g., rather than spurious links introduced by unlabeled third-party services).

First, we examine the distribution of coin creation within the top 1% most active clusters, ranked by the number of coins created. In the median cluster, the most active address accounts for 26.33% of the cluster's coins, while the three most active addresses together account for 50.33%. Conversely, the median number of addresses needed to account for 90% of a cluster's coins is 25.5. These results suggest that a single dominant address does not typically explain high-output clusters. Instead, coin creation is distributed across many addresses within the cluster.

Second, we evaluate cluster connectivity through adversarial deletion defined in §5.1. For the top 1% most active clusters, after the worst-case deletion, the largest remaining component retains a median of 80.4% of the cluster's addresses and 89.4% of its coin output. Our validation results indicate that the concentration patterns documented above are unlikely to be driven solely by a single address (e.g., unlabeled third-party entity).

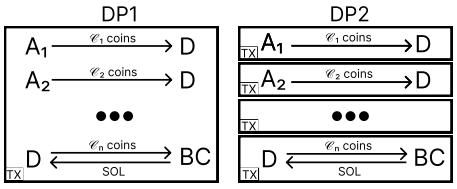


Figure 6: Dumping heuristics representation.

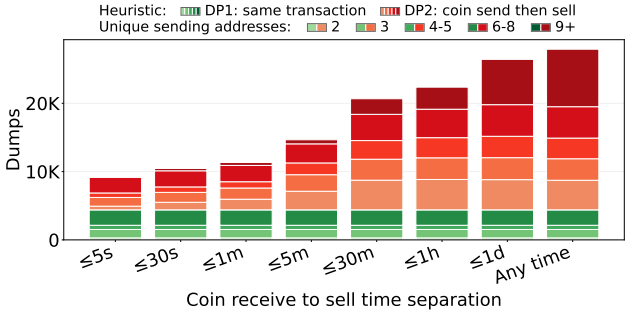


Figure 7: Nr. of dump instances for DP1 and DP2 across time thresholds (the x-axis) for 1% coin sample

6 Coordinated sell

This section investigates the scenario where a manipulator 1) deploys multiple addresses that buy tokens individually, 2) funnels all bought tokens to a single address, and 3) sells all tokens at once to profit. In particular, we focus on the "dumping" part (2, 3) that often happens consecutively.

6.1 Method

We define two heuristics: **DP1** and **DP2**, to identify coordinated sell, or "dumps." DP1 provides a conservative lower bound, while DP2 captures a broader set of patterns.

DP1: In Figure 6 (left), DP1 captures the most direct form of coordinated dumping. Multiple "sender" addresses $A_1, A_2, \ldots$ transfer tokens to a single "dumper" address D, which sells all received coins. All of this happens atomically, i.e., **within a single transaction**. Transactions that meet these criteria are unlikely to yield false positives. Indeed, all senders need to sign the transaction with their private keys—a strong sign that they belong to the same entity (or at the very least, colluding entities). Furthermore, manipulators need to directly interact with the blockchain to execute this, just like **WT1** in §4.

DP2: In Figure 6 (right), we relax the atomicity requirement, allowing for a time window between the transfers and the sell. We collect all coin transfers from $A_1, A_2, \ldots$ to D that occur within 24 hours of D's dumping. If D receives coins from at least two distinct senders and then sells at least the received amount, we classify it as a DP2 dump. When the window size is large, this heuristic may introduce false positives where two distinct traders happen to transfer their tokens to another trader, who sells them later.

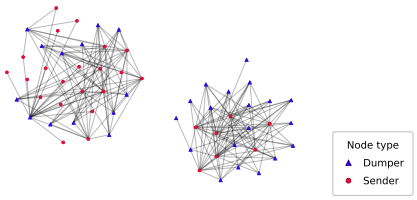


Figure 8: Token transfers (edges) between senders and dumpers (nodes) in the two largest clusters.

6.2 Results

Just like in §4, we only report our findings for the **1% coin sample**. Results for the **5-day sample** are similar, and can be found in the extended version [48, Appendix G]. Figure 7 shows the number of dumping instances (i.e., transfers and a sell pair) across different time window sizes (the x-axis). The green bars indicate the dumping instances flagged by DP1 (i.e., no change across the x-axis), and the red bars are the dumping instances flagged by DP2 (but not by DP1). The color gradation corresponds to the number of senders involved, as shown in the legend.

We find that dumping is prevalent. Under DP1, we detect 4,402 dumps. Under DP2, with time windows of 5 seconds, we capture 9,270 dumps. The number increases to 66,706 and 88,730 dumps for 1 hour and 1 day, respectively. We also show that there are many senders involved in dumping. Under DP1, the median number of senders is 7 (max 7). Under DP2, the median number is 7 (max 17). In the **5-day sample**, we find one extreme case, a dump involving 312 senders, highlighting a high degree of coordination.

We next show that manipulators reuse senders and dumpers. We construct a graph where each node is an address (either a sender or a dumper) and each edge represents a transfer from a sender to a dumper. Figure 8 shows the graphs corresponding to the largest two clusters flagged by DP1. Many senders repeatedly interact with many dumpers; the manipulators employ the same senders and dumpers repeatedly across manipulations.

Finally, we investigate dumping transaction volume under a conservative lower bound. Using DP1, the dumpers sell 33,393 SOL in total. In addition, considering the range of SOL price, which was roughly averaging USD 160 and never went below USD 80, we can estimate that the total dump volume is between USD 2.5M and 5M in this 1% sample.

7 Copycat coins

We describe how we detect "copycat" coins (i.e., coins impersonating another asset), and the strategies surrounding them.

7.1 Methods

We use three heuristics for identifying the set of original and copycat coins. The first and strictest approach detects coins with a shared **name**, **symbol**, **description** (can be empty), and **profile image**. The second heuristic detects coins with a shared **name**, **symbol**, and **profile image**, because the description is not mandatory and is the easiest to modify. The third and broadest approach detects coins that share only **name** and **symbol**, since a manipulator can also crop or edit an image or change a file extension. To identify image duplicates, we rely on the structure of IPFS, which includes the file hash in the URL[3] [23]. This approach gives us a straightforward mechanism for capturing exact image duplicates and mitigates risks (e.g., downloading offensive or illegal materials), but limits our ability to detect image similarity.

We define the original coin as the coin with the earliest creation timestamp and label the rest as copycats. If the two coins share an identical creation timestamp, we consider the coin with the higher current market cap to be the original coin. Finally, we exclude any pair of original and copycat coins that are created by the same address or belong to the same creator cluster (derived in §5), since such pairs likely belong to the same entity and do not exhibit impersonation intent. We separately analyze these coins with identical metadata and creator in the full version [48, Appendix H].

Table 3: Copycat detection results by the three heuristics

	Nr. originals	Nr. copycats (ratio)	Ori. no suffix rate	Copy. no suffix rate
1	647,986	1,490,123 (0.10)	0.14	0.26
2	757,930	1,866,178 (0.12)	0.14	0.25
3	1,270,892	5,392,081 (0.36)	0.14	0.20

7.2 Results

Table 3 summarizes the detection results for all three heuristics. The first and second methods capture over 1.5 million and 1.9 million copycats (10%, 12% of all coins), respectively. Most copycats that are caught by the second heuristic but not by the first either leave the description empty, or have a description associated with a token-deploying platform, e.g., "Deployed using https://[Service-C] ($n = 9,047$).

The broad third method identifies 5.4 million copycats (36%) that share the same name and symbol. This increase in detected coins comes from 1) copycat coins that modify either the description or image, or both, or 2) false positives of coins that happen to use the same name and symbol. The second case often appears when coin creators use the same concept (e.g., coins created after celebrities). We do not capture coins with slight modifications to their names and/or symbols, as seen in typosquatting research [16, 27, 40]. As a conservative bound, we use the first heuristic for the rest of the analyses.

We first investigate the competition between originals and copycats. The original coins have a significantly higher graduation rate, 9.20% (compared to the baseline 1.02%), whereas the copycat coins are less likely to graduate, 0.86%. The signal here is that the presence of copycats is an endorsement of the original coin, reflected in the higher-than-average graduation rate of originals. In Figure 9 (left), the x-axis is the order of arrival ($x = 1$ is the original, $x = 2$ is the earliest copycat), and the y-axis is the likelihood of graduation (§2). In this figure, we only include cases with at least 9 copycats. Becoming the original coin confers a significant first-mover advantage, and this benefit quickly disappears. Nevertheless, 17.7% of graduated coins are copycats. Earlier copycats have a slightly higher probability of graduation than later ones.

[3] IPFS generates a different identifier even if an image differs by a single pixel.

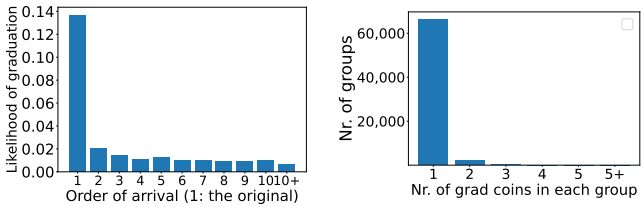


Figure 9: 1) Order of arrival and graduation likelihood and 2) Distributions of groups by nr. of graduated coins.

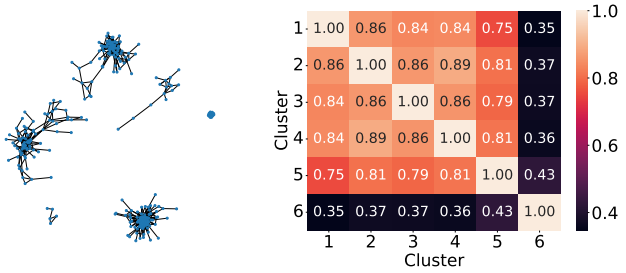


Figure 10: (Left) user collusion visualization; (Right) text similarity among the clusters.

We also examine if the original and the copycat(s) can both graduate. Figure 9 (right) further illustrates the number of graduated coins per "group" (i.e., the original and its copycats combined). 68,985 (10.65%) of groups graduate at least one coin. Only 2,613 groups manage to graduate multiple coins, accounting for just 0.40% of the total groups. While competition usually eliminates copycats, an original and the copycat(s) occasionally split users' attention and both graduate.

We will next estimate the level of automation present in the deployment of copycats. When coin creators launch a coin through the website's UI, pump.fun has been generating a vanity address [46], which ends with "pump" (lowercase) since (roughly) May 30, 2024. Thus, if a coin address ends with a random string ("no suffix"), it means the creator directly interacted with a blockchain using their custom smart contract. Since the manipulators could brute-force an address ending with "pump" (requiring about $58^4 \approx 11$ million attempts on average, which can be done in less than a second with the latest GPUs [53]), we consider our estimate as a lower bound. Table 3 shows that 13.7% of the original coins' addresses do not have a "pump" suffix. On the other hand, 26.5% of the copycat coins' addresses do not include the "pump" suffix. This result indicates copycat deployment is much more automated than regular coin creation: for comparison, the baseline rate of addresses without the "pump" suffix, over all coins, is 16.05%.

8 Social manipulation

This section covers two channels that actors use for influence: 1) pump.fun comments and 2) social media.

8.1 Comment manipulation

We first investigate automated bots that post promotional comments on pump.fun's coin profile pages.

Methods: Traders incorporate social signals (e.g., comments about coins) into their future price expectations.[4] Strategic actors can therefore influence the outcome of a coin by using multiple accounts to "astroturf" publicly available social signals.

pump.fun provides a chatbox on each coin's profile page where users can freely post comments from their blockchain addresses; they do not need to own the token. Using the comments from the 1% coin sample, we construct a graph (Figure 10, left) in which all nodes represent users (i.e., blockchain addresses), following the "lockstep" approach used in He et al. [21]. An edge between two users indicates that both users posted within $\tau = 0.1$ seconds of each other on at least *two* different coin pages. We also confirm that the graph remains structurally the same as τ varies from 0.1s to 1s. We exclude clusters with fewer than three nodes for visibility.

Results: Figure 10 (left) shows six closely-knit clusters: 111, 80, 78, 25, 15, and 5 users, respectively, suggesting potential collusion to boost social engagement. We expect to find more linkages or a denser central cluster if we expand our dataset beyond the 1% coin sample. While Tsuchiya et al. [52] find that reputation manipulation on financial articles uses *short* positive comments, most messages from these clusters are longer (median: about 23 characters for most groups) and appear to mimic authentic user comments. The common practice includes capitalization for emphasis (e.g., "HOW LONG I'VE WAITED FOR THIS! I BOUGHT IT AND I'M WAITING FOR THE PUMP."), slang, abbreviations, typos (e.g., "lmaoooo the idea is crazy , have u cjeck the website guys ?"). While most comments are positive, manipulators appear to intentionally include negative comments (e.g., "That 10% splitted wallet is not good," "I smell a rug").

We next examine comment similarity across clusters. Figure 10 (right) shows the Jaccard similarity, i.e., the fraction of text shared by all pairs of the six clusters. We see significant overlap in the text used *across* clusters, especially among the top five clusters, which suggests two plausible explanations. Each cluster could represent different actors that rely on the same software or services identified in §9. The fact that they are both tight-knit and distinct from each other, and that this remains robust to timing threshold changes ($0.1 \leq \tau \leq 1$ seconds) support this interpretation. Alternatively, all clusters could belong to the same entity: using a 1% sample of all coins could have caused us to miss linkages.

These results suggest that content-based moderation (e.g., detecting positive short comments) advocated earlier [52] may no longer be effective. Metadata-level detection (e.g., when and where users post) appears to be useful, especially given the rise of LLMs.

8.2 Manipulations from social media

We next investigate two kinds of manipulation strategies that influence social media signals about coins.

Community-based manipulation. A manipulator can create a public or private channel on social media to advertise their coins or invite users to coordinate manipulation. We look at Telegram groups and Twitter communities whose links appear in pump.fun coin metadata and analyze their strategies and outcomes.

[4] A line of pump-and-dump studies confirms the impact of external channels (e.g., Telegram or Discord groups) on coin prices [38, 51].

Post-based manipulation. Social media "influencers" (i.e., users with highly visible profiles) can post about something "memeable," and simultaneously create a meme coin to profit from it. Even if a manipulator does not hold any social status, they can closely monitor influential social media posts and immediately launch related coins.

For both manipulation types, our data cannot assert malicious *intent* to coordinate price manipulation or extract money. However, looking at market outcomes (e.g., graduation, price jumps) allows us to estimate how much a manipulator could *potentially* reap.

Methods: On `pump.fun`, coin creators can write anything in three (optional) open fields: "`twitter`," "`telegram`," and "`website`" for each coin. Notably, `pump.fun` does not validate the entered content (e.g., a Twitter username) against the corresponding services. We first use the `tldextract` library [29] to extract domain names from the links creators include in each field. 59.6%, 20.6%, 43.8% of coins link domains in the `twitter`, `telegram`, and `website` fields, respectively. This result indicates the importance of studying external influences outside `pump.fun`. We then aggregate all domain names (e.g., `youtube.com` and `youtu.be`) for each platform. We report the aggregated statistics for linked external platforms in the extended version [48, Appendix I].

We next focus on Twitter and Telegram, accounting for the majority of coins in the social media fields, and extract users, groups, and posts. We also look at Truth Social whose structure is highly similar to Twitter's.

Twitter links are either users, communities (i.e., public chat groups), or specific user posts. We distinguish those three based on the URL structure (i.e., `i/communities/` or `/status/`). We find 1,271,698 users, 434,383 communities, and 1,447,668 unique post links. We similarly get user accounts and their post links in Truth Social, accounting for 372 users and 4,662 unique posts.

Telegram links are either public (i.e., groups where messages are public), private (i.e., groups that require invitations), or bots (i.e., programmable chatbots, typically trading bots). We differentiate private channels that start with a plus sign ("+") after the domain and bots whose usernames ends with "bot" (case-insensitive) [54]. There are 800,441 public channels, 235,850 private channels, and 5,220 bots. We will use Telegram groups (both public and private) and Twitter communities to study community-based manipulation, and Twitter and Truth Social posts for potential post-based manipulation.

Results: Community-based manipulation. We first focus on community-based manipulation and analyze their strategies. We show that most groups are associated with only a few coins and are short-lived, while some groups are highly active. Table 4 illustrates the distribution of the number of coins each community has and the number of active days (i.e., days when they have at least one coin). Most groups have 1 coin, and 95% of groups only generate up to 5 coins (2 days). On the other hand, the top 0.1% (1,478 groups, ranked by the number of coins created) have 60 or more coins constantly over time.

We then turn our focus to these top 0.1% of highly active groups. We observe variations in coin creation timing and automation level. Figure 11a illustrates the distribution of the median coin creation interval between two consecutive coins generated by the same group; some groups generate a coin within a second, while others spend more than a day. Figure 11b shows the "automation rate" distribution, that is, the distribution of the ratio of coins without the `pump` suffix (likely to have been directly instantiated on the blockchain, through some automated mechanism, see §7) over all coins created by the same group. In other words, when all coins have the `pump` suffix, $x = 0$; when none of them do, $x = 1$. The distribution is bimodal. Many channels appear to create coins via the website UI (low x), while others systematically interact directly with the blockchain ($x = 1$).

Table 4: Community statistics: nr. of coins and active days.

Variable	mean	50%	95%	99%	99.9%	max
Nr. of coins	1.91	1	5	13	60	7,630
Nr. of active days	1.29	1	2	6	21	346

We then examine these variables against the "graduation rate," the number of graduated coins divided by the total number of coins created by each group, reflecting the "skillfulness" of each group. We provide three scatter plots against the graduation rate: Fig. 11c is the number of coins (i.e., the number of attempts), Fig. 11d is the automation ("no suffix") rate, and 11e is the median creation interval. The graduation rate generally rises with fewer attempts, less automation, and a medium to longer creation interval. For instance, some groups with 100 attempts, no automation, and 30 min interval achieve nearly a 14% graduation rate, 14 times higher than the baseline. The groups appear to be more successful when they do not have too many coins, use the website UI, and do not spam coins in a short period of time. In our extended version [48, Appendix J], we also show that the graduation rate drops significantly after the first coin, suggesting that continuing to create many coins in the same channel may not yield additional profits.

Finally, we qualitatively examine posts and messages from Telegram public groups and Twitter communities, document their activity level, and infer their coordination intent. We look at the 20 most active communities (by number of coins) and the 20 most successful communities (by graduation rate). For each community, we manually review at least 50 messages and record the community activities (e.g., number of members, number of messages, presence of Solana addresses, or bots broadcasting coin information). The two samples reveal distinct patterns. Communities with many coins tend to focus on trading and coin announcements rather than their underlying projects (e.g, channel owners and members hype upcoming coin releases). One notable scheme appears in Twitter communities (5 out of the 10 most active ones) that automatically generates a coin when someone joins; their average automation rate (i.e., no-pump suffix) is 96.1%. In contrast, successful communities tend to post project announcements (with an occasional use of a bot to broadcast coin status), and have coins that are aligned with the project theme. An exception is an AI agent channel that randomly deploys coins after famous figures or characters. We documented the detailed method and results in Appendix C.

Results: Post-based manipulation. We next study post-based manipulation. 3,571,170 coins (23.5% of all coins) are created (shortly) after 1,452,330 Twitter or Truth Social posts, highlighting the strong influence of these platforms on `pump.fun`. We are interested in estimating a potential profit from creating a coin based on a specific

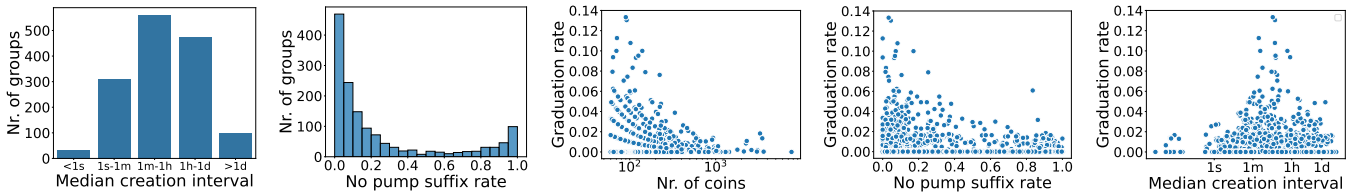


Figure 11: For each community, the distribution for (a) medium creation interval and (b) automation ("no suffix") ratio. Each community's graduation rate vs (c) the number of coins, (d) automation ratio, and (e) medium coin creation interval.

Table 5: Post statistics: nr. of coins and the total *EV* in USD.

Variable	mean	50%	95%	99%	99.9%	max
Nr. of coins	22.52	12	76	149	338	3,406
Total *EV* (USD)	28,260	135	12,748	126,445	3,683,136	113,633,854

post. We define the "extractable value (*EV*)" as the largest price difference multiplied by the amount of tokens the manipulator initially holds. The largest price difference is the highest daily price after the coin creation[5] minus the price of the coin creation in USD. While manipulators can own up to 1 billion tokens (i.e., the maximum token supply for each coin), the more they own, the lower the returns they can expect since this depletes the coin supply available to potential buyers. In this analysis, we set $x = 0.1$ (i.e., the manipulator possesses 10% of the initial share), though the number is easily adjustable.

We estimate *EV* for all graduated coins[6] for each post and aggregate *EV*s if the post has more than one graduate coin. We focus on 12,544 posts (0.86% of all posts) with a positive *EV*. Table 5 summarizes the distribution of the number of coins (including non-graduates) and the total of *EV* per post. The median post has 12 coins, and a total *EV* of 135 USD. We do not find any significant correlation between the number of coins and *EV*. 31 posts (the top 0.002% of all posts) have $EV \geq 1$ million USD, suggesting that the post-based manipulation can be highly competitive but lucrative.

To understand how social media posts generate profits, we manually review the 31 posts with $EV \geq 1$M USD. These posts together account for 83.4% of the total *EV* across all posts. For each post, we record engagement statistics (e.g., number of comments, reposts, likes, and number of followers the creator has), and whether the post (and the original post if the post is a reply or repost) includes pictures or videos. The complete annotation results are in the full version [48, Appendix K].

These profitable posts tend to come from high-profile accounts. The post creators have a median of 42,600 followers, ranging from 5,000 to 240 million followers. Their posts are highly engaging, with an average of 1,001 comments, 4,053 reposts, and 20,246 likes, and a median of 56 comments, 93 reposts, and 413 likes. These numbers are an order of magnitude higher than those of average Twitter posts: 2.6 comments, 6.7 reposts, 32.9 likes [47].

We also find that these successful posts generally fall into one of three categories. Inspired by Long et al. [34], we classify these posts into Culture, News, and Animal. Culture posts ($n = 13$) typically feature internet jokes often involving AI or celebrities (e.g., Elon Musk). News posts ($n = 8$) generally refer to announcements or real-world events (e.g., Ross Ulbricht's pardon). Animal posts ($n = 5$) often introduce pets or AI-themed characters. 20 out of 27 posts contain pictures or videos, suggesting that visual content is an important component.

As a case study, we choose five notable posts (out of 31): `truth_terminal`, `bitcoinmagazine`, `d33v33d0`, `tong0x`, `tonyplasencia3` to illustrate 1) the content of each post and 2) how it led to the creation of its most profitable coin. In all five cases, the original posters did not seem to have created the coin; posters may have no malicious intent, yet others can still profit from the post. Some posters endorse the token (e.g., `truth_terminal`) or receive some coin shares (e.g., `d33v33d0`). In contrast, other posters explicitly disassociate from the coin by 1) replying within the post's thread to state that the coin does not belong to the poster (e.g., `tong0x`, `tonyplasencia3`), 2) stating in the post itself that the poster has no intention of creating a meme coin. The time from post to coin creation varies: from 4 minutes (a news announcement, `bitcoinmagazine`) to a few months (a dog picture, `tonyplasencia3`). Details for each coin case study are in Appendix D.

9 Market-manipulation-as-a-service

In this section, we identify and investigate a set of websites and software tools that *could* enable users to perform the above manipulations with ease: Market-Manipulation-as-a-Service (MMaaS).

9.1 MMaaS websites

We first identify websites or apps with complex feature support and sophistication, e.g., those that require access to blockchain nodes or social media feeds with low latency. As described in §3, `pump.fun` typically hosts coin images on IPFS. However, some external tools for launching coins host images on their own infrastructure, which we can use to identify a lower bound on potential MMaaS activity related to `pump.fun`.

We follow the same process as §8.2 to extract (non-IPFS) domains from the coin metadata and manually infer potential websites from these domains (e.g., [platform_x] from [metadata.platform_x.com]). We look at the top 30 domains by frequency and only include those

[5] On very rare occasions, `pump.fun` appears to miscalculate the price, causing unrealistic (1,000,000×) high price jumps. We exclude these data points by taking the maximum daily *low* across the whole interval (i.e., the historical "high" not accounting for intra-day variations) as a conservative estimate.
[6] We assume $EV = 0$ for non-graduated coins.

that 1) are currently available and identifiable (e.g., some are in cloud services, such as Deutsche Telekom, Digital Ocean), 2) do not require authentication or registration, and 3) provide at least two functionalities defined below. These criteria reduce the list to four distinct services.

For each website, we check whether it *claims* to 1) provide low-latency services, 2) create multiple addresses (e.g., to buy coins in one transaction "bundling"), 3) copy existing coins (i.e., often called "vamping"), and 4) track social media feeds in real time. Importantly, we did *not* purchase or use these services to validate their claimed functionality for ethical reasons in Appendix B. Two authors annotated all four services on Apr. 14–15, 2026. Both authors agreed on 15 out of 16 cases (4 functionalities across 4 services). They met up to resolve the one disagreement. The annotation codebook and full results are in the extended version [48, Appendix N].

All services provide low-latency features, while some services help create multiple addresses, copycat coins, and track social media in real time, which corroborates our findings in §5, §7, and §8.2, respectively. Furthermore, one service allows users to create multiple addresses from a mixer (to obfuscate an identity potentially), and another service enables users to generate a vanity contract address ending with "pump" (to imitate coin creation on the `pump.fun` UI potentially).

9.2 MMaaS software

We next investigate open-source software that primarily supports simple automation (e.g., software that does not require access to blockchain nodes, for instance, software used to post comments automatically). We run the keyword search "pump fun comment bot" on GitHub, which identifies 29 repositories. We manually investigate all projects and exclude obvious duplicates and those without enough information in the `README` file. In total, we confirm 14 repositories *advertise* `pump.fun` comment bots. Similar to He et al. [21], we look at the `README` file, since the repositories often 1) redact some files or 2) redirect to external links. To avoid engaging in malicious activity ourselves, we cannot run the software to verify its advertised functionality.

We follow the same procedure as above (i.e., the website analysis) to annotate software functionalities. We check whether the repository *advertises* to 1) create fresh wallets for commenting, 2) use both fresh wallets and coin creator wallets for commenting (for a more genuine look), 3) provide predefined comment sets, 4) use AI to generate user profiles or comments, and 5) deploy anti-detection (e.g., proxies, CAPTCHA solvers). The two authors agree on 57 out of 70 questions (5 functionalities, 14 services). The major discrepancy comes from 1) ambiguity in the `README` file (e.g., whether the services or users prepare the predefined comments), 2) vague codebook definitions (e.g., terminology confusion), and 3) simple mistakes by one of the authors. If the description is unclear, we do not select that functionality (conservatively). Our codebook and annotation results are in the full version [48, Appendix N].

Overall, at least 8 repositories create fresh wallets (i.e., accounts) for users, while 4 repositories mix fresh accounts with developer or creator accounts to create even more natural social interaction (often called "shill" mode). 6 repositories provide a predefined set of comments for easier execution; the `README` file often claims that the software owner handpicked comments from the wild. On the other hand, two repositories use AI to generate comments or fake profiles (e.g., using OpenAI). 10 repositories have anti-detection tools such as CAPTCHA solving or proxy rotations (e.g., when creating accounts and posting comments on `pump.fun`).

With these services, users can execute manipulation without any technical knowledge or infrastructure. They can easily create multiple blockchain addresses to obfuscate token ownership, copy existing tokens, and monitor social media feeds in real time with low latency. Users can also create realistic social interactions (e.g., a coin developer talking to other accounts) through custom or AI-generated comments. These tools can even circumvent platform defenses (e.g., CAPTCHAs) or network-based defenses by hiding their IP addresses behind proxies. The websites and GitHub repositories we identify are publicly accessible: we may find more of these services in Telegram channels or underground marketplaces, which are considerably harder to find.

10 Discussion and Conclusion

Our findings have two clear takeaways. First, **market manipulation activities are widespread, even under conservative estimates**. Our analysis shows that at least 17% of all trading transactions are wash trades, and at least 10% of all coins on `pump.fun` are copycats. Several manipulation strategies are demonstrably effective: coins with high levels of wash trading activity are more likely to succeed. Similarly, some social media groups tend to have higher success rates. While other strategies, such as copycat creation, appear less effective on average, they remain important components of broader manipulation campaigns designed to mislead others.

Second, **manipulation strategies are sophisticated**. We have consistently observed that strategic actors deploy custom smart contracts that directly interact with the Solana blockchain in a highly automated manner. For instance, manipulators frequently bundle multiple trades within a single transaction (WT1 in §4 or DP1 in §6), which is not possible through the `pump.fun` web interface. Similarly, we find that some copycat creators (in §7) and Telegram or Twitter groups (in §8.2) consistently generate coins outside the `pump.fun` web interface (i.e., no pump suffix). The existence of Manipulation-as-a-Service (MMaaS) tools (§9) further facilitates the execution of sophisticated manipulation strategies.

As other competitors follow similar web interfaces and on-chain implementations, we believe most of the manipulation strategies here are also applicable to them. We leave the investigation of such manipulations' prevalence on other platforms to future work.

Mitigations. Our findings suggest that strategic actors on `pump.fun` manipulate a variety of signals, including trading volume, price charts, and comments. These strategies highlight the need for interventions that both raise costs for manipulators and help users better interpret the signals they observe. We outline interventions along two dimensions: technical and regulatory interventions.

Technical interventions. Overall, we believe there is a need for better quality and integrity of displayed signals. Platforms in the launchpad ecosystem, such as `pump.fun`, wallets, chain scanners, and statistics aggregators can detect or even suppress manipulative activity such as wash trading. Such detection can increase the friction and cost for manipulators and make it harder to mislead users.

For example, atomic wash trading is easy to detect and unlikely to contain false positives. While adversaries may adapt to evade detection, doing so increases their transaction costs.

In a similar vein, platforms should expose more robust provenance signals. For instance, they can create the funding graph (in §5) for each coin creator and show if a creator uses multiple addresses for obfuscation. Moreover, pump.fun can also flag coins whose addresses do not end with the "pump" suffix (i.e., likely creating coins outside pump.fun's UI), which suggests deviation from the typical coin creation process and may deserve more scrutiny. Additionally, pump.fun can show the number of copycat coins, e.g., "this coin has x coins with the same name, symbol, description, and image." This additional information would help prevent users from accidentally buying the wrong coin. Simultaneously, we advise users to adjust their expectations about simple indicators such as trading count or coin comments that are easy to manipulate.

Regulatory interventions. Regulatory bodies, such as the U.S. Securities and Exchange Commission (SEC) or its counterparts in other jurisdictions, can also play a role in helping address manipulation in the ecosystem. For example, they can create a regulatory framework to account for new forms of manipulation. We show that social media influencers *could* tweet and create a coin simultaneously to extract millions of profits on pump.fun potentially—this may be very similar in spirit to insider trading. Regulators can also consider taking down MMaaS websites or coordinate with GitHub moderators to remove repositories that enable manipulation at scale. However, software-level interventions may be less effective given the ease of creating new websites and repositories.

Acknowledgement

We thank Bryan Routledge, Ariel Zetlin-Jones, Viraj Shah and Kamya Singh for research discussions, review of relevant literature and assistance with preliminary data collection. This work was supported by the Cylab Presidential Fellowship, CyLab Security and Privacy Institute seed funding grant, the Hasler Foundation, Nakajima Foundation, and Google Academic Research Award.

References

[1] Arkham Intelligence. 2026. Arkham Intel API Guide. https://intel.arkm.com/api/docs. Version 1.1.0. Accessed Apr. 9, 2026.
[2] BankkRoll. 2026. Pump.fun API Documentation. https://github.com/BankkRoll/pumpfun-apis. Accessed Jan. 20, 2026.
[3] Bitquery. 2026. Understanding pump.fun: from launchpad to PumpSwap. https://docs.bitquery.io/docs/blockchain/Solana/Pumpfun/pump-fun-to-pump-swap/. Accessed Apr. 20th, 2026.
[4] Blockworks. 2026. Pump.fun: onchain metrics, activity and charts for Pump.fun. https://blockworks.com/analytics/pumpfun/pump-fun-trading-volume. Accessed Apr. 20, 2026.
[5] Blockworks. 2026. Raydium: onchain metrics, activity and charts for Raydium. https://blockworks.com/analytics/raydium/raydium-launchlab. Accessed Apr. 20, 2026.
[6] Elijah Bouma-Sims, Hiba Hassan, Alexandra Nisenoff, Lorrie Faith Cranor, and Nicolas Christin. 2024. "It was honestly just gambling": investigating the experiences of teenage cryptocurrency users on reddit. In *Proceedings of the 20th Symposium on Usable Privacy and Security (SOUPS'24)*.
[7] Federico Cernera, Massimo La Morgia, Alessandro Mei, and Francesco Sassi. 2023. Token spammers, rug pulls, and sniper bots: an analysis of the ecosystem of tokens in ethereum and in the binance smart chain (BNB). In *32nd USENIX Security Symposium (USENIX Security 23)*. 3349–3366.
[8] U.S. Commodity Futures Trading Commission. 2026. Futures glossary: wash trading. https://www.cftc.gov/LearnAndProtect/AdvisoriesAndArticles/CFTCGlossary/index.htm. Accessed Apr. 6, 2026.
[9] Lin William Cong, Xi Li, Ke Tang, and Yang Yang. 2023. Crypto wash trading. *Management Science* 69, 11 (2023).
[10] Will Cong, Campbell Harvey, Daniel Rabetti, and Zong-Yu Wu. 2025. An anatomy of crypto-enabled cybercrimes. *Management Science* 71, 4 (2025).
[11] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. 2020. Flash Boys 2.0: frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy*. IEEE.
[12] Anirudh Dhawan and Tālis J Putniņš. 2023. A new wolf in town? pump-and-dump manipulation in cryptocurrency markets. *Review of Finance* 27, 3 (2023), 935–975.
[13] Wenzhi Ding, Chen Lin, Yichen Luo, and Jiahua Xu. 2025. Decompose market manipulation strategies: evidence from on-chain meme coin market. (2025).
[14] Brett Hemenway Falk, Gerry Tsoukalas, and Niuniu Zhang. 2023. Can AI detect wash trading? evidence from NFTs. *arXiv preprint arXiv:2311.18717* (2023).
[15] Edward Felten. 2020. Front-Running as a Service. https://medium.com/offchainlabs/front-running-as-a-service-334c929c945a. Accessed Apr. 20, 2026.
[16] Evgeniy Gabrilovich and Alex Gontmakher. 2002. The homograph attack. *Commun. ACM* 45, 2 (2002), 128.
[17] Rundong Gan, Le Wang, Liang Xue, and Xiaodong Lin. 2024. Exposing stealthy wash trading on automated market maker exchanges. *ACM Transactions on Internet Technology* 24, 17 (2024), 1–30.
[18] Bingyu Gao, Haoyu Wang, Pengcheng Xia, Siwei Wu, Yajin Zhou, Xiapu Luo, and Gareth Tyson. 2020. Tracking counterfeit cryptocurrency end-to-end. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3, Article 50 (2020), 28 pages. https://doi.org/10.1145/3428335
[19] Shixuan Guan and Kai Li. 2024. Characterizing ethereum address poisoning attack. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. 986–1000.
[20] JT Hamrick, Farhang Rouhi, Arghya Mukherjee, Amir Feder, Neil Gandal, Tyler Moore, and Marie Vasek. 2021. An examination of the cryptocurrency pump-and-dump ecosystem. *Information Processing & Management* 58, 4 (2021).
[21] Hao He, Haoqin Yang, Philipp Burckhardt, Alexandros Kapravelos, Bogdan Vasilescu, and Christian Kästner. 2026. Six million (suspected) fake stars on GitHub: a growing spiral of popularity contests, spam, and walware. In *Proceedings of the 48th International Conference on Software Engineering (ICSE'26)*.
[22] Arkham Intelligence. 2026. How Arkham Labels Wallets. https://intel.arkm.com/api/docs#concepts/labeling. Accessed Apr. 9, 2026.
[23] IPFS. 2026. Content Identifiers (CIDs). https://docs.ipfs.tech/concepts/content-addressing/. Accessed Apr. 14, 2026.
[24] jhackworth. 2026. Dune Dashboard Pump.Fun. https://dune.com/jhackworth/pumpfun. Accessed Apr. 20, 2026.
[25] Josh Kamps and Bennett Kleinberg. 2018. To the moon: defining and detecting cryptocurrency pump-and-dumps. *Crime Science* 7, 18 (2018), 18.
[26] Daisuke Kawai, Kyle Soska, Bryan Routledge, Ariel Zetlin-Jones, and Nicolas Christin. 2024. Stranger danger? investor behavior and incentives on cryptocurrency copy-trading platforms. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–20.
[27] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in plain sight: a longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 569–586.
[28] David B Kramer. 2005. The way it is and the way it should be: liability under §10(b) of the Exchange Act and Rule 10b-5 Thereunder for making false and misleading statements as part of a scheme to "pump and dump" a stock. *University of Miami Business Law Review* 13, 2 (2005).
[29] John Kurkowski. 2026. tldextract 5.3.1. https://pypi.org/project/tldextract/. Accessed Apr. 27, 2026.
[30] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. 2023. The doge of wall street: analysis and detection of pump and dump cryptocurrency manipulations. *ACM Transactions on Internet Technology* 23, 1 (2023).
[31] Cornell Law School Legal Information Institute. 2026. 15 U.S. Code § 78i – manipulation of security prices. https://www.law.cornell.edu/uscode/text/15/78i. Section 9(a)(1), Securities Exchange Act of 1934; accessed Apr. 6, 2026.
[32] Tao Li, Donghwa Shin, and Baolian Wang. 2025. Cryptocurrency pump-and-dump schemes. *Journal of Financial and Quantitative Analysis* 60, 8 (2025), 3622–3659.
[33] Yueyao Li, Nanjun Yao, Yuhui Huo, and Wei Cai. 2025. Trust dynamics and bot-driven responses: an approach to rug pulls in solana meme coin markets. In *Proceedings of the 17th ACM Web Science Conference 2025*. 106–116.
[34] Hou-Wan Long, Nga-Man Wong, and Wei Cai. 2025. Bridging culture and finance: a multimodal analysis of memecoins in the web3 ecosystem. In *Companion Proceedings of the ACM on Web Conference 2025*. 1158–1161.
[35] Yichen Luo, Yebo Feng, Jiahua Xu, and Yang Liu. 2026. Resisting manipulative bots in meme coin copy trading: a multi-agent approach with chain-of-thought reasoning. In *Proceedings of the ACM Web Conference 2026*. 3414–3425.
[36] James Martin and Nicolas Christin. 2016. Ethics in cryptomarket research. *International Journal of Drug Policy* 35 (2016), 84–91.

[37] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. 2013. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet Measurement Conference*. 127–140.
[38] Mehrnoosh Mirtaheri, Sami Abu-El-Haija, Fred Morstatter, Greg Ver Steeg, and Aram Galstyan. 2021. Identifying and analyzing cryptocurrency manipulations in social media. *IEEE Transactions on Computational Social Systems* 8, 3 (2021), 607–617.
[39] Alberto Maria Mongardini and Alessandro Mei. 2026. A midsummer meme's dream: investigating market manipulations in the meme coin ecosystem. *35th USENIX Security Symposium* (2026).
[40] Tyler Moore and Benjamin Edelman. 2010. Measuring the perpetrators and funders of typosquatting. In *International Conference on Financial Cryptography and Data Security*. Springer, 175–191.
[41] Leonardo Nizzoli, Serena Tardelli, Marco Avvenuti, Stefano Cresci, Maurizio Tesconi, and Emilio Ferrara. 2020. Charting the landscape of online cryptocurrency manipulation. *IEEE Access* 8 (2020), 113230–113245.
[42] United States Department of Justice. 2014. Corporate executives, registered brokers and an attorney indicted for orchestrating a $300 million market manipulation scheme involving four publicly traded companies. https://www.justice.gov/usao-edny/pr/corporate-executives-registered-brokers-and-attorney-indicted-orchestrating-300-million. Accessed Apr. 6, 2026.
[43] Pump.fun. 2025. Fees. https://pump.fun/docs/fees. Last updated 2025-10-07. Accessed Apr. 9, 2026.
[44] Kaihua Qin, Liyi Zhou, and Arthur Gervais. 2022. Quantifying blockchain extractable value: how dark is the forest?. In *2022 IEEE Symposium on Security and Privacy*. IEEE.
[45] The Securities and Exchange Commission. 2024. SEC charges three so-called market makers and nine individuals in crackdown on manipulation of crypto assets offered and sold as securities. https://www.sec.gov/newsroom/press-releases/2024-166. Accessed Apr. 9, 2026.
[46] Kyle Soska, Jin-Dong Dong, Alex Khodaverdian, Ariel Zetlin-Jones, Bryan Routledge, and Nicolas Christin. 2021. Towards understanding cryptocurrency derivatives: a case study of BitMEX. In *Proceedings of the Web Conference (WWW'21)*.
[47] Statista. 2026. Average engagement on posts on X (formerly Twitter) in 2024 and 2025, by type of interaction. https://www.statista.com/statistics/1483834/. Accessed Apr. 27, 2026.
[48] Nicolas Szwajcok, Taro Tsuchiya, Enze Liu, Kyle Soska, Mathias Payer, and Nicolas Christin. 2026. Meme coin factories: uncovering large-scale manipulations on pump.fun. arXiv:2609.10246 [cs.CR] https://arxiv.org/abs/2609.10246
[49] Mia Tanner. 2025. Who Gets to Have Fun? The Alternative Earmarking of a Memecoin Marketplace. https://digitalcommons.macalester.edu/soci_honors/79. Sociology Honors Projects, Macalester College.
[50] Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. 2011. Suspended accounts in retrospect: an analysis of Twitter spam. In *Proc. ACM IMC 2011*. 243–258. https://doi.org/10.1145/2068816.2068840
[51] Taro Tsuchiya. 2021. Profitability of cryptocurrency pump and dump schemes. *Digital Finance* (2021), 149–167. https://doi.org/10.1007/s42521-021-00034-6
[52] Taro Tsuchiya, Alejandro Cuevas, Thomas Magelinski, and Nicolas Christin. 2023. Misbehavior and account suspension in an online financial communication platform. In *Proceedings of the ACM Web Conference 2023 (WWW'23)*.
[53] Taro Tsuchiya, Jin-Dong Dong, Kyle Soska, and Nicolas Christin. 2025. Blockchain address poisoning. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security'25)*.
[54] Taro Tsuchiya, Haoxiang Yu, Tina Marjanov, Alice Hutchings, Nicolas Christin, and Alejandro Cuevas. 2026. A large-scale study of Telegram bots. In *Proceedings of the 20th International Conference on Web and Social Media (ICWSM'26)*.
[55] District of Massachusetts U.S. Attorney's Office. 2024. Founder of cryptocurrency financial services firm "MyTrade" pleads guilty to market manipulation and fraud conspiracy. https://www.justice.gov/usao-ma/pr/founder-cryptocurrency-financial-services-firm-mytrade-pleads-guilty-market-manipulation. Accessed Apr. 9, 2026.
[56] Friedhelm Victor. 2020. Address clustering heuristics for ethereum. In *International conference on financial cryptography and data security*. Springer, 617–633.
[57] Friedhelm Victor and Tanja Hagemann. 2019. Cryptocurrency pump and dump schemes: quantification and detection. In *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 244–251.
[58] Friedhelm Victor and Andrea Marie Weintraud. 2021. Detecting and quantifying wash trading on decentralized cryptocurrency exchanges. In *Proceedings of the Web Conference 2021 (WWW'21)*. 23–32.
[59] Anh V. Vu, Ben Collier, Daniel R. Thomas, John Kristoff, Richard Clayton, and Alice Hutchings. 2025. Assessing the aftermath: the effects of a global takedown against DDoS-for-hire services. In *34th USENIX Security Symposium (USENIX Security 25)*. 3595–3612.
[60] Pengcheng Xia, Haoyu Wang, Bingyu Gao, Weihang Su, Zhou Yu, Xiapu Luo, Chao Zhang, Xusheng Xiao, and Guoai Xu. 2022. Trade or trick? Detecting and characterizing scam tokens on Uniswap decentralized exchange. *SIGMETRICS Perform. Eval. Rev.* 50, 1 (July 2022), 23–24. https://doi.org/10.1145/3547353.3522636
[61] Jiahua Xu and Benjamin Livshits. 2019. The anatomy of a cryptocurrency pump-and-dump scheme. In *28th USENIX Security Symposium (USENIX Security 19)*. 1609–1625.
[62] Guoyi Ye, Geng Hong, Yuan Zhang, and Min Yang. 2024. Interface illusions: uncovering the rise of visual scams in cryptocurrency wallets. In *Proceedings of the ACM Web Conference 2024*. 1585–1595.

A Open Science

We release the blockchain data and the scripts needed to reproduce our analysis of trade manipulation. Our data includes coin minting transactions (e.g., the coin's first transaction) for all 15 million coins, as well as the entire transaction history for both samples: the **1% coin sample** and the **5-day sample**.

This consists of a large (approximately 2 TB uncompressed) Postgres database, available at https://www.doi.org/10.1184/R1/33420628. The page also contains a set of scripts to help interact with the database. We cannot release the data we collected from the `pump.fun` website because `pump.fun`'s Terms of Use (ToU) prohibits data (re)distribution.

B Ethical Considerations

This section discusses the ethics of our data collection and the potential impact of publishing our results.

Blockchain data. We collected the blockchain transaction data from the Solana blockchain using third-party RPC service providers such as Chainstack and Syndica. This blockchain data is publicly accessible to everyone. We also use Arkham Intelligence's API [1] data to supplement wallet information. As the Solana blockchain wallets are pseudonymous (i.e., base58 strings), we do not consider them as Personally Identifiable Information (PII).

`pump.fun` website data. We collected coin information, the coin comments, and the price data from the `pump.fun` website. These data are critical to identify new classes of attacks in §7, §8, and §9. We only use `pump.fun`'s APIs [2] (no web scraping), following their ToS at the time of data collection. `pump.fun` did not provide `robots.txt` at that time. To minimize load, we leave at least 4–5 seconds between each request. For instance, when we retrieve coin information for 15.2 million coins, we have made about 304,000 requests (i.e., 50 coins for each query) and spread them across 25 days. We did not scrape web pages or download media files (e.g., images and videos). `pump.fun` accounts are also blockchain wallets (or occasionally user-defined usernames), which are not considered personally identifiable information (PII). Therefore, the IRB at our institution(s) do not consider this human subject research.

MMaaS. We investigated the potential websites or software tools that could facilitate the execution of market manipulation. We only view the website or the GitHub page, but do not interact with these services or execute their software to avoid supporting their services or risk installing malicious code. We opted not to report these services for the reasons below. First, some features have benign uses. For instance, users can create multiple blockchain addresses for privacy-preserving purposes. Second, we could not verify what these services actually do without using them or executing their software. Accordingly, we focus on documenting what they *advertise* and discuss implications for regulators in §10. When reporting our qualitative analysis in this paper, we anonymize their service names (e.g., Service *A*) to prevent misuse.

Social impact Malicious actors could misuse our findings and refine their attack strategies to prey on inexperienced traders. Simultaneously, our work helps inexperienced traders to recognize manipulation schemes and take precautionary measures. As discussed in §10, our findings could also inspire defenses for many actors in this space, including chain scanners, wallet providers, and financial regulators. Following the utilitarian ethics advocated in cryptomarket research [36], we believe the advantages of measuring the pump.fun ecosystem and reporting our findings outweigh the potential harms. As a part of responsible disclosure, we reported and discussed our findings with pump.fun team.

C Case study: community-based manipulation

We document the case studies for Telegram and Twitter communities in §8.2. This complementary section explains the detailed method and results.

We select 1) the 20 most active communities, ranked by the number of coins launched, and 2) the 20 most successful communities, ranked by graduation rate. We exclude Telegram private channels. Although the invitation links are publicly available, we believe these communities are intended to remain private (i.e., not searchable on Telegram), and joining them risks us exposing members' private information.

We manually visit each Telegram or Twitter community and read at least 50 messages (or tweets) from each, and document its activity levels (on Jul. 27th to 28th, 2026.) The questions include

- Is a community still accessible?
- Is it a Telegram channel (C) or a group (G)[7], or Twitter Community (T)?
- How many members does it currently have?
- Does it have more than 50 or 20 messages?
- Does it contain any Solana addresses?
- Does it use bots to broadcast trading information?

We have the deatiled annotatation results in the extended version [48, Appendix I], mask the channel name to avoid misuse, and refer to each community by index (e.g., Community 1).

20 most active communities. These channels primarily discuss coin symbol or address announcements and trading; only a few channels (e.g., Community 4) discuss the underlying project. For instance, some channel owners (e.g., Communities 6, 7, 10, 13, 15) or group members (e.g., Community 8) hype the upcoming token release. Example messages include "ACTION!" "Notifiastions ON!" (from Community 6) or "Moon" or "buy bro" (from Community 8).

In the most active Twitter community (Community 2), a new coin is automatically generated every time someone joins. For instance, we confirm in our pump.fun dataset that their coin name is "[new member's username] joined [community name]" and the symbol is "[new member's username]." By additionally looking at the 10 most active Twitter communities, we find that five follow the same scheme. They also show a significantly higher no-pump suffix rate, averaging 96.1%, as evidence of automation. In Community 2, members also post coin addresses, ask others to buy them and create a Telegram group to coordinate purchases.

20 most successful communities. Unlike the most active communities, most successful communities tend to discuss the underlying projects more often, though we cannot confirm their legitimacy. Some channels still use bots to broadcast financial information when someone buys the coin (e.g., Communities 25, 36, 40). In these channels, most coins appear to be created around the channel's theme, featuring different aspects of the project, or variations in symbols or images. Most channels follow this pattern, but we find an exception (e.g., Community 35) where agents randomly deploy coins based on famous figures or characters and post the corresponding coin information.

Limitation. Telegram private channels may likely include communities focused more on hyping coins (e.g., asking people to buy them). We also could not access some information when 1) the owner deleted the channel, 2) the owner deleted messages to avoid moderation. For instance, some channels have a significantly smaller number of messages relative to their number of subscribers (e.g., Community 14), which may suggest message deletion.

D Case study: post-based manipulation

We conduct the case studies on post-based manipulation in §8.2 and illustrate how a single social media post helps generate numerous coins and profits. This supplementary section describes each example in detail.

truth_terminal and **$GOAT** coin: Truth Terminal, a fully autonomous AI bot created by Andy Ayrey, posts its inner monologue on X, and became obsessed with the 1990s meme "Goatse." On Oct. 10th, 2024, an anonymous user created pump.fun coin, Goatseus Maximus ($GOAT), about an hour after one of the bot's posts. The bot then endorsed the coin and its price rose.

bitcoinmagazine and **$Ross** coin: On May 25th, 2024, Donald Trump stated that he would commute the sentence of Ross Ulbricht (founder of the anonymous marketplace, Silk Road) once he became president. Four min after the post, the coin was created.

d33v33d0 and **$SOL** coin: In Dec. 2025, Martin DeVido experimented with using Claude to autonomously grow tomato plants, giving Claude access to sensors to control water or humidity. The community created a coin based on this project and sent 10% of its supply to Martin's Solana address.

tong0x and **$AVA** coin: On Nov. 13th, 2024, Tong, co-founder of Hologram Labs, posted a picture of Ava AI captioned "Newest Intern at HoloworldAI." The coin was created about 30 minutes after the post (with the coin address posted in the reply), but Tong denied that the coin belongs to Tong.

tonyplasencia3 and **$ZOEY** coin: Tony from Griffin AI's posted the picture of his dog, Zoey, on Twitter on Jul. 15th, 2024. The coin was created on Dec. 16th, 2024. Tony did not seem to endorse the coin ("Do not speculate on my dog"). Similarly, when famous figures post their pets, the post often leads to the creation of numerous coins. For instance, Former Binance CEO CZ also posted pictures of his dog, Broccoli on Twitter, while stating that he did not issue any meme coin in his post. In total, the communities have created 1,960 pump.fun coins following his post.

[7] Telegram public groups fall into two types: channels (one-to-many) and groups (many-to-many). We only read 20 messages for Telegram channels, as the channel owners typically post a single long formal message, and 20 messages are sufficient to characterize their activities.