Decompose Market Manipulation Strategies: Evidence from On-chain Meme Coin Market *

Wenzhi Ding† Yichen Luo‡ Jiahua Xu¶

Abstract

This study unravels how different market manipulation strategies affect meme coin project performance and participant profits. Using a clean and observable on-chain setting on the meme coin platform Pumpfun, we collect granular on-chain transaction data and off-chain comment data for 6,000 meme coins and analyze the heterogeneous effects of rat (concealed accumulation/front-running), sniper (concealed accumulation/front-running), wash trading (fabricated activity), and comment bots (fabricated sentiment). Creators or snipers profit from concealed low-cost accumulation; attention manipulation, such as wash trading and fake comments, could effectively attract more traders and affect wealth redistribution. Avoiding manipulation and often timing the dump in early-stage projects, outperformers are disproportionately creators/snipers; underperformers are loss-sensitive and learn little from their past trading. Overall, our findings show how inventory concentration and attention fabrication drive participation, timing, and wealth redistribution. These insights inform transparency and anti-manipulation policy in both crypto and traditional financial markets to better protect innocent traders.

*The author gratefully acknowledges Ievgen Gerasymchuk for his insightful comments and valuable discussions on trading dynamics in the meme coin market. The author also thanks Lin William Cong for invaluable and constructive discussions.

†Hong Kong Polytechnic University, wenzhi.ding@polyu.edu.hk

‡UCL Centre for Blockchain Technologies, yichen.luo.22@ucl.ac.uk

¶UCL Centre for Blockchain Technologies, Exponential Science jiahua.xu@ucl.ac.uk

1 Introduction

Market manipulation represents a foundational challenge to market efficiency (Goldstein and Guembel, 2008). Classic strategies such as wash trading, concealed transactions and holding, rumor-mongering, and hyping have been prevalent across equity and commodity markets for decades. However, empirically measuring the distinct causal impact of each manipulative tactic has remained difficult due to a lack of account-level transaction data, especially those accounts that commit market manipulations, and difficulty isolating fundamental and non-fundamental (where manipulation belongs to) shocks from price movement. This study leverages a unique scenario with (1) clean and comprehensive account-level activity data and (2) no fundamental shock to asset price to address a central question in market microstructure: how do different manipulation strategies distinctly affect an asset's performance and the distribution of profits among its participants?

We use the on-chain meme coin market as a natural laboratory, where the absence of fundamental value isolates the impact of speculative and manipulative behaviors, and a standardized meme coin lifecycle allows for clean, comparative analysis. We provide the first granular analysis of how distinct strategies—including execution bots designed to conceal transaction and ownership, wash trading bots engineered to attract attention, and comment bots that fabricate social sentiment—independently shape market outcomes and redistribute wealth of manipulators and noise traders.

Meme coins, fueled by social hype and viral celebrity endorsements (such as $PEPE and $DOGE), have surged in popularity on platforms like Pumpfun, where a meme coin project can be launched with minimal barriers. One of the most prominent meme coins is $TRUMP coin, launched by President Trump, which ignited a wave of meme coin frenzy on Solana following its debut on January 17, 2025. The meme coin market, especially the platform Pumpfun, has attracted many speculative traders to participate actively in millions of projects, which can be regarded as a market manipulation experiment repeated a million times with large variations in participant constituents and strategies deployed.

We assemble a novel, account-level dataset from the Solana blockchain (via Flipside) covering 6,000 Pumpfun meme coin projects selected by stratified sampling—3,000 launched before and 3,000 after the January 17, 2025 $TRUMP debut—and balanced across three lifecycle outcomes ("no one cared," "unsuccessful," "migrated"), with project-level weights to recover population representativeness. For each project, we retrieve complete launch and migration records and the full transaction and transfer history, including block height and timestamp, wallet addresses, trade side (buy/sell), token and SOL amounts, USD values, and on-chain transfers (sender, receiver, amount). We complement these with off-chain Pumpfun comment streams. The granularity enables trader-level profit construction (clustering wallets via transfer links) and detection of manipulation strategies: rat bots (creator-controlled wallets transacting in the creation block), sniper bots (buys within the first five blocks post-launch), wash trading (a flip-based score that flags repeated equal-quantity round trips), and comment bots (LLM-based classification using few-shot prompts). We also build minute-level panels of participation and returns around bot-adoption times to study dynamic, causal effects.

We execute a stratified, population-representative study of 6,000 Pumpfun meme coin projects on Solana, combining complete on-chain transaction and transfer histories with off-chain comment streams. We identify manipulation tactics at scale using transparent account-level traces: rat bots (creator-controlled wallets transacting in the creation block), sniper bots (buys within the first five blocks), wash trading bots (a flip-based score capturing repeated equal-size round trips), and comment bots (LLM-classified hype/FUD messages). Wallets are clustered at the user level via transfer links to compute profits, and all project-level analyses are weighted to recover population shares. Our empirical strategy pairs project-level Weighted Least Squared (WLS) regressions with event-time stacked difference-in-differences around the minute of bot adoption to separate association from causality for attention-based manipulations.

On market outcomes, rat and sniper activity mechanically lift peak returns under the bonding-curve launch design and draw in more traders (about 35% more for rat/sniper projects), while shortening the time to migration when dumps do not occur early. Rat bots lengthen the pump but shorten the dump phase, consistent with creators building large, low-cost positions and then exiting aggressively at the peak; snipers exhibit qualitatively similar but weaker patterns due to higher average entry prices. We directly identify who triggers the selloff and find that creators are the most frequent dumpers (roughly half of cases), with an inverse relation to sniper dumping: the party with the larger low-cost inventory is more likely to start the dump.

To establish causal attention effects, we run stacked DID at the minute level around first adoption events. Introducing a wash trading bot nearly doubles contemporaneous trader participation (∼98%), and introducing a comment bot more than doubles it (∼116%). These spikes occur despite no improvement in average trader profitability, confirming that fabricated activity and sentiment can meaningfully pull in noise traders in the very short run.

On the distribution of profits, creators earn on average \$104 per project absent ratting, and an additional \$161 when using rat bots, validating the value of concealed low-cost accumulation. Sniper presence reduces creator profits by about \$54, reflecting competition for early low-price inventory. Wash trading bots (used in ∼4% of projects) boost creator profits by ∼\$432 on average, while simultaneously shrinking average losses for non-creators from −\$6.37 to −\$4.47 as larger crowds both push prices higher and lengthen windows to exit; comment bots have qualitatively similar but smaller effects. Overall, attention fabrication enriches originators while diluting per-trader losses among a larger pool of entrants.

Finally, we document stark heterogeneity in skill and strategy. "Winners" (positive, significant PnL t-stats) are far more likely to be creators (19.2%) or snipers (16.0%) than normals (4.4%, 3.3%) or losers (0.8%, 1.0%) are. Winners also avoid manipulated, crowded, high-profile coins, instead selecting smaller, earlier-stage projects with lower maxima but greater controllability, and they are frequently the dump initiators (>50%) in pre-mature projects. Losers are sensitive to recent losses and exit more, and while they show some ability to recognize wash/comment manipulations over time, learning plateaus below the winners' selectivity. These findings imply that low-cost position concentration and attention fabrication are key levers of extraction, that noise traders are highly susceptible to short-run attention shocks, and that passive learning is unlikely to protect vulnerable participants without stronger transparency and manipulation warnings.

The most closely related literature is the pump-and-dump scheme in the cryptocurrency market (Xu and Livshits, 2019; Dhawan and Putniņš, 2023; Li et al., 2025). Both this literature and our research build on the scenario of pump-and-dump. Existing literature on pump-and-dump answers the question specific to the phenomenon itself: why it happens and how it affects participants. Our differential focus is to answer how general market manipulation strategies affect assets and participants. Pump-and-dump is a scenario with account-level data and no fundamental shock that can be used to decompose the impact of several general market manipulation strategies. Our research tries to lend implications on evaluating and regulating market manipulations in both cryptocurrency and traditional financial markets.

Our study contributes to the literature on market manipulation by providing the first empirical decomposition of the distinct effects of different manipulative strategies. While foundational theories establish that manipulation distorts market efficiency (Allen and Gale, 1992; Jarrow, 1992; Goldstein and Guembel, 2008; Gandal et al., 2018; Griffin and Shams, 2020), empirically isolating individual tactics has been notoriously difficult in traditional markets due to opaque data and commingled strategies. We overcome these long-standing identification challenges by using the transparent, on-chain meme coin market as a natural laboratory. This unique setting allows us, for example, to advance the literature on sentiment and hype-based manipulation (Gurun and Butler, 2012; Degeorge et al., 2007; Fan et al., 2020; Li et al., 2022) by cleanly separating the impact of fabricated social engagement (via "comment bots") from that of fabricated trading volume (via "wash trading bots"). We can thus measure the distinct marginal effect of each tactic, a distinction that is nearly impossible to make in traditional settings.

Second, we contribute to the broader literature on crypto market manipulation. Existing research has primarily documented the existence of wash trading and analyzed its impact on centralized exchanges (CEXs), rather than on investors (Pennec et al., 2021; Cong et al., 2023; Aloosh and Li, 2024; Amiram et al., 2025). Gandal et al. (2018) and Aloosh and Li (2024) provide evidence of manipulation by individual traders on the Mt. Gox exchange. Griffin and Shams (2020) study institutional manipulative behavior in Bitcoin and Tether. In addition, Li et al. (2025) investigate pump-and-dump schemes across various cryptocurrencies. These studies do not document the on-chain wash trading bot activities. Utilizing the transparency of on-chain data on Solana, our unique dataset enables us to examine the impact of different wash trading bots on different participants.

2 Institutional Background

In this section, we introduce the concept of meme coin, Pumpfun's business model, and why it could be a scenario for studying market manipulation strategies, which can generate implications for the broader financial market.

2.1 Meme Coin and Pumpfun

A meme coin is a cryptocurrency derived from internet memes. It is typically characterized by low liquidity and a volatile price driven by community hype, social media trends, and celebrity endorsements.

Pumpfun, Solana's largest meme coin crowdfunding platform, lets users create meme coins for free. As of July 2025, Pumpfun has launched 11 million meme coins on its marketplace for free trading.[1] The timeline for launching a meme coin project on Pumpfun is shown in Figure 1.

First, users can upload an image and select a name and ticker (Figure 2), and Pumpfun will use this information to create a new token (so-called meme coin) via a standard smart contract on blockchain with a total supply of 1 billion. Of the total supply, 793,100,000 tokens are made available for trading, while 206,900,000 remain locked. Immediately afterward, the creator can choose whether to purchase the token at the lowest available price. This is similar to the Initial Public Offerings (IPO) application stage in the stock market, but with (1) an extremely simple prospectus, (2) no regulatory approval, and (3) no subscription process.

Next, the meme coin enters the "launching stage". It will be displayed on Pumpfun's front page and open to subscription. At this stage, the creator and other traders could buy or sell 793,100,000 meme coins from or to the contract at a designated price. The Automatic Market Maker (AMM) mechanism, introduced in the next subsection, sets the price. Therefore, the launching stage is essentially a secondary market instead of a primary market.

After 793,100,000 meme coins are purchased and 85 SOL are deposited, the meme coin project automatically enters the next stage, which is called "migration". Migration means

[1] https://dune.com/jhackworth/pumpfun

the meme coin project receives sufficient purchase and will be listed on a decentralized exchange (DEX). At this stage, the trading on DEX is nothing essentially different from the trading on a stock exchange, where marginal traders set the price, and the assets have a fixed circulation amount.

Besides the financial market, Pumpfun also provides a social platform. Any user with a Solana wallet can comment on any coin, similar to StockTwit. Each trade bumps the coin's name to the front page with a brief jiggle, as illustrated by Figure 3. The platform also flags potential bot activity and displays meme coin holding details, as shown in Figure 4.

2.2 Meme Coin Pricing Mechanism

The meme coin's price is variable and determined by the AMM mechanism, which generally reflects the principle "more demand, higher price". Specifically, at the launch stage of a meme coin project, traders can trade the 793,100,000 meme coins from the smart contract using SOL, Solana's native cryptocurrency, following the AMM mechanism. This mechanism defines the relationship between the total SOL deposited and the total meme coins received, as shown in Figure 5a and Figure 5b. It shows that when there are more demands for the meme coin (i.e., traders keep buying meme coin with SOL), the price of meme coin rises. More importantly, traders can sell meme coins back to the contract to claim back SOL. Therefore, this is essentially a secondary market in which traders can profit through market timing.

Technically, the AMM requires that the product of the quantities of two assets remains constant, expressed as $k = x'y'$, where x' and y' denote the respective asset quantities (Milionis et al., 2022). When Pumpfun's smart contract creates a meme coin, no SOL is initially deposited, and thus AMM cannot directly determine its price. To resolve this, Pumpfun assigns predetermined virtual values of $x' = 30$ and $y' = 1,073,000,191$. Accordingly, the relationship between the amount of SOL deposited and number of meme coins received for the 793,100,000 meme coins traded in Pumpfun is given by:

$$y = y' - \frac{k}{x + x'}, \tag{1}$$

where x is the amount of SOL deposited, and y is the corresponding number of meme coins received. x' and y' are 30 and 1,073,000,191, respectively. Based on Equation 1, we can derive the relationship between the price of meme coin and the SOL deposited as follows:

$$P = \frac{dx}{dy} = \frac{(x + x')^2}{k}, \tag{2}$$

where p is the price of the meme coin in terms of SOL. The AMM mechanism ensures prices rise with increased demand.

Similarly, after migration, Pumpfun unlocks 206,900,000 meme coins, deducts 6 SOL as a migration fee from the total 85 SOL deposited, and supplies the remaining 79 SOL together with the 206,900,000 meme coins as liquidity to the DEX. Therefore, the relationship between the amount of SOL deposited and meme coins received follows the same form as Equation 1, but with $x' = 79$ and $y' = 206,900,000$, respectively. All these numbers are predetermined in the smart contract to ensure that the price of the migrated coins remains constant at the onset.

2.3 Players

The meme coin ecosystem studied in this paper involves a diverse set of participants, each playing a distinct role in the lifecycle of a meme coin. We identify six key types of players:

2.3.1 Pumpfun

Pumpfun is the primary market for meme coins. It is the most prominent meme coin launchpad at the time this paper is written. It allows anyone with a Solana wallet to create and trade meme coins in a standard way. Pumpfun gains from charging a 1% fee on each buy or sell transaction, as well as a 6 SOL one-time migration fee deducted from the liquidity pool.

2.3.2 DEX

DEX is the secondary market for meme coins. After a meme coin reaches its migration threshold on Pumpfun, it is migrated to a DEX, Raydium, for secondary market trading. The DEX relies on AMM for liquidity provision and meme coin pricing. AMM is a way to determine trading price on DEX, to accommodate blockchain's features. It does not make this market fundamentally different from the traditional secondary market.

2.3.3 Meme Coin Creator

The creator is the user who initiates a meme coin by uploading metadata and deploying the smart contract via Pumpfun. Creators can profit from early token purchases at low prices, and some employ bots to conceal their ownership, inflate trading volume, or manipulate sentiment. Thus, creators often act as both originators and strategic manipulators within the system.

2.3.4 Traders

Traders are market participants who buy and sell meme coins either during the Pumpfun launch stage or on the DEX after migration. They can be affected by both transparent and hidden manipulative activity, and their profitability varies depending on timing and bot presence.

2.3.5 Snipers

Snipers are advanced traders who monitor the blockchain to exploit inefficiencies created by bots or inexperienced token creators. Using automated scripts, they front-run transactions within seconds of a meme coin launch so that they can secure a favorable position at very low cost ahead of other traders, especially when the creator does not buy many at the lowest price when launching the meme coin.

2.3.6 Bot Providers

Bot providers develop and rent or sell automated scripts to actors—typically creators or traders—who wish to manipulate market dynamics, as shown in Figure 6. These bots include rat bots (to hide ownership), wash trading bots (to simulate liquidity), and comment bots (to fabricate community sentiment).

2.4 Manipulation Bots

The speculative nature of meme coins makes them highly susceptible to rug pulls—sudden exit scams where the meme coin creator and early participants cash out and abandon the project. To make the scam appear organic, actors often deploy multiple fake wallets to simulate active trading and widespread interest while concealing their positions, as illustrated in Figure 6. This fabricated activity attracts new buyers, enabling early actors to profit. Typically, three types of automated scripts or bots are used: rat bots for concealed ownership (or rat trading), wash trading bots for wash trading, and comment bots for social media trolling. These scripts control multiple wallets to execute coordinated transactions.

2.4.1 Rat Bot

A rat bot is utilized when a creator launches a meme coin. During a launch on platforms like Pumpfun, the creator can purchase the meme coin at the lowest price based on a bonding curve mechanism, thereby driving up the price by subscribing to more tokens. However, these transactions, including the creator's holdings, are publicly visible on the Pumpfun dashboard (Figure 4). If the creator engages in heavy purchasing to inflate the price, it may signal concentrated ownership to other traders, deterring them from investing due to fears of a rug pull.

To counteract this visibility, the rat bot enables the creator to generate, fund, and control multiple wallets (e.g., Wallet A and Wallet B), which simultaneously buy the meme coin during the same creation block. This masks centralized ownership and create an illusion of organic demand, as illustrated in Figure 7a. This technique is analogous to *concealed ownership* strategies in stock markets, where traders may split their orders across multiple accounts or brokers to obscure their intentions. However, Pumpfun flags transactions occurring within the same block as potential bot activity, making this approach easily detectable. Moreover, only the coin creator can insert transactions into the creation block, further exposing such behavior as suspicious.

Alternatively, to avoid detection, the creator might employ a gradual rat bot, where multiple wallets purchase the meme coin incrementally over time, as depicted in Figure 7b. This approach reduces the likelihood of being flagged as bot activity. However, it introduces the risk of being *sniped* by other traders (e.g., Wallet X), who may front-run these transactions to secure a more favorable price. This creates unique challenges in meme coin trading due to its variable subscription price mechanism, which is absent in traditional markets.

2.4.2 Sniper Bot

Due to the meme coin pricing mechanism explained in subsection 2.2, acquiring positions at the initial stage of a meme coin launch is always beneficial because the price is almost at its lowest possible. Therefore, if the creator of a meme coin does not buy a sufficient amount himself, there is room for someone else to take this profitable opportunity. And these kinds of traders are called "sniper" since they seem to "sniping" those creators. Given the market competition, there only exists a very short time window (first to fifth blocks or 0.4 to 2 seconds) to snipe a newly launched meme coin, so snipers unavoidably use an automated script to detect and exploit the opportunities. Such a script is called "sniper bot".

Essentially, rat bots and sniper bots are similar. Both try to secretly accumulate positions at the lowest price, front-running other ordinary traders. Therefore, we treat it as the same type of manipulation strategy as "rat trading". The only difference is that meme coin creators deploy rat bots, while non-creators deploy sniper bots.

2.4.3 Wash Trading Bot

The primary goal of wash trading is to grab attention by creating the illusion of high trading volume and demand. In traditional markets, this involves submitting offset buy and sell orders to fake high liquidity and volume to attract investor interest. Similarly, wash trading bots automate this strategy in meme coin trading by repeatedly buying and selling tokens through multiple wallets. These offset transactions simulate organic trading activity without changing actual holdings but instead inflate the coin's perceived popularity.

On Pumpfun, every transaction by a wash trading bot bumps the coin to the platform's front page, increasing visibility and drawing attention from other traders (Figure 3). This heightened exposure acts as a form of advertising, enticing users to engage with the coin under the assumption of genuine interest. By automating the creation, funding, and operation of wallets, wash trading bots ensure a continuous flow of artificial activity, making the market appear more active than it actually is.

This behavior closely maps to traditional market wash trading, which aims to manipulate trading metrics to deceive other participants. While wash trading in traditional markets often seeks to influence prices or meet exchange volume requirements, meme coin trading's primary purpose is to capture attention in a competitive, speculative environment. Although effective in creating visibility, wash trading misleads traders and distorts market data, making it a harmful practice in both traditional and decentralized markets.

2.4.4 Comment Bot

A comment bot is an automated script designed to fabricate user engagement by generating fake comments and simulating community interest and enthusiasm around a meme coin. The bot posts short, context-free, and easily mass-producible positive messages by controlling multiple wallets. These comments often lack substantive information but are crafted to create the illusion of widespread support and social validation, such as phrases like "This coin is mooning!", "Don't miss out!", or "Huge potential here!". The goal is to mislead real users into perceiving the coin as having strong community backing, encouraging them to invest under the assumption that the project has popular traction.

This artificial engagement exploits the psychological principle of social proof, where traders are more likely to act when they believe others are already doing the same. Comment Bots amplify the perceived buzz around a project by flooding discussion forums, dashboards, or social media platforms associated with a coin. This can be particularly effective in the speculative environment of meme coin trading, where decisions are often driven by hype and momentum rather than fundamental analysis.

In addition to promoting specific coins, bots may also be deployed to post negative or misleading comments targeting competing projects. These are commonly referred to as fear, uncertainty, and doubt (FUD) bots. By spreading rumors, casting doubt on a rival project's legitimacy, or emphasizing potential risks, these bots aim to erode confidence in competing coins and divert attention toward their own promoted tokens. This tactic mirrors the concept of social media trolling in traditional financial markets, where fake accounts or coordinated campaigns are used to manipulate public sentiment about stocks or assets.

While comment bots can effectively create the appearance of community engagement, their presence undermines the authenticity of market interactions and misleads genuine users. Unlike traditional financial markets, where such behavior might rely on anonymous social media accounts, meme coin trading leverages the transparency of the blockchain, making it easier to trace the activity back to specific wallets. However, the decentralized nature of these platforms makes enforcing anti-manipulation measures challenging, allowing comment bots to remain a persistent issue in the ecosystem.

2.5 Mapping to Traditional Market

This study on meme coin trading offers insights into market manipulation strategies that also exist in traditional financial markets. While meme coins lack fundamental information, making them speculative and unconventional assets, this very feature allows for a clear and measurable decomposition of three key manipulation strategies: concealed ownership through rat bots or sniper bots, wash trading via wash trading bots, and social media trolling via comment bots. By isolating these strategies, the study provides a unique framework to analyze their effects on market performance, trader behavior, and participants' profits.

Furthermore, the findings are highly relevant to various types of assets in traditional markets, particularly those with speculative or lottery-like characteristics such as penny stocks, small-cap stocks, and other volatile investments. Much like meme coins, these assets are often prone to manipulation due to their relatively low liquidity and high price sensitivity. Additionally, this study is similar to the dynamics of post-IPO trading, where newly listed stocks frequently experience speculative trading, price pump-and-dump, media hype, rat trading, and all sorts of manipulations.

3 Data

3.1 Sample

We use account-level transaction and transfer data obtained from Flipside, a blockchain data indexing platform that maintains archive nodes with fully indexed, historical access to the entire Solana ledger.

Given that the overall number of meme coin trading records is in the hundreds of billions, it is beyond our capacity to analyze the universe of meme coins. To ensure representativeness, we sample six groups of meme coin projects stratifiedly and use weighted methods to carry out our analysis. Our dataset comprises 6,000 meme coin projects, evenly split between 3,000 tokens launched before the introduction of $TRUMP on January 17th, 2025, and 3,000 launched thereafter. This ensures that our sample comes from both normal and heated market sentiment. Then for each of the two time spans, we break the meme coin projects into three groups: (1) "no one cared" group represents those meme coins that receive less than 20% subscription at maximum; (2) "unsuccessful" group represents those meme coins that receive more than 20% subscription but not reach migration threshold (800 million) in the end; (3) "migrated" group represents those meme coins that migrated to secondary market (i.e., DEX). For each group, we randomly sample 1,000 projects. This stratified sampling strategy is mutually exclusive and collectively exhaustive, so we have sufficient observations for each type of project. We then obtain the distribution of these three types of projects in pre- and post-Trump. The distribution is shown in Appendix A. For the pre-Trump period, there are 67.2% "no one cared", 31.1% "unsuccessful", and 1.7% "migrated". For the post-Trump period, these ratios are 58.4%, 40.7%, and 0.9%, respectively. We then use these ratios to proportionally weight our sample in the following project-level analyses, so that the coefficients estimated are representative of the population.

We fetch the launch time and block, creator account address, transaction, and transfer histories for each meme coin. We retrieve the block, timestamp, trader account address, type (buy/sell), meme amount, SOL amount, and transaction dollar value for each transaction. We obtain the block, timestamp, sender, receiver, and coin amount for each transfer.

3.2 Bot Detection

To detect the presence of a rat bot, we analyze the block in which the meme coin is launched. Given Solana's block time of approximately 0.4 seconds and the absence of a global memory pool (mempool)[2], it is technically impossible for any trader other than the token creator to execute transactions within the same block as the meme coin creation. Therefore, any non-creator wallet activity observed in this block can only plausibly originate from the creator's use of an automatic script to create, fund, and purchase their own meme coins, thereby concealing their true position. We illustrate this process in Figure 7a. We define a dummy variable, $\textit{Rat Bot}_i$, which equals 1 when such activity is detected for meme coin project i and 0 otherwise.

Sniper bots are characterized by their ability to execute meme coin purchases at exceptionally high speeds, far exceeding typical human reaction times. To detect the presence of a sniper bot, we examine transactions occurring within the first five blocks following a token's launch (approximately 2 seconds, given Solana's 0.4-second block time). If a buy transaction is observed during this interval, we classify the meme coin as having been sniped by a sniper bot. We define a dummy variable, $\textit{Sniper Bot}_i$, which equals one if a naïve rat bot is detected for meme coin project i and zero otherwise.

The wash trading bot is characterized by repeated buy and sell transactions of identical amounts. To heuristically quantify wash trading behavior, we define a wash trading score $W_{i,j}$ for each trader j in a given meme coin i. Let the sequence of swaps executed by trader j in meme coin i be represented as:

$$\mathcal{S}_{i,j} = \{(d_{i,j,k}, q_{i,j,k})\}_{k=1}^{N_{i,j}}, \tag{3}$$

where $d_{i,j,k} \in \{\text{buy}, \text{sell}\}$ is the direction of k-th transaction. $q_{i,j,k} \in \mathbb{R}_+$ is the corresponding trading quantity. $N_{i,j}$ is the total number of transactions. A flip is defined as two consecutive trades in $\mathcal{S}_{i,j}$ that are in opposite directions and involve identical quantities. Let $\#\textit{Flip}_{i,j}$ denote the number of such pairs. The wash trading score $W_{i,j}$ of a trader j in meme coin i is defined as

$$W_{i,j} = \frac{\#\textit{Flip}_{i,j}}{|\textit{Position}_{i,j} + 1|}, \tag{4}$$

[2] A mempool is a temporary storage area where unconfirmed transactions wait before being included in a block. On Ethereum, the global mempool allows all participants to observe pending transactions, enabling practices such as front-running. By contrast, Solana does not maintain a global mempool; transactions are sent directly to validators, reducing transparency prior to block inclusion but also limiting the ability of external traders to compete within the same block.

where $Position_{i,j}$ denotes the final net token position (in units of the meme coin) of trader j for meme coin i.

The intuition behind this score is that it tends to be inflated if a trader executes a high number of flip transactions without changing the final net position. For example, if a regular trader buys 1,000 coins, sells 1,000 coins, and then buys 500 coins, the score is around 0.002. In contrast, a wash trading bot that buys and sells the same 1,000 coins 50 times yields a score of 99. We define a dummy variable, $Wash\ Trading\ Bot_i$, which equals 1 if any trader in meme coin project i has $W_{i,j} > 50$, and 0 otherwise. We visualize the distribution and threshold of wash trading score $W_{i,j}$ in Appendix B.

To analyze comment bot activity, we collect off-chain comment data for the 1,000 meme coins from Pumpfun. We observe a small proportion of comments that reference other users (e.g., "#88857219 show screenshot as proof pls?"), which are highly likely human-generated. In contrast, the majority consists of short, contextless, and mass-producible slogans highly indicative of bot activity. These two types of comments are semantically different: users operating bots or posting fake comments typically face a trade-off between length and variation, often resulting in more generic and shorter slogans that lack contextual relevance than human-generated ones. Moreover, bot-generated comments tend to express strong hype or hostility.

To capture these semantic differences and detect bot-generated content, we employ a Large Language Model (LLM). Specifically, we manually collect bot-generated comments from a YouTube advertisement video for a Pumpfun comment bot.[3] We also collect and annotate human-generated comments. These examples are then included as few-shot prompts in the context window of GPT-4o-mini to classify whether a comment is bot-generated and to determine its sentiment. In Appendix C, we show the system instruction and few-shot learning prompts for the comment bot detection. For each comment, we classify each comment as either bot- or human-generated. We define a dummy variable, $Comment\ Bot_i$, equal to 1 if at least one comment of meme coin project i is classified as bot-generated and 0 otherwise.

[3] https://www.youtube.com/watch?v=E_9fhdNo96c

3.3 Project Performance

We construct a set of variables to capture distinct dimensions of meme coin project performance. Specifically, *Ln(Max Ret)*$_i$ proxies the return potential of project i, defined as the maximum log return observed over the coin's lifecycle. *Ln(Number of Traders)*$_i$ measures the extent of market participation, calculated as the natural logarithm of the number of unique non-bot traders—aggregated at the individual level across wallets—who actively engage in trading the meme coin. *Ln(Pre-Migration Duration)*$_i$ reflects the project's time-to-migration, defined as the natural logarithm of the number of seconds elapsed between the coin's launch and its subsequent migration to DEX. *Ln(Pump Duration)*$_i$ captures the velocity of price appreciation, measured as the natural logarithm of the number of seconds from launch to the peak price. Finally, *Ln(Dump Duration)*$_i$ characterizes the speed of the price collapse, defined as the natural logarithm of the time interval between the peak price and the point at which the circulating supply decreases by 90%. Collectively, these variables capture the return potential, the breadth of trader participation, and the temporal dynamics of the pump-and-dump cycle in meme coin projects.

3.4 Trader Profit

We also examine the profits of traders for each meme coin project. Since a trader may control multiple wallets for a given meme coin, it is essential to cluster wallets at the user level when calculating profits. We rely on the assumption that it is unlikely for different users to transfer meme coin directly, so those wallets that transfer meme coin could be reasonably regarded as being controlled by the same user. We treat all wallets with transfer history involving that meme coin or SOL as a single trader when calculating profits. For each trader, the profit is calculated as follows:

$$Profit_{i,j} = S_{i,j}^{\text{USD}} - B_{i,j}^{\text{USD}}, \tag{5}$$

where $S_{i,j}^{\text{USD}}$ is the total USD proceeds from selling, $B_{i,j}^{\text{USD}}$ is the total USD spent on buying. The definitions of all variables referenced above are provided in Appendix D.

3.5 Descriptive

Table 1 shows the summary statistics of the overall 6,000 meme coin projects. All statistics are adjusted by weights as introduced in subsection 3.1. On average 24% of project creators adopt rat bots, indicating that this strategy is not that prevalent. In contrast, 84% of projects have sniper bots, showing the sniper strategy is competitive and a majority of projects are exploited. The presence of wash trading bots is rare - only 4% projects detect wash trading bots, probably due to the cost of wash trading being relatively high or the deployment of the wash trading bot being sophisticated. Comment bots show in 26% of the projects. The median of the maximum return of a project is around 15%. The median number of participants in a project is 9.

4 Empirical Results

In this section, we analyze the effects of different market manipulation strategies on project and participant performance.

4.1 Effects on Meme Coin Projects

In this subsection, we investigate the impact of different bots on the market performance of meme coin projects. We run a regression of meme coin projects' characteristics against indicators capturing bot activities:

$$Y_i = \alpha + \boldsymbol{\beta}\mathbf{X}_i' + \varepsilon_i, \tag{6}$$

where Y_i denotes one of the characteristics of the meme coin project as described in subsection 3.3. $\mathbf{X}_i'$ is a vector of the dummies indicating various bot activities. ε_i is an error term. We report the Variance Inflation Factor (VIF) of independent variables in Appendix E. The results show that multicollinearity is not a concern (average VIF < 5). We then estimate the model at the project level using WLS regression.

The results are shown in Table 2. The existence of all bots boosts the project's maximum returns (column 1). It is intuitive that projects with rat bots and sniper bots are mechanically more likely to have more purchases from the meme coin creators and snipers. Given the bonding curve mechanism, these projects mechanically have higher maximum returns. While it seems to be a higher likelihood of migration, projects with rat and sniper bots also attract 35% more noise traders (column 2) despite Pumpfun marking rat bot activities on each project's webpage. This result indicates that a certain fraction of noise traders may be extremely innocent and unsophisticated, so that even with alerts, they still step into traps. They might just see a relatively higher demand for such projects and consider it appealing.

Migrated projects exhibiting the presence of rat or sniper bots display significantly shorter pre-migration durations. This pattern confirms that both bot types artificially inflate perceived demand, thereby attracting a larger pool of noise traders and accelerating the migration process if neither bot engages in premature dumping before migration.

Projects involving rat bots tend to exhibit longer pump duration and shorter dump duration. This occurs because rat bots draw in more traders, prolonging the upward price movement until a higher peak is reached. Once this peak is achieved, the token creator usually initiates a large-scale sell-off. Given the creator's substantial token holdings, this aggressive dumping exerts significant downward pressure on the price, prompting other traders to close their positions rapidly.

Although sniper bots perform a similar function, they typically acquire smaller token positions because their average purchase price is higher than that of rat bots when spending the same amount of SOL. Consequently, sniper bots are likely to exert a weaker influence on the subsequent pump-and-dump dynamics of the project.

We identify the "dumper" and calculate the fraction of projects where the creator or sniper initiates the dump, as shown in Table 3. Technically, we define the dumper as the trader responsible for the largest sell-off among the first ten sell orders following the price peak. Our analysis reveals that creators are the most likely dumpers under all conditions, with approximately 50% of projects being dumped by creators. This is intuitive, as creators acquire meme coins at the lowest possible cost, making it profitable to dump at almost any point. Snipers also have a significant likelihood of being dumpers, particularly in the absence of rat bots, but to a lesser extent compared to creators. This finding supports our explanation of why sniper bots have similar, albeit weaker, effects on project performance compared to rat bots.

It is noteworthy that the likelihood of a creator or sniper being the dumper is inversely related—when one rises, the other falls. This finding highlights two key points: (1) whoever accumulates more low-cost holdings has more incentive to initiate the dump and (2) snipers usually act only in the absence of rat bot, i.e., they could identify rat bot and avoid them.

Back to Table 2, we further find that wash trading and comment bots positively correlate with projects' market performance. The correlations are positive across all performance metrics. Again, for this strong positive correlation, we could not rule out the reverse causality that projects with stronger performance attract more wash trading and comment bots. We conduct a difference-in-difference (DID) analysis better to estimate wash trading and comment bots' effects.

4.1.1 Dynamic Effects

In this subsection, we break the lifecycle of a meme coin project into 1-minute intervals and evaluate the extensive and intensive margin of introducing wash trading and comment bots to cope with the reverse causality problem in project-level analyses.

In light of recent advancements in econometric methodologies that address the limitations of the staggered DID framework, we adopt a stacked DID design to alleviate concerns regarding heterogeneous treatment effects following Cengiz et al. (2019) and Deshpande and Li (2019).

For each project i, we define $Treat_i$ to be 1 if it ever adopts a certain bot and 0 if otherwise. For each project i's bot adoption event, we first calculate the number of minutes passed after the launch of the project i, denoted as $t_i \in \mathbb{N}_0$. We then group all projects $\{i|t_i = t_c\}$ together to form the treated group of a cohort c. The projects forming the control group are those projects with $Treat_i = 0$. The trader number data for cohort c includes data within 1-hour event window $\{Number\ of\ Traders_{i,t}|t_c - 30 \leq t \leq t_c + 30]\}$. We subsequently stack all cohort samples to construct a cohort–project–time panel with 10 million and 6 million observations for the wash trading bot and comment bot, respectively. These two stacked panels serve as the basis for estimating the following stacked DID model:

$$Number\ of\ Traders_{i,t} = \beta\, Treat_i \times Post_{c,t} + \delta_{c,i} + \delta_{c,t} + \epsilon_{c,i,t}, \tag{7}$$

Number of Traders$_{i,t}$ is the number of traders of project i at t minutes since its initial launch. $Post_{c,t}$ is a dummy variable equal to 1 for the period after the event time ($t \geq t_c$) in corhort c and 0 otherwise. β captures the effect of bot adoption on the dependent variable. $\delta_{c,i}$ and $\delta_{c,t}$ represent the cohort-project and cohort-time fixed effects, respectively. $\varepsilon_{c,i,t}$ is an error term.

The results are shown in Table 4. The introduction of wash trading and comment bots attracts around 98% and 116% more traders within a short window. These stacked DID estimate prove that wash trading and comment bots have strong causal effects in attracting noise traders, instead of all effects being from reverse causality. And a more intuitive comparison between the treatment and control groups, shown in Figure 10, depicts how strongly traders are attracted by these two attention-based manipulations, despite the fact that these two manipulations do not let noise traders earn money (shown in the next subsection).

This stacked DID estimation provides a reference to the traditional market, to what extent wash trading and social media trolling could induce more noise trading.

4.2 Effects on Meme Coin Traders

We then examine the effects of different bots on the profit of meme coin project participants. We estimate the creator's profit via the following model specifications:

$$\begin{aligned} \mathit{Profit}_{i,j} = \beta_1 \cdot \mathit{Creator}_{i,j} + \beta_2 \cdot (\mathit{Creator}_{i,j} \times \mathit{Bot}_i) \\ + \beta_3 \cdot \mathit{Non\text{-}Creator}_{i,j} + \beta_4 \cdot (\mathit{Non\text{-}Creator}_{i,j} \times \mathit{Bot}_i) + \varepsilon_{i,j}, \end{aligned} \tag{8}$$

where $\mathit{Profit}_{i,j}$ denotes each trader's profit from a given meme coin between the token's launch and 12 hours after its migration; $\mathit{Creator}_{i,j}$ is the dummy variable equal to 1 if the trader is the meme coin's creator, 0 otherwise; Bot_i represents indicators capturing different bot activities. We estimate the model at the trader-project level using WLS regression.

From Table 5, column (1) indicates that, in the absence of a rat bot, meme coin project creators on average earn \$104 per project. Using the rat bot boosts the creator's profit by \$161 per project. The project creator earns a surplus from the rat trading manipulation strategy, indicating the effectiveness of accumulating positions at the lowest cost.

For the sniper bot (column 2), in the absence of such an attack, the creator can earn \$187, and the presence of the sniper bot makes the creator earn less \$54. Intuitively, a sniper attack is essentially someone else taking the role of "rat trading", taking advantage of naive creators. This shows a competition between the creator and snipers.

Wash trading bots primarily benefit creators, boosting their earnings by an average of \$432 per project. Specifically, for the 4% of projects that utilize wash trading bots, creators earn an average of \$556. However, this analysis does not account for the costs associated with deploying such bots. Given their low adoption rate, it is likely that the cost of deploying wash trading bots is considerable since wash trading requires a large number of transactions, which may consume too much transaction cost.

Interestingly, wash trading bots also reduce losses for non-creators. Without these bots, non-creators lose an average of \$6.37 per project. In contrast, with the presence of wash trading bots, which attract more noise traders, the average loss decreases to \$4.47 ($= -\$6.37 + \$1.90$) per project. It seems contradictory that everyone is better off. Actually, it is because there are more noise traders attracted by the wash trading bot, so that, on average, everyone contributes less to the creator's profit, but in aggregate, creators earn from more participants, so it also gets a higher profit. Combining the analysis of wash-trading bots on project performance in Table 2, the reason why noise traders could lose less is probably because when there are more participants, the project not only goes higher, but also endures longer, so on average, traders have more chances to liquidate their position at a better time.

Comment bots have similar effects to wash trading bots, though with a much smaller economic impact. This makes sense, as both wash trading and comment bots are designed to attract noise traders, i.e., attention manipulation. The weaker effect of the comment bot compared with the wash trading bot also makes sense. Wash trading has a first-order attention-grabbing effect due to Pumpfun's mechanism - when a project has a new trade, it is bumped to the first place on Pumpfun's front page. Comment bots do not boost the project on the platform, but traders must first click on a project and then see the heated discussion.

4.3 Winners' and Losers' Strategies

From the 6,000 projects, we randomly sample 4,000 traders who control only one wallet and retrieve their entire transaction histories from the inception of their wallets on Solana. We then compute each trader's return for every project in which they participate. For traders who participate in more than two but fewer than one thousand projects (excluding algorithmic traders), we calculate the t-stat of all their returns and see whether this trader consistently outperforms or underperforms others. For the 2,459 traders that satisfy the above requirement, we plot the distribution of their t-stats in Figure 11. Traders with a return t-stat less than -2.65 underperform, and those with a return t-stat over 2.65 outperform. There are two takeaways from the distribution. First, there is huge heterogeneous performance among traders, indicating the existence of effective skills or strategies in trading meme coins. Second, the right-skewed distribution means that more traders robustly lose money than traders robustly make money. In other words, a small group of skillful traders earns profits from a large group of innocent traders. It implies that experienced traders can easily take advantage of innocent traders in a market that can be freely manipulated.

Then the next question is how these skillful traders take others' money. We further investigate the strategies that might be employed by winners ($t > 2.65$) and losers ($t < -2.65$). Since we can identify who creates a meme coin or deploys a sniper bot (introduced in subsection 3.2), we evaluate the proportion of winners and losers who act as creators or snipers. A trader is classified as a creator if they have ever created a project, and as a sniper if they have ever deployed a sniper bot in any project. Note that a trader could simultaneously be a creator and a sniper based on our definition. The trader-level descriptive statistics are shown in Table 6. Consistent with previous results on trader profits, we find that 19.2% winners create coins and 16% winners adopt the sniper strategy. In normal traders, the ratio is 4.4% and 3.3%, respectively, way much lower than winners. These ratios are even lower in losers, with 0.8% and 1%, respectively. This monotonic decline of the creator and sniper role from winners to losers shows that "rat trading" is indeed a very effective way to become an outperformer.

We further see how winners and losers choose projects differently. The results are shown in Table 7. The descriptive statistics are interpreted in the following way. Taking the winner's rat bot ratio 0.2 as an example, it says that on average, 20% of projects that winners participate in have rat bot. The results reveal rich insights.

Winners tend to excel at avoiding projects associated with manipulation. Overall, the projects chosen by winners exhibit the lowest ratios of rat bots, sniper bots, wash trading bots, and comment bots. Interestingly, winners do not select the best-performing projects. Instead, they opt for projects with lower maximum returns, less popularity, and shorter lifespans than those chosen by normal or loser traders. A clear monotonic pattern emerges across these groups, with winners strategically favoring pre-mature, small-cap projects that offer more predictable, albeit smaller, profits. In contrast, losers are often drawn to high-profile projects with greater uncertainty.

This explanation is further supported by Table 8. We define a "dumper" as the trader who initiates the sell-off of a project. Specifically, the dumper is the trader who places the largest sell order among the first ten sell transactions following a project's price peak. Our analysis shows that winners are highly likely ($> 50\%$) to act as dumpers in premature projects, while losers rarely initiate the dump ($< 10\%$). For more mature projects with larger pools of traders, the identity of the dumper appears significantly more random. In our migrated project sample, all dumpers were classified as normal traders—neither outperforming nor underperforming their peers.

In summary, the key takeaways from this section are threefold. First, winners are likelier to act as token creators or snipers, leveraging strategies like "rat trading" to gain an advantage. Second, winners can identify manipulations and avoid such projects. Third, winners prefer pre-mature projects, allowing them to control the timing of their exit through dumping. These findings have important implications for traditional financial markets: assets with concentrated or low-cost holdings should be closely monitored or excluded to mitigate predatory behaviors and protect vulnerable investors.

4.4 Trader's Learning

Given that manipulations heavily hurt noise traders, it is important to know whether traders can learn to protect themselves. If they cannot learn from the failure, then exchanges or regulators must take more action to protect them from traps.

We first explore whether traders quit the market based on recent profit and loss. Table 9 shows the results. Overall, we find that if a trader's recent profits are higher, the likelihood of exiting the market is lower, as shown in column 1. Interestingly, winners' stay-or-quit decisions are not affected by recent performance (column 2), while losers are strongly affected (column 4). These results show that winners might have persistent trading strategies while losers are sensitive to recent losses and quit.

Next, we examine whether traders learn to identify manipulations and avoid them. As shown in Figure 12, we do not find systematic differences in learning the identification of the rat bot and the sniper bot. However, there are strong differences between the wash trading bot (Panel c) and the comment bot (Panel d). Although lower learners learn to identify both bots from the very beginning, they reach a stable level, but still a much higher level compared with the winners. This pattern shows that although losers have a certain level of learning, it is not sufficient.

5 Conclusion and Implication

This paper decomposes the causal effects of three canonical manipulation strategies—concealed ownership/front-running (rat and sniper bots), fabricated trading activity (wash trading bots), and fabricated social sentiment (comment bots)—in a setting with account-level transparency and negligible fundamentals: the on-chain meme coin market on Solana's Pumpfun. Using stratified sampling, project-level WLS regressions, trader-level profit analysis, and stacked difference-in-differences around the minute of bot adoption, we isolate how each tactic shapes market outcomes and redistributes profits.

First, we find that traders who accumulate positions at low cost tend to initiate the dump and profit from other traders. This has implications for both meme coins and the traditional financial market, as disclosing the position distribution could prevent traders from falling into a trap.

Second, attention fabrication strategies, such as wash trading and fake comments, could effectively attract more traders and affect wealth redistribution. There are strong causal effects between attention manipulation and trader participation. We also find that experienced traders can identify such manipulations and avoid these projects. This generates implications for the traditional financial market, which might help protect ordinary traders by limiting attention manipulations. For example, many stock trading software list active trading or abnormal return stocks on the front page, which might attract innocent traders to step into traps.

Lastly, underperforming traders hardly learn from their failures - they keep making the same mistakes. This implies that it is unlikely to rely on traders to learn to protect themselves; instead, active protection, such as more manipulation detection and warning, is necessary.

We acknowledge that this study has certain limitations. Although we have a scenario without fundamental information to study the impact of market manipulations, traders may behave differently when there is a mix of fundamental and non-fundamental shocks. In that case, our findings may not be generalizable. Also, we study a scenario without professional or institutional participants, such as market makers and funds. The presence of these investors may change noise traders' responses to market manipulation.

References

Allen, F. and Gale, D. (1992). Stock-Price Manipulation. *The Review of Financial Studies*, 5(3):503–529.

Aloosh, A. and Li, J. (2024). Direct Evidence of Bitcoin Wash Trading. *Management Science*, 70(12):8875–8921.

Amiram, D., Lyandres, E., and Rabetti, D. (2025). Trading Volume Manipulation and Competition Among Centralized Crypto Exchanges. *Management Science*.

Cengiz, D., Dube, A., Lindner, A., and Zipperer, B. (2019). The Effect of Minimum Wages on Low-Wage Jobs. *The Quarterly Journal of Economics*, 134(3):1405–1454.

Cong, L. W., Li, X., Tang, K., and Yang, Y. (2023). Crypto Wash Trading. *Management Science*, 69(11):6427–6454.

Degeorge, F., Derrien, F., and Womack, K. L. (2007). Analyst Hype in IPOs: Explaining the Popularity of Bookbuilding. *The Review of Financial Studies*, 20(4):1021–1058.

Deshpande, M. and Li, Y. (2019). Who Is Screened Out? Application Costs and the Targeting of Disability Programs. *American Economic Journal: Economic Policy*, 11(4):213–248.

Dhawan, A. and Putniņš, T. J. (2023). A New Wolf in Town? Pump-and-Dump Manipulation in Cryptocurrency Markets. *Review of Finance*, 27(3):935–975.

Fan, R., Talavera, O., and Tran, V. (2020). Social media bots and stock markets. *European Financial Management*, 26(3):753–777.

Gandal, N., Hamrick, J., Moore, T., and Oberman, T. (2018). Price manipulation in the Bitcoin ecosystem. *Journal of Monetary Economics*, 95:86–96.

Goldstein, I. and Guembel, A. (2008). Manipulation and the Allocational Role of Prices. *The Review of Economic Studies*, 75(1):133–164.

Griffin, J. M. and Shams, A. (2020). Is Bitcoin Really Untethered? *The Journal of Finance*, 75(4):1913–1964.

Gurun, U. G. and Butler, A. W. (2012). Don't Believe the Hype: Local Media Slant, Local Advertising, and Firm Value. *The Journal of Finance*, 67(2):561–598.

Jarrow, R. A. (1992). Market Manipulation, Bubbles, Corners, and Short Squeezes. *The Journal of Financial and Quantitative Analysis*, 27(3):311–336.

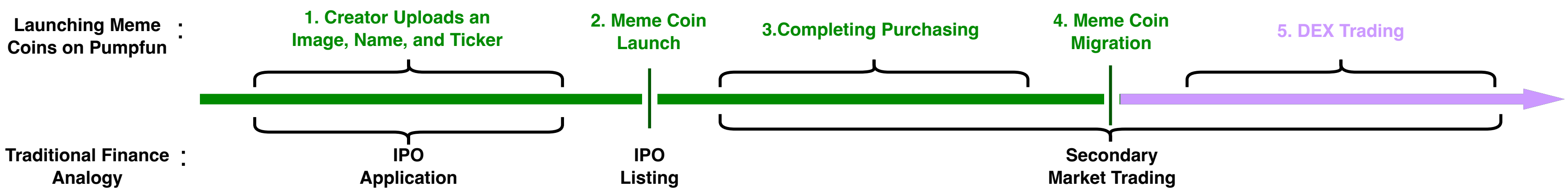
Li, T., Shin, D., Sun, C., and Wang, B. (2022). The Dark Side of Decentralized Finance: Evidence from Meme Tokens.

Li, T., Shin, D., and Wang, B. (2025). Cryptocurrency Pump-and-Dump Schemes. *Journal of Financial and Quantitative Analysis*, pages 1–59.

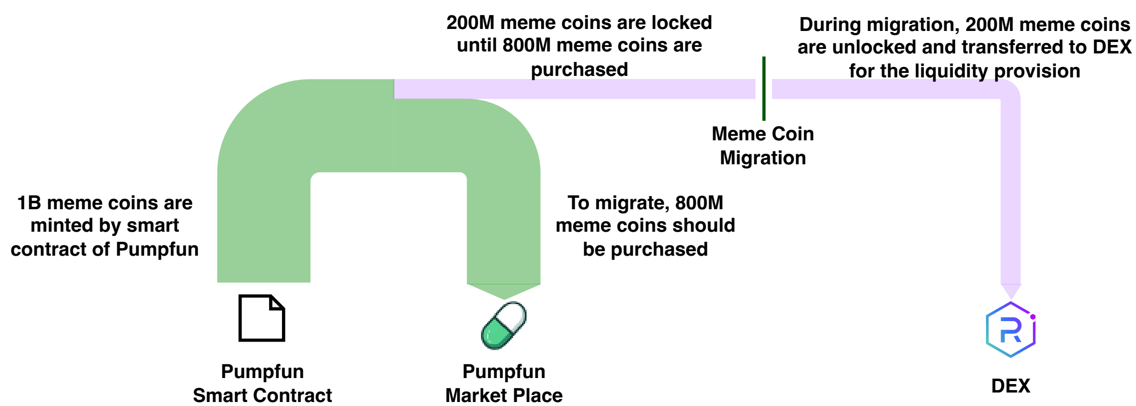
Milionis, J., Moallemi, C. C., Roughgarden, T., and Zhang, A. L. (2022). Automated Market Making and Loss-Versus-Rebalancing.

Pennec, G. L., Fiedler, I., and Ante, L. (2021). Wash trading at cryptocurrency exchanges. *Finance Research Letters*, 43:101982.

Xu, J. and Livshits, B. (2019). The anatomy of a cryptocurrency pump-and-dump scheme. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pages 1609–1625. USENIX Association.



(a) Timeline for Launching a Meme Coin Project



(b) Meme Coin Flow

Figure 1: Meme Coin Project Life Cycle

This figure illustrates the analogy between launching a meme coin project and an IPO in traditional finance. First, the creator uploads an image, name, and ticker to launch a meme coin, analogous to the IPO announcement and application process. Next, the meme coin is launched and displayed on Pumpfun's front page. Once launched, the meme coin appears on Pumpfun's front page with a total supply of 1 billion, 200 million locked, and 800 million available for trading. After the 800 million coins are purchased, this meme coin will be migrated to the DEX. Finally, it becomes tradable on the DEX, mirroring secondary market trading.

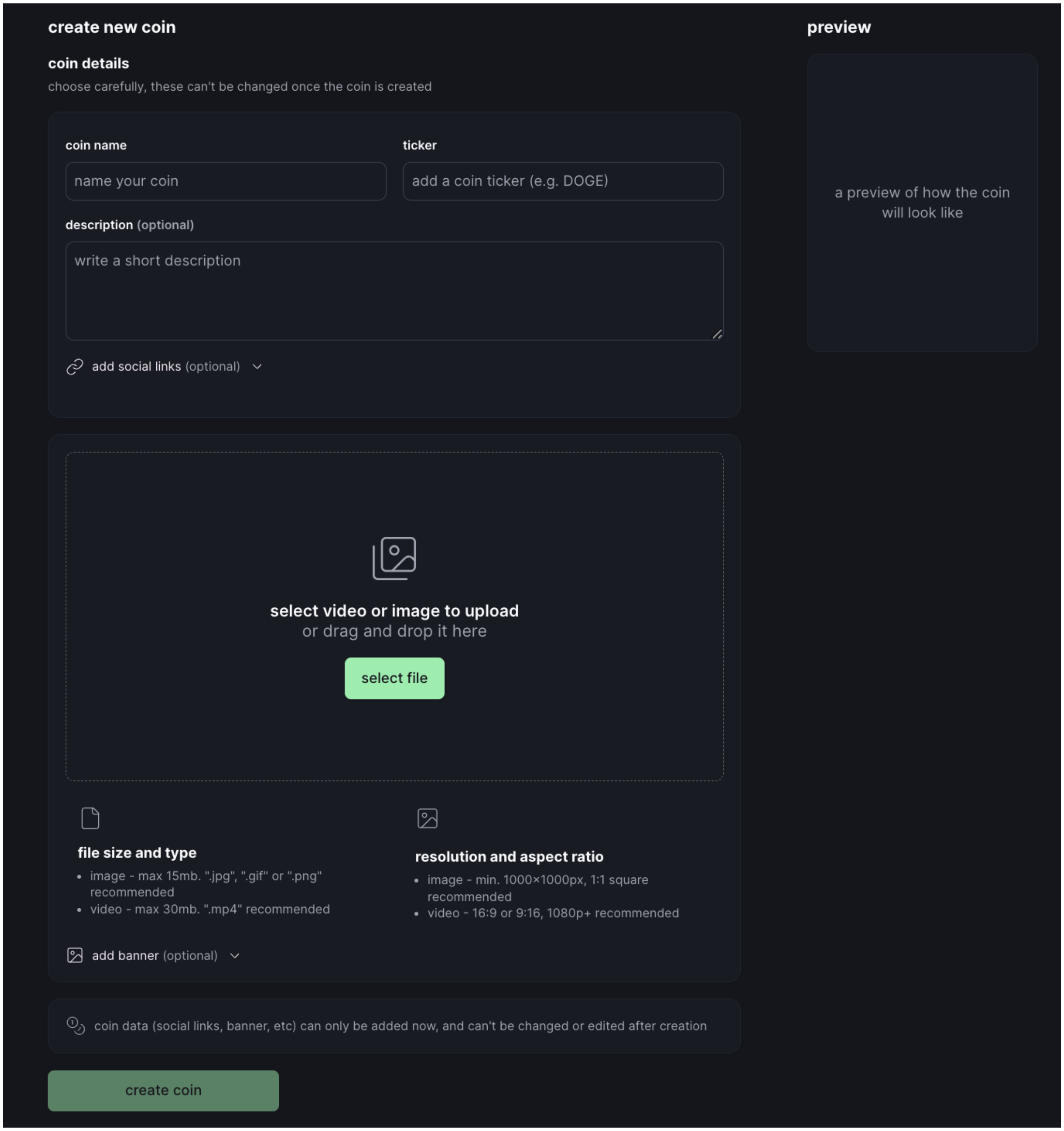


Figure 2: Meme Coin Creation Illustration

Users can upload an image and select a name and ticker. Pumpfun will use this information to create a new meme coin.

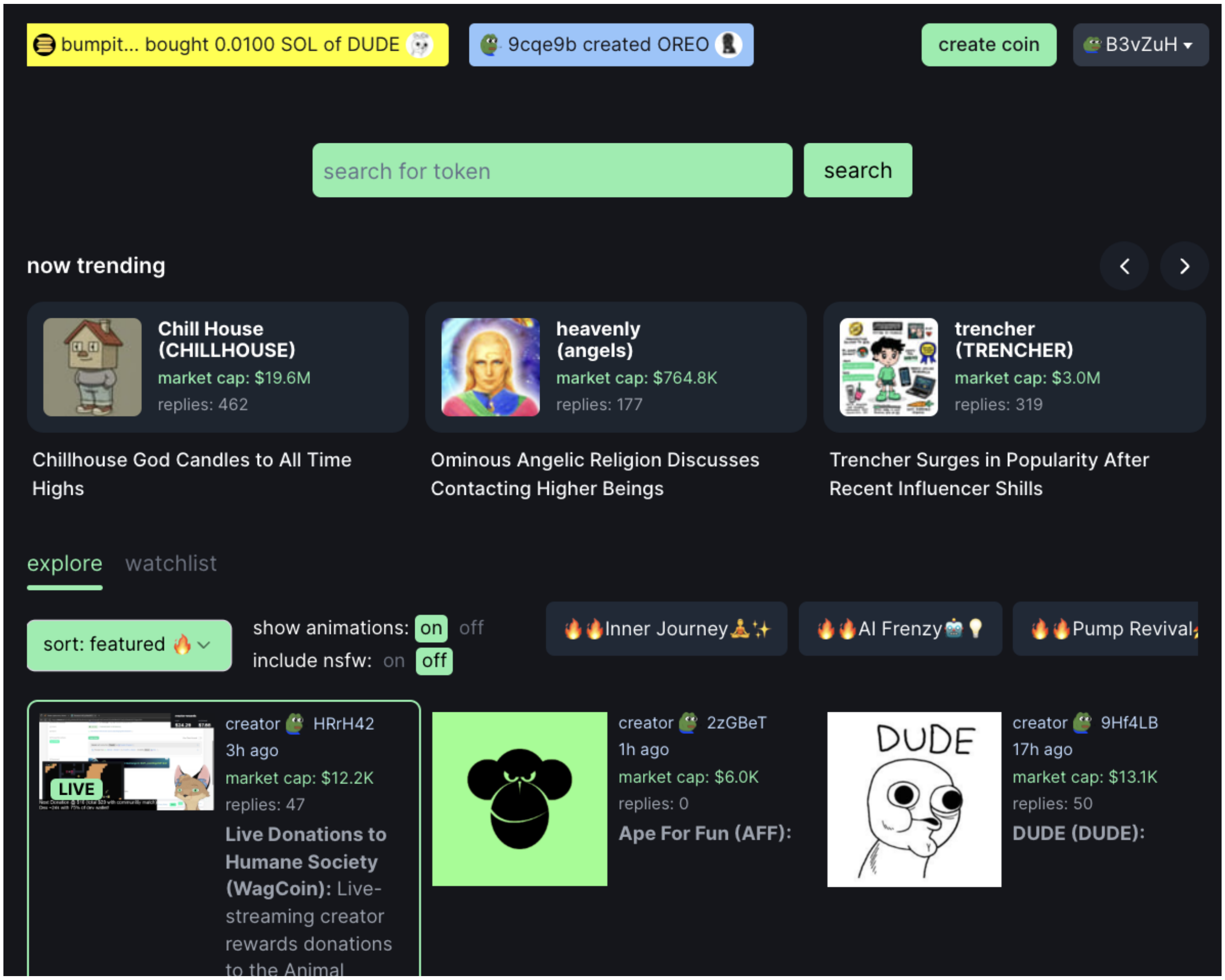


Figure 3: Pumpfun Frontpage Meme Coin Bumping

Each time a transaction occurs for a meme coin, it gets bumped to the front page and briefly jiggles, increasing its visibility and exposure.

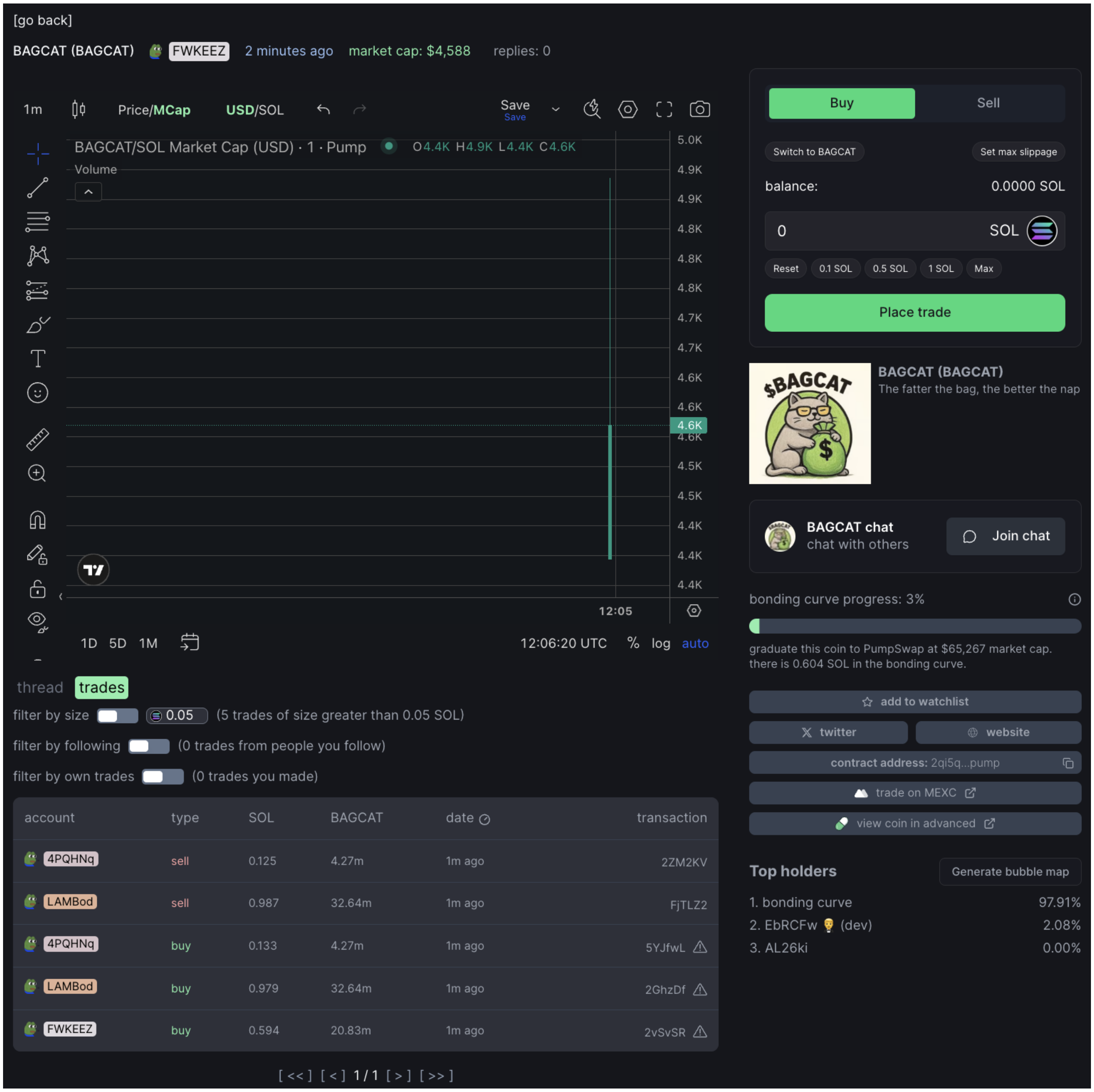
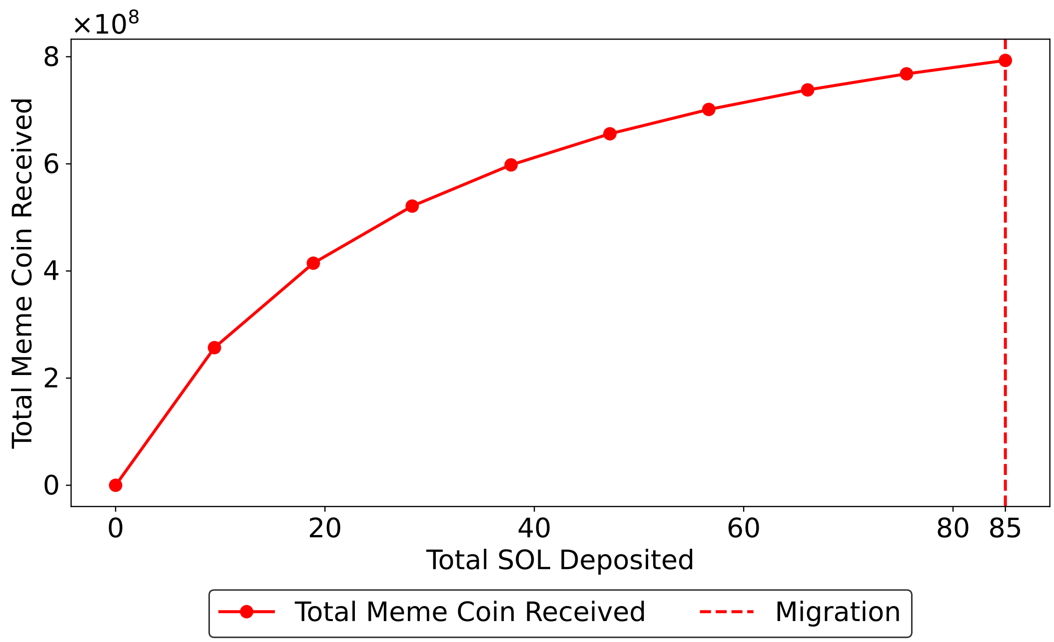
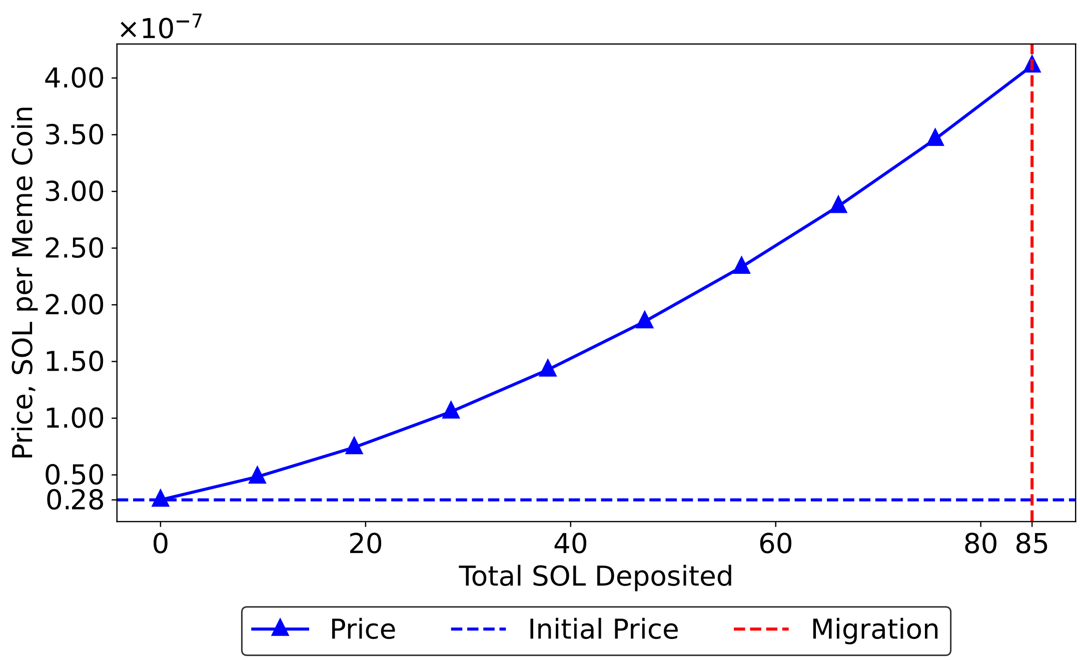


Figure 4: Pumpfun Trading Page

Pump.fun marks transaction bundles, groups of transactions occurring within the same block, as possible bot activity. On the right-hand side, it also displays token holders and their corresponding holdings.



(a) Relationship between the total SOL deposited by traders



(b) Relationship between the total meme coins received

Figure 5: Meme Coin Pricing Mechanism in Pumpfun

Red solid line describes the relationship between the total SOL deposited by traders and the total meme coins received in Pumpfun. From this, the price of the meme coin in terms of SOL in Pumpfun can also be derived (blue solid line). Blue and red dotted lines are the initial price and migration price.

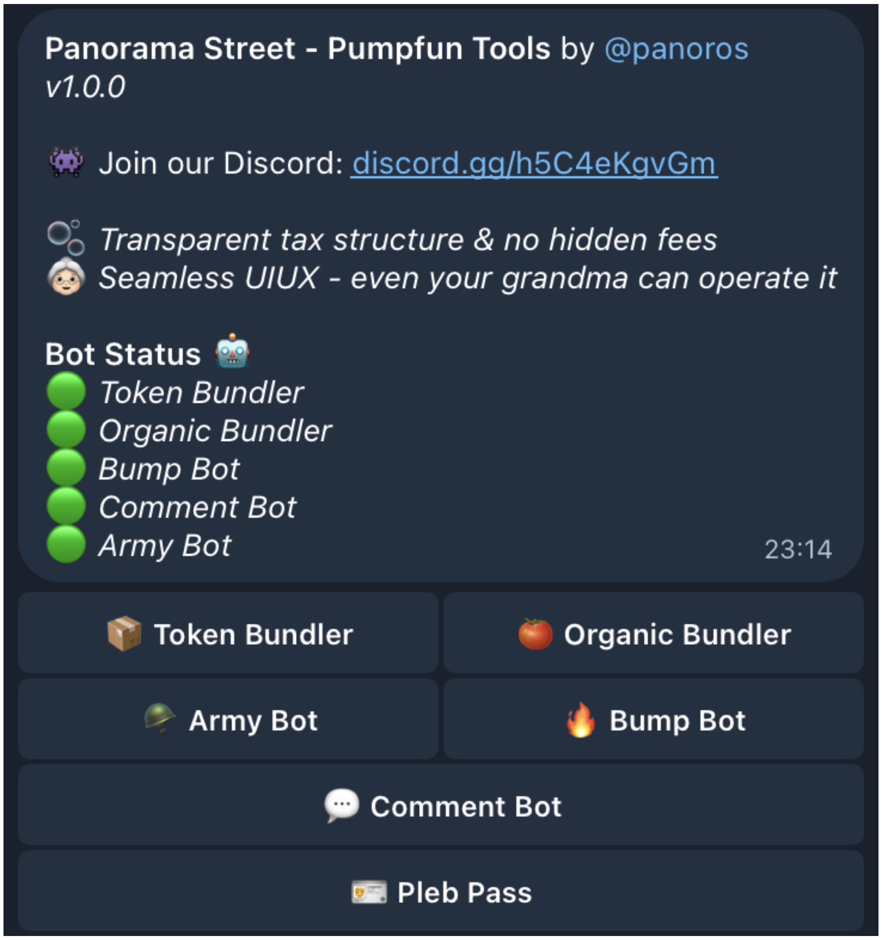
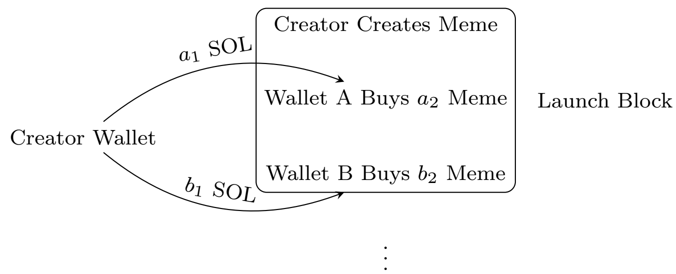
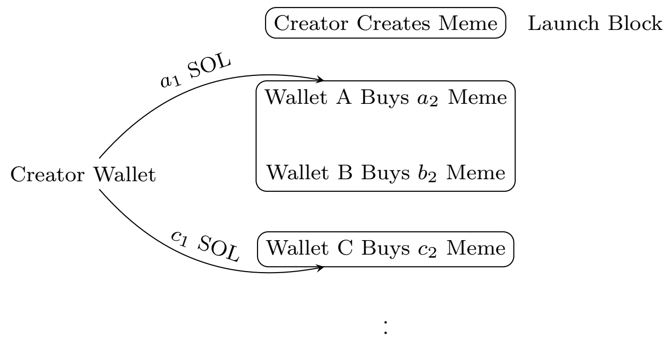


Figure 6: A Telegram bot for fabricated comments, wash trading, and creator position concealment

Bot providers offer Telegram bots as the user interface for different types of automated strategies. In the figure, the "Token Bundler" corresponds to the rat bot, the "Bump Bot" to the wash trading bot, and the "Comment Bot" to the comment bot.



(a) Naïve Rat Bot.



(b) Gradual Rat Bot.

Figure 7: Heuristics for Naïve and Gradual Rat Bots

This figure illustrates the heuristics for rat bots. The rat bot enables the creator to generate, fund, and control multiple wallets (e.g., Wallet A and Wallet B), which simultaneously buy the meme coin during the same creation block. This masks centralized ownership and creates an illusion of organic demand.

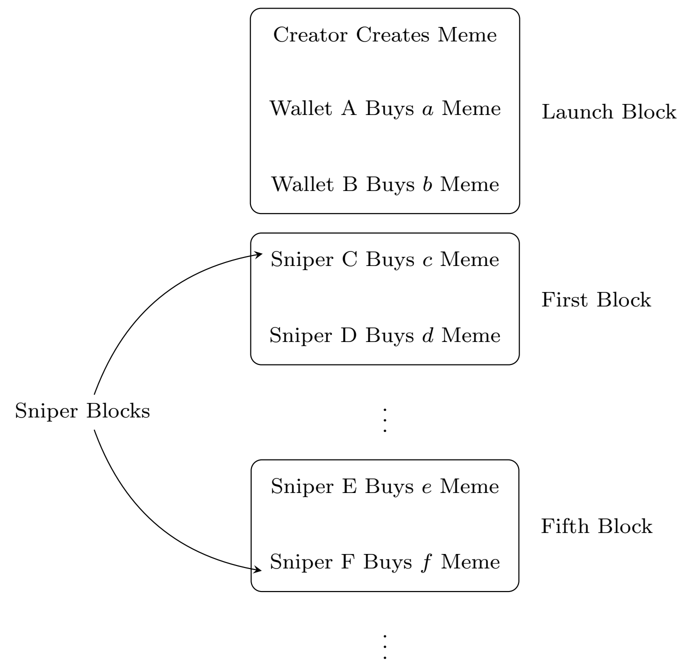
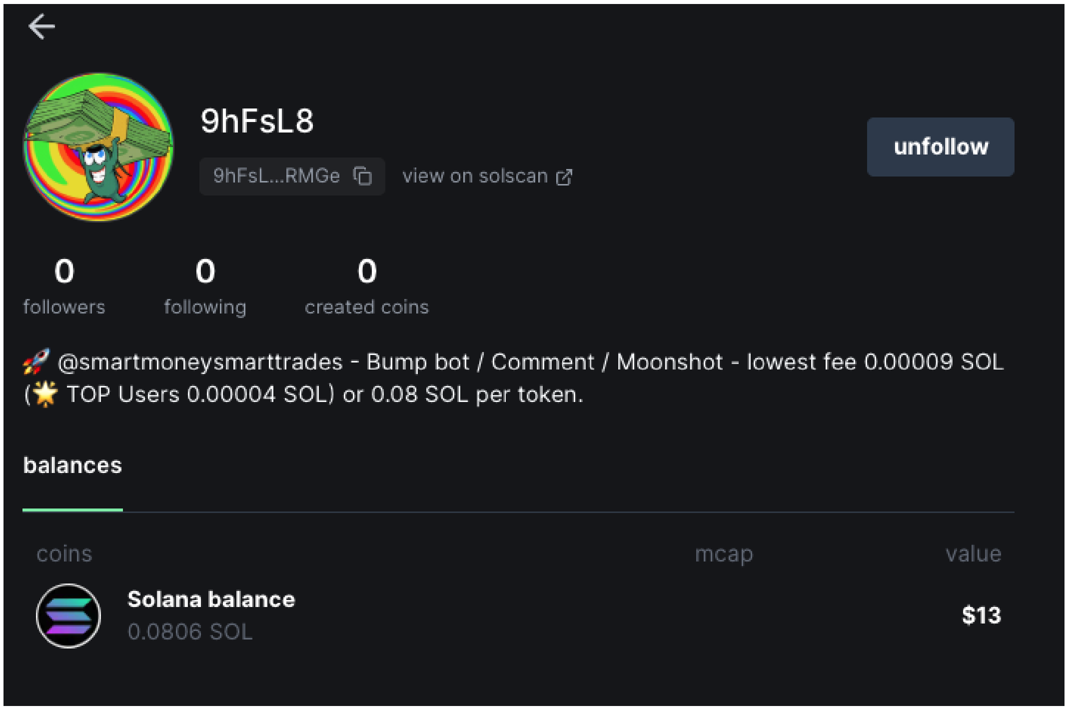
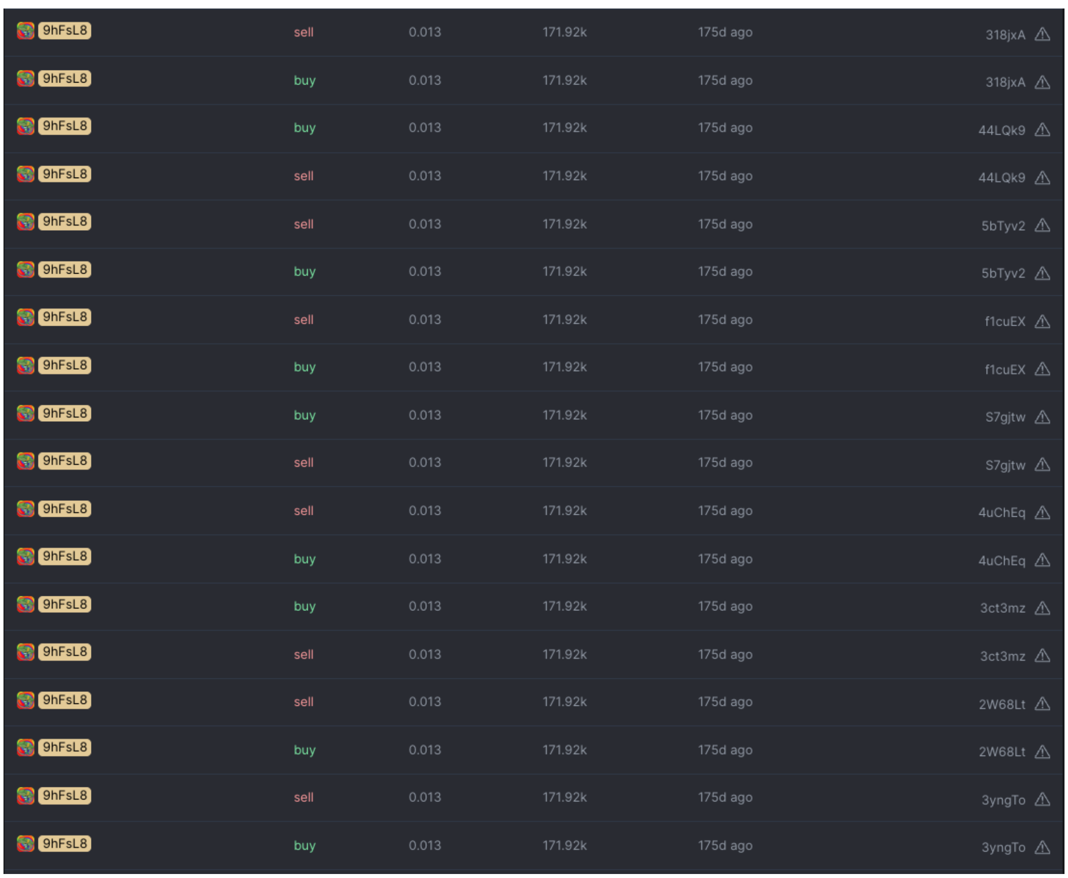


Figure 8: Heuristics for the Sniper Bot

This figure illustrates the heuristics for the sniper bot. The sniper bot enables non-creators to generate, fund, and control multiple wallets to buy meme coins within a very short time window since the coin launch (first to fifth blocks or 0.4 to 2 seconds).



(a) Profile



(b) Transaction History

Figure 9: Wash Trading Bot Identified by High Wash Trading Score

Table 1: Summary Statistics

This table presents summary statistics for the key variables that are used in our analyses. We resample three types of projects based on the real distribution. The variables are defined in the Appendix D.

Variable	Num. Obs.	Mean	Std. Dev.	P10	Median	P90
Rat Bot$_i$	6,000	0.24	0.43	0	0	1
Sniper Bot$_i$	6,000	0.84	0.37	0	1	1
Wash Trading Bot$_i$	6,000	0.04	0.19	0	0	0
Comment Bot$_i$	6,000	0.26	0.44	0	0	1
Ln(Max Ret)$_i$	6,000	0.38	0.63	0.02	0.14	1.01
Ln(Number of Traders)$_i$	6,000	2.51	1.08	1.61	2.20	3.99
Ln(Pre-Migration Duration)$_i$	2,000	7.39	2.72	5.09	6.45	11.07
Ln(Pump Duration)$_i$	6,000	3.40	3.28	0.69	2.30	7.77
Ln(Dump Duration)$_i$	6,000	4.53	2.67	1.61	4.20	8.01

Table 2: Meme Coin Projects' Characteristics

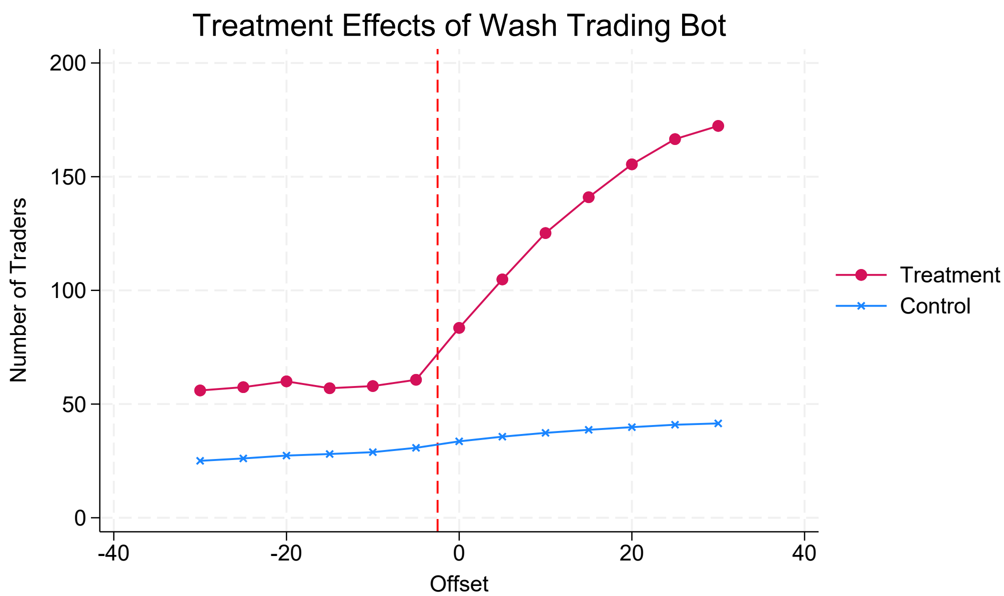
This table presents estimates of how bots affect the meme coin projects. The dependent variables are $Ln(Max\ Ret)_i$, which is maximum log return of the meme coin, $Ln(Number\ of\ Traders)_i$, which is the natural logarithm of the number of non-bot traders who control one or multiple wallets and trade the meme coin, $Ln(Pre\text{-}Migration\ Duration)_i$ is the natural logarithm of seconds between the migrated meme coin's launch and its migration, $Ln(Pump\ Duration)_i$ is the natural logarithm of seconds between the launch and the peak price of the meme coin, and $Ln(Dump\ Duration)_i$ is the natural logarithm of the seconds between the peak price and the 90% drop of the circulating supply. $Rat\ Bot_i$, $Sniper\ Bot_i$, $Wash\ Trading\ Bot_i$, and $Comment\ Bot_i$ are dummy variables indicating the presence of each bot type. We report t-statistics in parentheses. ***, **, and * denote the statistical significance levels at 1%, 5%, and 10%, respectively.

	$Ln(Max\ Ret)_i$	$Ln(Number\ of\ Traders)_i$	$Ln(Pre\text{-}Migration\ Duration)_i$	$Ln(Pump\ Duration)_i$	$Ln(Dump\ Duration)_i$
	(1)	(2)	(3)	(4)	(5)
$Rat\ Bot_i$	0.11***	0.35***	-0.93***	0.27**	-0.27***
	(5.15)	(10.56)	(-6.30)	(2.47)	(-3.14)
$Sniper\ Bot_i$	0.05**	0.36***	-1.08***	-0.13	0.15
	(2.26)	(9.25)	(-7.23)	(-1.04)	(1.51)
$Wash\ Trading\ Bot_i$	0.54***	1.45***	2.12***	2.77***	2.64***
	(12.34)	(20.61)	(15.66)	(12.11)	(14.31)
$Comment\ Bot_i$	0.25***	0.53***	1.09***	0.85***	0.79***
	(13.51)	(17.38)	(8.91)	(8.69)	(10.01)
Constant	0.22***	1.93***	7.22***	3.12***	4.16***
	(8.88)	(48.13)	(44.07)	(24.05)	(39.66)
Observations	6,000	6,000	2,000	6,000	6,000
R^2	0.08	0.17	0.20	0.05	0.07

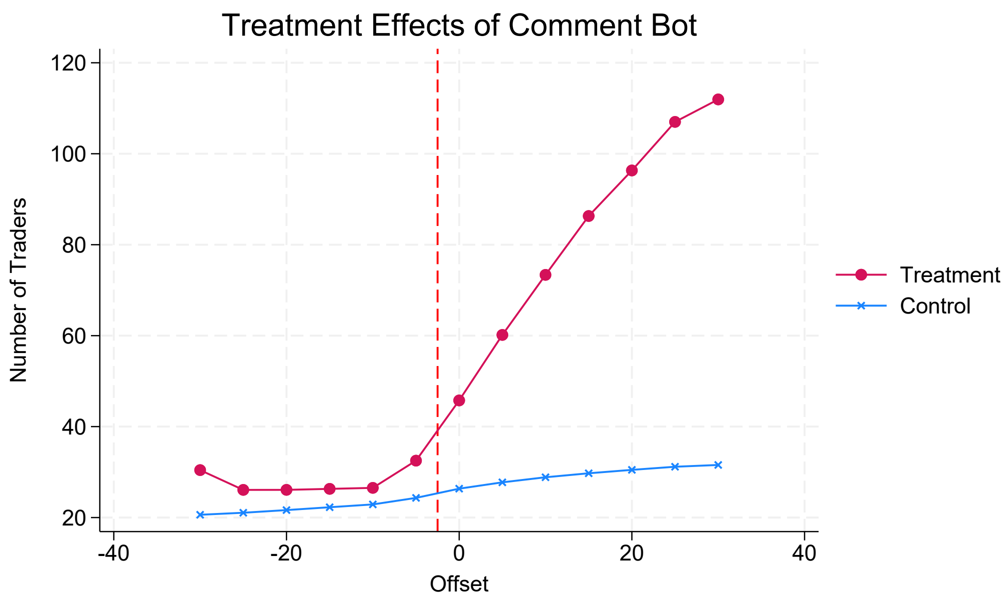
Table 3: Who Executes the Dump?

This table reports the identity of the market participant responsible for placing the largest sell order among the first ten sell transactions following the price peak.

	Rat Bot		Sniper Bot	
	No	Yes	No	Yes
Creator Dump	48.33%	62.10%	62.17%	49.67%
Sniper Dump	15.42%	4.06%	0.00%	15.06%
Other Dump	36.26%	33.84%	37.83%	35.27%



(a) Wash Trading Bot



(b) Comment Bot

Figure 10: Time Dynamics

This figure illustrates the effect of the initial introduction of a wash trading bot or a comment bot on the number of meme coin traders. The horizontal axis measures minutes relative to the bot introduction, and the vertical axis measures the number of unique traders.

Table 4: Stacked DID of Wash Trading Bot and Comment Bot

This table presents the relationship between the bots and the number of traders. The dependent variable $Number\ of\ Traders_{i,t}$ is the number of traders of project i at time t. $Treat_i$ is a binary variable that equals 1 if there are bots present in the project. $Post_{c,t}$ is a binary variable that equals 1 if the time is after the present of the bot and 0 otherwise. We report t-statistics based on standard errors that are clustered at the cohort-project level. ***, **, and * denote the statistical significance levels at 1%, 5%, and 10%, respectively.

	$Number\ of\ Traders_{i,t}$	
Bot :	Wash Trading Bot	Comment Bot
	(1)	(2)
$Treat_i \times Post_{c,t}$	0.685***	0.769***
	(6.64)	(9.27)
Observations	10,208,466	6,284,375
Pseudo R^2	0.937	0.928
Cohort-Time FE	Y	Y
Cohort-Project FE	Y	Y

Table 5: Profits of Creators and Non-Creators

This table presents estimates of how bots affect the profits of creators and non-creators in the full sample. The dependent variable $Profit_{i,j}$ is the profit and loss of trader j in a meme coin project i. $Rat\ Bot_i$, $Sniper\ Bot_i$, $Wash\ Trading\ Bot_i$, and $Comment\ Bot_i$ are dummy variables indicating the presence of each bot type. We report t-statistics based on standard errors that are clustered at the project level. ***, **, and * denote the statistical significance levels at 1%, 5%, and 10%, respectively.

	$Profit_{i,j}$			
Bot :	$Rat\ Bot_i$	$Sniper\ Bot_i$	$Wash\ Trading\ Bot_i$	$Comment\ Bot_i$
	(1)	(2)	(3)	(4)
$Creator_{i,j}$	103.69***	186.58***	124.32***	131.87***
	(9.06)	(6.30)	(17.14)	(12.43)
$Creator_{i,j} \times Bot_i$	161.29***	-54.28*	431.76**	5.74***
	(5.95)	(-1.72)	(2.05)	(2.68)
$Non\text{-}Creator_{i,j}$	-5.27***	-5.58***	-6.37***	-5.89***
	(-9.30)	(-5.01)	(-23.61)	(-12.49)
$Non\text{-}Creator_{i,j} \times Bot_i$	-1.48*	-0.05	1.90*	0.02**
	(-1.80)	(-0.04)	(1.78)	(2.44)
Observations	4,353,316	4,353,316	4,353,316	4,353,316
R^2	0.00	0.00	0.00	0.00

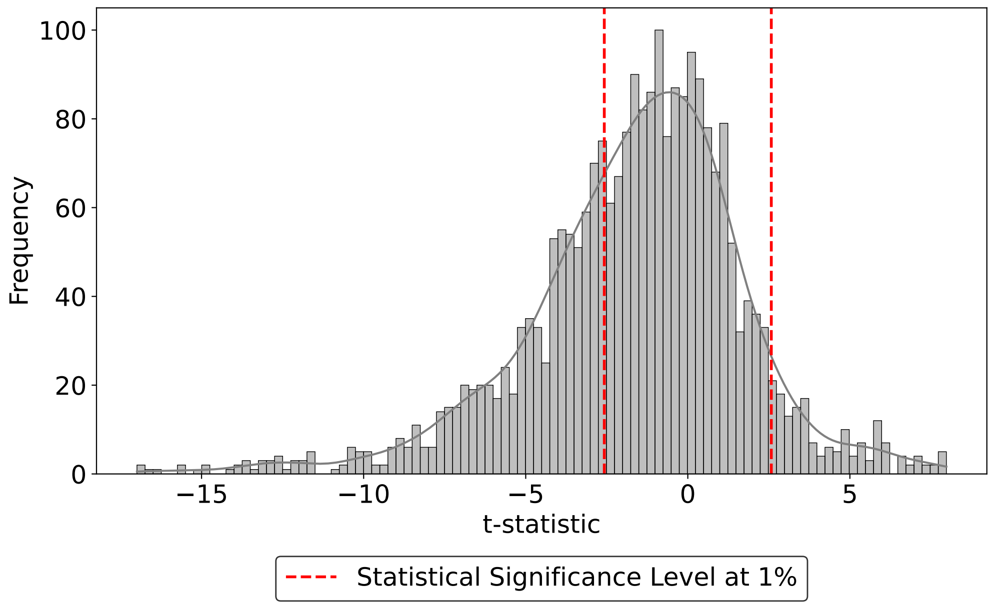


Figure 11: Distribution of T-Statistic for Meme Coin Traders' Historical Project Returns

This figure illustrates the distribution of t-stats for the historical project returns of meme coin traders. We select traders who have participated in fewer than 1,000 projects to exclude algorithmic traders. For each trader, we first compute project-level returns by dividing total profits by the total costs within each meme coin project in which the trader participated. Then, we calculate the t-stat for the set of project returns associated with each trader.

Table 6: Winner, Loser, and Neutal Trader Proportion in Creators and Snipers This table reports the winner, normal, and loser trader count in creators and snipers. The percentage in parentheses is the ratio of creators or snipers within the trader group.

Trader Type	Total	Creator	Sniper
Winner	182	35 (19.2%)	29 (16%)
Normal	1,436	63 (4.4%)	47 (3.3%)
Loser	830	7 (0.8%)	8 (1.0%)
Total	2,448	105	84

Table 7: Winner and Loser Trader Meme Coin Project Choice

This table reports the winner, normal, and loser trader participated meme coin project characteristics, aggregated at the trader level. The variables are defined in the Appendix D.

	Mean		
	Winner	Neutral	Loser
$Rat\ Bot_i$	0.20	0.25	0.28
$Sniper\ Bot_i$	0.90	0.84	0.83
$Wash\ Trading\ Bot_i$	0.16	0.22	0.28
$Comment\ Bot_i$	0.45	0.57	0.58
$Ln(Max\ Ret)_i$	1.60	2.78	3.30
$Ln(Number\ of\ Traders)_i$	4.26	5.64	6.18
$Ln(Pre\text{-}Migration\ Duration)_i$	7.68	7.52	7.34
$Ln(Pump\ Duration)_i$	5.07	6.44	6.91
$Ln(Dump\ Duration)_i$	6.27	7.58	8.05
$Creator_i$	0.19	0.04	0.01
$Sniper_i$	0.16	0.03	0.01
Observations	182	1,436	830

Table 8: Who Executes the Dump?

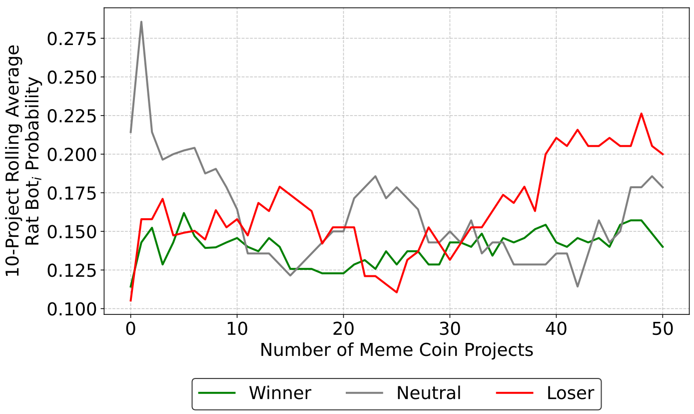
This table reports the identity of the market participant responsible for placing the largest sell order among the first ten sell transactions following the price peak.

	No One Cared	Unsucessful	Migrated
Winner Dump	50.84%	57.30%	0.00%
Loser Dump	9.50%	3.37%	0.00%
Neutral Dump	39.66%	39.33%	100.00%

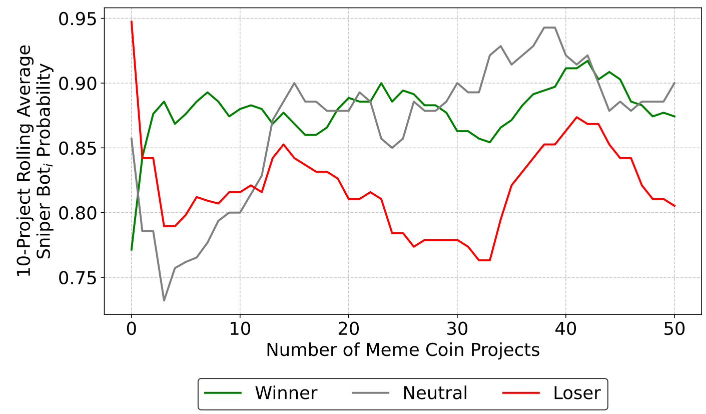
Table 9: Exit Probability

This table presents regression estimates of how past returns influence the exit probability of different trader groups. The dependent variable $Exit_{i,t}$ is a dummy variable equal to one if trader j in project i does not make another trade within one month after the t-th order transaction. The independent variables $\bar{r}_{i,t-m,t-n}$ represent average project returns between the $(t-m)$-th and $(t-n)$-th orders. Column (1) includes all traders, while Columns (2)–(4) report results separately for Winners, Neutrals, and Losers. ***, **, and * denote statistical significance at the 1%, 5%, and 10% levels, respectively.

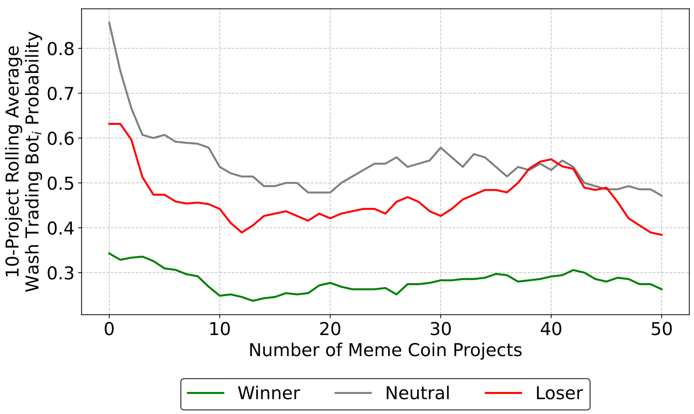
	$Exit_{i,t}$			
	All	Winner	Neutral	Loser
	(1)	(2)	(3)	(4)
$\bar{r}_{i,t-15,0}$	0.0002	0.0054***	0.0000	0.0004
	(0.0002)	(0.0016)	(0.0002)	(0.0007)
$\bar{r}_{i,t-15,t-10}$	-0.0002**	0.0001	-0.0002*	-0.0008**
	(0.0001)	(0.0006)	(0.0001)	(0.0003)
$\bar{r}_{i,t-10,t-5}$	-0.0003***	-0.0001	-0.0002**	-0.0014***
	(0.0001)	(0.0006)	(0.0001)	(0.0003)
$\bar{r}_{i,t-5,t-1}$	-0.0004***	-0.0000	-0.0003***	-0.0026***
	(0.0001)	(0.0005)	(0.0001)	(0.0003)
Constant	0.0032***	0.0016***	0.0040***	0.0019***
	(0.0001)	(0.0003)	(0.0001)	(0.0002)
Observations	561,888	67,143	266,558	227,127
R^2	0.0001	0.0002	0.0001	0.0006



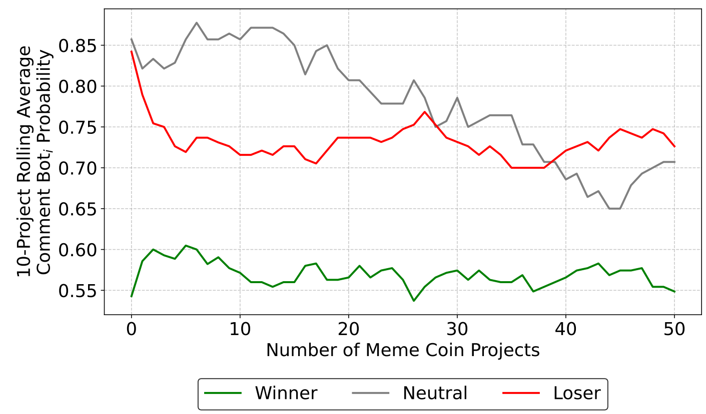
(a) Rat bot



(b) Sniper bot



(c) Wash trading bot



(d) Comment bot

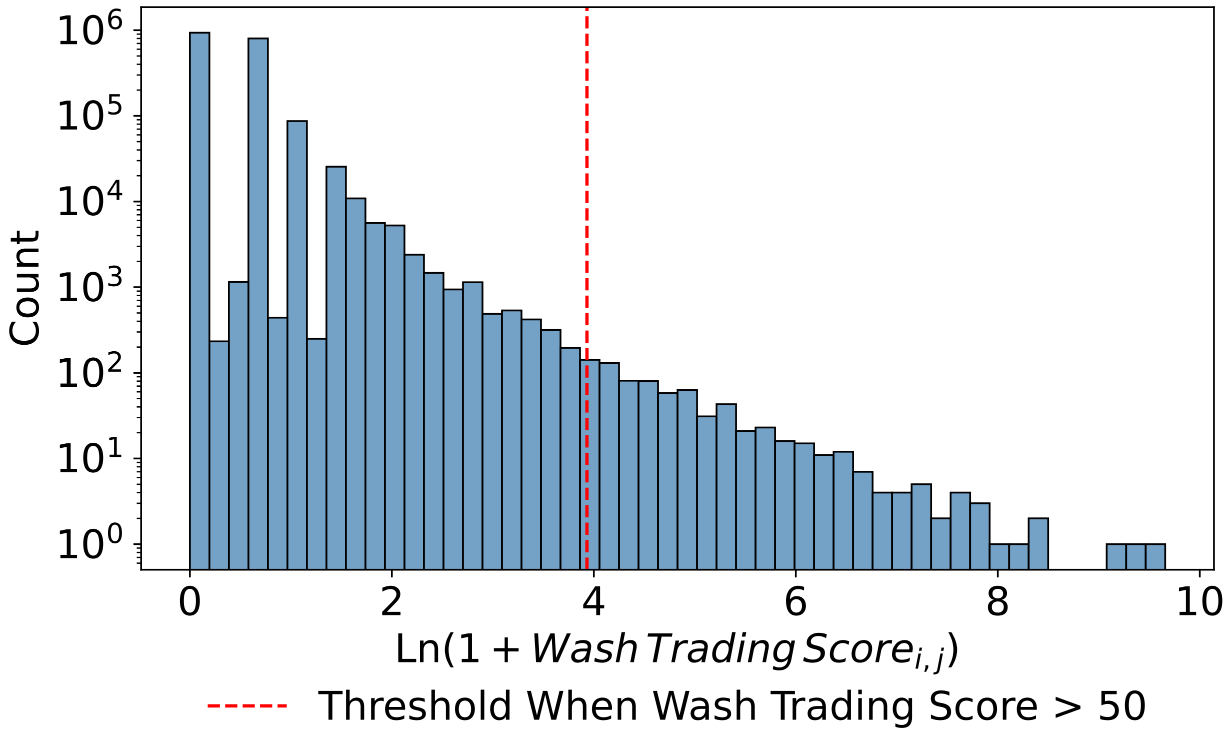
Figure 12: Learning Curve of Winners and Non-Winners in terms of Bots

The y-axis plots the 10-project moving average of the proportion of projects with a specific type of bots. The x-axis denotes the number of meme coin projects. For the initial 10 projects, a simple average is used instead of a rolling average.

A Distribution of Three Types of Projects

Project Type	Pre-Trump (%)	Post-Trump (%)
No one cared	67.2	58.4
Unmigrated	31.1	40.7
Migrated	1.7	0.9
Total	100.0	100.0

B Distribution and Threshold of Wash Trading Score



C LLM Prompt to Detect Comment Bot

C.1 System Instruction

You are a meme coin comment analyzer. Your task is to classify a given comment as bot-generated or human-generated, and assess its sentiment (positive or negative) toward the coin. Bot-generated comments are often short, context-less, and mass-producible slogans that express hype or hostility. In contrast, human-generated comments tend to be more personalized and nuanced, containing context or opinion with reasoning. Comments that reference other users (e.g., #89009679) are typically human-generated, although not all human comments contain such references. Respond with id and true (if the comment is bot-generated) or false (if it is human-generated). Your response should follow this format: '{"bot": <true/false>, "sentiment": <positive/negative>}'

C.2 Prompt for Few-Shot Learning

Table 10 presents examples used in the LLM few-shot learning prompt. These examples are provided in the LLM's context to help it learn the semantic differences between bot-generated and human-generated comments. Then, the comment to be classified is appended.

Table 10: Few-Shot Learning Prompt Examples

User	Assistant	
	bot	**sentiment**
ill pay for dex	true	positive
let the jeeters go and next stop the fuckin moon	true	positive
send it to the moon	true	positive
TO THE MOON!!! READYY	true	positive
Go on jeet now! We want no paper hands. The community is holding strong and we'll get there! LFG	true	positive
this is going to send	true	positive
Dev is selling	true	negative
Worst token	true	negative
Run	true	negative
I will send 5 sol to the one who CTO	true	negative
#89009679 Lmao WATCH THIS RISES TO MEGA MILLIONS	false	positive
#89016342 what in the actual fuck is this atrocity?	false	negative
scam mfq I invested 8 BUCKS MF I'm not stupid u won't be rich by my 4 USD thejacket MF	false	negative
#88857219 show screenshot as proof pls?	false	negative
Fake web bros, not same ca	false	negative
Obvious scam but derivatives of trump coin should go up as people who missed the boat buy into the hype	false	negative

D Variable Description

Variable	Description
$Rat\ Bot_i$	Dummy variable equal to 1 if the meme coin has rat bot, 0 otherwise.
$Sniper\ Bot_i$	Dummy variable equal to 1 if the meme coin has sniper bot, 0 otherwise.
$Wash\ Trading\ Bot_i$	Dummy variable equal to 1 if the meme coin has wash trading bot, 0 otherwise.
$Comment\ Bot_i$	Dummy variable equal to 1 if the meme coin has comment bot, 0 otherwise.
$Ln(Max\ Ret)_i$	Maximum log return of the meme coin.
$Ln(Pre\text{-}Migration\ Duration)_i$	Natural logarithm of seconds between the migrated meme coin's launch and its migration.
$Ln(Pump\ Duration)_i$	Natural logarithm of seconds between the launch and the peak price of the meme coin.
$Ln(Dump\ Duration)_i$	Natural logarithm of seconds between the peak price and the 90% drop of the circulating supply.
$Ln(Number\ of\ Traders)_i$	Natural logarithm of the number of non-bot traders who control one or multiple wallets and trade the meme coin.
$Profit_{i,j}$	Each trader's profit and loss from a given meme coin
$Creator_i$	Dummy variable equal to 1 if the trader is the meme coin's creator, 0 otherwise.
$Sniper_i$	Dummy variable equal to 1 if the trader is a sniper bot, 0 otherwise.
$Winner_i$	Dummy variable equal to 1 if the t-statistic of the trader's profit is greater than 2.576, 0 otherwise.
$Loser_i$	Dummy variable equal to 1 if the t-statistic of the trader's profit is less than -2.576, 0 otherwise.
$Neutral_i$	Dummy variable equal to 1 if the t-statistic of the trader's profit is between -2.576 and 2.576, 0 otherwise.

E Variance Inflation Factor

The VIF analysis suggests that multicollinearity is not a concern among the bot dummy variables (average VIF smaller than 5).

Variable	VIF	1/VIF
$Rat\ Bot_i$	1.19	0.84
$Sniper\ Bot_i$	1.79	0.56
$Wash\ Trading\ Bot_i$	1.30	0.77
$Comment\ Bot_i$	2.04	0.49
Mean	1.58	0.67