# A Sea of Coins: The Cryptocurrencies Proliferation in UniswapV2

Manuel Naviglio[1], **Francesco Tarantelli**[2], Fabrizio Lillo[1,2]
[1]Scuola Normale Superiore, Pisa, Italy,
[2]Dipartimento di Matematica, Università di Bologna, Bologna, Italy

francesco.tarantell3@unibo.it

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Finanziato dall'Unione europea
NextGenerationEU

MUR
Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

# Introduction



- The rise of DEXs like Uniswap has enabled the creation and trading of a huge number of new cryptocoins, revolutionizing how assets are exchanged globally.

- We analyze financial properties (e.g., returns, liquidity) and security challenges (e.g., smart contract risks, transaction vulnerabilities) to understand the dynamics of these tokens.

# Blockchain

A blockchain is a distributed ledger maintained by a network of computers (nodes).

**Key Properties:** Decentralization, immutability, and transparency.

Each block holds a batch of transactions and a reference (hash) to the previous block.

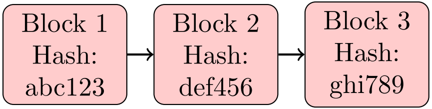


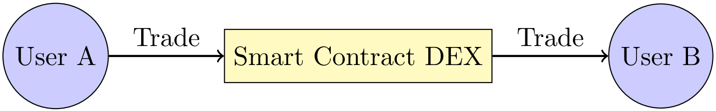

Figure 1: Blockchain

# DEX



- **What is a Decentralized Exchange (DEX)?**
  A peer-to-peer marketplace where crypto traders transact directly, eliminating the need for a central intermediary.

- **Smart Contracts – The Backbone of DEXs:**
  Automated code programs that facilitate trades, handle funds, and ensure trustless transactions. These self-executing contracts operate based on predefined rules coded into them.



# AMM & Mempool



Automated Market Makers (AMMs): A type of DEX that uses a mathematical formula to price assets ( **Constant Product Formula**);

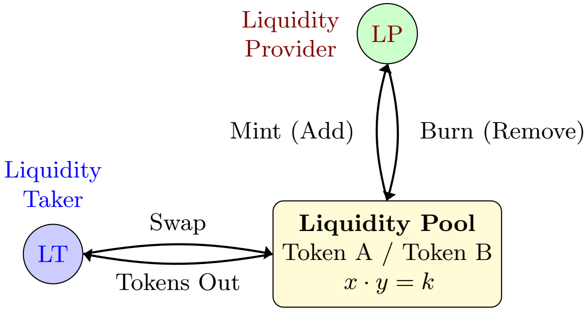
A mempool is a node's "waiting room" for unconfirmed transactions before they're added to a block.

Miners or validators typically pick transactions offering higher gas fees first (in many blockchains).During high activity, the mempool grows, increasing gas fees and wait times.

# UniswapV2



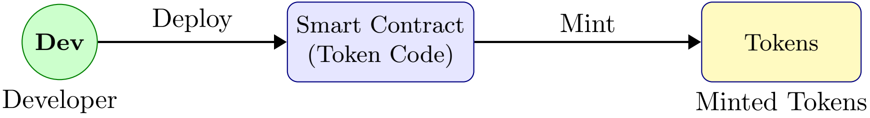
- **Automated Market Maker (AMM)** using the formula between the quantity of Token A $x$ and of Token B $y$ in the reserves: $x \cdot y = k$.
- **Pool Pair**: Smart contract holding two tokens (e.g., Token A, Token B).



# Token Creation



- **Smart Contract Code**: Written in a language (e.g., Solidity).
- **Deploy to Blockchain**: The contract is published (deployed) on an EVM-compatible chain (e.g., Ethereum).
- **Minting**: Upon deployment or via a function call, the token supply is created (minted).
- **Token Standard (e.g., ERC-20)**: Defines functions like `totalSupply()`, `balanceOf()`, `transfer()`, etc.
- **Ownership & Distribution**: Deployer or specified addresses receive the initial tokens.



# Token Creation Smart Contract



```
contract DOPE is Context, IERC20, Ownable {
    mapping (address => uint256) private _balances;
    address payable private _taxWallet;
    uint8 private constant _decimals = 9;
    uint256 private constant _tTotal = 1000000000 * 10**_decimals;
    string private constant _name = unicode'Decentralization obligatory, practicality
↪ essential';
    string private constant _symbol = unicode'DOPE';
    // ... other variables ...
    constructor () {
        _taxWallet = payable(_msgSender());
        _balances[_msgSender()] = _tTotal;
       emit Transfer(address(0), _msgSender(), _tTotal);
    }
    function symbol() public pure returns (string memory) {  return _symbol;}
    function totalSupply() public pure override returns (uint256) { return _tTotal;}
    // ... other functions ...
    function _transfer(address from, address to, uint256 amount) private {
        // Swap Rule function
    }
}
```

# Dataset Description



Retrieve information on:

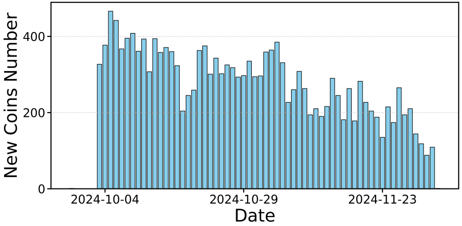
- Newly created pairs (pair contract address, token0, token1) $\leftrightarrow$ Token creation
- Swaps (amount swapped in/out) $\rightarrow$ Price behavior
- Mints and Burns (amount minted/burned, liquidity tokens) $\rightarrow$ Liquidity behavior

**Approach:**

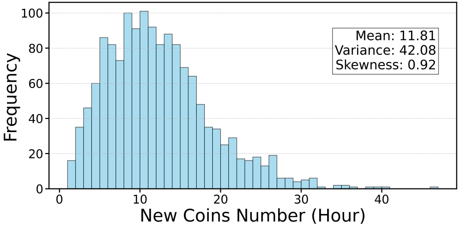
1. **Connect to an Ethereum Node:** - Use a node or an RPC provider (e.g., Infura, Alchemy) via web3.
2. **Data:** - Use 51582 new coins created between the 10th October 2024 and the 2nd of December 2024.

# Number of New Tokens Created






(a) Number of tokens created in a day for 2 months

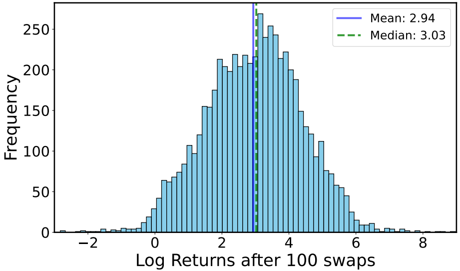



(b) Distribution of the new token number in one hour.

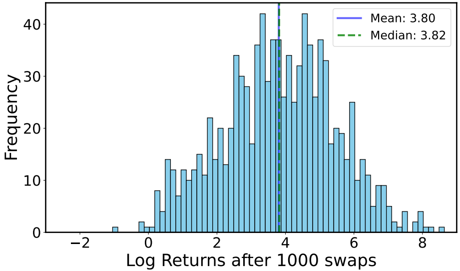
We focused on the cases in which one of the two pairs of the pool was WETH.

# Positive Returns






(a) Histogram of lagged log returns after 100 swaps.

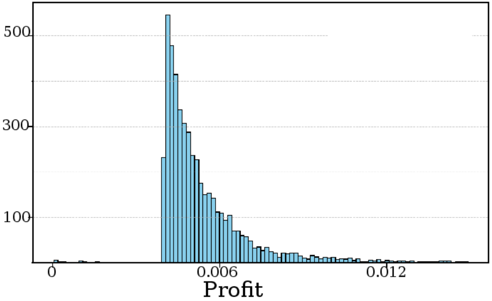



(b) Histogram of lagged log returns after 1000 swaps.

Figure 3: Histograms of the lagged log returns of the new tokens after different number of swaps. We stress that we have a medium percent returns value $\langle \boldsymbol{\eta} \rangle = \mathbf{10^5}\%$.

# Profit distribution for a simple Buy & Hold strategy



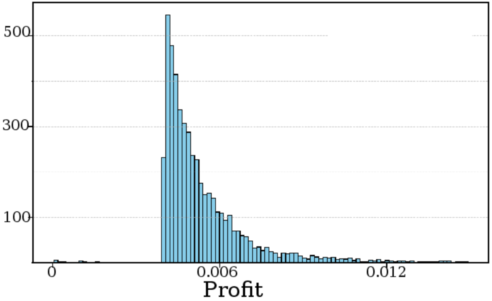



We do not have considered:

1. Are the purchased tokens secure?
2. Presence of sandwich attacks
3. Swap time instead Physical time

# Profit distribution for a simple Buy & Hold strategy






We do not have considered:

1. **Are the purchased tokens secure?**
2. Presence of sandwich attacks
3. Swap time instead Physical time

# Not All Tokens Are Sellable: Honeypots



**Definition 1 (Honeypot)**
*A honeypot is a smart contract that pretends to leak its funds to an arbitrary user (victim), provided that the user sends additional funds to it. However, the funds provided by the user will be trapped and at most the honeypot creator (attacker) will be able to retrieve them.*[1]

**Possible Scam Honeypot Elements:**

- Ownership & Renouncement
- Liquidity Addition
- Minting & Burning
- Slippage Tax Modifiable
- Blacklist/Whitelist
- Token Supply
- Swapping Mechanism
- Taxes & Fees
- Liquidity Burned/Locked

[1] Christof Ferreira Torres, Mathis Steichen, and Radu State. The art of the scam: Demystifying honeypots in ethereum smart contracts, USENIX Security 19, pages 1591-1607.

# Token Classification: Safe vs. Honeypot



**Motivation:** Not all tokens remain safe throughout their lifecycle. Some evolve into honeypot contracts, making them effectively unsellable.

**Our Approach:**

- **Time-Based Analysis:** Observe each token over its entire history.
- **Security Detector Queries:** Use external security checks from:
  - honeypot.is — Honeypot detection service;
  - GoPlus Labs — Token security scoring;
- **Final Categorization:** At the end of its observed lifetime, each token is classified as:
  - **Safe**: No honeypot behavior detected $\sim 6240$ tokens $\simeq 12\%$ of the total;
  - **Honeypot**: Displays locked or blocked selling $\sim 45342$ tokens $\simeq 88\%$ of the total.

# Token Capitalization



The Token Capitalization (TK) of the token $i$ is given by:

$$\mathrm{TK}_t^i = \left( \sum_{t=1}^{N_t} v_t^i \right) \cdot p^i(t = N_t) \tag{1}$$

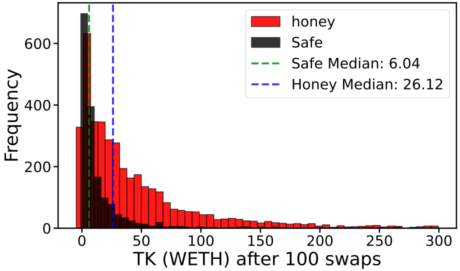
where:

- $v_t^i$ represents the signed token amount exchanged in the $t$-th swap for the $i$-th token;
- $N_t$ is the number of swaps considered;
- $p^i(t = N_t)$ is the token price recorded at the $N_t$-th swap for the $i$-th token.

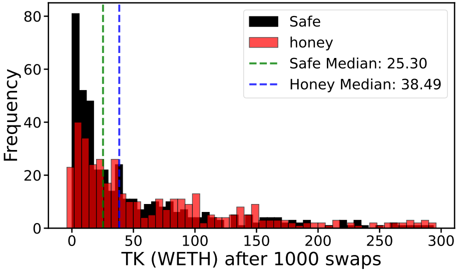
We are interested in the amount of money circulating around these new tokens.

# TK Distribution






(a) Histogram of capitalization after 100 swaps.




(b) Histogram of capitalization after 1000 swaps.

Figure 4: Histograms of the capitalization of the new tokens after different number of swaps.

# TK Property



TK is also an indicator of a sudden lack of liquidity in the market that is artificially triggered.

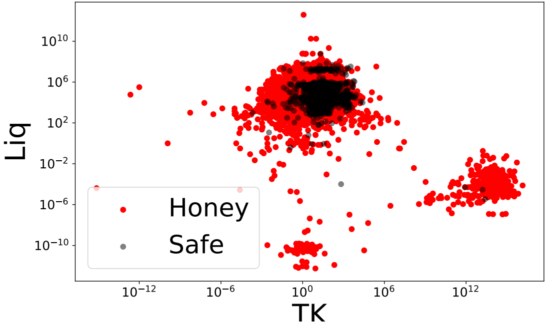
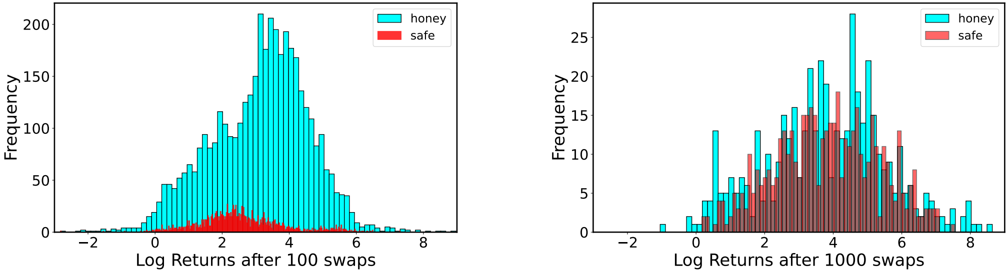



Figure 5: Scatter plot between the maximum value of TK in the lifetime of all new tokens and the corresponding value of Liquidity in that time.

# Honeypot Returns





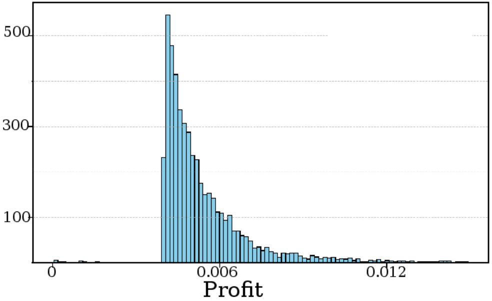

(a) Histogram of lagged log returns after 100 swaps.

(b) Histogram of lagged log returns after 1000 swaps.

Figure 6: Histograms of the lagged log returns of new safe and honey tokens.

# Profit distribution for a simple Buy & Hold strategy






We do not have considered:

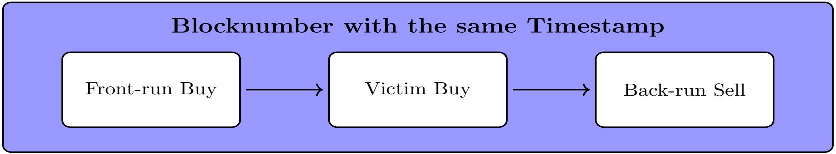
1. Are the purchased tokens secure?
2. **Presence of sandwich attacks**
3. Swap time instead Physical time

# Sandwich Strategy Overview



High Profit for the presence of sandwich whose Txs are deployed in the same blocknumber.

- **Mempool Monitoring:** Identify a pending swap transaction (buying token $y$ for some ETH $x$).
- **Front-Run Attack:** Submit a transaction with higher gas to buy token $y$ first.
- **Victim's Purchase:** The original, slower transaction then occurs, pushing the price up.
- **Back-Run Attack:** Finally, sell the token $y$ at the new higher price, capturing profit.



# Optimal Sandwich



With a initial reserve $x$ of the ETH in the LP in the pair:

- The first swap of the front-run attack sells $\Delta x = a$ ETH;
- The swap of the slower trader sells $\Delta x_\epsilon = \epsilon$ ETH;
- The second swap of the back-run attack sells all of his tokens bought and obtains $\Delta x_{tot}$ ETH.

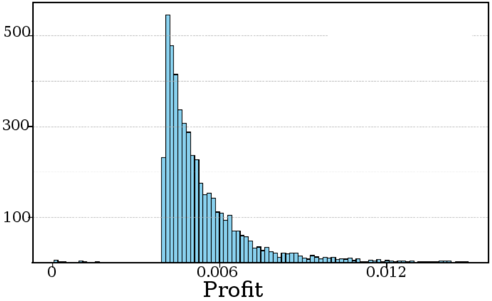
The maximum of the gain $s = \Delta x_{tot} - a$, with a pair pool fee $r = 1 - f$ ($f = 0.3\% \ll 1$) and in the approx $fa \simeq O(\epsilon)$:

$$a_{\max} = \frac{\epsilon}{f} - x \; ; \tag{2}$$

$$s_{\max} = \frac{(\epsilon - fx)^2}{\epsilon} \; , \tag{3}$$

# Profit distribution for a simple Buy & Hold strategy





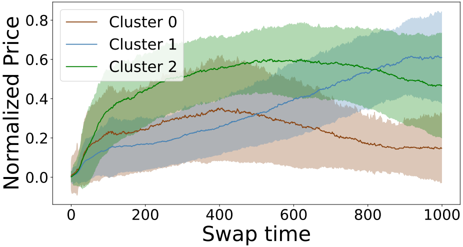

We do not have considered:

1. Are the purchased tokens secure?
2. Presence of sandwich attacks
3. **Swap time instead Physical time**

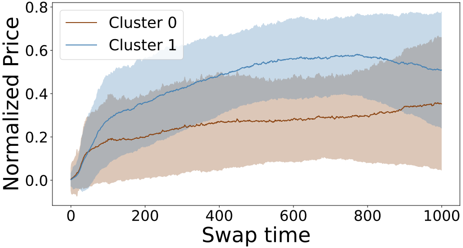
# Clustering in swap time



The k-means algorithm partitions the trajectories into $k$ distinct clusters by minimizing the within-cluster variance, thereby grouping similar trajectories based on their features.




(a) Clustering resulting trajectories for $k = 3$.




(b) Clustering resulting trajectories for $k = 2$

Figure 7: The figures show the mean and the one standard deviation bands of the trajectories of length $10^3$ resulting from the clustering, with $k = 2$ and $k = 3$.

# Clustering in physical time



Now, we applied a dynamic time warping (DTW) method to categorize tokens into clusters based on their temporal price evolution.

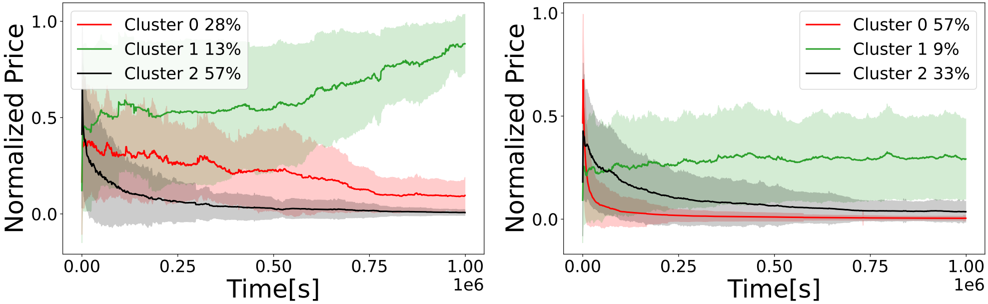



Figure 8: At left, the Honeypot Tokens; instead, at right, the Safe Tokens.

The initial pick is associated with the BOT presence.

# Conclusions



- The positive profits of new token returns suggest it may be convenient to trade in this currency.

- However, traders must consider:
    - The presence of honeypots.
    - The risk of sandwich attacks.
    - The necessity of a physical time strategy.

- The emergence of BOT activity significantly reduces potential profits, making careful strategic planning essential.

# Optimal Sandwich: Example



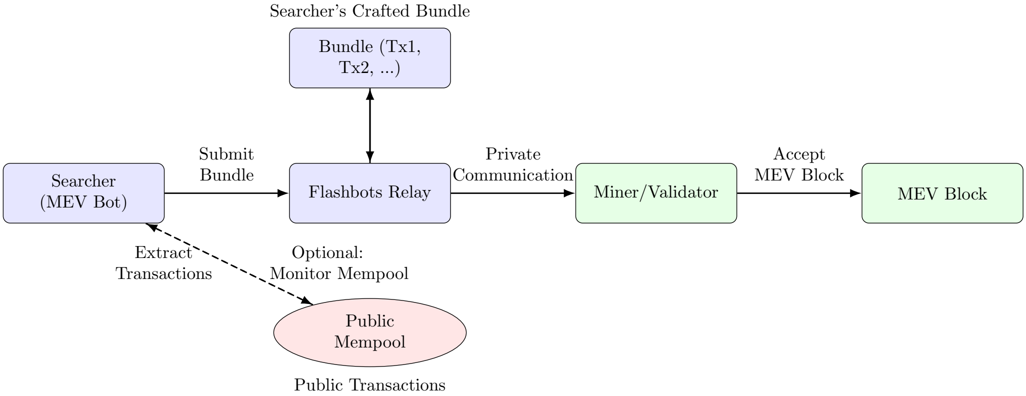
| $\Delta x$ | $\Delta y$ | $x_{\text{new}}$ | $y_{\text{new}}$ | **New Price** | **Old Price** | **Liquidity** |
|---|---|---|---|---|---|---|
| 25.0000 | -268464000 | 28.3199 | 35758537 | 7.920e-07 | 1.091e-08 | 31823 |
| 0.0500 | -62833 | 28.3699 | 35695704 | 7.948e-07 | 7.920e-07 | 31823 |
| -25.0316 | 268463996 | 3.3382 | 304159700 | 1.098e-08 | 7.948e-07 | 31865 |

In this case, $a_{\max} = 10$ ETH and $s_{\max} = 0.032$ ETH, with an initial reserve $x = 3.32$ ETH.

Sandwich on the token AIDLE in the pair WETH/AIDLE executed in the block number $21560868$.

# PNL




Searcher's Crafted Bundle
Bundle (Tx1, Tx2, ...)
Searcher (MEV Bot)
Submit Bundle
Flashbots Relay
Private Communication
Miner/Validator
Accept MEV Block
MEV Block
Extract Transactions
Optional: Monitor Mempool
Public Mempool
Public Transactions