Whitepaper

Fama One

AI powered investment vaults

Authored by Seif Khedija & Adam Greenberg

7th April 2023

1. Background

Financial markets are complex and given the differences of crypto assets compared to commonplace web2 technologies, this often presents an additional layer of complexity for users. Many people don't have the skill or experience to successfully navigate financial and crypto markets and suffer losses as a result.

With the 24/7/365 nature of Crypto markets, even those with experience can struggle to keep on top of their ideal strategies and execution.

With the increase of incidents in crypto custodianship in recent times, more people are looking for non-custodial and decentralized solutions.

Furthermore, we believe that the recent acceleration of Artificial Intelligence (AI) advancement is widening the tooling and data gap between well resourced people and institutions compared to the general public. This limits the opportunity of financial freedom, something which we believe everyone should have the chance to pursue in financial markets.

2. Vision and Mission

The vision of Fama one is to help provide a world where there is a level playing field and equal opportunity to financial freedom. In essence, we want to make investing easy for everyone.

No matter your experience and capital level, we aim for Fama One to provide value to you - all you need is a web3 wallet.

In order to achieve this vision, Fama is working on its mission of automating trading strategies through the use of AI and providing a robust blockchain infrastructure that is non-custodial and accessible for anyone to use.

We believe that this combination of AI and its automation of decision making alongside the decentralized nature of blockchain technology creates a powerful dynamic that resolves many of the challenges that users face in the financial environment today.

3. Technical Abstract

In this whitepaper, we outline a novel crypto DeFi vault infrastructure that leverages reinforcement learning (RL) to provide automated trading strategies in a decentralized and non-custodial manner. Our vaults receive execution instructions from an AI model and utilize Uniswap V3 for performing swaps. We will discuss the key aspects of this infrastructure including our models, vault mechanics and swap execution.

4. Overview of Reinforcement Learning Models

Reinforcement learning (RL) is a subfield of artificial intelligence (AI) that focuses on training agents to make decisions by interacting with their environment. At the core of a reinforcement learning model are three key components: the agent, the environment, and the reward signal. The agent is the decision-making entity that interacts with the environment, which represents the external context within which the agent operates.

The agent takes actions that influence the environment, and in return, receives feedback in the form of a reward signal. This reward signal quantifies the desirability of the agent's actions and guides its learning process.

Reinforcement learning models generally follow a trial-and-error approach, with the agent exploring the environment and refining its actions based on the received rewards. The learning process can be guided by various algorithms, such as Q-Learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO), which define the strategy for updating the agent's decision-making policies.

While Fama will explore and develop models utilizing each of these algorithms and more, our initial focus will be Q-learning models.

Q-Learning is a model-free reinforcement learning algorithm that seeks to learn the optimal action-selection policy for an agent interacting with an environment. In the context of trading strategies, Q-Learning can be applied to train an agent that can adaptively execute trades based on historical and real-time market data. In this section, we will discuss the process of training a Q-Learning model and applying it to trading strategies.

5. Data Infrastructure:

Our data infrastructure comprises several components that work together to gather, process, and store data for our AI model. These components include:

5.1 Data Sources for Q-Learning Models in Trading

To create effective trading strategies using Q-Learning, it is crucial to feed the model with relevant and diverse data sources. Incorporating various data types can enhance the model's ability to capture different market dynamics and improve its performance. Below we outline the key data sources that the Q-Learning models can draw upon:

5.1.1 OHLCV Data

OHLCV (Open, High, Low, Close, Volume) data is a widely used data source in trading strategies and consists of the following components:

- Open: The opening price of an asset for a given period (e.g., daily, hourly).
- High: The highest price reached by an asset during the period.
- Low: The lowest price reached by an asset during the period.
- Close: The closing price of an asset for the period.
- Volume: The total quantity of the asset traded during the period.

OHLCV data provides a summary of the asset's price movements and trading activity over a specified time frame, which can help the Q-Learning model capture price trends and identify patterns related to market behavior.

5.1.2 Tick Data

Tick data refers to the information about each individual trade executed in the market, including the price, quantity, and timestamp of the trade. Tick data provides a granular view of market activity, allowing the Q-Learning model to capture short-term price fluctuations and micro-level market dynamics. By incorporating tick data, the model can potentially identify high-frequency trading patterns and exploit transient market inefficiencies.

5.1.3 Technical Indicators

Technical indicators are mathematical calculations based on historical price and volume data, which are used to identify patterns and trends in the market. Some common technical indicators include:

- Moving Averages (MA): A smoothed representation of an asset's price over a specified time period, used to identify trends and potential support/resistance levels.
- Relative Strength Index (RSI): A momentum indicator that measures the speed and change of price movements, used to identify overbought or oversold conditions.
- Bollinger Bands: A volatility indicator that consists of a moving average and two standard deviations above and below it, used to identify potential breakouts or trend reversals.
- MACD (Moving Average Convergence Divergence): A trend-following momentum indicator that calculates the difference between two moving averages, used to identify potential bullish or bearish momentum shifts.

Incorporating technical indicators into the Q-Learning model can help capture market patterns and trends, which can be used to generate more informed trading signals and improve the model's overall performance.

5.1.4 News and Sentiment Data

News and sentiment data capture qualitative information about the market, such as news articles, press releases, social media posts, and expert opinions. This data can be processed using natural language processing (NLP) techniques to extract relevant information and quantify market sentiment.

Incorporating news and sentiment data can help the Q-Learning model account for external factors that may influence market behavior, such as macroeconomic events, regulatory changes, or market rumors. By considering this qualitative information, the model can potentially anticipate market reactions and adjust its trading strategies accordingly.

By integrating these diverse data sources into the Q-Learning model, we can create a more comprehensive and robust trading strategy that adapts to different market conditions and captures various aspects of market behavior. This data-driven approach can enhance the performance of our crypto DeFi vault infrastructure and maximize the overall return for users.

5.2 Data Collection:

We use batch data collection methods to gather data from our sources. Batch methods involve downloading historical data at regular intervals and storing it in a database. We collect data on various timeframes, including daily, hourly, and minute-by-minute data, depending on our trading strategy's requirements.

5.3 Data Processing:

Once the raw data is collected, it must be processed and transformed into a suitable format for the AI model. Data processing includes tasks such as:

- Data cleaning: Removing or correcting erroneous or incomplete data points to ensure data quality.
- Feature extraction: Calculating relevant features from the raw data, such as moving averages or other technical indicators.
- Feature engineering: Creating new features or aggregating existing features to enhance the AI model's input.
- Data normalization: Scaling and standardizing data to ensure consistent and comparable values across different data sources.

Any or all of these techniques may be used depending on the status of the data and/or intention of the trained model.

5.4 Data Monitoring and Updates

To ensure the AI model remains up-to-date with the latest market information, the data must be regularly updated. This involves continuously collecting new data, processing it, and appending it to the existing data.

Our data infrastructure for our AI model is composed of data sources, data collection, data processing, data storage and data monitoring. We collect data from various sources and process it to make it compatible with our AI model, and store the processed data for easy access and analysis. A robust data infrastructure is crucial for the success of our trading strategy in order to remain informed and optimal in performance.

6. Training a Q-Learning Model for Trading

The first step in training a Q-Learning model for trading is to define the environment, which in this case, consists of historical and real-time market data, as defined in the previous section. The agent's objective is to learn a policy that maximizes the expected cumulative reward, which may be defined as the total profit generated by the trading strategy.

The agent has a set of predefined actions it can take in each state which primarily consists of buying, selling, or holding the asset.

During the training process, the agent iteratively interacts with the environment by choosing actions based on its current state and a Q-function, which estimates the expected cumulative reward for taking a specific action in a given state. The agent updates the Q-function using the observed rewards and the following update rule:

$$Q(s, a) = Q(s, a) + \alpha * (r + \gamma * max\_a' \, Q(s', a') - Q(s, a))$$

Here, α is the learning rate, γ is the discount factor, s is the current state, a is the current action, r is the immediate reward, s' is the next state, and a' is the next action.

The agent explores the environment by balancing exploitation, where it chooses the action with the highest Q-value, and exploration, where it selects a random action with a probability ε. This ε-greedy approach allows the agent to explore new strategies and avoid getting stuck in local optima.

The training process continues until the Q-function converges or a predefined number of episodes are completed.

7. Applying the Q-Learning Model to Trading Strategies

Once the Q-Learning model is trained, it can be applied to real-time trading by continuously updating the agent's state based on the latest market data and selecting the optimal action according to the learned Q-function.

When applied to trading strategies, the Q-Learning agent seeks to optimize its actions to maximize the cumulative profit generated by the strategy. The agent can adapt to changing market conditions by continuously updating its Q-function based on real-time market data and the observed rewards.

By incorporating a Q-Learning model into our crypto DeFi vault infrastructure, we can create adaptive trading strategies that can optimize the execution of the swaps, generate returns for users and depending on the model properties, reduce the risk of drawdowns. This approach allows the vault to remain competitive and efficient in the ever-changing landscape of DeFi and digital asset markets.

8. Further Use Cases for Reinforcement Learning (RL) In asset management

Beyond trading strategies, there are a number of additional use cases we intend to explore to add value to users via Fama in the future:

- Portfolio management

RL agents can be trained to optimize asset allocation and manage investment portfolios by learning from historical market data and adjusting their strategies based on current market conditions. By maximizing risk-adjusted returns or other predefined reward signals, RL agents can help investors make more informed decisions and achieve their financial goals.

- Market making

RL agents can be used to develop market-making algorithms that provide liquidity to markets by quoting bid and ask prices for financial instruments. By learning from their interactions with the market, these agents can optimize their quoting strategies to maximize profits while minimizing inventory risk.

- Algorithmic execution

Reinforcement learning models can be employed to optimize the execution of large orders by breaking them down into smaller trades and intelligently timing their execution. This can help reduce market impact, minimize trading costs, and improve overall trade performance.

9. Vault Framework

Our vault infrastructure is a smart contract-based system that automates the process of investing, managing, and optimizing user assets in the blockchain ecosystem. Our vaults aim to maximize fee efficiency by pooling user funds and implementing a standardized security framework that minimizes the risks associated with manual asset management. Below we outline the key functions of the vault system:

9.1 Vault Smart Contract

The smart contract is the backbone of our DeFi vault infrastructure. It is responsible for storing user deposits and executing trades based on signals received. The smart contract's design is optimized for security, efficiency, and scalability. We use the Solidity programming language to implement the smart contract and deploy on the Ethereum blockchain. The smart contract includes the following key features:

9.1.1 Depositing and Withdrawing

Users can deposit their assets into the vault by interacting with its smart contract. Typically this is through a web3 wallet in which a user calls actions and signs approval with their private key. Upon deposit, the vault will store the user's assets and follow predetermined processes outlined within the value smart contract. Users can also withdraw their assets at any time, subject to being able to pay the gas withdrawal fee defined by the ethereum blockchain.

To track each user's assets and respective balances, the vault will utilize a Liquidity Provider (LP) token model. Upon depositing assets into the vault, users will receive an equivalent amount of LP tokens, which represent their share of the total assets in the vault. These tokens are compliant with the ERC-20 standard.

The number of LP tokens a user receives upon deposit is calculated based on the current value of the assets in the vault and the value of the user's deposit. The formula for calculating the LP tokens to be issued is as follows:

$$LP\ tokens\ to\ be\ issued\ =$$
$$(User's\ deposit\ value\ /\ Total\ value\ locked\ in\ the\ vault)\ *\ Total\ LP\ tokens\ in\ circulation$$

This calculation ensures that the proportion of a user's deposit relative to the total assets in the vault is accurately represented by the LP tokens issued to them.

When a user wishes to withdraw their assets, they will need to return the equivalent amount of LP tokens to the vault. The vault will then calculate the value of the assets to be returned based on the user's share and distribute the corresponding assets accordingly.

9.1.2 Trading

The trade function is responsible for executing trades based on signals received from the AI model. The function interacts with the cryptocurrency exchange to execute the trade and updates the contract's balance accordingly.

When a signal is received, the trade function validates it and interacts with the API of a decentralized exchange to place an order. It then waits for the order to be filled and updates the contract's balance accordingly. The trade function also calculates the transaction fee charged by the exchange and deducts it from the contract's balance.

Once the trade is completed, an event is emitted on the blockchain, indicating the trade's execution details, the traded cryptocurrency pair, the trade type, the trade quantity, and the transaction fee.

The trade function is designed to handle different types of orders and handle errors and exceptions that may occur during the trade execution process.

9.2 Signal Delivery

In preliminary versions of Fama we will use an off-chain service that generates and signs the signals to ensure secure delivery to the smart contract. The service is hosted on a secure server and is accessible only to authorized personnel. The service generates the signals based on the AI model's output and signs them with a private key. The signed signals are then sent to the smart contract using an API.

Further work will be done to add further security and decentralization to signal delivery to reduce single point of failure risk in the future.

9.3 Security Measures:

To ensure the security of our DeFi vault infrastructure, we have put in place several security measures, including:

9.3.1 Multi-Signature Wallet

We use a multi-signature wallet to store the private keys required to access the smart contract. The wallet requires the approval of multiple personnel to initiate any transaction, adding an extra layer of security.

9.3.2 Code Audits

We conduct regular code audits to identify and mitigate potential vulnerabilities in the smart contract's code.

9.3.3 Penetration Testing

We conduct periodic penetration testing to assess the smart contract's security and identify any vulnerabilities.

10. Fama Token

$FAMA is a utility and governance token and will have use case in the following ways:

Premium Vaults: Basic vaults will be free to access but additional vaults will be available on a subscription paid in FAMA. Some vaults may also require staking of FAMA to access.

DAO Voting: FAMA staking will be used as part of the weighting algorithm used for DAO proposals.

An additional paper will be produced on the Fama DAO closer to its release due to the novel nature of our DAO design, however in essence, the DAO will be responsible for:

- Managing fees and rewards
- Investing in new models and infrastructure proposals
- Approving technical upgrades and modifications

Fama's success will come from the strength of its community and contributors, therefore we are keen to distribute Fama tokens based on merit to those who will support Fama on its journey and share its vision and values. Further details will be provided in due course.