Protocol Economy Assessment

An analysis of financial stability of the Bancor v3 protocol

V1.1 - October 2022

Disclaimer

The content is for informational purposes only and nobody should construe any such information or other material as legal, tax, investment, financial, or other advice.

Version

Version	Date	Description
1.0	Sept. 2022	Initial publication
1.1	Oct. 2022, 7th	Minor typos corrections following DAO feedback

Introduction

About Almanak

Almanak is an institutional protocol risk management and governance platform engaging in performance-based, long-term partnerships through the provision of data-driven recommendations and C-level intelligence to DAOs and management teams of blockchain-based projects.

Almanak uses agent-based modeling and machine learning to optimize Decentralized Finance (DeFi) protocols and drive sustainable growth, competitiveness, and innovation.

Pertaining to the field of sensitivity analysis, Almanak's scientific approach aims to determine the system input variables (i.e. optimization levers) which contribute to interest quantities depending on system output. The goal is to produce a set of dynamic, data-driven rules, integrable to the client's architecture to enhance protocol risk management. We rely on expert domain investigation and numerical validation using screening, measures of importance, and deep exploration of variation range over mixtures of historical and simulated data. By building convex sets of parameters and factors, we use evolutionary optimization schemes to design representative sets of scenarios informing the relationships between input and output.

Almanak's intelligence therefore ensures sustainable DeFi protocols' economic soundness alongside profitability enhancement.

Objective and Scope

The report presents Almanak's approach to design an optimization solution for Bancor's V3 risk-reward ratio between impermanent loss (IL) and protocol revenue. The solution improves Bancor's short-term deficit time-to-recovery and contributes to the overall protocol's long-term sustainability.

We first describe the ongoing economic and technical situation Bancor is currently in. We then explain our methodology to design the optimization solution. We discuss the protocol levers and agent design as well as the simulation engine architecture and solution validation. Training of agents' interactions with the environment and the simulation process itself are also outlined.

The numerical assessment is performed on the following assets: ETH, WBTC, LINK and DAI. Due to the structure of Bancor's Omnipools, BNT is necessarily included, though not analyzed separately.

Bancor V3 Overview

What is Bancor V3?

Bancor V3 is a protocol that brings an innovative approach to the AMM space. Two key innovations developed by the protocol are single-asset exposure, which allows users to deposit only one asset into an automated market maker, and impermanent loss protection, which protects users from any value loss during deposit in the protocol. Both features go hand in hand with Bancor's unique design. Firstly, the protocol pairs all assets against BNT, its native token. Secondly, it is able to mint BNT in order to match the demand for the deposits of any other asset. Finally, Bancor is able to mint BNT to compensate for impermanent loss protection costs if necessary.

After having briefly introduced all key differences that Bancor boasts, the innovations and design features are discussed in detail in the following sections.

Single-Asset Exposure

Constant product market makers (CPMM) utilize the formula $x * y = k$ to operate a market of two assets. To become a market maker of asset X, the liquidity provider has to match the deposited value of asset X (TKN) with the deposit of the same value of asset Y. To avoid this two-asset constraint, Bancor V3 allows users to deposit just one asset X (TKN) while the protocol matches the Y amount of the deposit with BNT. The BNT is either minted by the protocol or withdrawn from the Bancor staking contract. When asset X is withdrawn, the protocol burns the BNT or transfers it back to the staking contract [1].

The Bancor staking implementation (Omnipool) allows users to stake their BNT in the single staking contract that collects BNT trading fees and distributes them pro rata to BNT stakers. The protocol itself manages where BNT should be allocated

based on the demand for single-asset exposure from other assets and trading liquidity size as explained in the subsequent sections.

Impermanent Loss (IL) Protection

Impermanent loss (IL) occurs when a liquidity provider has deposited one asset into a Bancor pool and the exchange rate of the assets diverge compared to the rate at the time of deposit. The bigger the divergence, the more the position is exposed to impermanent loss. IL is expressed as the net difference between the value of two assets in an AMM liquidity pool after a change in the exchange rate compared to the value of holding these assets in an external wallet instead of the liquidity pool (without the additional revenue from swapping fees).

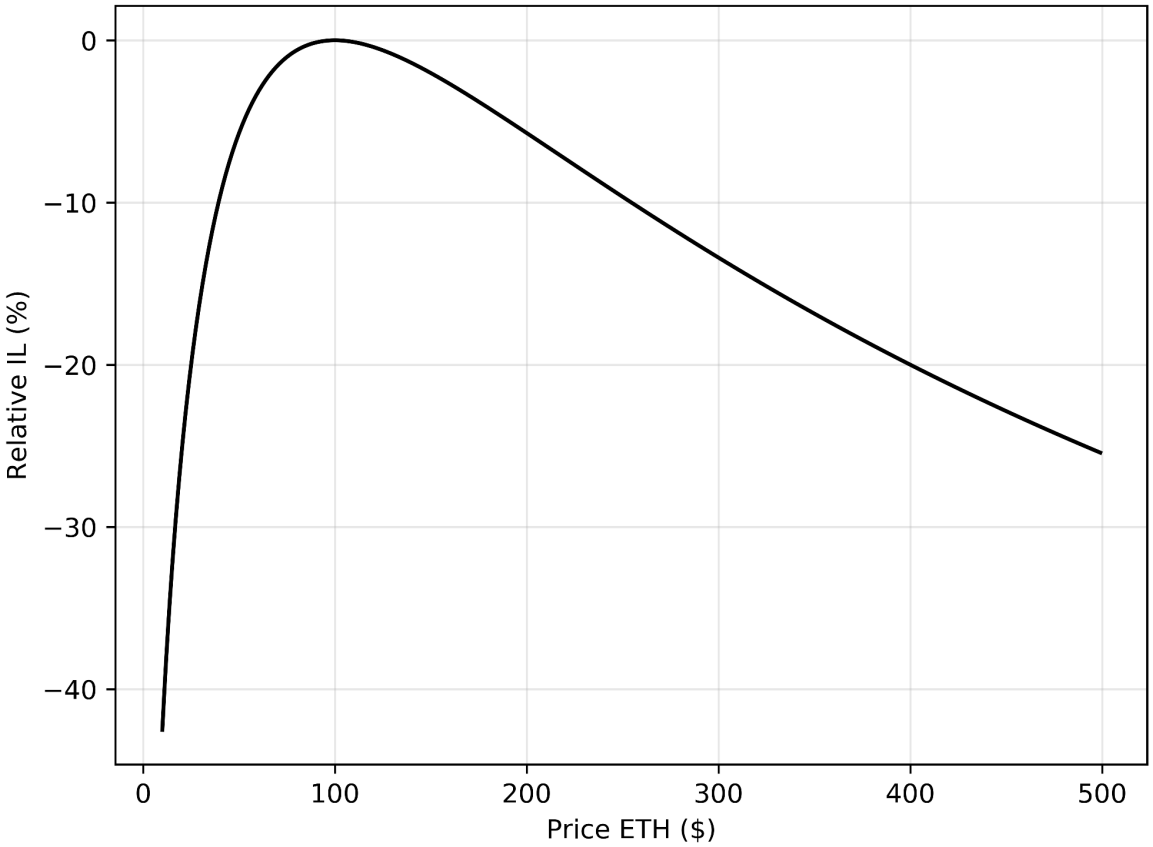

Fig. 1: Display of IL for ETH starting at $100

Impermanent loss is calculated using the following formula:

$$IL = 2 * \frac{\sqrt{\frac{r_1}{r_2}}}{1 + \frac{r_1}{r_2}} - 1 \quad (1)$$

Where r_1 is the rate of the assets at deposit and r_2 is the rate of the assets at withdrawal (**Fig. 1**). One of Bancor V3's key innovations is impermanent loss protection that allows liquidity providers not to be exposed to impermanent loss. Bancor V3 achieves this by monitoring the surplus and shortage of the assets deposited in the pool. If the pool has a shortage of the asset X (TKN) when the liquidity provider is withdrawing their deposited asset, the protocol covers the difference in BNT by either minting or trading it from the surplus in exchange for TKN.

Pool Surplus and Deficit

Under Bancor V3's unique design, the protocol uses Pool Deficit and Pool Surplus to express how the particular asset is affected by impermanent loss. To understand the nature of Pool Surplus and Deficit, we need to better understand the influence of rate divergence on the pool balance.

Let's go over an example on a simple CPMM pool. There is a deposit of Ethereum to the Bancor V3 pool, the price of ETH is $1,000 and BNT $1, and there was 10 ETH deposited to the Bancor pool. The assets' current rate is 1,000/1, which means that 1 ETH is worth 1,000 BNT, so the protocol had to co-invest 10,000 BNT in order to meet the value of ETH on the other side of the pool.

The current state of the inventory of the pool is 10 ETH and 10,000 BNT. After appreciation of the price of ETH to 1,200, the rate has changed to 1,200/1, which means that ETH price appreciated to $1,200. Due to the nature of constant product pools design, traders buy cheaper than the market price ETH in exchange for BNT, changing the inventory balance and leaving the pool with 9.13 ETH and 10,954.45 BNT, thus leaving Bancor with 0.87 ETH in deficit and 954.54 BNT in surplus.

The situation looks similar when the BNT price depreciates. Following the similar example of the deposit of 10 ETH and 10,000 BNT, if the ETH price is the same and the BNT price depreciates to $0.5 BNT, the pool inventory balance would indicate 14,141 BNTs and 7.07 ETH.

On-Curve Trading Liquidity (TL)

In order to decrease the exposure to impermanent loss, Bancor V3 introduces trading and non-trading liquidity. Trading liquidity is a virtual representation of asset X (TKN) and Y (BNT) on the curve that participates in market making. Non-trading liquidity consists of assets that do not participate in market creation and stay idle in the staking contract. This allows Bancor V3 to accept an unlimited number of asset X (TKN) and have limited exposure to impermanent loss.

Regardless of the liquidity split between trading and non-trading, depositors are rewarded with trading fees pro rata and are equally exposed to the effect of impermanent loss. In an ideal world, Bancor V3 would supply the optimal amount of asset X (TKN) and match it with Y (BNT) to limit the exposure to impermanent loss and meet all trading demand, hence optimizing the protocol risk-reward ratio.

Note that, in the current Bancor V3 live version, all assets participate in on-curve trading liquidity (TL), therefore all assets are available for market making, exposing all of them to impermanent loss.

Bancor V3 Recent Updates and Current State

During the May 2022 market downturn, Bancor V3 experienced a high volume of withdrawals that triggered IL protection and covered it with BNT. This instantly triggered liquidation, causing a significant BNT price divergence from other assets.

Due to protocol exposure to impermanent loss with all liquidity being placed on-curve, it followed a cascading effect of further increasing the protocol deficit, triggering more BNT liquidations. To stop the death spiral, Bancor governance decided to temporarily turn off IL protection, leaving liquidity providers with an estimated $42M dollars in loss as of early August 2022. The protocol still holds $25M worth of BNT in surplus, implying a shortfall of $17M worth of assets.

Dune user *ferdinand_hodl* created a Dune dashboard that offers several visuals in regard to the current state of Bancor.

To quickly respond and tackle the recovery, the protocol was updated (July 24th community probe), allowing for 90% of fees collected from TKN to be exchanged to DAI and used to replenish the protocol deficit. Additionally, to address the BNT supply crisis, 90% of BNT-side aggregated trade fees are being used to be burnt by the Vortex.

Investigation Methodology

Given Bancor V3 observed challenges and axes of improvement, Almanak has investigated potential optimization levers within Bancor V3 to mitigate the risk of *realized* (confirmed withdrawals of user deposits) and *unrealized* (value at risk due to changes in asset prices) impermanent loss and to maximize the protocol's profit. To take into account ongoing protocol updates, this investigation focuses on building an optimal recovery plan to achieve protocol sustainability.

To ensure governance continuity, Almanak also prioritizes parameters that Bancor DAO can act upon. Levers are broadly screened and filtered through domain expertise involving fundamental and qualitative analysis to isolate the most potent parameters affecting the two main metrics of interest: **protocol capital efficiency** and **protocol deficit**.

Solvency and capital efficiency form a Pareto front: the more capital is exposed to trading, the higher the chances of losing assets and therefore being unable to repay users. The two goals are nevertheless not easily tractable, as both have no direct dimension associated with them. We therefore need to elaborate tractable metrics, which will be optimized to achieve protocol recovery.

As a first order iteration, we investigate multivariate configurations of predefined protocol levers such as on-curve trading liquidity size and dynamic swapping fees. Note that additional variables were part of the optimization process, but excluded for limited impact on the main metrics. These are excluded from this analysis but can be introduced at a later stage of cooperation.

Our investigation leads the optimization framework to focus on the following two axes:

- **Maximization of the revenue of the protocol** (Protocol Capital Efficiency)
- **Minimization of protocol exposure to unrealized IL** (Protocol Deficit)

With these axes as north stars, the following parameters are retained as subject to optimization:

- **Pool swapping fees:** Introduction and optimization of dynamic fees per pool to maximize protocol revenue from trading.

- **Pool depths:** Introduction and optimization of dynamic on-curve trading liquidity in order to minimize protocol exposure to further impermanent loss.

Note that another avenue of optimization could be to directly bring the pool rates to their pre-crash level. However, this is premature at the time of writing given the uncertainty of BNT economy updates. We thus prefer to rule it out in the current version of this report and will reassess the pertinence once more clarity is gained. Overall, we can write the following two multi-horizon objective functions over which we need to optimize for protocol capital efficiency and protocol deficit:

$$\begin{cases} \max \sum_{t \in T} fee_t^{swap} - Incentives_t - IL_t^{realized} & (2) \\ \min \sum_{t \in T} IL_t^{unrealized} & (3) \end{cases}$$

where t represents the current simulation step and T represents the total number of steps.

Optimization Solution Design

This section first describes the modeling and calibration choices made to bring the protocol levers into the optimization problem. It then explains the choices made in terms of agents and simulation design before providing a high-level overview of the architecture we built to implement the solution. Finally, we discuss the solution validation approach.

Protocol Levers Specification

This section introduces the protocol levers Almanak identified, varied within the simulation environment and optimized through different scenarios.

Dynamic Pool Depths

Bancor has three different instances of capital: Trading Liquidity (TL), the vault, and the staking ledger. As only TL is subject to IL, the protocol is able to allocate funds to the vault to limit the exposure or leave deposits up to market movement.

The pool depth of either pool defines the liquidity and thus how much capital of all deposits can be traded. As of the time of writing, Bancor has allocated all deposits

into trading liquidity. As the price of BNT diverges, more imbalance is created and more IL is accrued by the protocol.

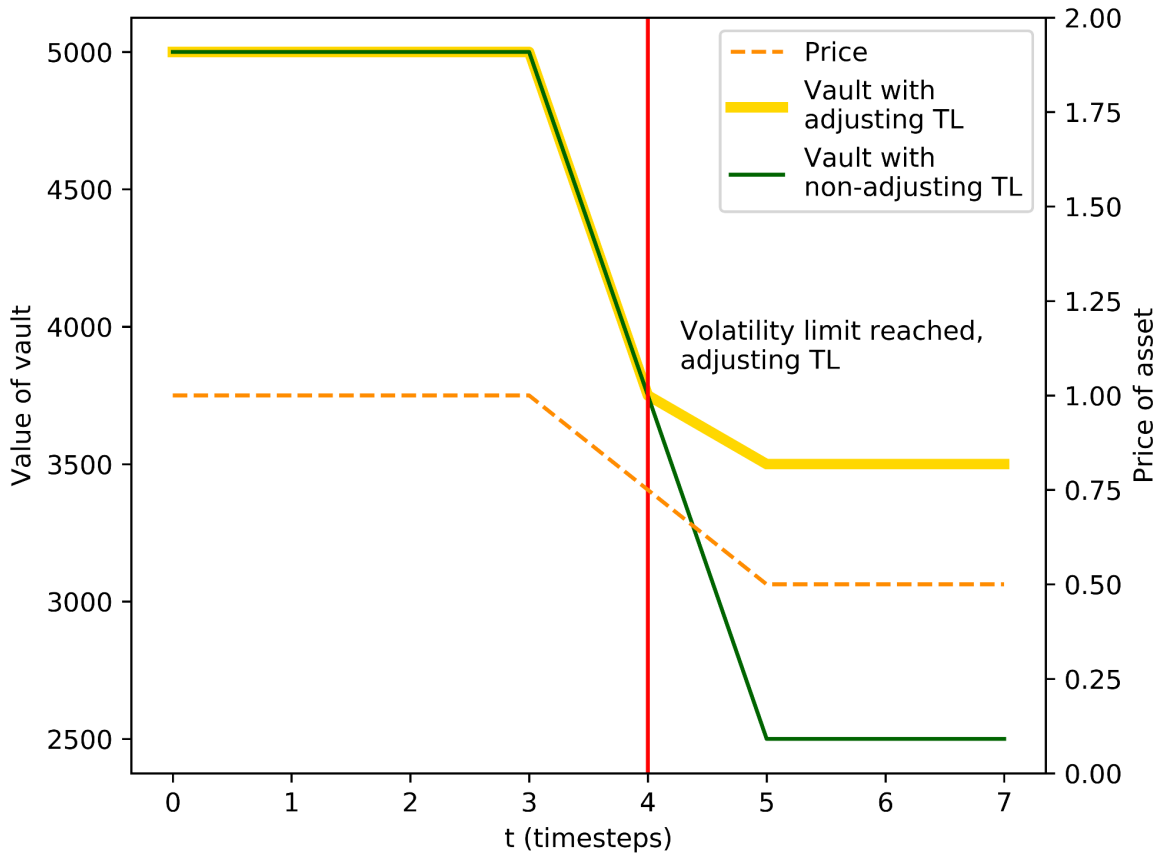

Fig. 2: Adjusting TL with higher volatility helps limit the impact on total deposits (vault)

One avenue to limit IL in the future is to dynamically change the pool size by moving the trading liquidity to an idle, staked-liquidity state, thus changing the factor of the CPMM pool equation, which can be interpreted as TL. As mentioned above, moving capital from the TL to the vault can be beneficial to reduce IL.

When limiting TL, Bancor exempts itself from being exposed to IL: once the risk is deemed over, restoring the initial, maximum value of TL should be the next step. Any transfer of liquidity based on ongoing risk will be executed in order to facilitate an environment where the protocol can stabilize and reduce the impact of price changes on user deposits. We nevertheless need a variable to track the ongoing risk and directly correlated with IL. Recent changes in market price and the risk associated with an asset can be detected and observed via the volatility σ_x (which is denoted in %).

Having set the metric to be measured, an outline on the recurrence of changing TL has to be defined. One possibility to limit the impact is to change TL atomically, i.e., with every action happening on-chain. This bears the risk of high market slippage paired with a negative user experience, which has to be avoided. Instead, TL changes are limited to hard thresholds σ_x^{lim} defined for every asset. The adjustments to TL are taken into account through a fixed adaptation factor $\delta_x^{tl,lim}$ given a fixed σ_x^{lim} (**Fig. 2**).

Based on the assumptions taken here, we formulate the mathematical description of this on-chain contract the following way:

$$\Delta_x^{tl} = \begin{cases} \delta_x^{tl,lim} & \text{if } \sigma_x \geq \sigma_x^{lim} \\ \frac{1}{\delta_x^{tl,lim}} & \text{if } \sigma_x < \sigma_x^{lim} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

with Δ_x^{tl} being the change in trading liquidity for asset x, σ_x the observed volatility of an asset, and $\delta_x^{tl,lim}$ the adaptation factor relative to the original trading liquidity. Updates are made based on observations of the 24-hour volatility. Subsequently, the pool is either enlarged or shrunk.

Why Monitoring TL Dynamically Makes Sense

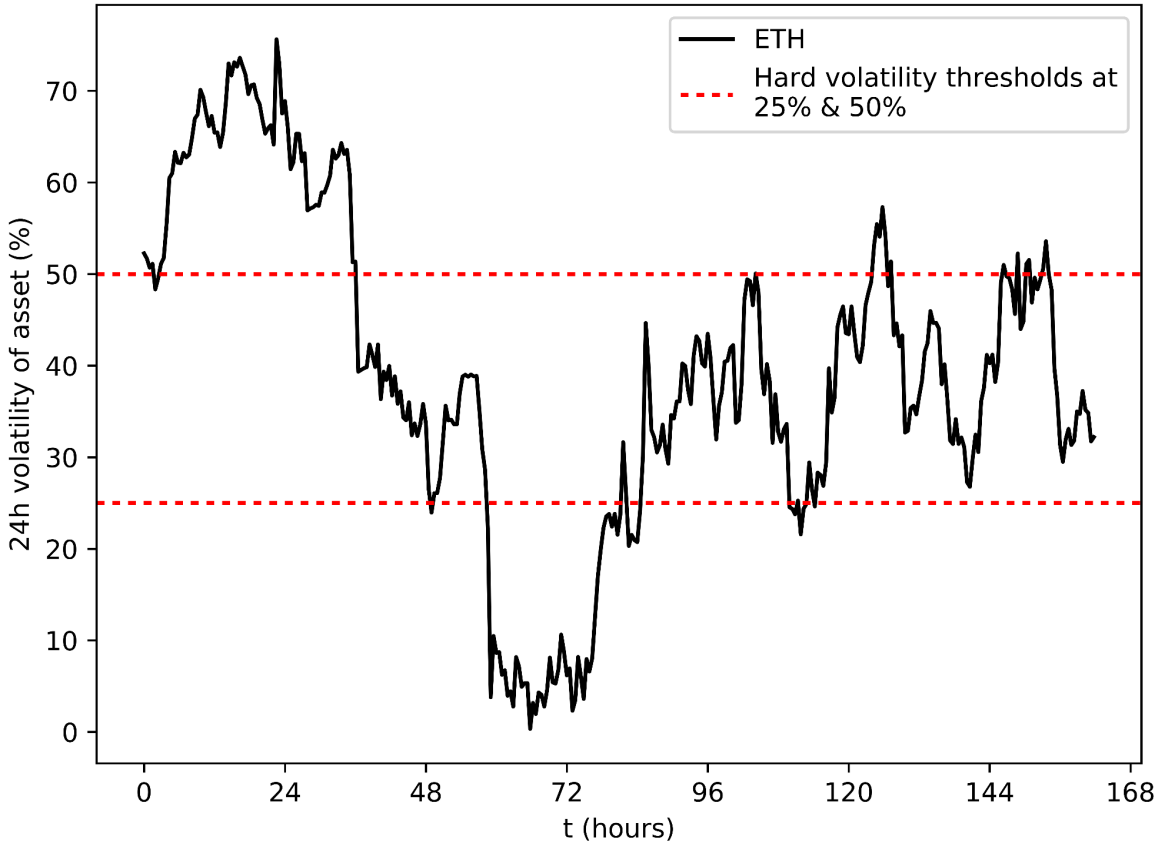

Fig. 3: Changing volatility of ETH over a week

Trading liquidity is a vital component which needs to be protected in regard to price movements. Especially with falling prices, taking components out leaves less capital at risk but would hurt protocol returns over time.

To act swiftly, the protocol can check if and by how much the volatility has changed over the past 24 hours and adapt the trading liquidity accordingly. In the example above, hard volatility thresholds are set at 25% and 50% and the protocol would have changed the trading liquidity, if possible instantly, 14 times, whereas a weekly update would have brought only a single change (**Fig. 3**). One might observe that, at the end of the time interval, the protocol had no need to adapt, as the reactive change of trading liquidity would have resulted in the same level as only changing it once per week. Due to the lack of reaction to asset volatility, the protocol accrues more IL than if TL was reactively changed.

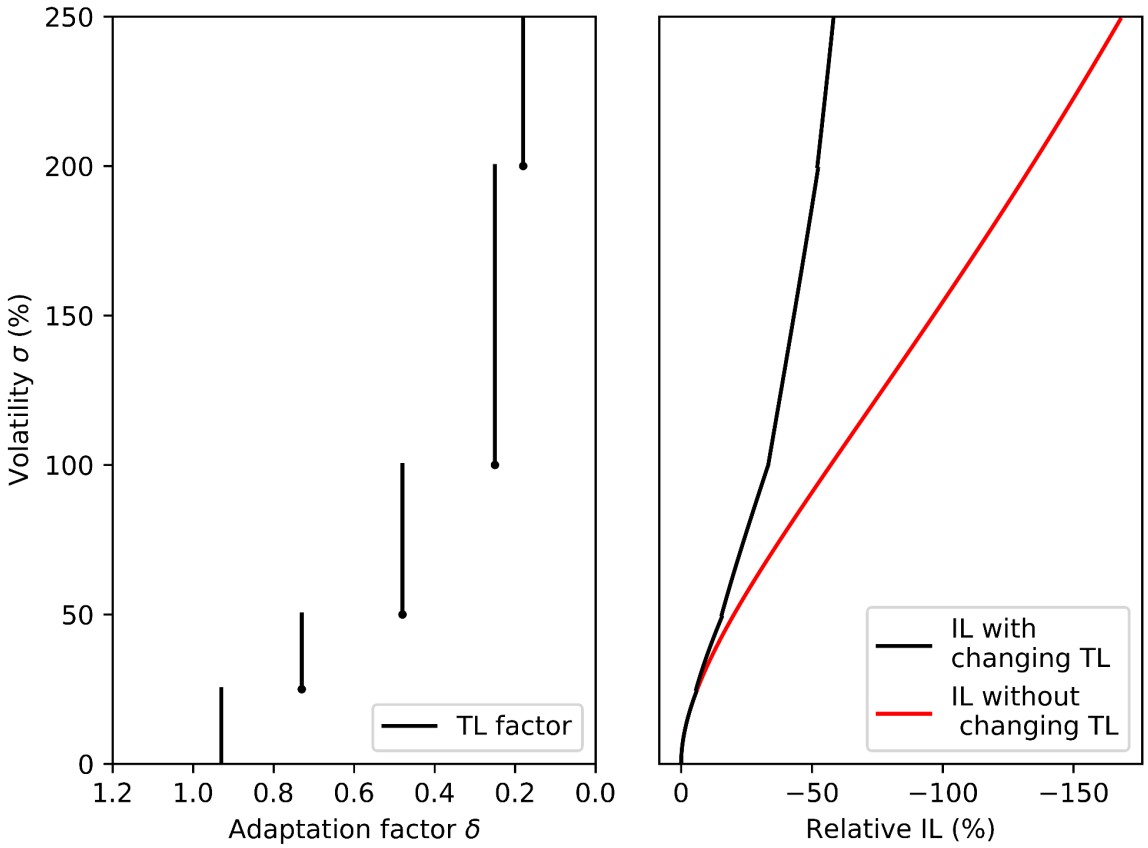

Fig. 4: Impact of changing TL on IL in volatile environments

Additionally, we observe a change in IL through the TL change based on volatility/price change. **Fig. 4** shows two separate instances. The left side is an example of the evolution of the adaptation factor $\delta_x^{tl,lim}$ based on σ_x. Due to the definition of $\delta_x^{tl,lim}$, the representation is non-continuous, i.e., hard cuts in values are observable. On the right side, IL is shown based on the current volatility and the capital being exposed to IL: with increasing volatility, TL is shrinking and limits the impact of IL on the deposits of users.

Dynamic Swapping Fees

Swapping fees are by far the most important revenue stream for the protocol and are therefore one of the main parameters to tune.

This primarily boils down to answering the question of how much arbitrage trading "should" take place. For instance, with lower fees and a larger pool size, more fees will accrue to the protocol as arbitrageurs will be attracted to execute trades with a

higher profitability compared to other platforms. Given that the majority of trades are coming from this group of arbitrageurs, opting for higher fees will limit protocol profitability while adversely preventing the pools from being drained too quickly.

Fees are therefore designed to react to on-chain behavior without any external help, especially in a volatile environment, but with the internal help of a changing trading liquidity. Initial research has been published to analyze returns in CPMM pools for traders and their optimal fee structure [2] [3].

Factors are determined to balance the impact of features to take into account, providing flexibility to adjust efficiently enough to recent events as well as long-term interests. Recent events can first be grasped through the asset's 30-day volatility (recent changes are strongly related to past shifts dynamic). Moreover, additional internal factors are the volume and the revenue share of arbitrageurs, responsible for the protocol's main revenue stream. Based on how large the share is and the recurrence of the trades, swapping fees will adjust and impact the number of incoming trades (the higher the fees, the lower that number). The following equation mathematically represents these relations by defining the swapping fee as the sum of a Base Fee and an Risk Add-on:

$$f_x = f_x^{base} + A_x^v \tanh(\sigma_x) + A_x^s \tanh\left(\frac{V_x^{arb}}{V_x}\right) \quad (5)$$

where f_x is the swapping fee for token x. Every token has a base fee f_x^{base}, which starts at 1bp (0.01% or 0.0001). Second and third terms define the Risk Add-on. A_x^s and A_x^v are the amplification factors. A_x^s scales the arbitrage term, with V_x^{arb} and V_x being respectively the asset's arbitrage volume and total volume over the considered past time frame. A_x^v scales the volatility term, with σ_x being the 30-day volatility of the asset.

Remark on Maximal Extractable Value
Maximal Extractable Value happens when a third party takes advantage of information asymmetry about current open transactions in the mempool and performs sandwich trading (see here) to benefit from the immediate price action.
Note that MEV is procyclical as larger trades and arbitrage would be hindered or only partly executed given the potential sandwich trades. Based on the sample set from the past 30 days, no such profitable arbitrage could be found though (dataset here). As of now, MEV is not considered for swapping fee calculation.

Simulation Design

Agents and Environments

Virtual environments are simulated in order to replicate real world conditions in terms of both chain mechanisms and market conditions. For each of these virtual simulated environments, we copy the same mechanics of participants that are present on-chain. To achieve this, we define multiple programmatic agents who can act upon and receive feedback from virtualized simulated environments.
Agents are of three types: Liquidity Providers, Traders, and Arbitrageurs (see Appendix A. Agent Specifications for a detailed description).

Each agent's actions and interactions affect the live simulated environment which also affects other agents, their behavior, and subsequent actions. An agent's behavior in an environment is modeled after real world market participants via a combination of machine learning, probabilistic programming, and domain expertise.

Each run of the simulation is therefore unique as the combinations of agents and their parameters are different. The simulation environments are seeded by price trajectories, with a plethora generated to cover a maximum of potential scenarios and validate the protocol lever combination. Price trajectories can be a combination of historical prices, a recombination of prices from various time periods, or a variation of black swan events injected into data and simulated trajectories. These create innumerable possible simulation scenarios, with each simulation having unique agents competing with other agents and learning to play in multiple market environments that will predict real user behavior in different market environments.

Description of a Simulation Run

The following diagram presents the sequence of a simulation payload within an optimizer instance.

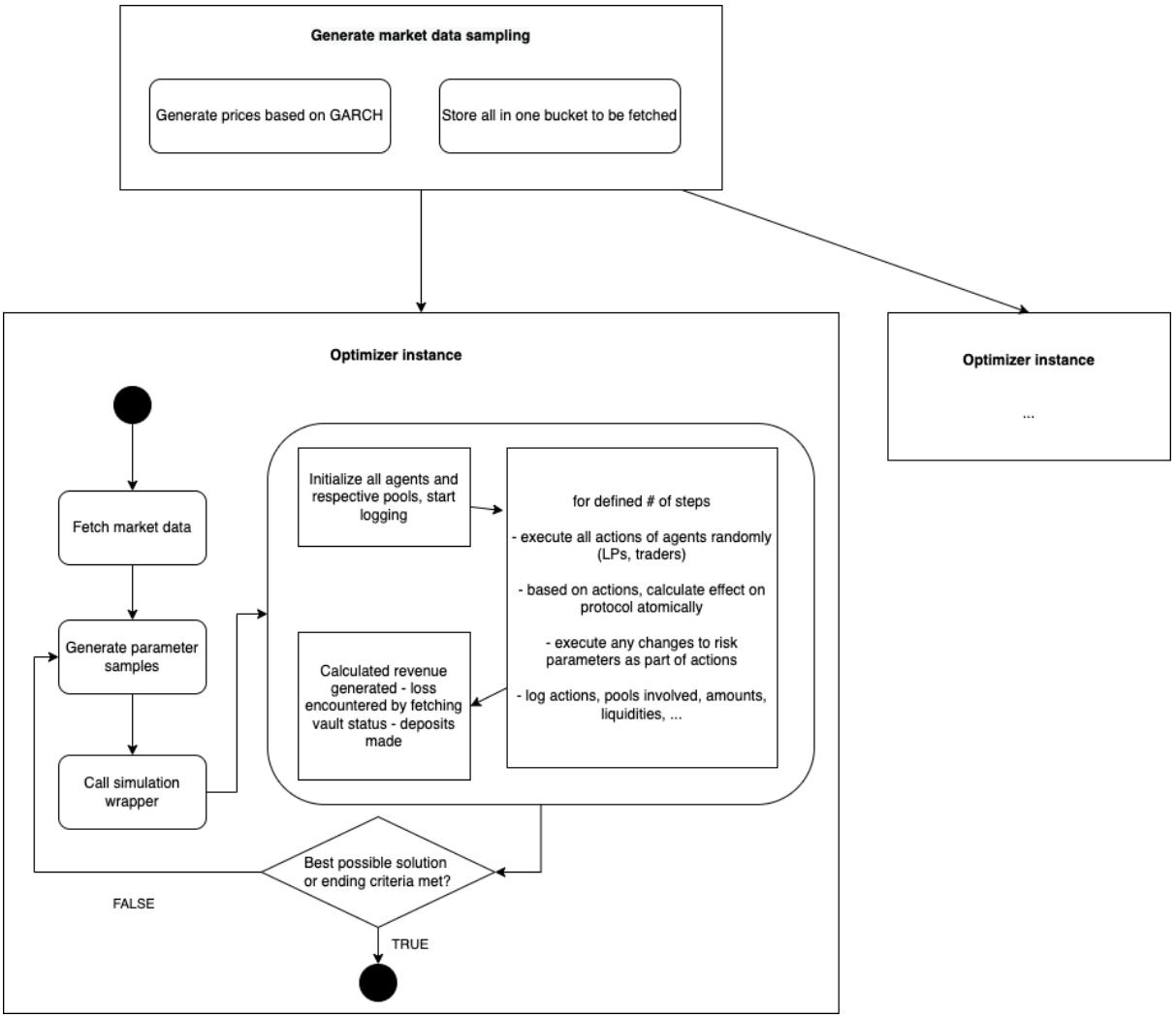

Fig. 5: Flowchart of a simulation run

We start a simulation with sampling agents and price trajectories under a specific combination of Bancor key parameters. Simulations are run simultaneously in parallel over samples of historical-based stochastic projections (see Appendix B. Price Trajectory Simulations for the mathematical framework). At the end of each simulation, a score is given which represents risk and profitability metrics. Evolutionary optimization is used to evolve and tune the best Bancor parameters that optimize solvency and profitability. To minimize overfitting, each recommended set of parameters is compared relative to their agents and price trajectories against other similar sets and their score.

Note that the *Bancor Simulation Environment* is agent-based, where agents execute actions within an epoch and are retrained offline prior to every simulation run. Their actions are a stochastic function of the set of protocol levers that are tilted at each run. This could be likened to an offline, off-policy, model-free reinforcement

learning framework where the main agent is Bancor and the reward function is given in Equations (2) and (3).

Solving for Overfitting and Robustness

Overfitting is a known problem in both algorithmic trading and machine learning. In traditional scenarios, the prediction of a model and decisions of trading strategies during optimization does not affect their simulated worlds. Repeated tuning of algorithms given identical environments and price trajectories results in the optimized parameters to overfit akin to memorizing specific future patterns. This produces good results during backtesting but results in bad results as soon as the environment changes.

In our scenarios, our agents learn to operate in live environments among many other agents, forming a unique dynamic environment. Therefore, by training and tuning the parameters in dynamic environments, our approach fundamentally produces robust recommended parameters.

As a next step, we validate these parameters by running thousands of simulations with different generated price trajectories, each representing a different possible future in parallel. This tunes our approach toward high reliability (see C. Optimization Framework for more details).

Architecture Design

The length of each simulation run can take over multiple hours, largely depending on the number of agents and pools as well as the length of the time period to be observed. Supporting thousands of runs becomes critical, especially with the macro-environment changing and an imminent solution needed.

To cope with the necessary rising number of price trajectories and optimizations paired with connecting individual price feeds with optimization instances, Almanak's architectural core capabilities are triggered as required to complete and enhance the protocol's internal build of DataOps, DevOps, and ChainOps.

Composable and Agnostic

The simulation engine has been designed in an elastic cloud-native environment consisting of ephemeral language-agnostic, container-based workloads at each step. While multiple agents are being trained across different simulation environments operating independently, the agents can communicate with each other and share learning via a global shared state and messaging system.

Modular and Auto-Scalable
Each task in the simulation workflow is a self-contained computing process running in a cluster and orchestrated by a few touch-points, with the added benefit of writing each task using any combination of dependencies, programming language, and specific hardware requirements such as memory, CPU, and GPUs best fit for the task. This allows both scaling up in hardware and scaling out in quantity when additional processing power is required, such as to accommodate more experiments, parameters, larger and richer datasets, parameter optimizations, and addition of new kinds of transactions on-chain or supported assets into Bancor.

Antifragile and Agile
Our design is also modularized and decouples components to be robust and agile. When new constraints are added, the simulation becomes more comprehensive and resilient, while agents automatically learn and adapt. For agility, new optimization algorithms and agent classes or new features can also be added and experimented on without requiring substantial investment or commitment. Should the new experimental features be detrimental, they can be removed. This agility is required for quick adaptation to changes in the ecosystem and future experimentation for improvement can be done at low risk.

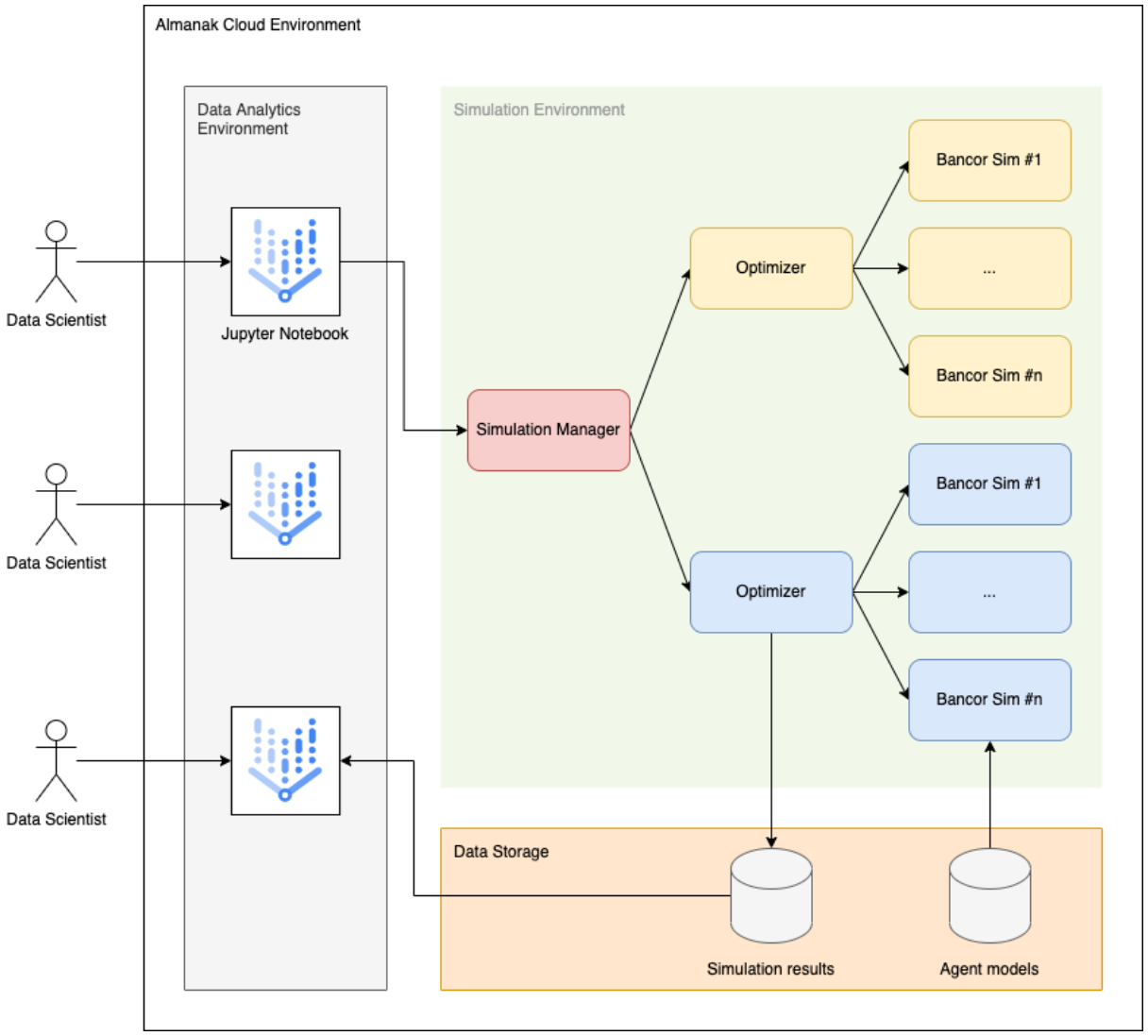

Fig. 6: Visual representation of the architecture design

Solution Validation Design

Validation of the efficacy of the simulation is a complex task. For each simulation run, simulation logs, agent logs, and operational metrics are generated. We validate the logs to ensure that the system is operating effectively as intended. To measure the efficacy of our output, it undergoes overfitting tests as mentioned above. To measure the efficacy of the simulation and agent-based modeling solution, we perform walk forward optimization and tests which not only validate our approach but help to refine hyperparameters.

Another hyperparameter optimization is to take data input as a parameter in itself. Data drift detection is used to ascertain the optimal amount and use of data while both walk forward and data drift will validate the "longevity" or strength of each

output before a new set of recommendations are required. Agents also undergo a similar validation, including the use of out-of-sample datasets to determine whether their fundamental approach has been effective, in addition to a comparison of real-life execution behaviors vs. simulation behavior with our domain expertise.

Numerical Assessment

This first assessment focuses on four main Bancor pools (ETH, LINK, WBTC, and DAI) with an hourly time step and simulated over a full month. Based on the analysis and optimization of these pools, the overall protocol simulation looks as follows.

Capital Efficiency

Visualization approach

Here, heatmaps are chosen to visualize 3-dimensional results over the simulated space of risk levers and protocol profit, stemming from all price feeds. This helps compare the performance of various optimal solutions. For further ease of legibility, risk levers have been weighed over the frequency of value changes with respect to volatility and arbitrage.

To illustrate this, let us imagine the following example scenarios of volatility and arbitrage. First, with regard to TL, the next table provides values of the volatility stepwise function and their simulated frequency over price feeds:

Volatility Threshold	Adaptation Factor	Frequency over the period
25%	0.93	50%
50%	0.7	50%

Using Equation (4) and the above values, the resulting **weighted TL value** is **0.815**. It is then brought on the x-axis of the heatmap.

Concerning swapping fees, the next table provides a scenario for the simulated average volatility and average arbitrage share stemming from price feeds, together with the corresponding amplification factors and risk add-ons.

Amplification Factor	Avg. Volatility	Avg. Arbitrage Share	Risk Add-on
0.04	78%	-	0.026
0.03	-	64%	0.017

Using Equation (5) and the above values, with a base fee of 1 bp, the resulting **weighted swapping fee value** is 0.053% or **5.3 bps**. It is then brought on the y-axis of the heatmap.
Doing so, each cell in the heatmap now represents a duplet of a set of risk levers and the averaged profit from all scenarios the set has been applied to.

Heatmaps for a single pool and multiple pools

1) **Single Pool: 14% increase in profitability compared to the current setup**

Fig. 7 shows a heatmap of the Protocol profit under Almanak optimization for the ETH pool (the largest and most profitable trading pool within Bancor) as a function of the risk levers (average swapping fee and average TL relative size) taken over multiple price feeds/scenarios, and aggregated following the previously introduced weighting method.

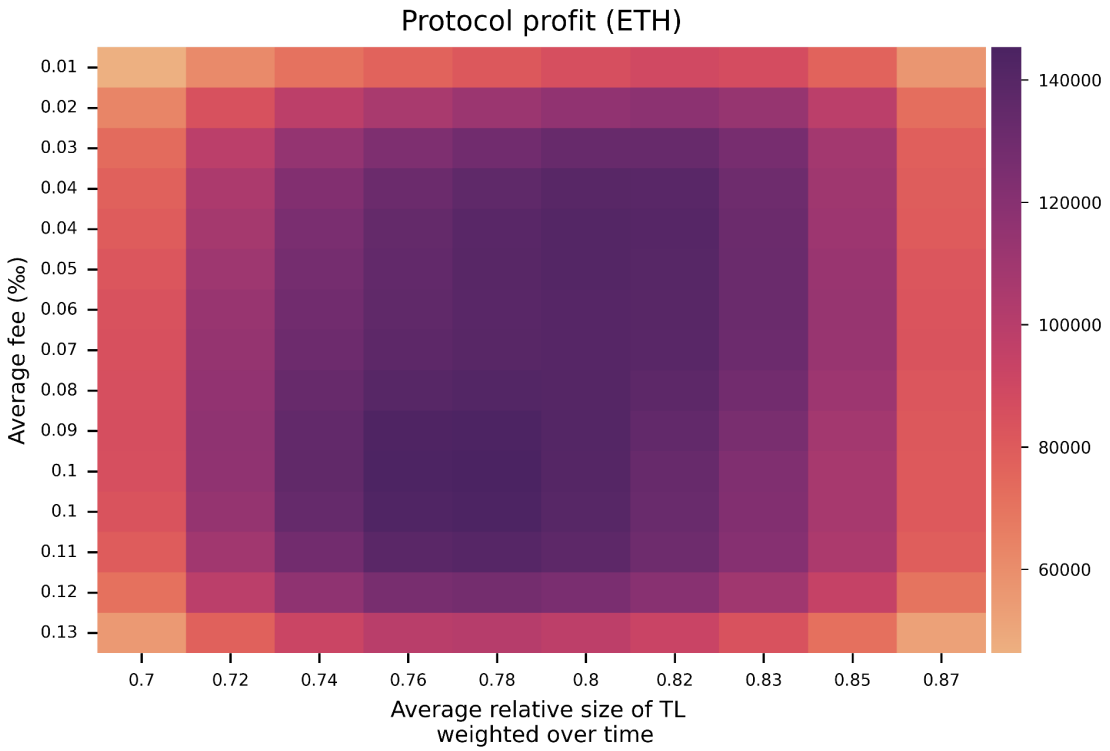

Fig. 7: Realized profit over 30 days through the ETH pool

By the shape of the dark colored cells one can observe a general tendency towards smaller pool depths and higher trading fees.

Extremes observed at top right and bottom left corners represent highly volatile market scenarios which can be schematically splitted in two opposite configurations.

(i) when a market breakdown is imminent: profits can be maintained with high swapping fees and smaller TL.

(ii) when no crash is present: profits are generated through high TL and low swapping fees.

These configurations being extreme, they are less frequent in the sampled simulation distribution, therefore also indicating less robust optimization solutions. Indeed, sets at the bottom left represent less than 0.01% each on average and top right 0.05%. The most common and robust solutions are located in the center of the heatmap, with an average representative share of 1.2%.

2) Multiple Pools: 12% increase in profitability compared to the current setup

Fig. 8 shows a heatmap of the Protocol profit under Almanak optimization aggregated over the four analyzed pools (ETH, LINK, WBTC, DAI).

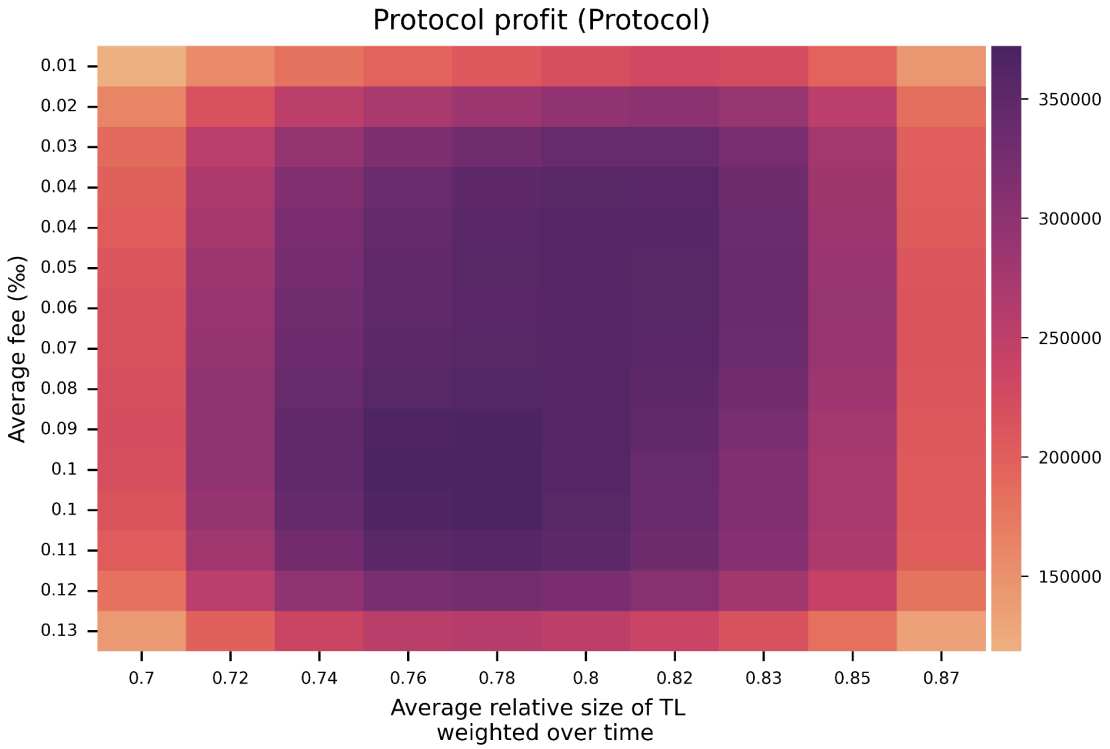

Fig. 8: Realized profit over 30 days through all considered assets

The observations made for the ETH single pool can be transferred to the general case. This is essentially due to ETH pool being a large contribution of the overall Vault in a majority of the simulation runs.

Overall, the same tendencies for all pools can be noted. This indicates that active risk management of IL protection can be optimized by scaling TL with regard to volatility and setting trading fees.

Daily Revenue

15% increase compared to the current setup

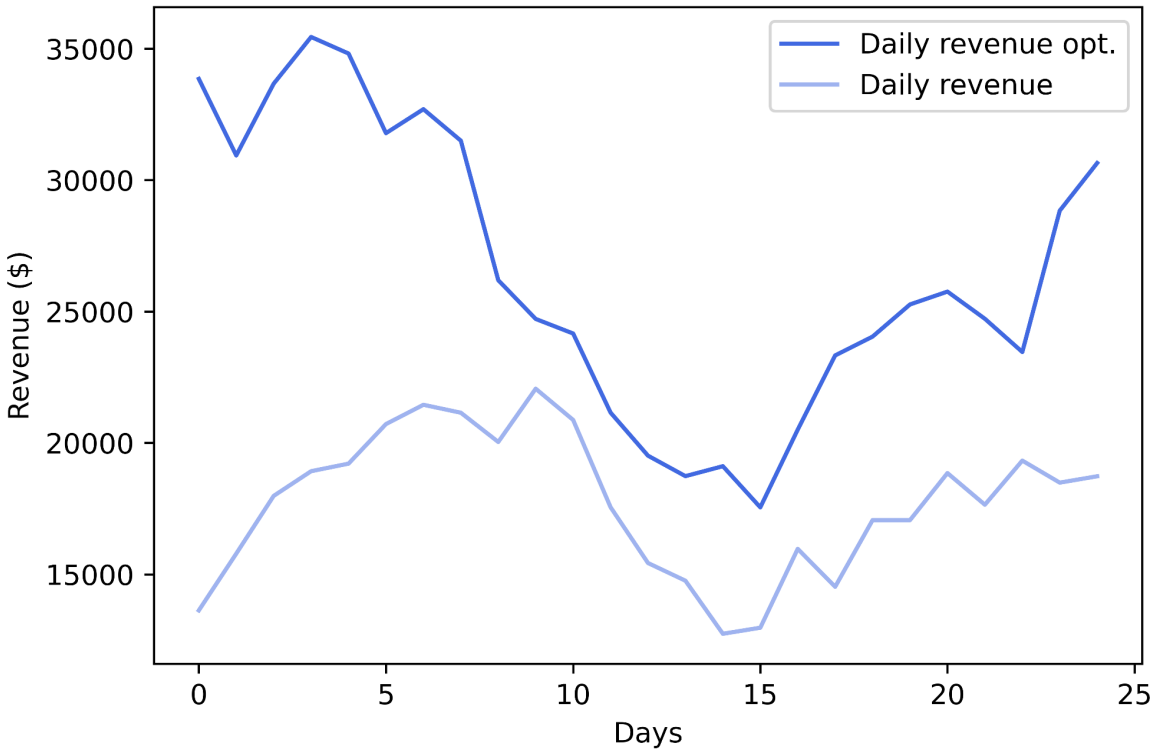

Fig. 9: Daily revenues for Almanak solution (opt.) vs current Bancor solution

When moving to a high-level view, revenue on a daily scale is an easier path to compare two different solutions and their impact on a metric. In this case, the Almanak solution outperforms the current Bancor solution on most days (**Fig. 9**). A main reason behind this is the limited TL following a volatile market. While the current Bancor solution relies on additional funds, the newly introduced risk levers help inject capital and offer lower slippage and better execution in these times.

Burning Ratio

Total vBNT burnt per pool increasing by 15% over 30-day time frame

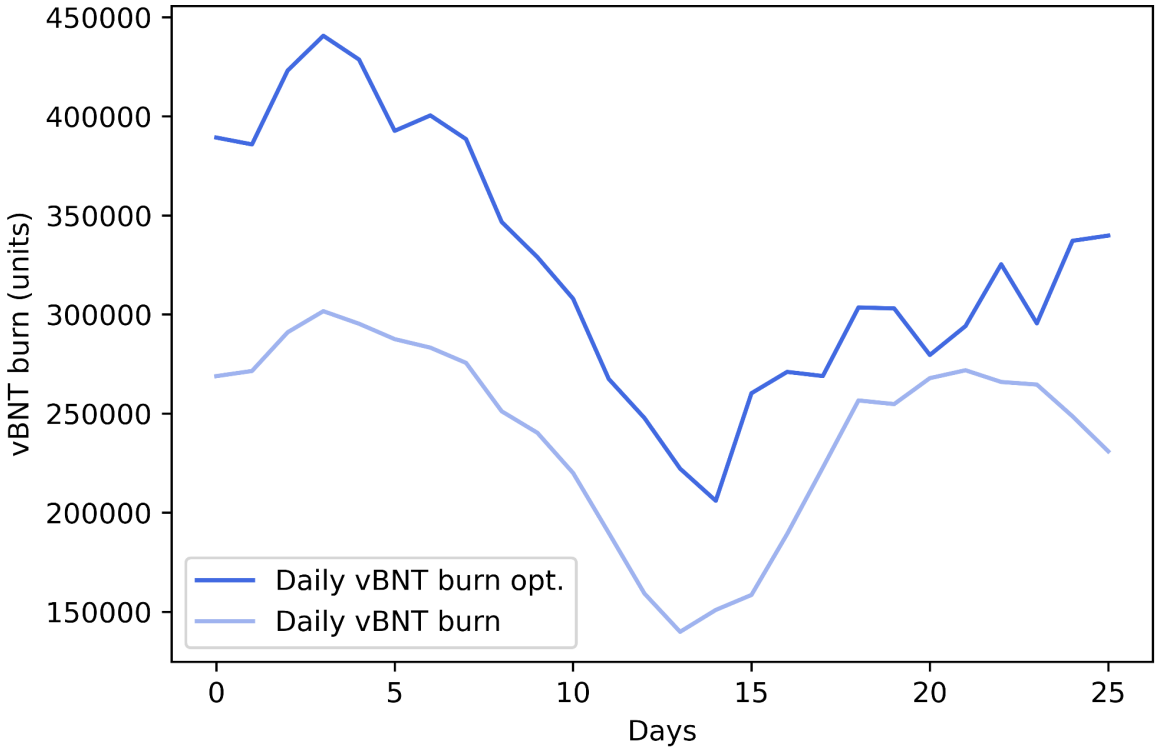

Fig. 10: Daily vBNT burn for Almanak solution (opt.) vs current Bancor solution

As the vBNT burn is pegged to a fixed split of the incoming revenue, **Fig. 10** shows high correlation of the burn of vBNT tokens with the latter.

Conclusion

This report outlined Almanak's assessment of the current Bancor V3 setup. After introducing its main features and current challenges in the aftermath of the May 2022 crash, we identified areas of improvements to both enhance the recovery plan and introduce sustainable protocol risk management mechanisms that optimize the ratio of Capital Efficiency and Deficit. Results of numerical assessment were discussed.

Despite Bancor's key competitive advantage over other existing decentralized trading venues on the market, recent market conditions and user behavior led to an emergency situation where Bancor ran into a deficit and was unable to guarantee a full-scale service. The main area of improvement identified concerns the management of the trading liquidity and the swapping fees as it can help increase the overall profitability of the protocol.

Almanak's proposed optimization methodology is based on the design of "protocol levers" and their dynamic calibration through an agent-based market simulation framework. The latter uses evolutionary algorithms and domain expertise to build a set of dynamic decision-making solutions, implemented within a scalable, modular and composable architecture. This offers the flexibility to design and deploy tailored data-driven agents, replicating actions on the Bancor protocol. Almanak framework also integrates validation methods to ensure proper generalization. The main result brought forward by the protocol risk management framework is that the protocol profitability is a function of volatility and arbitrage volume which can be actively managed through concerted tuning of the pools' dynamic swapping fees and trading liquidity.

Numerical application of the framework over simulated price trajectories and the resulting solution space were discussed. The general tendency of the preferred solution hints at larger trading liquidity and lower swapping fees at times of medium volatility. For extremely volatile markets, the solutions indicate that profitability may be optimized by maintaining either (i) very high fees and very low TL, in case the market is facing high volatilities, or (ii) very low fees and very high TL, in case of a flat market. Overall, the favored solution at this stage improves profitability by 12% and burns an additional 15% of vBNT over time.

With this first report, Almanak proposes a future collaboration with Bancor to design, deploy and maintain a customized protocol risk management framework that will ensure the protocol's sustainable growth and competitiveness.

Bibliography

[1] Hertzog, Eyal, Guy Benartzi, and Galia Benartzi. "Bancor protocol." Continuous Liquidity for Cryptographic Tokens through their Smart Contracts. Available online:https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en. pdf (accessed on 6 June 2020) (2017).

[2] Evans, Alex. "Liquidity provider returns in geometric mean markets." arXiv preprint arXiv:2006.08806 (2020).

[3] Evans, Alex, Guillermo Angeris, and Tarun Chitra. "Optimal fees for geometric mean market makers." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2021.

[4] Engle, Robert F. "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation." Econometrica: Journal of the econometric society (1982): 987-1007.

[5] Deb, Kalyanmoy, *et al.* "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation 6.2 (2002): 182-197

Appendix

A. Agent Specification

We used data from Uniswap, Sushiswap, Bancor V2.1, and Bancor V3 to create an agent model that reliably mimics the traders' and LPs' behavior. Agents read the environment and, based on their historical behavior, react accordingly, translating into an action. In our case, price trajectory simulations feed the environment with data that changes the environment. Agents then react to those changes and make decisions accordingly. Agents are continuously fed with new, fresh data and the models are retrained on a daily basis to react to market changes.

Each step involves a simultaneous action taken from all initialized agents. Given that four different pools are under responsibility and one agent can only execute one action per step, ten regular traders are initialized (one trade in either direction), where one per pool follows up with an arbitrage opportunity.

Most of the agents take features into account which might be unclear in a first instance and need clarification. Specifically, volatility and ground-truth price have different meanings, depending on the industry. We therefore want to clarify the internal definitions:

Volatility is a measurement metric to determine the risk of change of an asset x over time. The metric asserts the value spread in a timeframe and is calculated in our instance as the square root of the squared relative high-low difference (a shifted Parkinson volatility). Usually the reference timeframe is taken as an hourly time bar t.

$$\sigma_x^t = \sqrt{\left(\frac{p_{high}^t - p_{low}^t}{p_{avg}^t}\right)^2} \quad (6)$$

Ground truth might be more unfamiliar, as it is not a financial term. Ground truth relates to the actual value at a current time and not what is perceived by any player. In Bancor's environment, this translates to the value of a token on a given CEX (ground truth, simulated by Almanak) vs. the value of the token on Bancor (perceived). This becomes especially important for both arbitrageurs, seeking opportunities to rebalance and profit from a trade, as well as traders seeking the best possible price for their asset to swap.

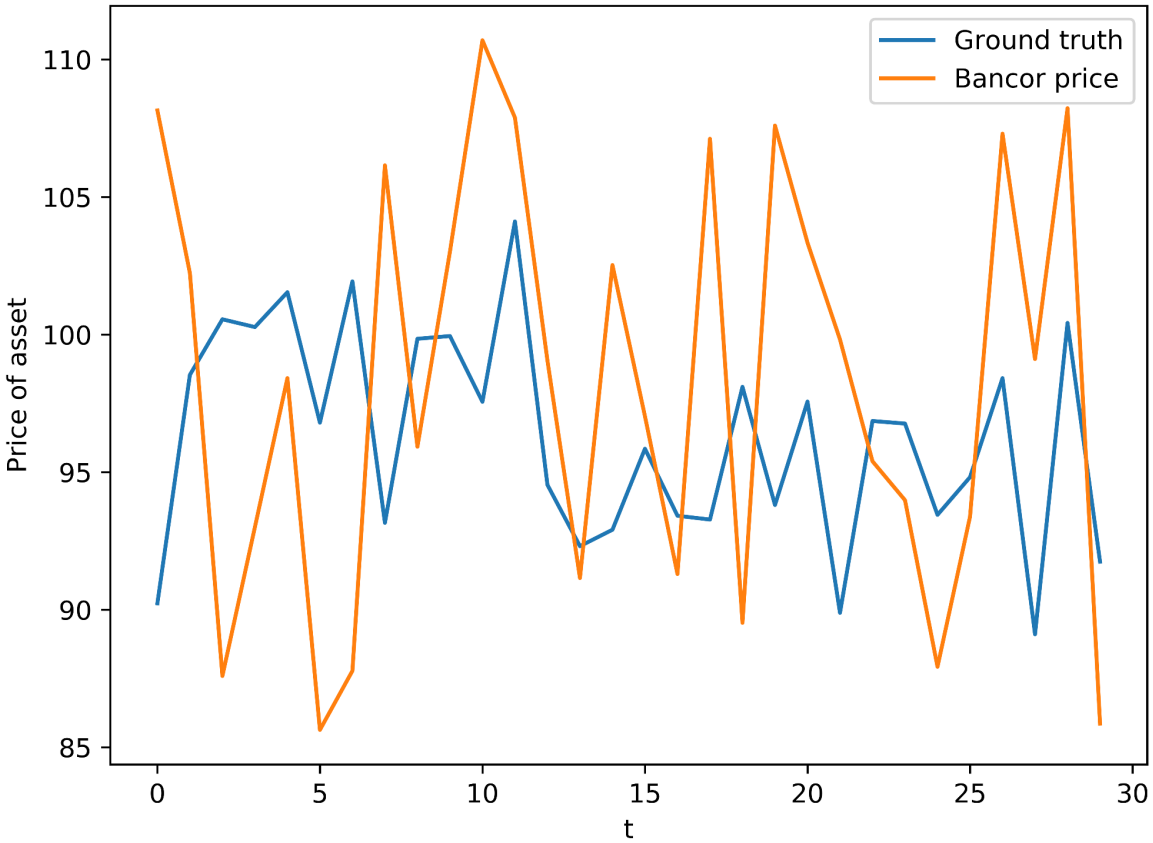

Fig. 11: Visualization of differences between ground truth & Bancor prices

Fig. 11 demonstrates the potential discrepancies between market and protocol, which can grow even further in more volatile environments.

A.1 User/LP Decision-Making

Liquidity providers (LPs) are the foundation of the protocol and offer their assets to gain a yield over time, which is generated through swapping fees. As part of the simulation, they are able to deposit assets, withdraw assets if there are any remaining, confirm their withdrawal, and stay idle. With each time step, i.e., each hour, the agent has the possibility to execute a single action, which is then executed within the same step.

The agent model takes into account the current 24-hour volatility of the asset and the price change of the asset in the past hour, among other features. Based on these features, the agent will decide what action to take as well as what amount is in play. In case of an idle action (nothing will be done), no amount is provided. This therefore results in a stacked model which provides two types of output. The first output is an action or decision, which defines what the agent will execute

afterward, followed by the amount associated with the action. This is expressed relative to the regular trade size involved or the current trading liquidity, whatever is smaller at the time.

Given the large number of LPs, a subpopulation is built by selecting statistically representative liquidity behaviors. Large and small holders/depositors (above $100 deposit) are picked along the pool distribution of the platform. The largest holders are added to provide the pool liquidity, whereas the smaller ones are integrated to simulate ongoing action in terms of deposits and withdrawals. Doing so ensures representativeness of the parent population distribution without having to actually cover the entire Bancor V3 trading liquidity. Note that additional users and their capital are progressively included to maintain a threshold.

A.2 Trader Decision-Making

To simulate regular volume on Bancor, trader agents are initialized and represent users directly trading on Bancor or through aggregators. Based on historical data, they execute a single trade within the current step with a certain amount of TKN/BNT, representing the hourly volume of either pool on Bancor. Once an EVM is integrated (currently under ongoing development), actions can be carried out on a per-block basis. A trader agent always has the possibility to trade and thus has access to funds to execute its trade.

The trading agent model takes into account the current 24-hour volatility of the asset, the price change of the asset in the past hour, and the swapping fee of the pool, among other features.

The volume emitted for each side of the pool predicts the normalized number of TKN/BNT relative to the current pool size to be traded, which is then applied to the number of tokens involved in either trade.

A.3 Arbitrageur

Arbitrage is the main income for the protocol, as agents benefit from additional knowledge and take advantage of this discrepancy to execute profitable trades. Overall, we include these agent models on the assumption that only profitable trades will be executed in regard to market conditions. We thus compare the price of assets within the pool with the values of the price simulation and determine the amount to trade given the swapping fee and additional slippage based on the total amount.

Slippage is defined as the additional impact on the price of the asset based solely on the trade, whereas the swapping fee linearly scales with the amount of tokens swapped. Based on the amount set, either the token or BNT within the pool are swapped and executed within the simulation.

Within the first iterations, we identified that arbitrage agents within the simulation do execute trades which might not be executed on-chain due to larger profit opportunities. We therefore opted to derive their behavior from previously identified arbitrage actions and condense them into the respective hourly volume. With constant monitoring, additional modeling of the arbitrage agent can be adapted in the future.

To execute a trade, the arbitrageur is eager to maximize its profits, i.e., to gain as much as possible out of the discrepancy between market and protocol, which can be condensed into:

$$p = (\Delta_{price} - \delta_q - \gamma)q - \alpha \quad (7)$$

where p is the total profit coming from a trade, Δ_{price} is the difference between pool price and market price of an asset, δ_q is the slippage applied based on the quantity q, γ is the swapping fee, and α is the transaction cost (i.e. gas and transaction fees).

A.4 Slippage

Price slippage is the change in executed price (*vs* quoted price) caused by external broad market movements (unrelated to a trade). Price slippage might be positive or negative, i.e. in favor or against the user-observed quoted price. This is not to be confused with *price impact*, which is the change in price directly caused by a specific trade itself on the current market price.

Both are dependent upon liquidity, volume, and the AMM engineering design as well as blockchain constraints such as confirmation time. These are factors to be taken into account within simulations. Particularly when considering volatile scenarios, strong differences between pool prices on Bancor and open market prices can loom, creating further potential slippage and price impact effects through liquidity and volume.

Within the Almanak simulation framework, slippage and price impact effects can materialize on revenue and deficit (unrealized IL).

- Revenue: taking advantage of market situations arbitrageurs bear maximum slippage up to a target profit. The greater the arbitrage spread, the more slippage they can bear and thus provoke. The net variational effect on profit is undetermined and will depend on the divergence between pool prices and open market prices, the latter being simulated in price trajectories.

- Deficit: unrealized IL being a function of liquidity, future trades and open market prices, slippage is also implicitly taken into account when estimating deficit and assessing how much risk is associated with each pool individually and the overall protocol.

A.5 Agent Data Sourcing

Agents within the simulation are trained on historical data, using such information and observations to learn and make decisions when running simulations. As previously seen, both LPs and traders rely on historical data and we therefore need to define how such information is retrieved and gathered.

We previously introduced volatility and how it is defined as information within the simulation. Hourly data is fetched from Binance as the primary data source. Retracing internal Bancor data is available throughout time and can be reconstructed, yet the variation in spot rates through flash trades can lead to unnecessary reactions due to high volatilities. The same can be said of price changes, as the same data is required to calculate this feature. Note that the drop is expressed in percent to guarantee comparability between assets.

Additional market data is fetched internally through on-chain transactions, especially to determine additional trade data such as amount involved or pools involved in a trade to reconstitute when and how a trade was executed. Especially for user-centered actions such as depositing and initiating a withdrawal, the size of the funds are relevant enough to determine the impact on the protocol. We thus have to rebuild the user's history of interactions with the protocol. Based on the time associated with an action, respective user data such as the share held within a pool and the amount involved are calculated and joined with the previously discussed features.

B. Price Trajectory Simulations

In order to simulate the Bancor V3 protocol path to recovery, we simulate price action for the five largest pools: ETH, LINK, WBTC, DAI, and BNT.

We consider a range of possible trading fees and on-curve trading liquidity and run price trajectory simulations under a variety of market conditions to trigger agents' behaviors. **Fig. 12** below shows 50 sample price simulations for ETH.

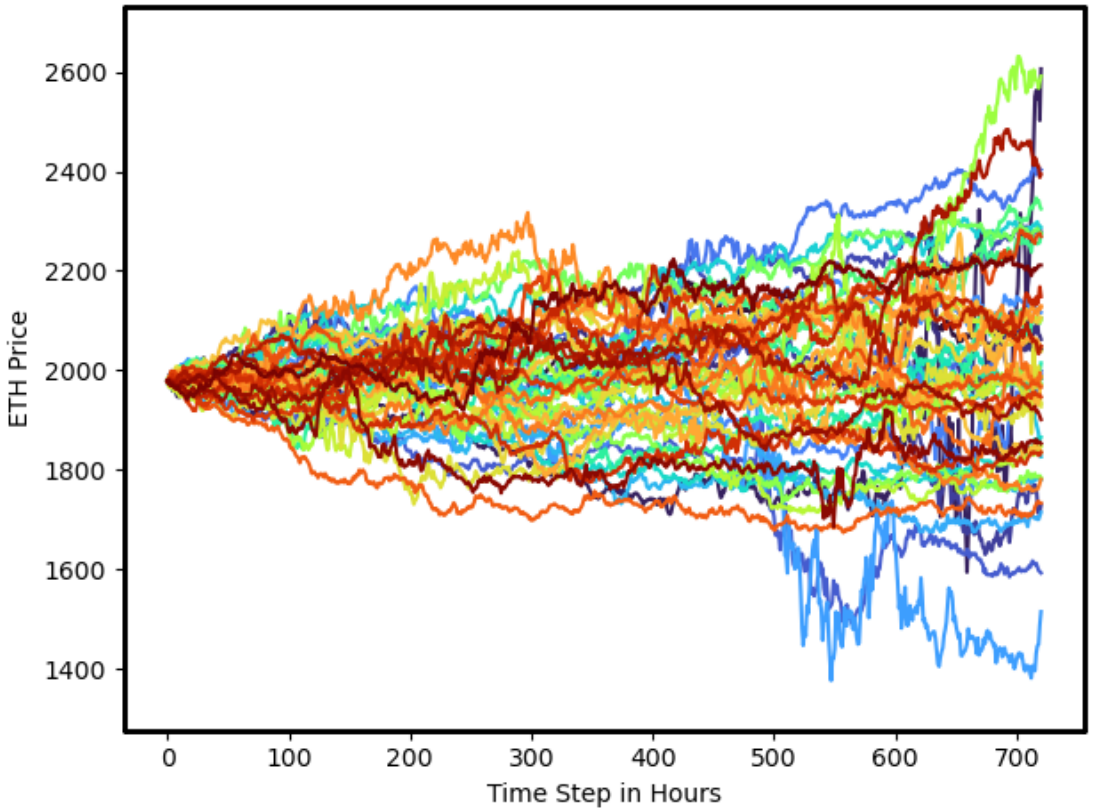

Fig. 12: Visualization of 50 different price trajectories of ETH

B.1 Asset Correlation

The current price simulation module uses asset-correlated Geometric Brownian motion (GBM) as the foundation to simulate price trajectories.

The standard formula for GBM at a given time step is as follows:

$$S + \mu S \cdot dt + \sigma S \cdot \sqrt{dt} \cdot z \quad (8)$$

Here, S is the price at the previous step, μ is the mean of the returns over the historical data (drift), $dt = 1/steps$, σ is the standard deviation of the returns over the historical data (volatility), and z is a sample from the normal distribution.

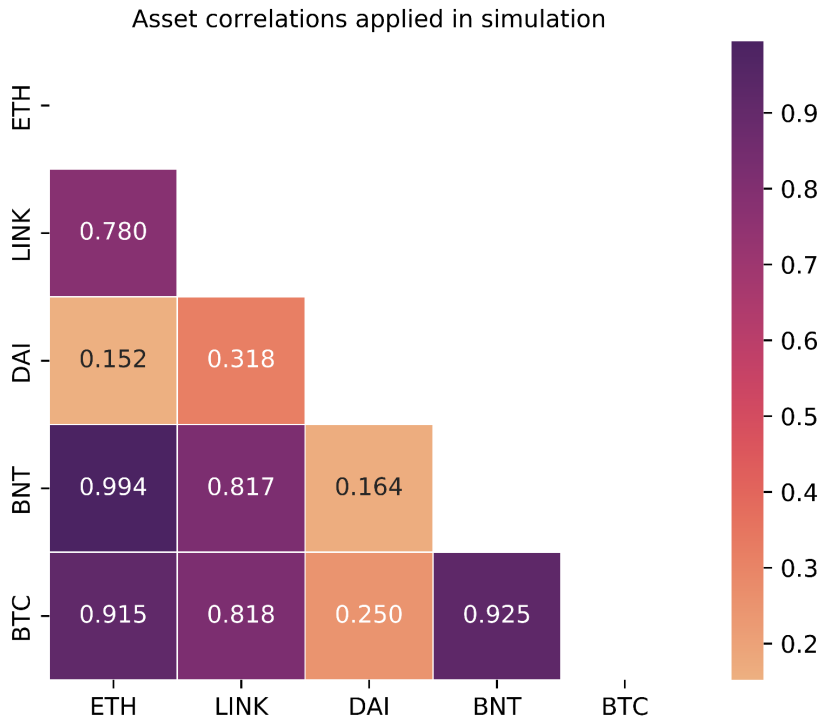

Fig. 13: Asset correlation of relevant assets over the past three months

The above generates a GBM price trajectory for a single asset, but there is a need to take into account multiple assets and how they correlate with each other over time (**Fig. 13**). This can be achieved by taking the Pearson correlation coefficient of prices over the historical data per the correlation coefficient formula, where σ_x and σ_y represent the standard deviation of either asset's price and $cov(X, Y)$ represents the covariance of the two assets:

$$\rho = \frac{cov(X, Y)}{\sigma_x \sigma_y} \quad (9)$$

We then use Cholesky factorization with the correlation matrix to generate the lower triangular covariance matrix. The dot product of this matrix and a matrix of random samples from the normal distribution produces a set of values that replace z at each time step.

B.2 GARCH Volatility

GBM uses a constant sigma derived from the standard deviation of historical returns to represent volatility. However, volatility may have a seasonal component or change over time. It would be ideal to capture this variance in volatility in addition to correlating each asset in the price simulation. GARCH, or generalized autoregressive conditional hetero-skedasticity, models time series and can simulate variance based on its estimated parameters [4]. CCC-GARCH, or constant conditional correlation GARCH, comprises multiple univariate GARCH processes related to one another with a constant conditional correlation matrix. At each time step, the volatility for the GBM price trajectory of a given asset is drawn from the fit CCC-GARCH model as opposed to depending on a constant calculated over the asset's historical volatility.

B.3 Black Swan Events

Considering recent market events and coming structural events like the merge, a price crash is incorporated into the simulation in order to predict the future behavior of agents under extreme and sudden market conditions. Due to the current lack of impermanent loss protection, this feature is not as relevant as it could be otherwise, since the Bancor bank run on BNT is not an issue at present. However, keeping in mind the willingness of the DAO to enforce external impermanent loss protection, it will be highly beneficial for the protocol to simulate bank runs on the funds stored in the external impermanent loss protection pool.

Moreover, the black swan event is still relevant as an indicator of future market sentiment and allows us to gauge market movements and react accordingly by adjusting trading fees and trading liquidity to minimize protocol exposure to loss and optimize revenue.

To simulate how the protocol responds to black swan events, each simulation trial must include a crash of varying intensity. The crash will initially adjust the price downward by a significant percentage followed by a sudden spike in volatility. This multiplier in volatility allows various trials to simulate subsequent market conditions following the initial sell-off, including dead cat bounces, continued downward pressure, or significant price adjustment to the upside, all reflecting conditions that have materialized in historical scenarios. This spike in volatility remains elevated compared to its trend pre-crash, but slowly scales downward as market sentiment reverts to the mean.

C. Optimization Framework

Based on the explanations prior and the different protocol levers to optimize, we need a structure to find the best possible solution and have a confidence interval around the proposed solution in order to offer reasonable insights.

The genetic optimization framework uses the NSGA-II (Non-Dominated Sorting Genetic Algorithm II) selection operator [5]. The fitness function is the profit, comparing the value of the vaults at the end of the time period against the initialized value. The values for each vault are then aggregated with the value of each token at initialization to ensure comparability.

The mating process executes a two-point crossover on the input sequence individuals. The mutation function applies an independent Gaussian mutation.
The optimization framework uses an evolutionary algorithm with an initial population (controlling for selection rate and crossover effect). A hall of fame is used to track the parameter sets that generate the most profit across the initial set of trials. Once the optimization has completed, top-performing candidates are then validated against an additional 1,000 trials to verify that performance is consistent across a variety of market and user conditions.

Individuals are initialized based on asset, swapping fees for each asset, and asset-based adjustments to trading liquidity. For each asset (ETH, WBTC, LINK, DAI and vBNT). For each asset, trading liquidity changes are randomly applied by buckets ensuring that trading liquidity can only decline or remain stagnant compared to its prior value. This constitues the input parameters for each asset's trading liquidity when calculating the individual's fitness value.

D. Authors

Philipp Enzinger is Lead Simulation Engineer at Almanak. He worked on predictive marketplaces in micro-mobility, enhanced data management platforms for R&D teams or focused on research in autonomous racing. He holds a master's degree in mechanical engineering.

Michael Herzyk is Almanak's CEO. He's worked 6 years in crypto as tokenomics expert and product designer for projects such as Bancor, Qredo, Stargate, Layer Zero and StarAtlas. He holds a master's degree in applied mathematics in economics.

Lars Suanet is Almanak's CTO. He has vast experience in designing, planning, building & scaling data-centric applications in the crypto space for tier one crypto firms. He also built a Machine Learning Operations software, empowering Data Scientists to deploy their ML models at scale.

Stephan Thomas is Almanak's CPO. He has worked in financial services and software for 17 years and held various positions at research, product, tech, and business levels. He ran his own R&D firm and led Quant teams at McKinsey and EY as an executive. He specializes in AI-ML operationalization and risk governance. He holds a PhD in decision theory and risk management.

E. Acknowledgments

The authors would like to thank Mark Richardson, Mike Casale and Steven Fox for helpful discussions. We would also like to thank the Almanak team, especially Nasrudin Salim, Koen van Marrewijk, Ren Xiang for their valuable contributions to the platform infrastructure.