QuantAMM: Utilising Temporal-Function Market Making for Blockchain Traded Funds

Matthew Willetts Christian Harrington

Version 0.1.0
17 November 2023

Abstract

QuantAMM is the first protocol to utilise Temporal Function Market Making (TFMM).[1] While still providing AMM functionality, core liquidity providing is not the primary objective on QuantAMM: TFMM is used to provide the rebalancing mechanism for on-chain quantitative asset management. This enables a new generation of DeFi passive products–Blockchain Traded Funds or *BTFs*. QuantAMM is targeting future institutional use by providing on-chain feature hooks that can likely enable necessary regional regulatory compliance.

Utilizing novel gradient and precision estimators, standard portfolio management strategies are outlined and simulated, demonstrating the strength of the TFMM approach outside of traditional liquidity providing.

We give a summary on TradFi fund construction and how QuantAMM utilises TFMM to provide the required on-chain infrastructure. Rigorous strategy testing approaches are also outlined to show approach efficacy.

1 Introduction

Automated Market Making and the constant-function model has been a key innovation for early adoption and liquidity, however impermanent loss [2] demonstrates how liquidity providing is a poor use of the innovation. This has led core liquidity providers to move away from AMM protocols and move towards Central Limit Order Books. Temporal Function Market Making (TFMM)[3] seeks to move past the issue of impermanent loss and LVR by continuously changing the portfolio vector of the pool. The TFMM approach leads to considerable architecture differences for fund construction compared to managed pools allowing stochastic updates to target weights.

The primary aim is to provide return through reallocation of capital rather than through trading fees. **QuantAMM is infrastructure for DeFi quantitative asset management not for core liquidity providing,** bringing a paradigm shift in terms of the primary utility of on-chain function-based exchanges. AMMs mechanics is used for price discovery and market efficiency of quantitative funds.

QuantAMM enables fund construction and demonstrates the utility of Temporal Function Market Making for running fully on-chain quantitative asset management. Here we explore the utility of TFMMs in various aspects of fund construction while determining if TFMM based fund construction can provide superior returns and reduced complexity compared to other centralised and decentralised blockchain asset management approaches.

[1] Implementing all aspects described in the TFMM litepaper.

[2] arxiv.org/ftp/arxiv/papers/2111/2111.09192.pdf. For more recent analysis see these two Twitter threads and associated Dune Analytics queries.

[3] See TFMM litepaper for mathematical background.

2 Bringing fund construction on-chain

QuantAMM complex fund construction: TFMM composite pools, for the first time, create a fully on-chain mechanism to structure a complex fund without losing price efficiency or introducing complex cross-chain trade routing stacks. BTFs are likely to be simple, however construction of an active managed fund requires many layers such as risk management and capital allocation.

QuantAMM applicable to both active and passive: Temporal Function Market Making provides the necessary rebalancing infrastructure for a fund on-chain, however it does not enforce what the fund should be. For instance on-chain products analogous to tokenized REITs can also be constructed using TFMMs. QuantAMM demonstrates textbook strategies to create Blockchain Traded Funds (see §3,4). As these strategies are known, the BTF is passively managed with low fees. Active managers require their strategies to be proprietary and secret to avoid replication.

ZK proofs can be used on QuantAMM to this end to run an encrypted strategies or encrypted proprietary tuning of a standard strategy. ZK proofs can also be used to ensure compliance with investor mandates and stated risk profiles without revealing strategy intricacies.

Bringing infrastructure for ETF-like products on-chain: ETFs of cryptocurrencies have represented a pivotal shift for institutional adoption. At QuantAMM, we believe that modern blockchain technology provides superior infrastructure to run such passive, algorithmic products on-chain, with the depositing and withdrawing to and from a pool also done directly on-chain. TFMM architecture can provide an automated, simple and efficient rebalancing mechanism – a key issue in previous attempts to bring asset management on-chain. As all aspects are on-chain rather than listed on an off-chain exchange *Blockchain Traded* Funds becomes a more appropriate categorisation.

Efficient rebalancing: A valid question of the use of TFMM rebalancing is: why not run the same strategy or fund on a CEX and rebalance as a taker. The intuitive answer is that, as TFMMs rebalance as a maker rather than as a taker, they have superior efficiency. The TFMM paper gives the theoretical treatment of this and a popular variant of the analysis is known as loss vs rebalancing.

A histogram Fig 2a is show for the simulated difference in total pool returns at the end of multiple economic cycles between QuantAMM and CEXs for a broad range of hundreds of parameter choices. Given parameter training will not select arbitrary negatively performing parameters, we can narrow down he selection to the top 5% absolute performance QuantAMM variable choices, Fig 2b, shows the efficiency is heavily in QuantAMM's favour for better strategies.

Regulatory Future-Proofing: As the regulatory momentum of cryptocurrencies and blockchain technology continues, we believe it is vital that this BTF-infrastructure-providing protocol has all the necessary compliance features, alongside a BTF creator's/manager's own regulatory requirements, to be compliant. MiCA and MiFID II are good pre-trade requirements to target despite their application to DeFi being unclear. QuantAMM also recognises that functionality is only part of the regulatory picture and company structure also important if institutional use is to be a target market.

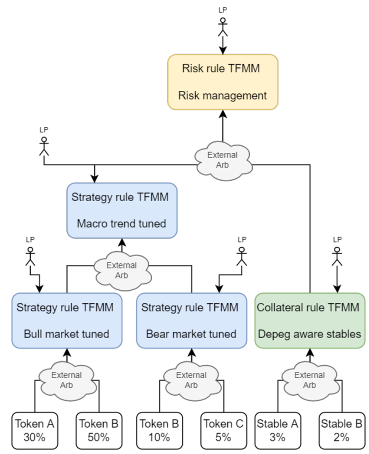
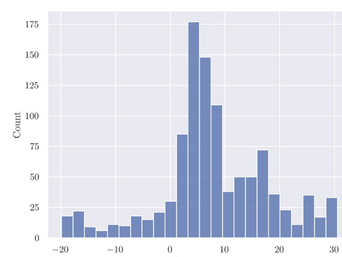
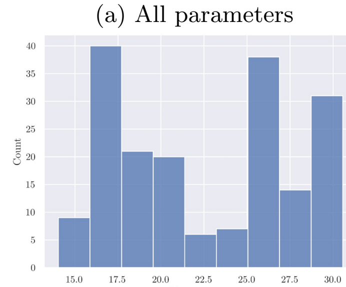


Figure 1: Example structure showing no increased trade stack complexity: different sub-pools here contain the same tokens but have different update rules (tuned to specific market conditions). A composite pool using a risk-based update rule allows for automatic risk management over sub-pools.



(a) All parameters



(b) Top 5 % parameters

Figure 2: Rebalance Efficiency vs CEX

3 QuantAMM TFMM Update Rules

BTFs are in essence baskets of assets that constantly rebalance given a mandate or strategy. The obvious next question to ask is: what are the rules that determine the weight vectors over time? Many classical strategies from traditional finance can be implemented using the tools available on QuantAMM, as well as many new approaches.

Vanilla Momentum: The father of this strategy once said far more money is made buying high and selling at even higher prices than buying low and selling high. Momentum trading is a staple of quantitative hedge funds and Commodity Trading Advisors. More recently, momentum ETFs have arisen as it can be argued that the level of active management needed is minimal.

Anti-Momentum / Mean Reversion: This economic principle states that given a long enough period of time prices tend to revert to a historic mean. This strategy involves identifying how long that period of time is and over what historic timeframe the mean should be calculated over. This is a much more refined strategy that can be considered market cycle-specific and likely should not be used by itself.

Channel Following: Often associated with weighted moving averages, this strategy expects a stable channel where prices can vary–allowing to buy low and sell high within the accepted range. Professional managers, however, understand that a price trajectory can become stale and if the price breaks out of the projected channel you need to act fast to exit a freefall position or increase the weight of a briskly-appreciating asset.

Power-Channel: Somewhere between vanilla momentum and channel following, there is an argument that small price movement noise distorts or delays acting on a good momentum signal. This strategy addresses that by ignoring those small price movements. Modelling has shown this is a particularly good approach for small cap tokens and for coins that can suddenly appreciate but whose day-to-day prices are erratic.

Minimum Variance: Mean-Variance Portfolio Theory is a foundational economic theory of asset management proposed by Harry Markowitz in 1952. Simply put, the theory uses statistical analysis comparing constituents' properties to each other to meet a given statistical target. In this case the aim is to reduce the variance of the portfolio returns as a whole.

3.1 Future potential rules

Given that TFMM *composite pools* allow you for the first time to create a fully on-chain mechanism to structure complex funds, the rules dictating the allocation of capital to subpools **do not** have to be alpha generating.

Risk management related rules All of the rules described above can be considered strategies trying to achieve a given alpha. These can be considered as types of strategies found at the base level of any fund construction. An example of standard fund construction can be multiple alpha strategy subpools being allocated capital based on a risk management tolerance. Risk can also be used as a core feature of the fund mandate, examples of this are target risk funds.

Trade Volume and volatility related rules Trade volume and volatility are other examples where changing the nature of the oracle data, while utilising the same rules, can provide a wide range of novel strategies and fund structures. While QuantAMM core update rules have been demonstrated to require signal intervals of hours or days, initial testing suggests that such rules would require a shorter interval time.

4 Testing TFMM strategies

4.1 Choosing BTFs and their Update Rules' parameters

Different strategies work best with different market dynamics. While we provide different pool examples, results for all strategies on all baskets will be shown even though some are clearly suited to a single strategy.

The QuantAMM pool SDK has over a large range tokens and allows users to experiment with any pool of their creation. This is a highly-sophisticated simulator, simulating no-arb regions, MEV protections, fees and much more. While this is the case, private beta testing and on-chain verification is important as the simulator does make some idealised assumptions that are not present in the real running of BTFs. While a full assumption description is provided in the QuantAMM whitepaper appendix and simulator code and data will be made open source, an example of such as assumption is not using real-time gas costs, rather using a fixed or periodic (e.g. weekly) average price. A React UX allows low barrier to entry.

4.2 Test variations and their biases

Historic back tests Historic simulations are almost always provided as a simple demonstration of how a product or strategy acts. A standard result set provided in the full whitepaper is a historic back test over multiple crypto economic cycles where HODL provides ∼flat returns despite multiple sustained bull and bear runs. The SDK allows you to pick any custom date range for simulation.

Stochastic gradient descent training with windowed time series Performance of historic backtests can be heavily impacted by the exact start time and the exact end time selected. A time series can be partitioned in a process called windowing. By training the model on different windows of data within the time series, this reduces start and end date bias that arises if solely performing fixed-period historic testing. See the full QuantAMM whitepaper for more detail.

Training with different window sizes can also be equated to training different pools to be performant for different LP time horizons. This allows for individual LP preference training. Other factors such as the frequency of strategy updates (e.g. hourly or daily) and MEV protection levels are also taken into account during the training process. Though initial results are tuned using SGD, other optimisation methods are explored in further technical notes. Comparisons are also made of training using feeless and fee based no-arb region modelling.

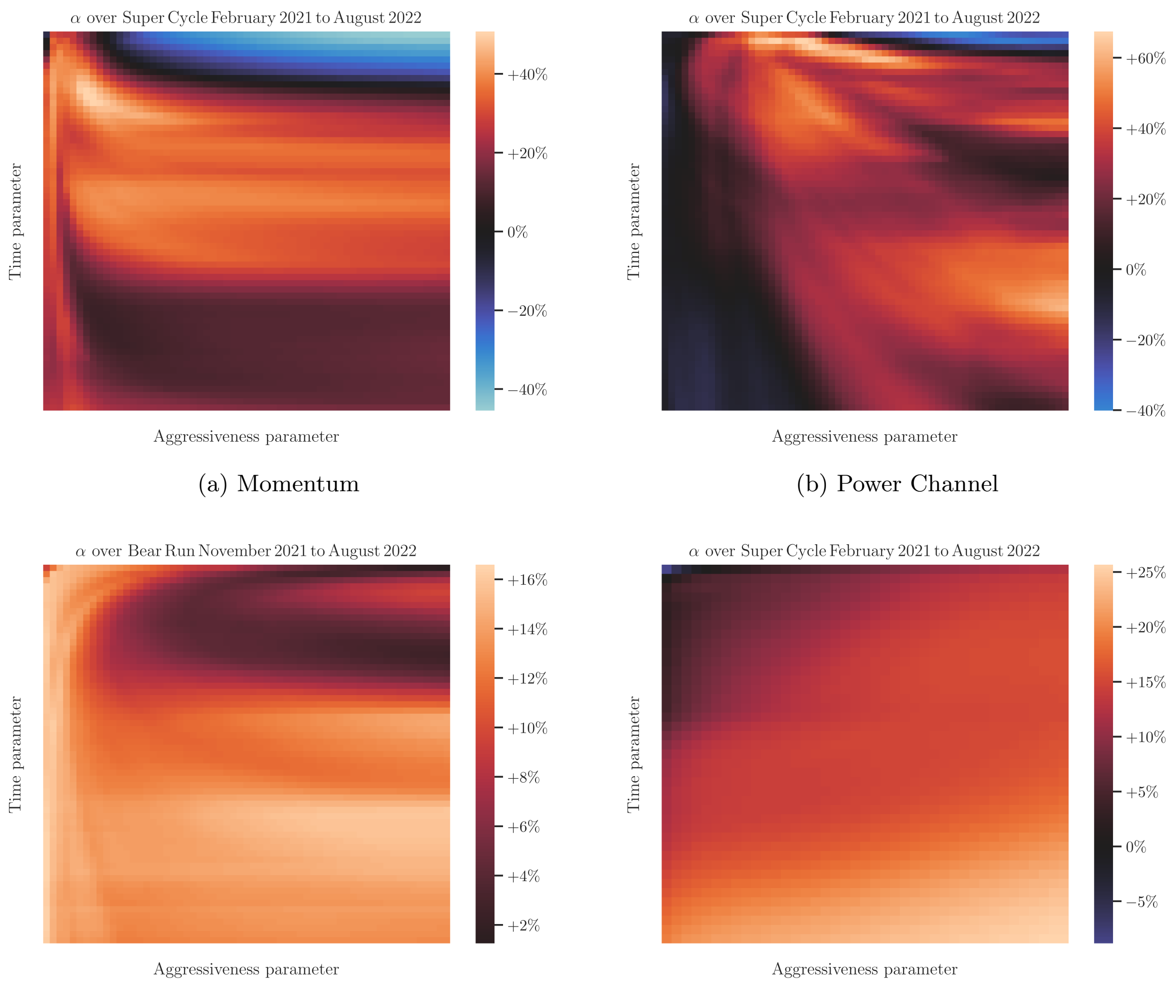
Monte Carlo testing Creating many variations of synthetic data and running Monte Carlo simulations aims to remove the bias of using only historic data. How these variations are generated is a deciding factor and is often very specific to the managers needs. A complex example of this is generation based on altered risk metrics such as Value at Risk. Monte Carlo testing can be done post training to test resilience of a configuration or the Monte Carlo prices can be used in the training itself. Our aim is to demonstrate performance over multiple economic cycles so a gated drift-diffusion model was initially chosen. More variations will follow in technical notes.

Strategy resilience training While training parameters can be done with a range of objectives (eg. absolute returns, Sharpe Ratio, Alpha, minimising maximum drawdowns etc) a range of techniques are explored to either improve strategy performance in post training window test data or tune a pool for a given risk profile. Two initial examples provided in this paper are higher order objectives and ensembling.

5 Example pool results

The full QuantAMM whitepaper has detailed results for all test variations and assumptions. As a visual example, alpha generated above a benchmark of counterfactual HODL is shown, Fig 3, for a range of update rule tuning. Tests here are performed over multiple crypto economic cycles between February 2021 and August 2022 that includes both bull and bear runs. We use the QuantAMM simulator to perform this analysis. No uninformed flow is modelled–these results are from modelling *only arbitrageurs* interacting with each QuantAMM pool.

The below heatmaps of Jensen's alpha for broad ranges of parameter (each pixel being a separate pool simulation) are visually digestible. However, this applies the parameter configuration uniformly to all constituents. By applying constituent level (vector) parameter choices one drastically increase performance. Channel following, perhaps the strongest update rule, has considerably more than two parameters so cannot be plotted easily.



(a) Momentum

(b) Power Channel

(c) Anti-momentum: Bear Run only

(d) Minimum Variance

Figure 3: wETH, wBTC, and DAI basket. We plot Jensen's Alpha, using HODL as the benchmark and a fixed $R(f)$ of 5% (even though this is much higher than the true $R(f)$ for the simulation period it is more realistic for today's environment). The simulator can be configured for individual runs to use daily DTB3 rates for $R(f)$. Each 'pixel' is a separate run over with different parameters (which can be read of the x and y axes). Anti-momentum is demonstrated in a specific market condition, to show how you can construct market-condition-specific base pools.

6 Improving strategy resilience

When tuning a strategy, focusing the training objective solely on increase returns may be detrimental to the future resilience of the strategy. In this section we explore a range of approaches and techniques to incorporate increased resilience to strategy training.

6.1 Ensembling parameters

The idea of 'ensembling' is common in applied statistics, where many small models are averaged to give an overall output, which has previously found use in finance.

The intuition is that when the update rules in the ensemble are making good decisions they are in agreement, but when they are making bad decisions their outputs tend to be bad in different ways. In other words, there are more ways to be wrong than to be right. Thus by averaging their outputs, we hope that the good choices are amplified and the bad choices cancel each other out (at least partially).

To test the effect of ensembling parameters in detail, in Fig 4 we provide the test results for ensemble momentum given the parameter set in Fig 3a.

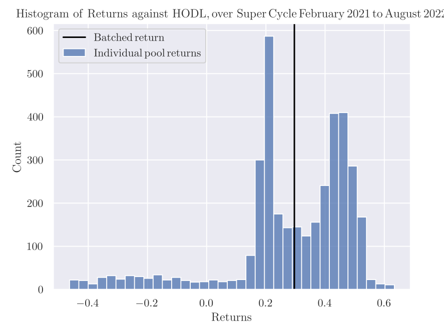


Figure 4: Histogram showing results for individual update rules in a set (blue) and the performance obtained from ensembling those update rules into one (black line).

6.2 Higher order methods

Here we will quickly showcase how the Hessian (the square matrix of second order derivative of a multi-input function) can be used to construct a metric for evaluating pools post-training. The Hessian's trace enables us to give us a rough, indicative metric for test-set performance that can be calculated purely from the pool's properties on training data.

In Fig 5 for trained parameters we plot true test-set performance against the trace of the Hessian. These hundreds of different parameter settings were found during the course of training an anti-momentum rule on a basket of wBTC, wETH and DAI over the same set of economic cycles and test period as in §4. As one would hope, we find a positive correlation between the Hessian's trace on training data and test set returns.

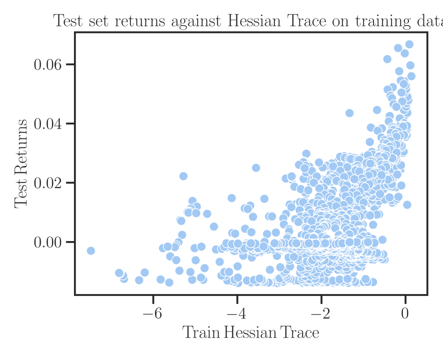


Figure 5: Plot showing test set performance against the trace of the Hessian calculated on training. This demonstrates how the Hessian's trace can be used as a training-time metric to predict test-time performance.

By utilising strategy resilience techniques, especially if combined with windowed training and training based on Monte Carlo data, test performance can be brought into positive alpha. Such techniques are especially important of a QuantAMM BTF has fixed parameters and is not configured to retrain periodically on more recent data.

7 Efficiency of QuantAMM fund construction approach

A key objective of TFMM architecture and design is to be able to run entire fund infrastructures on a single L1 without high gas costs nor complex protocol infrastructure. Below are efficiency figures assuming release before solidity transient store upgrades. These figures are for n calls (the first call may be more expensive due to access of cold storage). Some greater efficiencies may still be achieved, though the following gives a sense of running costs for QuantAMM BTFs.

It is important to emphasise that the following are total running costs. There are no off-chain steps and no additional chain calculation structures. The only variable is oracle call cost, which to begin with will be subsidised for Chainlink Oracles. Running costs are charged back to the LPs of the pool during the automated update calls using pool dilution mechanisms.

7.1 BTF running costs

Update Rule	Basket Size	Daily end to end running cost
Minimum Variance	2	164.5k
	4	221.9k
Momentum	2	167.3k
	4	221.6k
Anti-Momentum	2	167.2k
	4	221.5k
Power Channel	2	193.4k
	4	268.7k

7.2 Trading with QuantAMM

QuantAMM allows block trading functionality. This can drastically reduce the gas cost of large blocks of trades. While block trades are allowed, a standard lightweight swap function is also provided. We provide a 2 block trade example however one could have dozens of swaps within a single transaction.

	Swap Trade	1 Swap Block Trade	20 Swap Block Trade
Single Transaction Cost	114,985	161,438	701,608
Per swap cost	114,985	161,438	35,080

7.3 Vault design

The QuantAMM vault implements a single vault design for all pools. This design is employed by other protocols and means that QuantAMM supports inexpensive multi-step trades within a single transaction. QuantAMM allows pool invariants to be transiently broken within each transaction, as long as the invariants are satisfied at the end of the transaction.

This flexibility is important for composite pool trading: for example, in order to trade across the pools constituting a composite pool, it may be necessary to move liquidity between pools, and the optimal sequence of trades may transiently break the invariants of the underlying pools.

Other protocols span their vault functionality over many contracts due to their need to service many pool types. This drastically increases the number of smart contracts required and SLOAD/SSTOREs required. A single vault contract design where a trade only invokes the single vault contract allows for gas efficiency improvements. A technical downside to single vault design (especially when aiming to have reduced external library or contract calls) is having to decrease readability to maximise functionality within the contract deployment size limit.