Genetic lgorithms: Case Study on Generic Strategies

Ahbab Abeer, Ahmad Choudhary, Hemasai Suhas Kurapati
Duke University

1 Introduction

A central challenge in quantitative finance is dealing with a constantly changing market environment. Market regimes shift, correlations evolve, and strategies that used to work in a given time period suddenly do not anymore. Consequently, the difficulty lies not only in strategy development, but also in finetuning strategy parameters for specific market conditions, specific asset groups and more.

We begin with a generic momentum based strategy, and a generic mean-reversion strategy, and rebalance periodically with new ideal stocks for those strategies and new ideal strategy parameters for those new stocks. To choose the best parameters, we use a genetic approach that tests many possible strategies against recent data, and adapts to shifting market conditions. For weight allocation, we utilize hierarchical risk parity (HRP), combined with risk measures, such as a drawdown circuit breaker, individual strategy drawdown controls, portfolio level drawdown controls, a falling knife guard, and regime detection using the volatility of SPY, among others.

1.1 Genetic lgorithms

Genetic algorithms (GAs) are a class of algorithms inspired by biological principles of natural selection [1]. Over successive generations, candidate solutions are iteratively evaluated, and superior performers are kept while inferior candidates are eliminated. Through cycles of selection, crossover, and mutation, GAs are able to explore large, solution spaces where traditional optimization methods can struggle. We use a modified version of this basic framework of genetic algorithms. We summarize our algorithm here to give context for the literature review, but more details are covered in section 3.

Our trading system employs an adaptive optimization process that runs in 35-day rebalancing periods. In each period, the initial step is stock selection: for each of the two base strategy types (mean-reversion and momentum), 25 stocks are chosen based on their respective returns over the last 150 days (this selection process is discussed in **section**). This results in a total of 50 chosen stocks (25 for mean-reversion, 25 for momentum) that will be optimized.

For each of these 50 stock-strategy combinations, 300 candidate strategies are generated by sampling from an initial distribution of parameters. To select the best strategy for each stock, each candidate is rigorously evaluated using a custom two-stage backtest: first, over the stock's performance in the preceding 160 to 60 days (the training window), and then validated with its performance in the more recent 60 to 0 days (the validation window). A composite objective score is then calculated, which rewards favorable performance metrics, specifically total return and Sharpe ratio, while simultaneously penalizing unfavorable risk characteristics like drawdown.

Following this detailed evaluation, top 3 candidate strategies with the highest objective score for each stock are kept. For each stock, their set of 3 stocks will work together to trade for the duration of the upcoming 35-day cycle (See section 3.3.2 about ensamble learning). Thus, each of the 50 chosen stocks trades with the parameters that worked best during the training period. Crucially, the parameters of this top performer are used to update the initial parameter distribution, creating a feedback loop for continuous improvement. This entire process—of generating, evaluating, trading, and updating—is repeated every 35 days for the next generation. The core purpose of this structure is to ensure strategy parameters remain dynamic, adapting automatically as both market conditions and the ideal set of stocks for each base strategy change over time.

lgorithm 1 Multi-Strategy Trading System

```
 1: procedure EVERY35DAYS
 2:     Generate ideal 25 stocks for momentum
 3:     for each generated stock do
 4:         Generate 300 strategies
 5:         for each strategy do
 6:             Evaluate each strategy
 7:         this stock's ideal parameters $\leftarrow$ parameters of the top 3
 8:
 9:     Generate 25 stocks for mean-reversion
10:     for each generated stock do
11:         Generate 300 strategies
12:         for each strategy do
13:             Evaluate each strategy
14:         this stock's ideal parameters $\leftarrow$ parameters of the top 3
15:
16:     Update generating distributions
17:
18: procedure EVERYDAY
19:     (Weight generation)
20:     (Risk control)
21:     Trade the ideal 25 momentum stocks using their ideal parameters
22:     Trade the ideal 25 mean-reverting stocks using their ideal parameters
```

Stock selection, weight selection, and risk management are discussed in section 2.4 and section 3.

2 Related Works

Our proposed framework for optimizing algorithmic strategies has four major components: evolutionary algorithms, primitive trading strategies, portfolio optimization, and risk mitigation. This section of our paper reviews the foundations of each component and justifies their integration into our adaptive trading system.

2.1 Genetic lgorithms

The fundamental advantage of GAs lies in their ability to handle non-convex, discontinuous, and multi-objective problems without requiring gradient information [2, 3]. This makes them good for trading strategy parameter optimization, where the fitness landscape is typically rugged and characterized by multiple local optima. Empirical studies comparing GAs to conventional quadratic programming approaches in portfolio selection demonstrate that GAs outperform traditional methods in terms of Sharpe ratio and out-of-sample generalization, especially with cardinality and quantity constraints [2].

However, the use of GAs to trading is still new compared to use in general optimization. Salman et al. [4] demonstrate that GA based optimization of trading strategies produces statistically significant improvements over individually optimized strategies and conventional technical analysis approaches. Another practitioner-oriented study implementing GA for strategy optimization in financial modeling contexts reported higher-quality solutions and improved robustness compared to manual configurations [5].

Key hyperparameters that govern GA performance include population size, mutation rate, and crossover probability. The mutation rate must be carefully calibrated: rates that are too low result in premature convergence to suboptimal solutions, while rates that are too high introduce excessive randomness that prevents convergence altogether. Similarly, the crossover probability (typically between 0.6 and 0.9) balances the exploration of new parameter combinations against the exploitation of promising regions of the solution space [6]. Within the context of our project, there are also what we call **implied hyperparameters**, which refer to the time periods in which the genetic algorithms run and replicated. We explore this concept further in the Project Plan section.

Importantly, GAs are great at maintaining population diversity across generations, which prevents convergence to local optima – a huge challenge in algo trading. Population heuristics inherently explore multiple regions of the solution space simultaneously, a property that single-point optimization methods do not replicate [5, 2]. Standard guidance on GA design emphasizes tuning mutation and selection mechanisms to sustain this diversity and avoid premature convergence [7].

2.2 Primitive Trading Strategies

The two base strategies employed (momentum and mean reversion) are selected for 3 reasons: their effectiveness is well-documented in academic literature; they have a relatively small number of parameters that need tuning (entry and exit thresholds); they exhibit complementary characteristics that justify portfolio-level diversification.

2.2.1 Momentum Strategies

Momentum strategies profit from the positive autocorrelation in stock returns over medium-term horizons (typically 3-12 months). The mechanism is straightforward – a strategy purchases assets that have exhibited strong recent performance (past winners) while short-selling assets with weak recent performance (past losers).

The empirical foundation for momentum is pretty well documented. Jegadeesh and Titman [8] provide evidence that momentum strategies generate statistically significant abnormal returns. Subsequent research decomposing momentum profits into constituent components reveals that the momentum effect can be linked to predictable patterns in returns that go beyond static cross sectional differences in expected returns [9].

This gives us a guideline of selecting stocks for our momentum strategies: select stocks with the **highest** autocorrelation!

2.2.2 Mean Reversion

Mean reversion operates under the hypothesis that asset prices fluctuate around a long-run equilibrium level. When prices deviate significantly from their historical mean (whether through fundamental undervaluation or behavioral overreaction), mean reversion strategies anticipate a reversion toward the mean. Operationally, strategies short overvalued assets and long undervalued assets.

The theory of mean reversion is multifaceted. Barberis et al. [10] say that short-term mean reversion is an underreaction, when conservative investors fail to fully incorporate available information into prices. Daniel et al. [11] propose an alternative mechanism: overconfidence followed by attribution bias generates initial overreactions in the short term and mean reversion over longer horizons. Regardless of the underlying mechanism, mean reversion is detectable in oscillating markets.

This gives us a guideline of selecting stocks for our mean-reversion strategies: select stocks with the **lowest** autocorrelation!

2.3 Objective Function Design

The effectiveness of genetic algorithms depends on the specification of the fitness function, which gives the objectives for the optimization process. In real trading, multiple performance objectives at the same time can conflict (ex: maximizing absolute returns, minimizing drawdowns, maximizing win rates, and achieving consistent risk-adjusted performance).

Research comparing alternative fitness functions for strategy optimization shows us that Compound Annual Growth Rate (CAGR), correlation-based metrics, and profit-per-trade metrics, and Sortino ratio are often better choices because they each measure fundementally different things [12]. The chosen objective function must balance rewarding consistent performance, penalizing excessive drawdowns, incorporate risk adjustment, etc. These metrics tend to produce strategies with the highest average net profit and superior risk-adjusted returns on **out of sample data**. We also found through empirical testing that our Sharpe ratio much higher with using an objective function utilizing these new metrics, rather than simply utilizing Sharpe ratio and/or return.

2.4 Overfitting Mitigations

One of the most pervasive pitfalls in algorithmic trading is overfitting—strategies that perform well on historical data but fail to generalize to future periods. For our system, overfitting presents a particularly pressing concern due to the nature of genetic algorithms. These algorithms operate as a "black box," optimizing parameters to maximize an objective function without providing explainable rationale for parameter selection. Consequently, chosen parameters may optimize recent historical data with no inherent capacity to generalize across time periods. Without proper mitigation, future performance could be severely degraded as parameters hyperoptimize to the idiosyncrasies of the recent past.

We address this challenge through five complementary mechanisms:

- **1) Train-Test Split:** The fundamental challenge is ensuring that optimal parameters generalize beyond the training period. We explicitly validate this by partitioning data into training and testing sets. Parameters are optimized on the training period, then evaluated on a more recent test period. Parameters that hyperoptimize to the training set are filtered out through their poor performance on unseen test data, ensuring robust selection.
- **2) Frequent Rebalancing:** Parameters are optimized for recent market conditions. As time progresses beyond the training period, these parameters become progressively less relevant to current market dynamics. By rebalancing every 35 days, we maintain recency between the data used for parameter optimization and the data being traded, reducing the risk of parameter drift.
- **3) Ensemble Voting:** Rather than committing to a single optimal parameter set per asset, we retain the top three candidates. During trading, each of these 3 parameter sets generates a signal, and a weighted combination of these individual signals becomes our trading signal. This reduces the reliance on a single hyperoptimized strategy.
- **4) Distributional Validation:** We continuously monitor whether stocks retain the characteristics that qualified them for selection. For momentum strategies, stocks are selected based on high autocorrelation. During the trading period, we cease trading any stock whose autocorrelation falls below 0.02. This ensures that the distributional properties of live trading data remain consistent with those of the training data, preserving the validity of optimized parameters.
- **5) Objective Function Design:** Our objective function is designed to reward conservative strategies and penalize excessive turnover. This design choice inherently discourages hyper optimiza-

tion of raw returns alone, pushing the genetic algorithm toward parameter sets that exhibit stability, risk mitigation and greater reproducibility rather than fitting historical noise.

- **6) Strategy-level Risk Control:** Despite these preventive measures, markets are always unpredictable. We implement deep strategy level risk metrics, (ex: drawdown informed position weights), to ensure that any strategy exhibiting harmful behavior cannot have large damage on the overall portfolio.

2.5 Diversification

As shown in class, incorporating more uncorrelated stocks into a portfolio can boost the Sharpe ratio. The following mathematical proof demonstrates this relationship, assuming equal weights, equal returns, and no covariance across stocks:

$$\text{Sharpe}_{\text{port}} = \frac{r_{\text{port}}}{\sigma_{\text{port}}} = \frac{r}{\overline{\sqrt{n}}} = \sqrt{n} \;\; \frac{r}{\sigma}\bigg) = \sqrt{n}\,\text{Sharpe}_{\text{stock}}$$

This boost persists even when returns exhibit non-zero covariance, though the effect is weaker. However, diversification presents a trade-off. While adding more stocks reduces portfolio variance through diversification, it may also dilute signal quality. Stocks with weaker momentum characteristics—such as lower autocorrelation—contribute less to overall returns. **Through empirical testing, we determined that 25 stocks per strategy optimally balances diversification benefits against signal strength, maximizing risk-adjusted returns without over-diluting the predictive power of our selection criteria.** Too much, and we found that those later stocks had relatively poorer returns.

Furthermore, diversification across strategies provides an additional Sharpe ratio enhancement. Our dual-strategy approach exhibits low inter-strategy covariance by construction. Momentum strategies profit when trends persist, while mean-reversion strategies profit when trends reverse. This allows us to simultaneously benefit from both stock-level and strategy-level diversification.

2.6 Regime Shifts

A critical element in the proposed framework is its ability to automatically adapt to shifting market regimes through reparameterization and evolving parameter distributions. This represents a **modification** of the traditional genetic algorithm structure, which seeks to converge toward a single optimal strategy for a fixed, unchanging task. In financial markets, such convergence is undesirable—no single apex-level strategy can perform optimally across all market conditions. Consequently, we cannot effectively utilize **mutation** in the traditional sense, where genetic algorithms iteratively refine existing strategies over multiple generations. For instance, during high volatility periods, parameters should adapt (**temporarily shift**) toward more conservative configurations, while low volatility periods may favor more aggressive positioning.

Our modification abandons inter-generational strategy inheritance. While we planned to utilize this, heuristic experimentation found that the following approach is more effective in our current context. Every 35 days, we completely discard all previously used strategies and generate 300 entirely new parameter sets for evaluation. This prevents convergence onto a single, static strategy that may become obsolete as market conditions evolve. However, we retain the concept of adaptive exploration through a shifting parameter distribution. If certain parameter regions produced strong performance in the previous period, we want to concentrate our search around those regions while still allowing for broader discovery. This naturally let us to the normal distribution, where the majority of samples are around the mean, but there is still a significant chance of outlier parameters.

To achieve this balance, we model each parameter using a Gaussian distribution whose mean and standard deviation are set to the mean and standard deviation of the top-performing parameter sets from the previous generation. This approach preserves mutation-like exploratory behavior—parameters can drift to new regions as market conditions change—and ensuring that the brunt of samples are sampled from relevant regions of the solution space.

While beyond the scope of our work, recent research on adaptive portfolio optimization using reinforcement learning and regime switching models shows that agents explicitly conditioned on latent macroeconomic regimes can possible achieve superior risk-adjusted returns and exhibit faster recovery following market crashes compared to blind approaches [13].

2.7 Risk Controls

2.7.1 Portfolio Construction

Generating alpha is only half of the problem – controlling the risk associated determines whether those gains can be realized consistently. For this project, we first focused on portfolio construction methods that would help manage this risk.

We began by exploring classical approaches like Markowitz's Mean-Variance Optimization (MVO) and the Critical Line Algorithm (CLA). However, these have shown in the literature to be numerically unstable when applied to covariance matrices of correlated assets. When correlations are high, MVO behaves like an "error maximization" machine: small estimation errors in the covariance matrix produce extreme, concentrated weights that fail OOS [14]. This is particularly problematic in equity universes where by-sector clustering creates natural correlation.

Hierarchical Risk Parity (HRP) was an interested alternative. This technique offers leverages the correlation structure rather than fighting it [14]. It uses hierarchical clustering to reorganize the covariance matrix, grouping assets by the similarity of their return series. Capital is then allocated via recursive bisection, moving strictly downward through the tree. This ensures that allocation decisions respect the hierarchy: capital is first divided between uncorrelated clusters (e.g., broad sectors) before being subdivided among correlated constituents within each cluster. This structure naturally prevents the failure mode where highly correlated substitutes produce offsetting extreme weights. Empirical studies suggest that HRP yields lower out-of-sample variance and superior diversification ratios compared to quadratic optimization methods, making it well-suited for our momentum-driven strategy where return forecasts are inherently noisy.

Beyond diversification across assets, position sizing must adapt to time-varying volatility. The literature on trend-following consistently emphasizes that scaling positions by the inverse of realized volatility – aka volatility targeting – is important for stable performance across market regimes [15]. By normalizing returns through their rolling standard deviation (σ), the approach ensures that each asset contributes constant risk over time, regardless of whether the market is in a calm or turbulent phase. We plan to implement this framework as a continuous rebalancing mechanism that adjusts exposure dynamically.

Another extension we find particularly promising is modulating exposure based on signal conviction. Rather than treating momentum as binary, we aggregate signals across multiple lookback windows (5, 10, and 20 days) and smooth the result through ensemble averaging. This multi-horizon approach addresses a fundamental challenge in trend-following: any single lookback period is arbitrary and noisy. By diversifying across time-scales, the strategy can extract more robust trend information [16]. We use a hyperbolic tangent transformation ($\tanh$) to bound the aggregated signal, creating a smooth mapping from statistical trend strength to capital deployment. In preliminary tuning, this sizing appears to improve Sharpe relative to fixed-weight momentum.

2.7.2 Exit Risk Management

While portfolio construction implements risk management at the preventative level, we also need mechanical exit rules to handle tail events that can cause huge fall. This is where the role of stops becomes critical.

The case for stop-loss rules depends on market structure. In markets exhibiting serial correlation or momentum persistence, stops can add value by cutting exposure during adverse trend reversals [17]. Momentum strategies are particularly vulnerable to sudden crashes where trend-following positions experience sharp, synchronized reversals [18]. A trailing stop , (ex: 15% from peak) truncates the left tail of the return distribution. This asymmetry preserves capital during extended downturns while allowing profits to run during favorable trends.

Complementing position-level stops, drawdown controls operate at the portfolio level and are an adaptive form of risk reduction. While stop-losses are tied to price levels, drawdown brakes respond to cumulative losses exceeding a predefined threshold [19]. This reduction of exposure embodies risk budgeting principles, where capital allocation depends not only on marginal risk but also on drawdown recovery potential (remembering the triangular rule of gains and losses!) [20]. We hypothesize that implementing these controls will isolate underperforming components and prevent cascading damage.

3 Project Implementation

Our implementation architecture has three primary layers: (1) an asset selection layer that identifies candidates exhibiting characteristics aligned with momentum or mean reversion behavior, (2) a genetic algorithm layer that continuously optimizes strategy parameters using walk-forward validation and ensemble learning, and (3) a portfolio construction layer that dynamically allocates capital using hierarchical risk parity while incorporating regime detection and trailing stops.

The system operates on a rolling rebalance cycle of 35 trading days. At each rebalance, we reconstruct the asset universe, re-run genetic algorithms to optimize parameters for both momentum and mean reversion strategies, and recompute portfolio weights. Between rebalances, continuous signal strength metrics driving position re-adjustments, regime filters to modulate overall exposure, and trailing stops to provide downside protection, we explore significant risk management. This design balances the need for parameter stability (avoiding excessive turnover from daily reoptimization) against the need for adaptive behavior in changing market conditions.

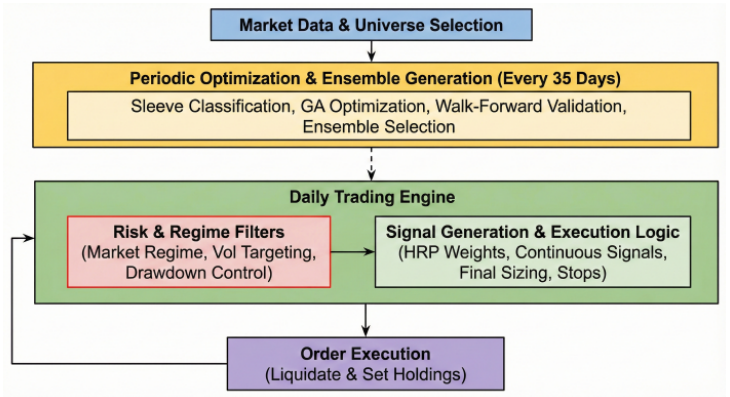

Market Data & Universe Selection
Periodic Optimization & Ensemble Generation (Every 35 Days)
Sleeve Classification, GA Optimization, Walk-Forward Validation, Ensemble Selection
Daily Trading Engine
Risk & Regime Filters (Market Regime, Vol Targeting, Drawdown Control)
Signal Generation & Execution Logic (HRP Weights, Continuous Signals, Final Sizing, Stops)
Order Execution (Liquidate & Set Holdings)

Figure 1: High-level architecture details. Charts the course of stock data through the trading system.

3.1 sset Universe and Selection

We begin with the QC500 universe, a liquid subset of the top 500 US equities. **t each rebalance, we restrict selection to a subset of up to 400 symbols by market cap drawn from the QC500 universe for computational tractability, and because it gave us better results (maybe because large cap stocks are less volatile)**. We take these 400 stocks and fetch the prices of the last 100 trading days for stock selection.

For each candidate asset, we compute rolling autocorrelation of daily returns as the primary selection criterion. Let $r_{i,t}$ denote the return of asset i on day t. The first-order autocorrelation is:

$$\rho_i = \text{Corr}(r_{i,t}, r_{i,t\ \ 1}) = \frac{\text{Cov}(r_{i,t}, r_{i,t\ \ 1})}{\sigma_{r_i}^2}$$

where $\sigma_{r_i}^2$ is the variance of returns for asset i over the lookback window.

Assets are then partitioned into two tracks:

- **Momentum Track**: We select the top 25 assets ranked by ρ_i in descending order. High positive autocorrelation indicates to us persistence in price trends, the empirical signature of momentum effects.
- **Mean Reversion Track**: We select the top 25 assets ranked by ρ_i in ascending order. Autocorrelation near zero or negative tells us this stocks have oscillatory behavior that makes them conducive to mean reversion strategies.

The use of two strategy primitives ensures that in addition to asset diversification, we are also able to explore strategy level diversification. While our initial plan also included the use of a third pairs-trading track, we found that this primitive did not optimize well with our genetic algorithm structure, often trading losses. Our hypothesis for this is that mean reverting strategies and momentum strategies typically operate on a similar time-scale (since their selection is so similar) which would allow for them to neatly work in conjunction given our fixed look-back periods and rolling variables while a pairs trade might often need a different set of parameters just for itself. **s discussed in class, pairs trading makes money when the gap between pairs close. This often takes an undefinite amount of time, which lends badly to our strategy of 35 day periods.**

As such, we chose to intentionally concentrate on the two above. The reason we chose exactly 25 stocks is discussed in section 2.5

Why is our look-back period exactly 90 days? Through empirical testing, we determined that 90 days optimally balances capturing persistent characteristics while remaining sensitive to recent trends. If the look-back period is too short, we risk selecting stocks with spurious autocorrelation patterns driven by random noise rather than genuine underlying behavior. Conversely, if the look-back period is too long, we capture historical characteristics but fail to detect recent regime changes in stock behavior. The 90-day window filters out noise while remaining responsive to evolving market dynamics.

3.2 Genetic lgorithms

3.2.1 Parameter Space and Initialization

Each strategy candidate is defined by a set of parameters that govern entry and exit thresholds. For momentum strategies, the parameter vector is $_{\text{mom}} = (k_{\text{upper}}, k_{\text{lower}})$, where:

- k_{upper}: the volatility-adjusted return threshold for initiating a long position
- k_{lower}: the threshold for exiting a position

For mean reversion strategies, this simplifies to $_{\text{rev}} = (k_{\text{lower}})$, representing the threshold below which a position is entered.

At each rebalance, we sample 200 candidate parameter sets per asset. For momentum strategies, we draw:

$$k_{\text{upper}} \sim \mathcal{N}(\mu_{\text{upper}} = 0.8,\ \sigma_{\text{upper}} = 0.15),$$
$$k_{\text{lower}} \sim \mathcal{N}(\mu_{\text{lower}} = 0.0,\ \sigma_{\text{lower}} = 0.15),$$

with the constraint that $k_{\text{upper}} > k_{\text{lower}}$; if this condition is violated, k_{upper} is shifted upward by 0.5 to enforce separation. **This is valid check because we exit when the momentum is exhausted (i.e, the returns are low)**. For mean reversion strategies,

$$k_{\text{lower}} \sim \mathcal{N}(\mu_{\text{lower}} = \quad 0.8,\ \sigma_{\text{lower}} = 0.2).$$

These sampling distributions are initialized with empirically motivated means (from our quarter project, which utilized a similar base strategy) and relatively tight standard deviations to balance exploration and robustness. At each rebalance, multiple candidate parameter sets are drawn per asset, and the empirical moments of the top-performing parameters are used to adapt the distribution means and variances across rebalance cycles.

After each rebalance, the sampling distributions are updated to reflect the empirical performance of recently selected parameters. Let $\hat{k}_j$ denote the best-performing parameter value (or vector) associated with asset j at the current rebalance. The distribution moments are then updated according to

$$\mu^{\ t+1)} = \frac{1}{N}\sum_{j=1}^{N}\hat{k}_j, \qquad \sigma^{\ t+1)} = \sqrt{\frac{1}{N \quad 1}\sum_{j=1}^{N}\ \hat{k}_j \quad \mu^{\ t+1)}\Big)^2},$$

where N is the number of assets in the corresponding strategy sleeve and t indexes the rebalance cycle. This update mechanism allows the parameter distributions to evolve over time, progressively concentrating probability mass around regions of the parameter space that demonstrate stable performance, while preserving stochastic variation necessary for continued exploration. In this sense, the procedure resembles an evolutionary search process in which successful parameter traits are propagated forward across generations. **Furthermore, through maximum likelihood estimation (MLE), one can find that the parameters (μ, σ) of the gaussian most likely to generate $\hat{k}$, the set of successful parameters, is found through above. That was our motivation for utilizing the above update rule.**

To note: a big concern with genetic algorithms is the issue of stochastic variability due to random sampling. Unlike other algorithmic trading strategies, genetic algorithms are non-deterministic – they vary over different runs. Because of this, we choose to be intentionally conservative with our parameter sets. We want to make sure that results are somewhat reproducible rather than "lucky".

We chose to use random sampling rather than grid search or other systematic parameter–generation methods because it allows us to easily scale the number of samples, giving us direct control over how precisely we estimate $\hat{k}$. Unlike grid search, which is constrained to evaluating parameters only at fixed grid points and can miss optimal values between them, random sampling can explore a highly irregular and unpredictable objective function more effectively. In addition, random sampling naturally concentrates more samples around the mean of the parameter distribution, which aligns with regions we already know perform well, while still allowing occasional exploration of the tails. This represents a deliberate tradeoff we were willing to make, as it both improves practical search efficiency and reduces the number of hyperparameters by eliminating the need to design and tune a complex grid scheme.

3.2.2 Signal Construction

Within the genetic optimization procedure, each sampled parameter set is evaluated by running a custom backtesting function, evaluating the given parameters on the previous 160 trading days. For a given asset and candidate parameter vector, this simulation applies a rule-based trading strategy to historical price data in order to determine, at each point in time, whether the strategy would have been invested or out of the market. The resulting simulated position sequence is then combined with realized returns to produce a hypothetical performance profile, which serves as the basis for computing fitness metrics and ranking candidate parameters during optimization.

To construct the simulated trading signal, we compute a volatility-adjusted return using a rolling z-score framework. Let $r_{i,t}$ denote the daily return of asset i on day t, and let $\sigma_{i,t}$ be the 30-day rolling standard deviation of returns, computed using information available up to day $t \quad 1$ to avoid look-ahead bias. The resulting standardized return is given by

$$z_{i,t} = \frac{r_{i,t}}{\sigma_{i,t}}.$$

For momentum strategies, the simulator maintains a binary position indicator $d_{i,t} \in \{0, 1\}$, representing whether the strategy is invested or out of the market. The idea is that because the chosen stock has high autocorrelation, if today's returns are high, chances are that tomorrow's will be as well, indicating momentum. The thresholds indicate when to enter and exit. The position evolves according to the rule

$$d_{i,t} = \begin{cases} 1 & \text{if } z_{i,t} > k_{\text{upper}}, \\ 0 & \text{if } z_{i,t} < k_{\text{lower}}, \\ d_{i,t \quad 1} & \text{otherwise.} \end{cases}$$

For mean-reversion strategies, it is the opposite. If the stock has low returns today, chances are that tomorrow, the return will be positive. We only have a single threshold, which represents both the entrance and exit threshold. We do not set a seperate exit threshold to be anything higher (unlike for momentum), because we do not plan to hold these stocks for very long. We seek to only capture the profit when the return flips signs tomorrow, and then liquidate.

$$d_{i,t} = \begin{cases} 1 & \text{if } z_{i,t} < k_{\text{lower}}, \\ 0 & \text{otherwise.} \end{cases}$$

In this case, a position is entered when returns are sufficiently negative relative to recent volatility, indicating an oversold condition, and exited once the signal reverts toward neutral. These simulated position paths are subsequently used to compute historical strategy returns and evaluate parameter fitness within the optimization framework.

While this simulation framework abstracts away from the full live-trading execution logic—which incorporates continuous position sizing, portfolio-level risk controls, and regime filters—it provides a computationally efficient and consistent baseline for assessing the relative strengths and weaknesses of candidate parameter sets during optimization. In other words, this is not the same logic used when trading.

Why were these two strategies chosen? While they are more basic in comparison to strategies implementing more complex metrics like RSI and/or Bollinger Bands, this lower parameter count is extremely helpful with the GA context. These other strategies generally have many more parameters that need tuning, such RSI entry, or Bollinger band lookback period, which, due to the curse of dimentionality, would be very computationally difficult to estimate. Additonally, this increases the risk of overfitting. We tried these more complicated strategies breifly, but we found that results differed heavily across runs due to the fact that sampling was not able to converge to similar parameters each run.

3.2.3 Fitness Function

The quality of a parameter set is evaluated using a composite fitness function that balances return, risk, turnover, and stability. Given a sequence of strategy returns $\{R_t^s\}_{t=1}^T$ generated by parameter set , where $R_t^s = d_{t \quad 1} \cdot r_t$, we compute:

- **Sortino Ratio.** The Sortino ratio evaluates risk-adjusted performance by focusing explicitly on *downside risk*, rather than overall volatility. In its basic form,

$$S_{\text{Sortino}} \approx \frac{[R_t]}{\sigma_{\text{down}}}$$

where σ_{down} measures variability of negative returns only. We employ the Sortino ratio rather than the Sharpe ratio in the fitness function to explicitly penalize harmful volatility. Whereas the Sharpe ratio treats positive and negative return variability symmetrically, the Sortino ratio isolates downside deviations and therefore focuses attention on losses rather than benign upside fluctuations. In earlier experiments using Sharpe-based optimization, we observed strong average returns and acceptable Sharpe ratios but consistently high drawdowns, indicating excessive exposure to downside risk. By emphasizing downside volatility through the Sortino ratio, the optimization process favors parameter sets that deliver more controlled risk profiles, even at the expense of marginally lower peak returns.

- **Calmar Ratio.** The Calmar ratio measures return relative to the largest peak-to-trough drawdown experienced over the evaluation period:

  $$C_{\text{Calmar}} \approx \frac{\text{Annualized Return}}{\text{Maximum Drawdown}}.$$

  By explicitly penalizing large drawdowns, the Calmar ratio complements downside-volatility measures such as the Sortino ratio and encourages parameter selections that achieve growth while maintaining control over severe equity declines.

- **Turnover Penalty.** Turnover quantifies how frequently a strategy changes its investment state over time. For a simulated position sequence $\{d_t\}_{t=1}^{T}$, where $d_t \in \{0, 1\}$, we approximate turnover as

  $$\text{TO} \approx \frac{1}{T} \sum_{t=2}^{T} |d_t \quad d_{t \ 1}|.$$

  This measure captures the average frequency with which the strategy enters or exits positions.

  High turnover is undesirable as it amplifies transaction costs, slippage, and execution risk, particularly in a daily-trading framework. This consideration is especially important in our setting, where momentum and mean-reversion strategies are optimized jointly under a genetic algorithm and are therefore naturally exposed to regime changes in market behavior. Without an explicit penalty, the optimization process may favor parameter sets that react excessively to short-term fluctuations, resulting in unstable trading behavior and degraded net performance. By penalizing turnover, we encourage the selection of parameter configurations that trade more selectively and maintain coherent positions across regimes, thereby reducing unnecessary trading activity and improving robustness in realistic execution environments.

- **Stability Bonus** Stability captures how consistent a strategy's performance is across time. We compute using rolling performance metrics, like

  $$S_{\text{roll}} = \frac{\mu_{\text{window}}}{\sigma_{\text{window}}},$$

  computed over fixed-length windows. Low variance in these suggests stronger generalization.

The composite fitness function then comes out to be:

$$F(\ ) = 0.45 \cdot S_{\text{Sortino}} + 0.25 \cdot C_{\text{Calmar}} \quad 0.20 \cdot \text{TO}_{\text{penalty}} + 0.10 \cdot B_{\text{stability}}$$

These weights reflect our priorities: Sortino ratio (45%) emphasizes risk-adjusted returns while penalizing downside volatility, Calmar ratio (25%) rewards drawdown control, turnover penalty (20%) discourages excessive trading, and stability bonus (10%) promotes robustness across market regimes.

While the specific coefficient values were refined through empirical experimentation, the overall weighting structure is also motivated by first-principles considerations. The dominant weight assigned to the Sortino ratio reflects our primary objective of maximizing risk-adjusted performance while explicitly penalizing harmful downside volatility. Drawdown control is addressed through the Calmar ratio, which discourages parameter configurations that generate strong returns at the cost of severe peak-to-trough losses. The turnover penalty further constrains the optimization by limiting excessive trading activity, which is particularly important in a daily trading framework and under a genetic algorithm that may otherwise overreact to short-term fluctuations. Finally, the stability bonus promotes consistent performance across time by favoring parameter sets with low variability in rolling performance metrics, helping to mitigate the inherent stochasticity of genetic optimization. Together, these components encode our preference for robust, explainable, and risk-aware strategies over fragile configurations driven by isolated periods of high returns.

3.3 Overfitting Prevention

As discussed earlier, a critical challenge in genetic algorithm optimization is overfitting risk. We wanted to ensure that parameter sets did not only excel on historical data but fail out-of-sample. We address this through two main mechanisms: walk-forward validation and ensemble learning.

3.3.1 Walk Forward Validation

For each candidate parameter set , we split the 160-day optimization window into a 60% training period ($t = 1, \ldots, 96$) and a 40% test period ($t = 97, \ldots, 160$). We compute the fitness function independently on both periods using the same parameters:

$$F_{\text{train}}(\ \ ) = F(\ \ \mid \{r_t\}_{t=1}^{96})$$

$$F_{\text{test}}(\ \ ) = F(\ \ \mid \{r_t\}_{t=97}^{160})$$

If $F_{\text{test}}(\ \ ) < \ \ 0.5$, the parameter set is rejected, as underperformance on unseen data mean overfitting. For surviving candidates, we compute composite score:

$$F_{\text{combined}}(\ \ ) = 0.4 \cdot F_{\text{train}}(\ \ ) + 0.6 \cdot F_{\text{test}}(\ \ )$$

Our higher weight on test performance (60%) means that we weigh generalization higher. In a sense, this process mimics cross validation in machine learning – but adapted to time series data like stock returns.

We chose 160 days as our evaluation period for our strategies because we wanted our testing set to have a time period similar to the time period used to choose stocks (100 days). Emperical results also showed that values much higher and much lower let to worse results, indicating that this is a length that captures recent trends, but also leads to generalizable parameters.

3.3.2 Ensemble Learning

Having identified high-performing parameter configurations through simulation and validation, we now turn to how these parameters are employed during live trading. Rather than committing to a single optimal parameter set per asset, we retain the top three candidates ranked by the combined fitness score F_{combined}. Let $\{\ _1,\ _2,\ _3\}$ denote these parameter sets with corresponding fitness scores $\{F_1, F_2, F_3\}$. This ensemble-based approach reduces sensitivity to estimation noise and mitigates the risk associated with relying on a single parameterization.

During live execution, each parameter set in the ensemble independently generates a continuous trading signal for the asset. These signals are then aggregated through a weighted average to form a single ensemble signal:

$$\text{Signal}_{\text{ensemble}} = \frac{\sum_{j=1}^{3} w_j \cdot \text{Signal}(\ _j)}{\sum_{j=1}^{3} w_j},$$

where the weights are defined as

$$w_j = \max(F_j, 0.1).$$

This weighting scheme ensures that parameter sets with stronger historical performance exert greater influence on the final signal, while maintaining a minimum contribution from all ensemble members to preserve diversification. The resulting ensemble signal is clipped to the interval $[\ \ 1, 1]$, allowing it to be interpreted as a bounded measure of directional conviction that directly informs position sizing in the live trading system.

3.4 Position Sizing

Between rebalance periods, we compute a continuous signal strength metric that allows us to modulate position sizes dynamically. This allows the system to scale exposure in proportion to a signal's quality, rather than rote allocation.

3.4.1 Multi timeframe Momentum

For asset i, we compute momentum over three horizons: 5-day, 10-day, and 20-day:

$$m_{i,5} = \frac{P_{i,t}}{P_{i,t\ \ 5}}\quad 1, \quad m_{i,10} = \frac{P_{i,t}}{P_{i,t\ \ 10}}\quad 1, \quad m_{i,20} = \frac{P_{i,t}}{P_{i,t\ \ 20}}\quad 1$$

where $P_{i,t}$ is the price of asset i on day t. These are volatility-adjusted:

$$z_{i,5} = \frac{m_{i,5}}{\sigma_{i,20}}, \quad z_{i,10} = \frac{m_{i,10}}{\sigma_{i,20}}, \quad z_{i,20} = \frac{m_{i,20}}{\sigma_{i,20}}$$

where $\sigma_{i,20}$ is the 20-day rolling standard deviation of returns. The composite signal is:

$$z_{\text{combined}} = 0.5 \cdot z_{i,5} + 0.3 \cdot z_{i,10} + 0.2 \cdot z_{i,20}$$

weighting recent momentum more heavily. To bound this in $[\ \ 1, 1]$, we apply a hyperbolic tangent transformation:

$$\text{Signal}_{\text{continuous}} = \tanh \ \ \frac{z_{\text{combined}}}{2}\Big)$$

For mean reversion strategies, we invert the signal: $\text{Signal}_{\text{MR}} = \ \ \text{Signal}_{\text{continuous}}$, as negative momentum (price dips) represents entry opportunities.

Using multiple momentum horizons allows the signal to capture trend information at different temporal scales. Shorter horizons react quickly to recent price movements, while longer horizons provide stability and reduce sensitivity to short-lived fluctuations. By combining these signals in a weighted manner, the resulting momentum measure balances responsiveness with persistence, reducing the likelihood of reacting to noise. Volatility normalization further ensures comparability across assets and market regimes, while the bounded transformation produces a smooth, interpretable signal that can be directly integrated into position sizing and risk management during live trading.

3.4.2 Ensemble Signal Integration

While the genetic algorithm evaluates parameters using simplified, threshold-based trading rules, the live trading system adopts a different approach to signal interpretation. Rather than applying hard entry and exit cutoffs, we transform the learned parameter values into a smooth signal-scaling mechanism that produces a continuous measure of conviction. This design allows the strategy to express varying degrees of momentum or mean-reversion strength, instead of relying on binary yes/no decisions.

For momentum strategies, the continuous signal derived from multi-timeframe momentum is modulated using the ensemble of GA-optimized parameter sets. Let $\ _j = (k_{\text{upper},j}, k_{\text{lower},j})$ denote the j-th parameter configuration in the ensemble. When the base signal is positive, its magnitude is amplified according to how far it exceeds the lower reference level, normalized by the separation between the learned thresholds:

$$\text{Signal}_j = \begin{cases} \text{Signal}_{\text{continuous}} \cdot \left(1 + \frac{\max\left(0, \text{Signal}_{\text{continuous}} \quad k_{\text{lower } j}\right)}{k_{\text{upper } j} \quad k_{\text{lower } j}}\right), & \text{if Signal}_{\text{continuous}} > 0, \\ 0.5 \cdot \text{Signal}_{\text{continuous}}, & \text{otherwise.} \end{cases}$$

This formulation reinterprets the GA-learned thresholds as sensitivity reference points that govern how aggressively the strategy responds to favorable conditions, rather than as fixed decision boundaries.

The final trading signal is obtained by aggregating the scaled signals from each ensemble member through a weighted average,

$$\text{Signal}_{\text{final}} = \frac{\sum_{j=1}^{3} w_j \cdot \text{Signal}_j}{\sum_{j=1}^{3} w_j},$$

and subsequently clipped to the interval $[\quad 1, 1]$ to maintain interpretability and compatibility with downstream position sizing. In preliminary implementations using hard threshold-based rules, distinct market conditions were often treated equivalently, leading to unstable behavior and elevated risk. The smooth signal-scaling approach consistently produced more differentiated responses and, critically, more stable performance in live trading.

3.4.3 Final Position Sizing

The final position size for asset i is determined by combining the ensemble trading signal with risk-based portfolio weights and multiple layers of exposure control. Specifically, the target portfolio weight is given by

$$\text{Weight}_i = \text{Signal}_{\text{final},i} \cdot \text{Weight}_{\text{HRP},i} \cdot \quad_{\text{regime}} \cdot \quad_{\text{drawdown}} \cdot \quad_{\text{leverage}},$$

where $\text{Weight}_{\text{HRP},i}$ denotes the hierarchical risk parity allocation for asset i, $\quad_{\text{regime}} \in [0,1]$ is a market regime multiplier, $\quad_{\text{drawdown}} \in [0,1]$ is a portfolio-level drawdown control factor, and $\quad_{\text{leverage}}$ is a dynamically adjusted leverage term derived from volatility targeting.

Positions are initiated when the final ensemble signal exceeds the conviction threshold ($\text{Signal}_{\text{final}} > 0.08$), which is chosen to balance opportunity capture against noise suppression. The dynamic leverage component scales exposure inversely with realized market volatility and is further constrained by predefined bounds, ensuring that overall portfolio risk remains controlled across changing market conditions.

3.5 Hierarchical Risk Parity (HRP)

Portfolio construction via mean variance optimization can be unstable when the number of assets approaches the number of observations. The sample covariance matrix $\hat{\ }$ becomes ill conditioned, leading to unstable weights. Hierarchical Risk Parity (HRP) is a way to address this by imposing structure through hierarchical clustering.

3.5.1 Distance Metric and Clustering

Given a set of N assets with return matrix $\mathbf{R} \in \mathbb{R}^{T \times N}$ (where $T = 63$ days, approximately 3 months), we compute the correlation matrix:

$$\mathbf{C}_{ij} = \text{Corr}(R_{\cdot,i}, R_{\cdot,j})$$

The distance between assets i and j is defined as:

$$d_{ij} = \sqrt{\frac{1 \quad C_{ij}}{2}}$$

This metric satisfies $d_{ij} \in [0, 1]$, with $d_{ij} = 0$ for perfectly correlated assets and $d_{ij} = 1$ for perfectly anti-correlated assets. We then convert the distance matrix to a condensed form and perform hierarchical clustering using single linkage (effectively, finding the nearest neighbor from both and their distance):

$$\text{linkage}(\mathbf{D}) = \text{single-link}(\{d_{ij}\})$$

This produces a graph capturing the hierarchical relationships among assets.

3.5.2 Recursive Bisection

HRP allocates weights via recursive bisection of the graph. At each node, we split the cluster into two sub-clusters and allocate weight inversely proportional to cluster variance.

Let $\mathcal{C}_{\text{left}}$ and $\mathcal{C}_{\text{right}}$ denote the left and right child clusters at a given node. For each cluster $\mathcal{C}$, we estimate cluster risk using the variance of an inverse-variance weighted portfolio:

$$\sigma_{\mathcal{C}}^2 = \mathbf{w}_{\mathcal{C}}^\top \quad {}_{\mathcal{C}}\mathbf{w}_{\mathcal{C}},$$

where $\quad {}_{\mathcal{C}}$ is the sample covariance matrix restricted to assets in $\mathcal{C}$, and

$$\mathbf{w}_{\mathcal{C}} = \frac{\text{diag}(\quad {}_{\mathcal{C}})^{\;\;1}}{\mathbf{1}^\top \text{diag}(\quad {}_{\mathcal{C}})^{\;\;1}}$$

assigns weights inversely proportional to individual asset variances. Correlations are incorporated implicitly through the hierarchical clustering structure rather than the intra-cluster weighting itself.

The weight allocated to the left and right sub-clusters is:

$$w_{\text{left}} = \frac{\sigma_{\text{right}}^{\;\;1}}{\sigma_{\text{left}}^{\;\;1} + \sigma_{\text{right}}^{\;\;1}}, \quad w_{\text{right}} = \frac{\sigma_{\text{left}}^{\;\;1}}{\sigma_{\text{left}}^{\;\;1} + \sigma_{\text{right}}^{\;\;1}}$$

This process recursively bisects the tree until reaching individual assets, at which point weights are assigned.

3.5.3 Position Concentration Limits

To prevent excessive concentration, we cap individual asset weights at 4%:

$$w_i^{\text{final}} = \max(0.005, \min(0.04, w_i^{\text{HRP}}))$$

and renormalize to ensure $\sum_{i=1}^{N} w_i^{\text{final}} = 1$.

Hierarchical Risk Parity offers a more robust and economically intuitive approach to portfolio construction than traditional mean–variance optimization. By grouping assets according to their empirical correlation structure, HRP allocates capital across distinct sources of market risk rather than allowing highly correlated positions to dominate exposure. This is especially important in equity portfolios, where apparent diversification can mask strong common drivers such as sector or macroeconomic effects. As a result, HRP produces more balanced and stable allocations that remain resilient across changing market conditions.

3.6 Regime Detection and Risk Management

Market regimes are a huge part of adaptation, and can fundamentally alter strategy performance. In our code, we implement a regime detection system that modulates exposure based on observed market conditions.

We use the S&P 500 (SPY) as a proxy for broad market conditions. A rolling window of $T = 22$ trading days (approximately one month) of daily prices is maintained, from which realized market volatility is computed. Let P_t denote the SPY price on day t. Daily returns are defined as

$$r_t = \frac{P_t \quad P_{t\ 1}}{P_t},$$

and realized volatility is calculated as the annualized standard deviation of these returns:

$$\sigma_{\text{real}} = \sqrt{252} \cdot \sqrt{\frac{1}{T \quad 1} \sum_{t=1}^{T\ 1} (r_t \quad r)^2}.$$

This measure captures the prevailing level of market uncertainty and is used to scale overall portfolio exposure.

To distinguish between trending and non-trending market regimes, we compare short-term and long-term simple moving averages computed from the most recent SPY price observations. The market is classified as being in a positive trend when the short-term average (5 trading days) exceeds the longer-term average (22 trading days), indicating sustained upward momentum in the broader market. This trend filter is used in conjunction with realized volatility to modulate overall portfolio exposure.

3.6.1 Regime Multiplier

The regime multiplier $_{\text{regime}}$ is then assigned based on volatility and trend:

$$_{\text{regime}} = \begin{cases} 0.0 & \text{if } \sigma_{\text{real}} > 0.35 \text{ and } \text{SMA}_5 \leq \text{SMA}_{22} \\ 0.4 & \text{if } \sigma_{\text{real}} > 0.35 \text{ and } \text{SMA}_5 > \text{SMA}_{22} \\ 0.5 & \text{if } \sigma_{\text{real}} \in (0.22, 0.35] \text{ and } \text{SMA}_5 \leq \text{SMA}_{22} \\ 0.7 & \text{if } \sigma_{\text{real}} \in (0.22, 0.35] \text{ and } \text{SMA}_5 > \text{SMA}_{22} \\ 0.65 & \text{if } \sigma_{\text{real}} \in (0.15, 0.22] \text{ and } \text{SMA}_5 \leq \text{SMA}_{22} \\ 0.85 & \text{if } \sigma_{\text{real}} \in (0.15, 0.22] \text{ and } \text{SMA}_5 > \text{SMA}_{22} \\ 1.0 & \text{if } \sigma_{\text{real}} \leq 0.15 \end{cases}$$

This piecewise function reflects the empirical observation that high volatility ($\sigma > 0.35$) typically indicates market stress (crashes, panics), during which systematic strategies often fail. By combining volatility with trend direction, we are able to filter a bit more between "healthy" trending volatility (where momentum can still profit) and "chaotic" mean-reverting volatility (where we would want to liquidate).

3.6.2 dditional Filters

For momentum strategies, we have an additional filter. If the autocorrelation of an asset's returns turns negative ($\rho_i < \quad 0.1$), the position is immediately liquidated. Since we do not employ shorting, this prevents momentum strategies from holding assets whose behavior has shifted to mean-reverting.

For mean reversion strategies, we compute the ratio of short-term (20 day) to long-term (60 day) realized volatility as γ_{vol}. If $\gamma_{\text{vol}} > 1.5$, we classify the asset as experiencing a "volatility expansion" event. This could mean a a structural issue, liquidity crisis, or more. What is important to us is that during such periods, the assumption of reversion to a stable mean is violated. Because of this, we do not enter mean reversion positions during them.

3.7 Trailing Stops

Trailing stops allow profits to run during favorable uptrends, but also cutting losses when positions move negatively.

3.7.1 Triggers

For each open position in asset i, we maintain:

- $P_{i,\text{entry}}$: the price at which the position was entered
- $P_{i,\text{peak}}$: the maximum price observed since entry

At each timestep, we update:

$$P_{i,\text{peak}} = \max(P_{i,\text{peak}}, P_{i,t})$$

The trailing stop is triggered if:

$$\frac{P_{i,t} \quad P_{i,\text{peak}}}{P_{i,\text{peak}}} < \quad 0.15$$

Essentially, if the current price falls more than 15% below the peak price observed during the holding period, the position is liquidated immediately. With this, we try to balance sensitivity (avoiding premature stops from intraday volatility) and protection (limiting tail risk from adverse moves).

3.7.2 Signal-Based Exits

Independent of trailing stops, positions are exited when the ensemble signal falls below a threshold:

- **Momentum:** Exit if $\text{Signal}_{\text{final}} < \quad 0.1$ (strong negative signal indicates trend reversal)
- **Mean Reversion:** Exit if $\text{Signal}_{\text{final}} < \quad 0.05$ (price has rebounded, reversion complete)

These thresholds are asymmetric to the entry thresholds, reflecting the asymmetry in risk management: we are more conservative in exiting losing positions than in entering new ones.

3.8 Margin Usage

The margin usage employed in this strategy is fundamentally different from discretionary or speculative leverage that introduces the risk of margin calls under adverse conditions. Leverage is applied at the portfolio level in a controlled manner and is continuously moderated by volatility targeting, regime detection, and drawdown constraints. Its primary purpose is to ensure that available capital is efficiently deployed during favorable market conditions, rather than remaining underutilized when high-quality signals are present.

By scaling exposure when risk conditions are stable, leverage allows the strategy to increase expected returns without proportionally increasing downside risk, which in turn improves risk-adjusted performance. Since the risk-free rate remains fixed, higher expected returns achieved through disciplined leverage translate directly into improved Sharpe and Sortino ratios. Empirically, the relatively shallow drawdowns observed in-sample justified the use of moderate leverage, enabling higher capital utilization while preserving robust risk control.

3.9 Profit and Loss Equations

Portfolio profit and loss (P&L) is computed as the aggregation of asset-level returns weighted by their dynamic position sizes. Let $r_{i,t}$ denote the daily return of asset i on day t, and let $w_{i,t \quad 1}$ represent the portfolio weight assigned to asset i at the close of day $t \quad 1$. The portfolio return on day t is then given by

$$R_t = \sum_{i=1}^{N_t} w_{i,t \quad 1} \cdot r_{i,t},$$

where N_t denotes the number of active positions at time t.

Weights are updated dynamically based on ensemble signal strength, hierarchical risk parity allocation, regime and drawdown multipliers, and volatility-targeted leverage. As a result, the portfolio may exhibit time-varying gross exposure, including periods where total exposure exceeds 100% due to controlled leverage usage.

Cumulative portfolio performance is computed via geometric compounding:

$$V_t = V_{t \quad 1} \cdot (1 + R_t),$$

where V_t denotes the total portfolio value at time t. This formulation naturally incorporates the effects of position resizing, leverage adjustments, and risk controls, providing a unified representation of strategy performance over time.

3.10 Implementation summary

Step by step, the full system works as follows

Initialization (Day 0):

1. Load QC500 universe, filter to top 400 by market cap
2. Compute autocorrelation, select top 35 momentum and top 35 mean reversion candidates
3. Run genetic algorithm (30 samples per asset, walk-forward validation, retain top-3)
4. Compute HRP weights
5. Initialize positions based on ensemble signals and regime filter

Daily operations (Day 1-34):

1. Update price history for all held assets
2. Compute continuous signal strength (multi-timeframe momentum)
3. Update regime multiplier based on SPY volatility and trend
4. For each position:
   - Check trailing stop (15% from peak)
   - Check correlation veto (momentum only)
   - Check volatility expansion (mean reversion only)
   - Compute ensemble signal
   - Adjust position size: w_i = Signal $\times$ HRP Weight $\times$ $_{\text{regime}}$ $\times$ 5
5. Execute rebalancing trades

Rebalance (Day 35):

1. Re-run asset selection (autocorrelation-based ranking)
2. Re-run genetic algorithms for new asset universe

3. Recompute HRP weights
4. Liquidate positions no longer in momentum/reversion tracks
5. Reset to Day 1

This structure allows us to address two huge vulnerabilities in algo trading: 1) overfitting (solved via Walk forward Validation and Ensembling) and Regime change (solved via the SPY volatility filter and HRP).

4 Results

Because our strategy is probabilistic and not deterministic (evolution relies on sampling from a distribution), we ran our strategy 10 times for each time period and took the average while keeping the parameters the same between all of them. The idea is that the average of the strategies represents the expected results and weight should be allocated equally among them.

4.1 In Sample

The general structure of a strategy being implemented over the whole in-sample period is as follows:

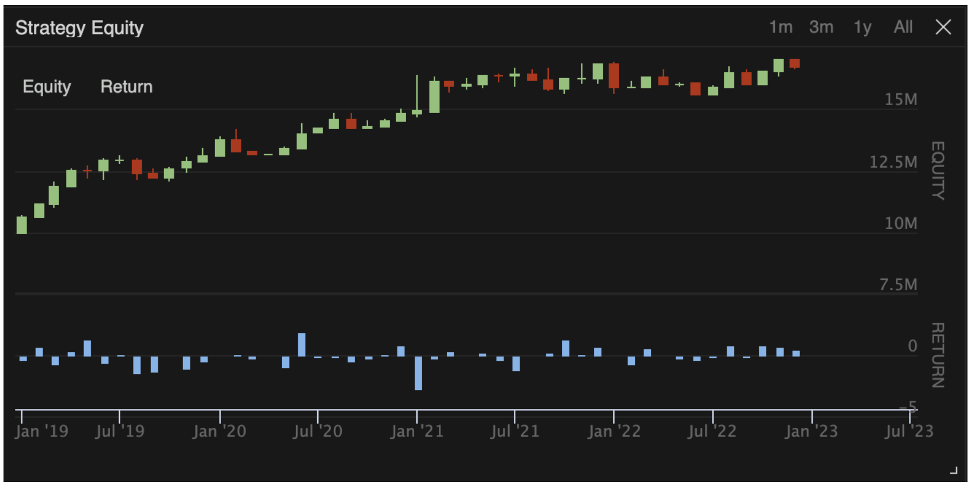


Figure 2: Skeleton of what the average strategy over IS looks like

The overall statistics of the backtest in sample periods were

Metric	verage	Minimum	Maximum	SPY
Sharpe Ratio	0.92	0.60	1.30	0.5
Drawdown (%)	9.30	7.00	13.00	33.6
Returns (%)	64.49	38.33	98.94	65.86
VaR (95%)	-0.84%			

Table 1: Results for In Sample period

4.2 OOS

4.2.1 OOS 1

Metric	verage	Minimum	Maximum	SPY
Sharpe Ratio	-0.44	-0.90	0.40	0.769
Drawdown (%)	11.40	7.00	14.00	10.3
Returns (%)	-0.45	-6.91	11.08	15.03

Table 2: Results for Out of Sample 1 period

4.2.2 OOS 2

Metric	verage	Minimum	Maximum	SPY
Sharpe Ratio	1.17	0.60	1.70	1.67
Drawdown (%)	6.50	5.00	8.00	9.5
Returns (%)	26.32	16.44	37.44	33.87
VaR (95%)	-1.06			

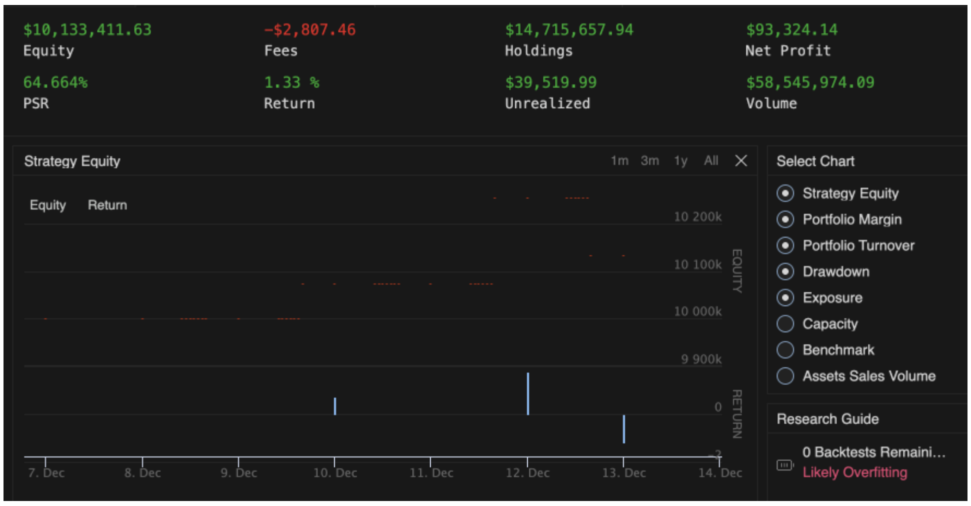
Table 3: Results for Out of Sample 2 period

4.2.3 OOS 3

Metric	verage	Minimum	Maximum	SPY
Sharpe Ratio	0.74	0.10	2.30	1.49
Drawdown (%)	8.80	6.00	12.00	5
Returns (%)	9.75	3.35	25.81	11.59
VaR (95%)	-1.41			

Table 4: Results for Out of Sample 3 period

4.3 Live Trading



$10,133,411.63	-$2,807.46	$14,715,657.94	$93,324.14
Equity	Fees	Holdings	Net Profit
64.664%	1.33 %	$39,519.99	$58,545,974.09
PSR	Return	Unrealized	Volume

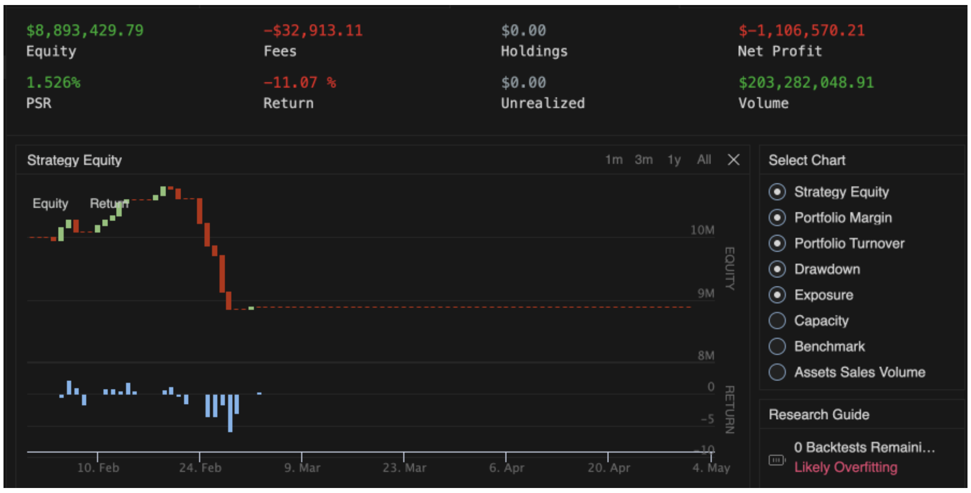
Figure 3: Skeleton of what the average strategy over IS looks like

There are no results to report on for the live trading at this point as that began on Friday and since our strategy runs daily and it is the weekend, there has been no opportunity to engage in trades. So, just to show that our strategy does operate on its own and requires no intervention and is without errors, we ran it over this past week.

We got returns of 1.33% along with a sharpe ratio of 5.289 and a drawdown of 1.2%.

4.4 Stress Test

4.4.1 Stress Test 1



$8,893,429.79 Equity	-$32,913.11 Fees	$0.00 Holdings	$-1,106,570.21 Net Profit
1.526% PSR	-11.07 % Return	$0.00 Unrealized	$203,282,048.91 Volume

Select Chart
- [✓ Strategy Equity]
- [✓ Portfolio Margin]
- [✓ Portfolio Turnover]
- [✓ Drawdown]
- [✓ Exposure]
- Capacity
- Benchmark
- Assets Sales Volume

Research Guide

0 Backtests Remaini...
Likely Overfitting

Figure 4: Skeleton of what the average strategy over ST 1 looks like

Metric	**verage**	**Minimum**	**Maximum**	**SPY**
Drawdown (%)	18.50	17.00	20.00	34
Returns (%)	-14.21	-18.34	-10.26	-13.39

Table 5: Results for Stress Test 1 period

4.4.2 Stress Test 2

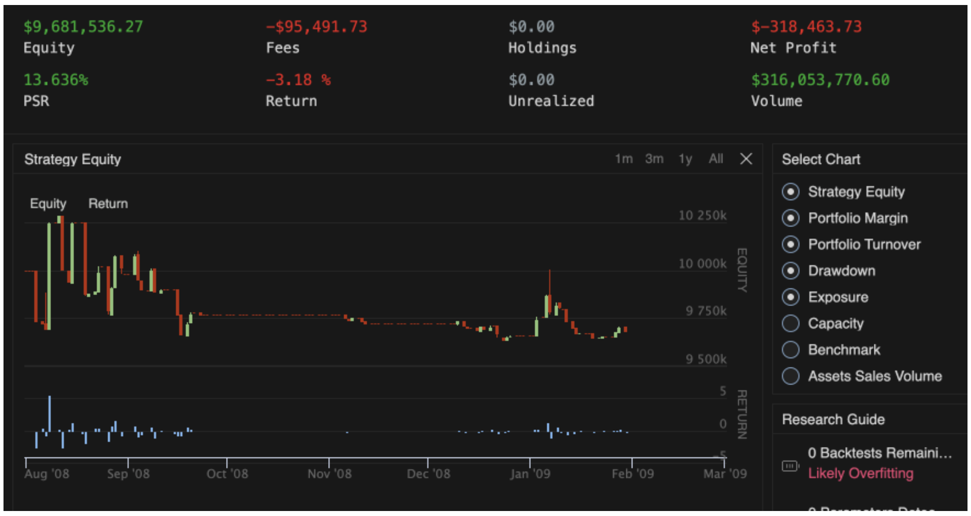
Metric	**verage**	**Minimum**	**Maximum**	**SPY**
Drawdown (%)	10.60	5.00	15.00	41.4
Returns (%)	-6.72	-15.00	0.74	-33.35

Table 6: Results for Stress Test 1 period

4.4.3 Stress Test 3

Metric	**verage**	**Minimum**	**Maximum**	**SPY**
Drawdown (%)	18.40	16.00	22.00	13.60
Returns (%)	-16.83	-21.59	-11.63	1.29

Table 7: Results for Stress Test 1 period


$9,681,536.27 Equity

-$95,491.73 Fees

$0.00 Holdings

$-318,463.73 Net Profit

13.636% PSR

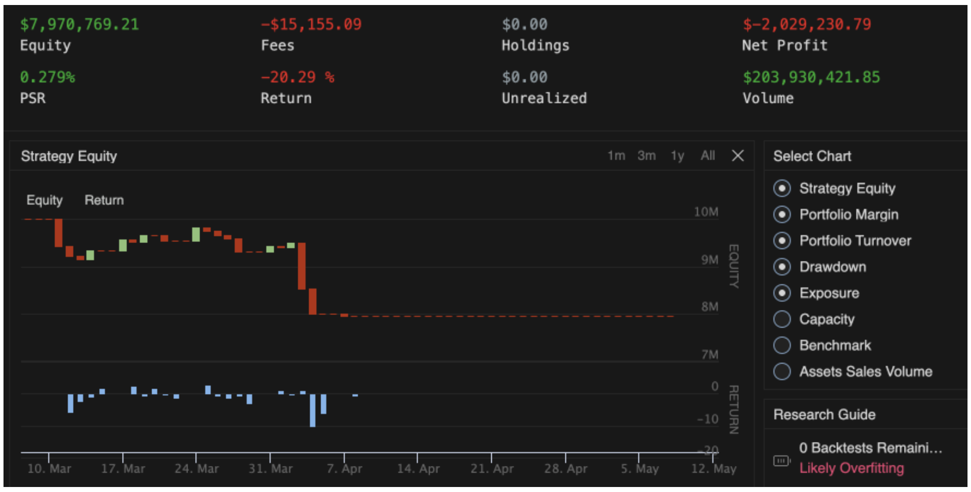
-3.18 % Return

$0.00 Unrealized

$316,053,770.60 Volume

0 Backtests Remaini…
Likely Overfitting

Figure 5: Skeleton of what the average strategy over ST 2 looks like


$7,970,769.21 Equity

-$15,155.09 Fees

$0.00 Holdings

$-2,029,230.79 Net Profit

0.279% PSR

-20.29 % Return

$0.00 Unrealized

$203,930,421.85 Volume

0 Backtests Remaini…
Likely Overfitting

Figure 6: Skeleton of what the average strategy over ST 3 looks like

4.5 Final

Dataset	Sharpe Ratio	Drawdown (%)	Returns (%)	nnualized (%)
IS	0.92	9.30	64.49	13.2
OOS 1	-0.44	11.4	-0.45	-50.8
OOS 2	1.17	6.50	26.32	26.32
OOS 3	0.74	8.80	9.75	23.5
ST 1	–	18.50	-14.21	-45.8
ST 2	–	10.60	-6.72	-13
ST 3	–	18.40	-16.83	-65.5

Table 8: All Results (ST missing sharpe because its clearly very negative)

Our strategy is overall good but sometimes shows mixed signals. On the one hand, it is pretty good during the in-sample period as it is able to produce just as much results as the market but with a sharpe ratio of near 1 and a drawdown of less than 10%, fitting both major requirements. However, when observing the out-of-sample periods, it gets complicated.

We observe that in the out of sample 1 period, the results are significantly worse than the market across every category. This, however, can be explained due to the nature of the algorithm: a momentum/mean-reverting based algorithm thrives in environments that aren't super volatile in a short period. In 2023, the market initially went down which, after observing our losses, the algorithm decided to play it safe and never caught the upswing immediately after. The out of sample 3 results were decent and expected as they are slightly worse than the in-sample averages.

More importantly, however, I believe our strategy works well for more longer periods of time. When observing the graphs above for each of the stress tests, we can see that when it notices a bad trend, it stops investing entirely. Although in the short term testing it never re-invests, over the longer term the stress tests should be mitigated after the algorithm re-invests in the market. Sometimes it might catch the signal too late like in stress test 3, but other times it can prevent a lot of further losses (as the algorithm beats SPY by a lot during 2008).

Also another interesting observed pattern is that because of the fact that our algorithm knows when to stop investing, our drawdowns are consistently low. We did put heavy emphasis on low drawdowns with a focus on the Sortino and Calmar Ratio. In addition, the VaR implies that we don't lose much at pretty high confidence.

5 Future Considerations

Sometimes, we struggle to achieve consistent out-of-sample returns, or when returns are positive, the Sharpe ratio remains below 1. We explored a few potential extensions to improve performance, which could be areas for future exploration:

- **1) lternative sset Classes:** While diversification boosts the Sharpe ratio, many S&P 500 constituents exhibit high correlation, limiting diversification benefits. Expanding into alternative asset classes—such as cryptocurrencies, real estate, or commodities—could enhance returns through reduced correlation with equity markets. However, we note that correlations between these asset classes and traditional equities have been increasing in recent years, potentially diminishing this advantage.
- **2) dvanced Strategy Implementation:** With greater computational resources and performance optimization, we could implement more sophisticated strategies beyond simple z-score-based momentum and mean-reversion. Established techniques such as pairs trading, RSI with Bollinger Bands, and delta-neutral hedging might yield more consistent returns. Allocating portfolio weight to these additional strategies would increase both expected returns and diversification benefits.
- **3) Improved Stock selection process:** Our current stock selection is only based on autocorrelation metrics. While we did explore alternatives through the experimentation process, more sophisticated selection methods like clustering algorithms, PCA, etc. could help us in identifying truly uncorrelated stocks. This would help us systematically improve diversification and risk.
- **4) Improve Regime detection:** Our current regime detection uses SPY volatility as a proxy for market status. Using advanced ML or statistics techniques to process this large data quantity would be extremely useful in making the regime filter more nuanced. Coupled with regime-specific asset allocation (ex: increasing gold exposure during high vol periods) this would allow the portfolio to maintain profitability across bear and bull markets.

References

[1] D. E. Goldberg. *Genetic lgorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[2] K. T. Lwin, R. Qu, and G. Kendall. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *pplied Soft Computing*, 24:757–772, 2014.[web:12][web:17]

[3] K. T. Lwin and R. Qu. A hybrid algorithm for constrained portfolio selection problems. *pplied Intelligence*, 39(2):251–266, 2013.[web:12]

[4] O. Salman, M. Kampouridis, and D. Jarchi. Trading strategies optimization using a genetic algorithm under the directional changes paradigm. In *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, 2023.[web:13][web:18]

[5] M. Majka. Leveraging genetic algorithms in financial modeling and forecasting. LinkedIn Article, 2024.[web:14][web:19]

[6] DataCamp. Genetic algorithm: Complete guide with Python implementation. Online tutorial, 2024. Available at: `https://www.d t c mp.com/tutori l/genetic- lgorithm-python`.[web:15]

[7] Algorithm Afternoon. Maintaining diversity in genetic algorithms to avoid premature convergence. Online article, 2024.

[8] N. Jegadeesh and S. Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48(1):65–91, 1993.[web:23]

[9] J. Lewellen. Momentum and autocorrelation in stock returns. *Review of Financial Studies*, 15(2):533–563, 2002.

[10] N. Barberis, A. Shleifer, and R. Vishny. A model of investor sentiment. *Journal of Financial Economics*, 49(3):307–343, 1998.

[11] K. Daniel, D. Hirshleifer, and A. Subrahmanyam. Investor psychology and security market under- and overreactions. *Journal of Finance*, 53(6):1839–1885, 1998.

[12] EasyLanguage Mastery. How to choose the best objective function for system trading. Online article, 2022.

[13] G. N. Raj. Adaptive and regime-aware RL for portfolio optimization. arXiv preprint arXiv:2509.14385, 2025.[web:5][web:10]

[14] M. López de Prado. Building diversified portfolios that perform well out of sample. *Journal of Portfolio Management*, 42(4):59–69, 2016.

[15] T. J. Moskowitz, Y. H. Ooi, and L. H. Pedersen. Time series momentum. *Journal of Financial Economics*, 104(2):228–250, 2012.

[16] Y. Lemp rière, C. Deremble, P. Seager, M. Potters, and J.-P. Bouchaud. Two centuries of trend following. *Journal of Investing*, 23(4):1–17, 2014.

[17] K. M. Kaminski and A. W. Lo. When do stop-loss rules stop losses? *Journal of Financial Markets*, 18:234–254, 2014.

[18] K. Daniel and T. J. Moskowitz. Momentum crashes. *Journal of Financial Economics*, 122(2):221–247, 2016.

[19] W. Hallerbach. Decomposing portfolio value-at-risk: A general analysis. *Journal of Risk*, 15(2):1–26, 2012.

[20] E. Qian. Risk parity and diversification. *Journal of Investing*, 20(1):119–127, 2011.

ppendix

.1 In Sample Backtests

https://www.quantconnect.cloud/backtest/5c549384575b797a769ac0f75e7f7788/?theme=darkly

https://www.quantconnect.cloud/backtest/c1a3cb4b95bcf596a30ff4cb3fccc626/?theme=darkly

https://www.quantconnect.cloud/backtest/3b87132a3e7404dedd5669b1afedf560/?theme=darkly

https://www.quantconnect.cloud/backtest/1d7d0fe4c62e349b69bef13c800ab1f1/?theme=darkly

https://www.quantconnect.cloud/backtest/8847eacca7ba91b9e00b9d4d01fd4c40/?theme=darkly

https://www.quantconnect.cloud/backtest/c14a32630f0d14961c0a4a3365d5d708/?theme=darkly

https://www.quantconnect.cloud/backtest/194ba2adf1c7d3248e651223abbafac7/?theme=darkly

https://www.quantconnect.cloud/backtest/8caf6a7690e9783d3bf745c81d383854/?theme=darkly

https://www.quantconnect.cloud/backtest/20f1a744a3e5cf83dc4e4d7ace942560/?theme=darkly

https://www.quantconnect.cloud/backtest/1cad09d13fae795484c8296b1dc59b6d/?theme=darkly

.2 OOS 2

https://www.quantconnect.cloud/backtest/57df4932b93b664bd633c450bf84195a/?theme=darkly

https://www.quantconnect.cloud/backtest/a6472429a96ac2187a529700bef70dba/?theme=darkly

https://www.quantconnect.cloud/backtest/7af7dbb289a079cea3e26439c30997f6/?theme=darkly

https://www.quantconnect.cloud/backtest/b1f81b6528d3ae86821050d63ac861de/?theme=darkly

https://www.quantconnect.cloud/backtest/4f6fac074d199f31f24b6b4ff6d6ba92/?theme=darkly

https://www.quantconnect.cloud/backtest/45f5957411ed645440da52449c1e6645/?theme=darkly

https://www.quantconnect.cloud/backtest/f6b595fd8cadfec9688be68ab1b12427/?theme=darkly

https://www.quantconnect.cloud/backtest/8c27538a4ad1d465ead85d280297c1ee/?theme=darkly

https://www.quantconnect.cloud/backtest/8057b1692d921cc5ac4fcae9842c9373/?theme=darkly

https://www.quantconnect.cloud/backtest/75a8ab39967680d4491fcdddb052f10a/?theme=darkly

.3 OOS 3

https://www.quantconnect.cloud/backtest/d54cd0f42f188c75b21ef280c7a9e32e/?theme=darkly

https://www.quantconnect.cloud/backtest/48c39813b737961223877a02fa5e9786/?theme=darkly

https://www.quantconnect.cloud/backtest/7d209891ec5841a1c674f6d1e5d51219/?theme=darkly

https://www.quantconnect.cloud/backtest/2cd7e2ad90f200ab6f81c4e02f067a5a/?theme=darkly

https://www.quantconnect.cloud/backtest/c2d96cfb9fb93513526ff3d04e9f1f5b/?theme=darkly

https://www.quantconnect.cloud/backtest/e2eed9baf0467a51ecc6c98beca2256e/?theme=darkly

https://www.quantconnect.cloud/backtest/564cfd419994249337b8f239cbdf0560/?theme=darkly

https://www.quantconnect.cloud/backtest/92ebde43b1aaf74c7803c98839770db6/?theme=darkly

https://www.quantconnect.cloud/backtest/1b774405e68a9de1436911ab0c30746f/?theme=darkly https://www.quantconnect.cloud/backtest/b1f4d2c38d21b840029ead363131b838/?theme=darkly

.4 ST 1

https://www.quantconnect.cloud/backtest/5f41840c1d2053c38fd6536ea0c8b357/?theme=darkly

https://www.quantconnect.cloud/backtest/28d3244a432899f052707654e8ac2386/?theme=darkly

https://www.quantconnect.cloud/backtest/735c2a44c9eaeb6d8cd6b84a18b5f534/?theme=darkly

https://www.quantconnect.cloud/backtest/514a18ec7e11a094fc60095c940bcb98/?theme=darkly

https://www.quantconnect.cloud/backtest/d756de029dbb9637b7e18f94f7311b83/?theme=darkly

https://www.quantconnect.cloud/backtest/907f5049b99e56c458d1897c2112b085/?theme=darkly

https://www.quantconnect.cloud/backtest/907f5049b99e56c458d1897c2112b085/?theme=darkly

https://www.quantconnect.cloud/backtest/40333ee1f339b701c79633db98deb251/?theme=darkly

https://www.quantconnect.cloud/backtest/998f1bd0a12375d69164dc034190fcba/?theme=darkly

https://www.quantconnect.cloud/backtest/77567a8c1bf9a465acaf3ca6594a0af4/?theme=darkly

.5 ST 2

https://www.quantconnect.cloud/backtest/98e74cd2e984e5884505c02f8ac06df2/?theme=darkly

https://www.quantconnect.cloud/backtest/3b2e15e3bf4e0097fbaea6f1f2ed2ef2/?theme=darkly

https://www.quantconnect.cloud/backtest/33a6f66979a5125f21a6aacbbf32269f/?theme=darkly

https://www.quantconnect.cloud/backtest/6b7bd4f1a6baef291dc73cf7460bbf2a/?theme=darkly

https://www.quantconnect.cloud/backtest/3564047a569ed81e1316afad90241a54/?theme=darkly

https://www.quantconnect.cloud/backtest/5fded78e84aa1949b00f1d6fd45ecd21/?theme=darkly

https://www.quantconnect.cloud/backtest/b1ad3abce06843eeaafe89cde54fa9e1/?theme=darkly

https://www.quantconnect.cloud/backtest/34102a4dab2d68b4836d3ef4d330c777/?theme=darkly

https://www.quantconnect.cloud/backtest/7ef1b140a6743707e5eefefcc9217d98/?theme=darkly

https://www.quantconnect.cloud/backtest/88f2d8261ad2575b42d088bef6736f4f/?theme=darkly

.6 ST 3

https://www.quantconnect.cloud/backtest/e3d6aaa5d893b57c119cb7436a86e3b5/?theme=darkly

https://www.quantconnect.cloud/backtest/4e48a3c234bdc481f26e0bb3d0db97d6/?theme=darkly

https://www.quantconnect.cloud/backtest/d1d43739e30450c3f8d1fd5037bacece/?theme=darkly

https://www.quantconnect.cloud/backtest/e90b36580848aee7008d123d99e813f1/?theme=darkly

https://www.quantconnect.cloud/backtest/319ea0a52a14eb5243e4938bf662569c/?theme=darkly

https://www.quantconnect.cloud/backtest/909518bd0c9cf661eddbe680e1a39354/?theme=darkly

https://www.quantconnect.cloud/backtest/a9e0a2091da45133dfa6ce9212b57ce0/?theme=darkly

https://www.quantconnect.cloud/backtest/511a6b436dac40c22347ef96d8bbdfe7/?theme=darkly

https://www.quantconnect.cloud/backtest/7ebe9a3ef0430995647b728346525c1f/?theme=darkly https://www.quantconnect.cloud/backtest/882774279cd11bfcba5f075837598291/?theme=darkly

.7 Live Trading

https://www.quantconnect.cloud/backtest/a5f9c5b7ebd4eaaa311f4d71f77d76c2/?theme=darkly

.8 SPY

In-sample

https://www.quantconnect.cloud/backtest/93511984b5048d742e80cdb23c633281/?theme=darkly

OOS 1

https://www.quantconnect.cloud/backtest/8743ceffd38a4518ba7a56bb4697a2ae/?theme=darkly

OOS 2

https://www.quantconnect.cloud/backtest/061574fe8ae994175c8d44defcc4b7d1/?theme=darkly

OOS 3

https://www.quantconnect.cloud/backtest/fc4715dbaf139445ced50a1813359ea0/?theme=darkly

ST 1

https://www.quantconnect.cloud/backtest/3eee59de1bb21c7a4bf1abb66a3fbc0e/?theme=darkly

ST 2

https://www.quantconnect.cloud/backtest/8fc23e20d86c3a2ba9417fcb6fed92d4/?theme=darkly

ST 3

https://www.quantconnect.cloud/backtest/b6af0bc585d6c730e19e0d5be058d570/?theme=darkly