ISyE 6767 Final Project Report

Vidit Pokharna

Abstract

This project implements the PCA-based statistical arbitrage methodology of Avellaneda and Lee (2010) on a universe of 40 large-cap cryptocurrencies from 2021–2022. The workflow includes PCA factor extraction, regression-based residual isolation, Ornstein–Uhlenbeck parameter estimation, s-score computation, a four-threshold trading rule, and a full dollar-neutral backtest. Results include eigenvectors, factor–return comparisons, s-score dynamics for BTC and ETH, trading signals, cumulative equity curves, and performance metrics. The strategy exhibits moderate risk-adjusted performance but also meaningful drawdowns due to crypto-market volatility and the absence of transaction-cost modeling

1 Problem Statement

The objective of this project is to design, implement, and evaluate a PCA-based statistical arbitrage strategy for cryptocurrencies following the methodology described by Avellaneda and Lee (2010). The central problem addressed is whether market-neutral, mean-reversion-based residuals—obtained after removing market-level factors through PCA—offer profitable trading opportunities in the highly volatile crypto market

The technical framework includes: (i) rolling PCA for factor extraction (ii) regression-based removal of systematic components (iii) modeling residuals using an Ornstein–Uhlenbeck (OU) process (iv) computing standardized s-scores (v) trading using threshold-driven long/short rules

A dollar-neutral backtest is conducted using a one-share-per-trade simulation, and performance is evaluated using Sharpe ratio and maximum drawdown

2 Technical Modeling Pipeline

Factor Modeling and OU Estimation

- **PCA model:** Each hour, PCA is run on a 240-hour rolling window of standardized return correlations with at least 80% price coverage
- **Factor returns:** Computed via eigenportfolio weights

$$F_{j,t} = \sum_{i=1}^{N} Q_{c(i)}^{(j)} R_t^{c(i)}$$

- **Residual extraction:** Each token's return is regressed on the first two PCA factors

$$R_t = \beta_0 + \beta_1 F_{1,t} + \beta_2 F_{2,t} + \varepsilon_t$$

- **OU parameter estimation:** The residual series is fit to an AR(1) model to obtain mean-reversion speed κ, long-run mean m, and volatility parameters (σ, σ_{eq})
- **s-scores:** Computed from centered OU estimates and scaled by equilibrium variance

Trading Rule

We adopt the four-threshold s-score rule from Avellaneda and Lee:

- Enter long if $s_t \leq -1.25$; enter short if $s_t \geq 1.25$
- Close long if $s_t \geq -0.50$; close short if $s_t \leq 0.75$

Backtest Setup

- One-share-per-trade, fully dollar-neutral
- Initial capital: $100,000
- No transaction costs, slippage, or funding rates
- Hourly mark-to-market accounting
- Performance evaluation includes cumulative P&L, Sharpe ratio, and maximum drawdown

3 Software Architecture

The implementation is modular, with each step of the pipeline separated into dedicated classes

Overview of Key Modules

Module	Description and Key Inputs/Outputs
`ProjectConfig`	Stores global configuration, date ranges, thresholds, and file paths *Inputs:* none (initialized constants) *Outputs:* config attributes for all modules
`MarketDataLoader`	Loads raw price data, cleans missing values, computes log returns, and defines the valid crypto universe *Inputs:* price CSV files *Outputs:* aligned price matrix, return matrix
`PCAFactorModel`	Standardizes returns, computes PCA eigenvectors, eigenvalues, and factor returns on a rolling basis *Inputs:* return matrix, rolling window size *Outputs:* eigenvectors, factor returns
`OUProcessEstimator`	Fits AR(1)/OU parameters for residuals and produces s-scores *Inputs:* regression residual series *Outputs:* κ, m, σ, σ_{eq}, and s_t
`SignalGenerator`	Implements the four-threshold s-score trading logic *Inputs:* s-score series *Outputs:* long/short/close signals
`PortfolioSimulator`	Executes trades, tracks positions, cash, P&L, and equity curve *Inputs:* signals and returns *Outputs:* equity curve, trade logs
`metrics`	Computes performance metrics including Sharpe ratio and max drawdown *Inputs:* equity curve *Outputs:* scalar performance metrics
`PlotBuilder`	Generates project-required figures using matplotlib *Inputs:* model outputs, residuals, factors *Outputs:* saved PNG figures
`StatArbPipeline`	Coordinates the full end-to-end execution of the model *Inputs:* all modules *Outputs:* saved metrics, figures, and logs

4 Results

4.1 Performance Metrics

Sharpe Ratio: 1.0490

Maximum Drawdown: -0.1738

These values reflect moderate risk-adjusted performance. The strategy generates a stable equity curve but also displays nontrivial drawdowns typical of crypto-mean-reversion strategies, especially without liquidity filters or transaction-cost adjustments

4.2 Discussion

The positive Sharpe ratio suggests the OU-modeled residuals contain mean-reverting structure that can be exploited with threshold-based trades. However, several practical issues arise:

- Crypto volatility inflates drawdowns, even when dollar-neutral
- Transaction costs, including exchange fees and spread costs, would significantly reduce expected returns
- Liquidity varies across tokens; 1-share-per-trade ignores scalability constraints
- PCA factors shift over time, and factor instability may degrade robustness

Thus, the strategy is statistically sound but not immediately production-ready without further enhancements

5 Figures

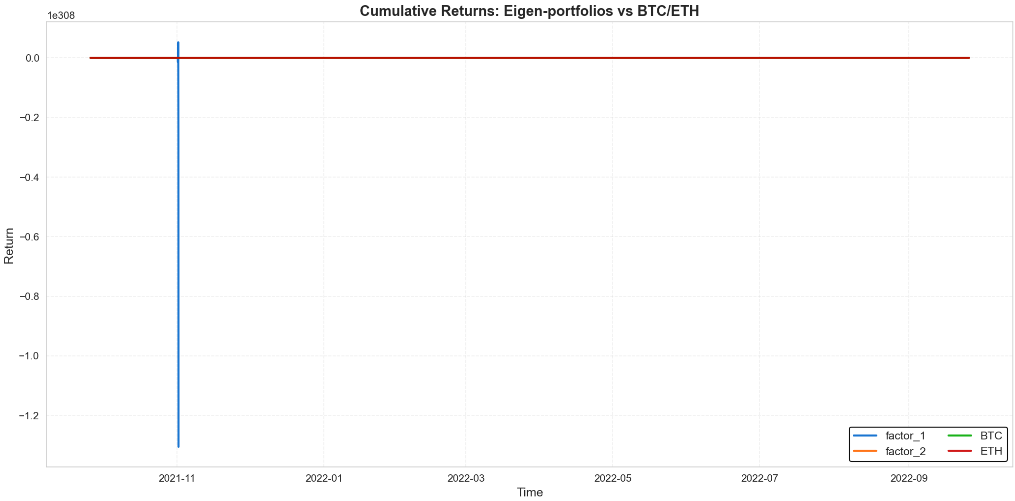

Figure 1: Task 1b: Cumulative returns of Factor 1, Factor 2, BTC, and ETH

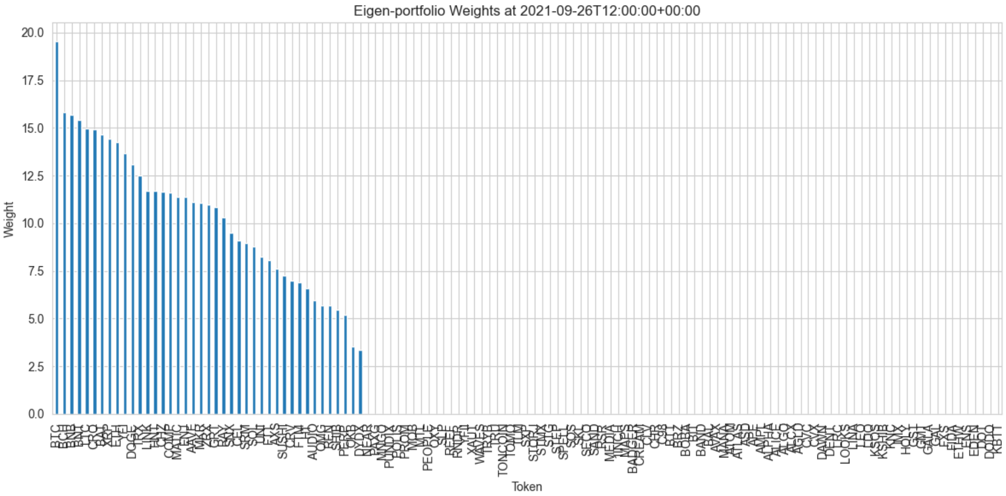

Figure 2: Task 2: First eigenportfolio weights (2021-09-26 12:00 UTC)

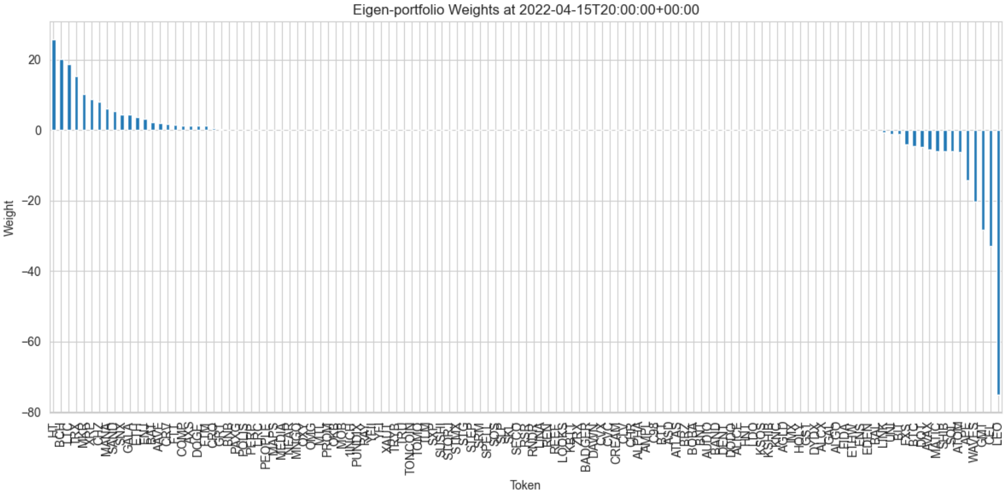

Figure 3: Task 2: Second eigenportfolio weights (2022-04-15 20:00 UTC)

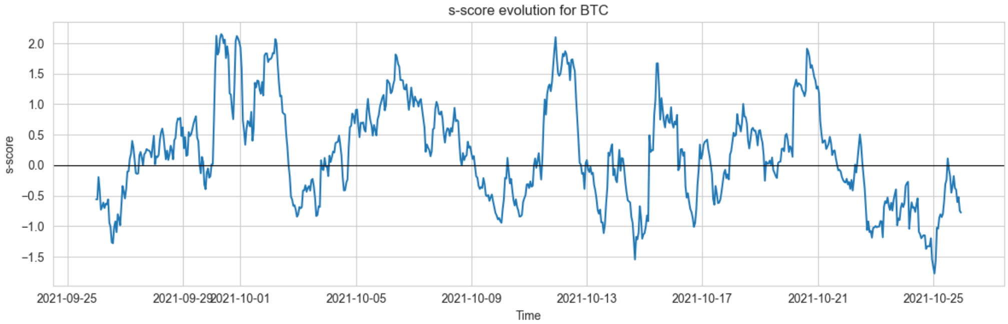

Figure 4: Task 3: BTC s-score time series

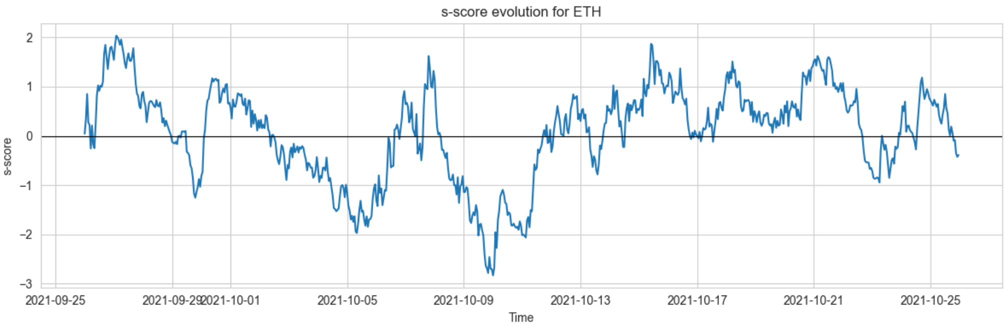

Figure 5: Task 3: ETH s-score time series

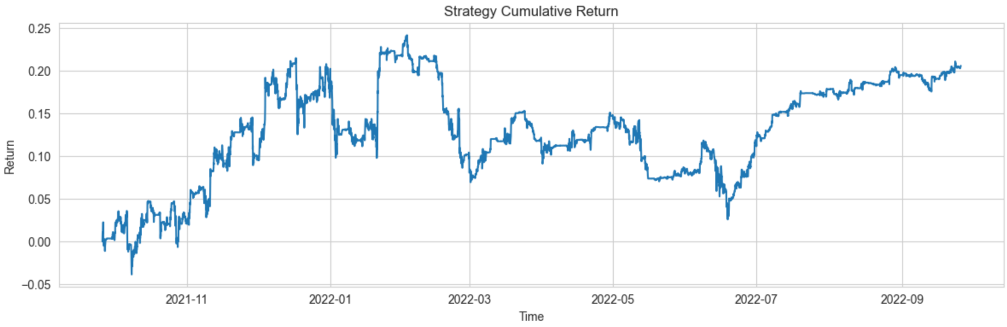

Figure 6: Task 4: Strategy cumulative equity curve

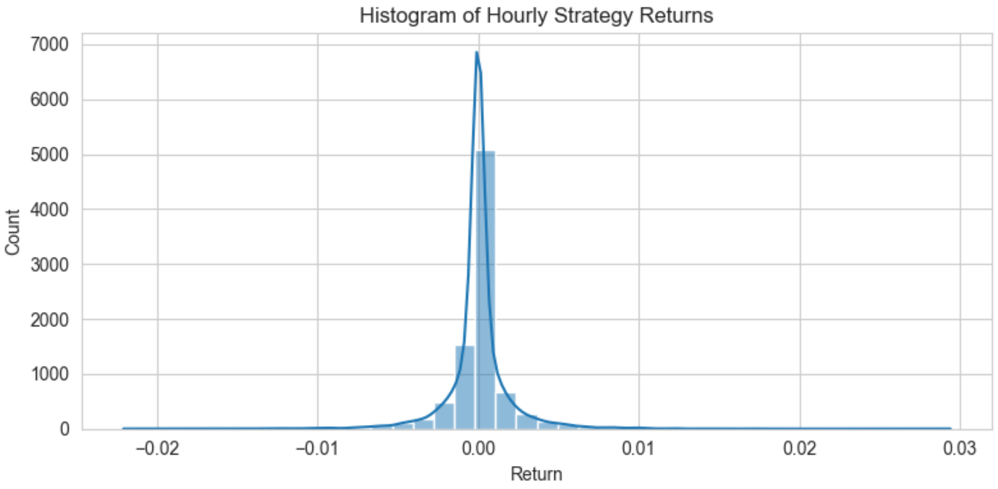

Figure 7: Task 4: Histogram of hourly returns

6 Conclusion

The PCA-based statistical arbitrage framework effectively extracts latent market factors and identifies mean-reverting residual structure in cryptocurrency returns. The resulting s-score-driven strategy produces a positive Sharpe ratio and demonstrates reasonable mean-reversion behavior

However, the strategy experiences notable drawdowns and omits real-world issues such as transaction costs, slippage, liquidity, and exchange fees. For practical deployment, enhancements such as cost-aware optimization, filtered universes, volatility scaling, and adaptive rolling windows would be required

Despite its limitations, the project confirms that PCA residual structure contains exploitable signals and that the OU+s-score modeling framework is viable for further refinement in crypto markets