Ablation Study for Correlation Matrix Clustering for Statistical Arbitrage Portfolios

Yifan Gu*
University of California, Los Angeles
Los Angeles, California, USA
yifangu@math.ucla.edu

Khang Nguyen*
University of California, Los Angeles
Los Angeles, California, USA
khang@math.ucla.edu

Lunji Zhu*
University of California, Los Angeles
Los Angeles, California, USA
lunji@math.ucla.edu

Abstract

Recently, signed graph clustering methods have been proposed as a valuable tool to construct portfolio strategies for statistical arbitrage. In this work, we explore the effect of several modifications of the portfolio construction pipeline in a recent paper, including winsorization, cluster selection, reweighting stock signals, and a novel evaluation metric. Results from our experiments show that winsorization of residual returns moderately improves portfolio performance, while reweighting stocks linearly generates significantly better profits than the baseline. Unfortunately, we do not find significant enough results for the performance of the cluster selection method from our current experiments.

CCS Concepts

• **Computer systems organization** → **Embedded systems**; • **Networks** → Network reliability; • **Applied computing** → Economics.

Keywords

Correlation matrix, Clustering, Portfolio management, Random matrix theory, Statistical arbitrage

ACM Reference Format:
Yifan Gu, Khang Nguyen, and Lunji Zhu. 2025. Ablation Study for Correlation Matrix Clustering for Statistical Arbitrage Portfolios. In *Proceedings of UCLA MATH 285J, Spring 2025.* ACM, New York, NY, USA, 7 pages. https://doi.org/1234567.7654321

1 Introduction

Financial data suffers from the curse of dimensionality and is well-known to be ill-conditioned [3]. To tackle these issues, techniques such as dimensionality reduction and clustering have previously been employed. Recently, [2, 3] proposed constructing portfolio strategies using signed graph clustering methods such as Signed Positive Over Negative Generalized Eigenproblem (SPONGE) [1]. These approaches generate profitable trading strategies with over 10% annualized returns and statistically significant Sharpe ratios above one. In this project, we expand on this idea by conducting further ablation studies to see the effect of using different techniques to construct the signed graph-based portfolios.

Main contributions. This work investigates the effect of several modifications to the statistical portfolios constructed by [2]. We

- Construct the stocks available in each trading window individually, thereby increasing the number of stocks in the trading universe by 4-5 times;
- Introduce winsorization to raw and residual returns to reduce outlier effects;
- Only construct portfolios using the best clusters, measured by a novel correlation density metric;
- Put more weights onto stocks that give the strongest signals in each cluster instead of using a uniform weighting scheme;
- Evaluate our portfolio performance via a new metric of arbitrage success rate.

Relevant results are reported for the effects of our modifications.

Paper outline. After Section 1 that gives an overview of our work, Section 2 explains in detail the methodology we use, including data preprocessing, graph clusterization, winner/loser identification, portfolio construction, trading strategy execution, and evaluation criteria. We also discuss aspects of modification as mentioned in the previous paragraph. Section 3 presents results from working on the financial dataset, that is, those stemming from our modifications. Section 4 concludes the paper with certain future directions of pursuit.

Notations. Given a matrix $A \in \mathbb{R}^{n\times n}$, we denote its eigenvalues and corresponding eigenvectors by λ_i and v_i, respectively, and order the eigenvalues such that $\lambda_1 \geq \cdots \geq \lambda_n$. For a graph $G = (V, E)$ with vertex and edge sets, we define $G^+ = (V, E^+)$ (resp. $G^- = (V, E^-)$) as the unsigned subgraph with only positive (resp. negative) edges. The positive and negative subgraphs have adjacency matrices A^+ and A^-, such that $A^+_{ij} = \max(A_{ij}, 0)$ and $A^-_{ij} = \max(-A_{ij}, 0)$. Note that the adjacency matrix for the original graph satisfies $A = A^+ - A^-$.

2 Methodology

2.1 Data preprocessing

2.1.1 Data preprocessing method. We use stock price data from the Center of Research in Security Prices (CRSP) daily returns database. Our data preprocessing method is based on that of [2], with several notable differences.

* These authors contributed equally to this work.

First, our sample period is between January 2000 and December 2021. [2] considers a slightly longer period ending in December 2022, but we do not have access to the data in 2022.

Second, for each trading period, we build our portfolio by looking into all stocks that have complete close price information within that period, yielding a universe of approximately $2000-3000$ stocks at any given time. [2] only takes into consideration stocks that have complete information over the 20 years period, leading to less than 700 stocks.

Third, to remove unreliable data, we drop all days for which more than 10% of stocks on that day have a zero or NaN close-to-close return but keep all stocks to be used in our portfolio.

The following steps are replicated exactly from [2]. To extract the idiosyncratic dynamic of each stock, we calculate its market residual return over time. This is the component of the stock's return that is not explained by the overall market movement. We simulate access to data over time by constructing the market residual return matrix $R^{res}_{i,t}$ via the formula

$$R^{res}_{i,t} = R_{i,t} - \beta_i R_{mkt,t},$$

in consecutive sliding windows of length T. Here, $R_{i,t}$ is the raw return of stock i at time t, β_i is the beta coefficient of the stock i calculated from data in the current sliding window, and $R_{mkt,t}$ is the market return at time t. The market is taken to be the SPY ETF.

To compute the correlation matrix at time $T+1$, we restrict the residual return matrix to its w most recent entries from time $T-w+1$ to T and calculate the entries of the correlation matrix C as

$$C_{i,j} = \frac{\sum_{t=T-w+1}^{T}(R^{res}_{i,t} - \overline{R}^{res}_i)(R^{res}_{j,t} - \overline{R}^{res}_j)}{(w-1)\sigma_i\sigma_j},$$

where $\overline{R}^{res}_i$ denotes the mean of the residual return of stock i, and σ_i, σ_j are the standard deviations of returns for stocks i, j over w days. The resulting correlation matrix contains the pairwise correlation coefficients between all stocks in the most recent w days.

2.1.2 Winsorization. During data preprocessing, to mitigate the effect of outliers, we also consider winsorization. Given a list of data, we set a threshold ϵ (typically 1% or 5%) so that the smallest and largest ϵ fractions of our data are clipped to the respective values exactly at the ϵ (or $1-\epsilon$) percentile in the cumulative data distribution. In our experiments, we try to winsorize either the raw return or the residual return. Because we are more interested in the collective behavior of stocks at each given time than the trend of individual stocks over its time series, we decide to winsorize across stocks. That is, for an $n \times T$ matrix, we perform winsorization for all columns.

2.2 Graph Clusterization

2.2.1 Reproduction of Clusterization Method from [2]. The correlation matrix that we constructed can be interpreted as a signed graph with edge values in $[-1, 1]$. Given this, we can cluster stocks based on pairwise correlations using signed graph clustering techniques. Specifically, we want to partition the graph nodes into k clusters such that most edges (correlations) within clusters are positive, and most edges between clusters are negative. Equivalently, we want to minimize violations, where negative edges exist within clusters or positive edges exist between clusters.

To this end, we use the SPONGE algorithm, first introduced in [1]. Given a graph G, let $L^{\pm}$ denote the graph Laplacians of $G^{\pm}$, and $D^{\pm}$ be diagonal matrices with the degrees of $G^{\pm}$. For suitably chosen hyperparameters $\tau^+, \tau^- > 0$, then, SPONGE first calculates $(L^- + \tau^+ D^+)^{-1/2} v_i$, where $i = n-k+1, \cdots, n$, for λ_i that are eigenvalues of $(L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}$. This generates k vectors, each of which is n-dimensional, leading to an embedding of each of the n stocks in $\mathbb{R}^k$. Finally, SPONGE clusters the lower-dimensional embedding using k-means++.

In the algorithm, the number of clusters, k, is predetermined with a separate process. In our experiments, we choose a k value ourselves. Alternatively, we could also find k through the percentage of explained variance. We first specify a lookback window of $d \le T$ days and obtain the residual return matrix in Section 2.1 sliced for these days, denoted as $R^{res}_d \in \mathbb{R}^{n \times d}$. Then, we form the matrix $Cov = \frac{1}{d} R^{res}_d (R^{res}_d)^T$, and let k be the smallest integer such that

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \ge P,$$

where λ_i are the eigenvalues of Cov and P is some threshold value.

2.3 Winners and losers identification

2.3.1 Reproduction of winners and losers identification from [2]. For each cluster that we created in Section 2.2 on a certain day t, we identify the winners and losers within that cluster, i.e. the stocks that are over-performing/under-performing when compared against the cluster mean. First, on each day s over the w-day look-back window (the same window as that used for calculating the correlation matrix in Section 2.1), we define the cluster mean return on that day as

$$\bar{R}_{C,s} = \frac{1}{|C|} \sum_{i=1}^{|C|} R_{i,s},$$

where $R_{i,s}$ represent raw returns. We now set a threshold p such that stocks whose returns cumulatively deviate by more than p from the mean are believed to revert to the mean in the long run. Thus, when

$$\sum_{s=t-w}^{t-1} (R_{i,s} - \bar{R}_{C,s}) > p,$$

stock i is identified as a winner, so we expect its value to decrease in the future and should short sell it. Conversely, when

$$\sum_{s=t-w}^{t-1} (R_{i,s} - \bar{R}_{C,s}) < -p,$$

stock i is a loser that we should buy.

2.3.2 Removing weakly correlated clusters. Before dealing with the big data set with more than 2000 stocks for more than 5000 trading days, we perform exploratory analysis on a return matrix with less than 700 stocks that already exists in the original dataset. After performing SPONGE clusterization on the correlation matrix, we find that different clusters have different intra-correlation patterns. Specifically, some clusters are "better" in the sense that the stocks in the clusters are more strongly correlated with each other. Meanwhile, some clusters are "worse" in the sense that they are less intra-correlated .

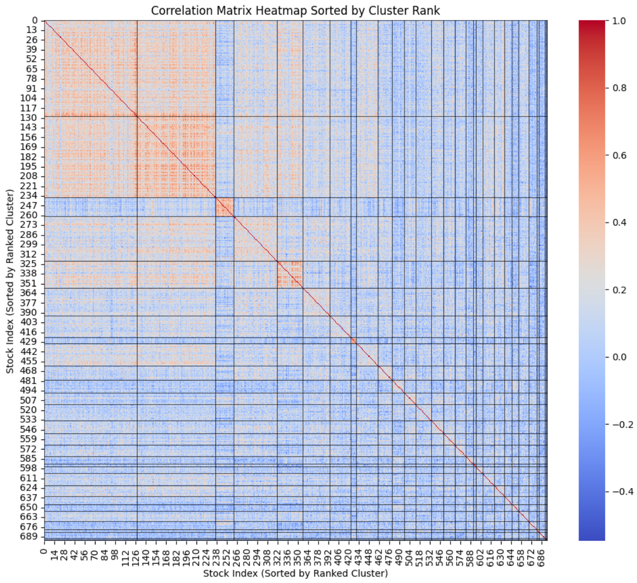

Figure 1: This is the heat map of the correlation matrix with the entries grouped by cluster and the clusters ranked by their performance with respect to the objective function for the SPONGE method. The top left cluster represents the "best cluster" and the bottom right cluster represents the "worst cluster" under the evaluation of the SPONGE objective function. We artificially set the number of clusters to be $k = 30$ (aligned with [2]) and $\tau^+ = \tau^- = 1$ as in the default setting.

Recall the loss function of SPONGE for each cluster, given by [1]:

$$\frac{\mathrm{cut}_{G^+}(C,\overline{C}) + \tau^- \mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C,\overline{C}) + \tau^+ \mathrm{vol}_{G^+}(C)}, \quad \text{which summarizes} \begin{cases} \dfrac{\mathrm{cut}_{G^+}(C,\overline{C})}{\mathrm{vol}_{G^+}(C)}, \\ \dfrac{\mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C,\overline{C})}, \end{cases}$$

where we have for an unsigned graph H with adjacency matrix W with non-negative entries, for any cluster $C \subset V$:

$$\begin{cases} \mathrm{cut}_H(C,\overline{C}) := \sum_{i \in C, j \in \overline{C}} W_{ij}, & \text{total edge weights crossing } C \text{ and } \overline{C}. \\ \mathrm{vol}_H(C,\overline{C}) := \sum_{i \in C} \sum_{j=1}^{n} W_{ij}, & \text{the sum of degrees of nodes in } C. \end{cases}$$

and for the whole graph G with adjacency matrix A:

$$\begin{cases} G^+ := (V, E^+), & \text{the unsigned subgraphs with only the positive edges} \\ G^- := (V, E^-), & \text{the unsigned subgraphs with only the negative edges} \end{cases}$$

each with adjacency matrix $A^+_{ij} = \max\{A_{ij}, 0\}, A^-_{ij} = \max\{-A_{ij}, 0\}$, respectively.

The heatmap sorted by SPONGE loss function is given in Figure 1. Indeed, it captures the "goodness" of the clusters in both the senses that it has high positive intra-correlation (via $\frac{\mathrm{cut}_{G^+}(C,\overline{C})}{\mathrm{vol}_{G^+}(C)}$) and high negative inter-correlation ($\frac{\mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C,\overline{C})}$). However, if we consider in terms of the usefulness in statistical arbitrage, we do not necessarily care about the negative inter-correlation between the clusters. What matters is just the strong intra-correlation, which is assumed to be the foundation of the expected mean-reversion. As the whole underlying idea of statistical arbitrage with clusterization is that we have the stocks evolved in a grouped pattern with stochasticity and the evolution of each stock would always be around it's group's average. If the group is ill-identified, then we should not expect the stocks would return to the mean of the clusters and the algorithm would not be making profits.

Therefore, it may be better if we define a new metric to evaluate the quality of a cluster designated for the purpose of statistical arbitrage.

To do this, we define the *Correlation Density* for each cluster C to be:

$$D_C = \frac{\sum_{i,j \in C, i \neq j} C_{i,j}}{|C|^2 - |C|},$$

where $C_{i,j}$ are the correlations from Section 2.1. It is just the average of inter-node correlation of the clusters, thus a pure and direct measurement of intra-correlation of a cluster.

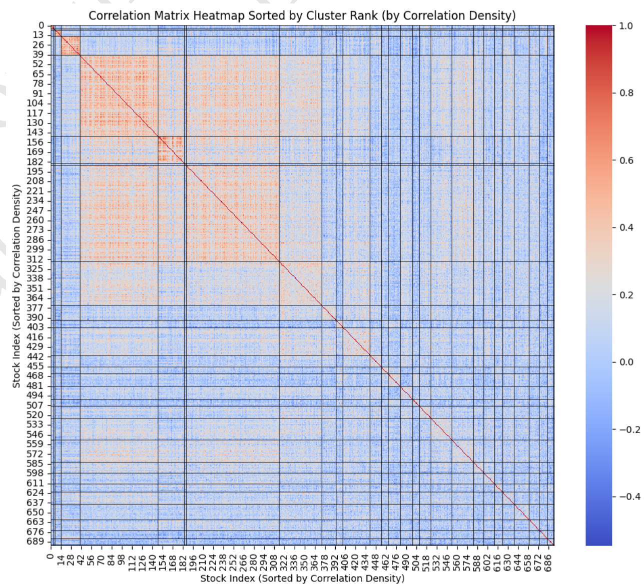

Figure 2: This is the heat map of correlation matrix with the entries grouped by the cluster and the clusters ranked by their performance with respect to Correlation Density function we just defined. The top left cluster represents the "best cluster" and the bottom right cluster represents the "worst cluster" under the evaluation of correlation density function.

We resorted the clusters in the correlation matrix and produced Figure 2. The cluster densities are given by Table 1. We can see that the metric of Correlation Density indeed captures the intra-correlation better than the SPONGE metric and there is indeed significant discprepancy between the intra-correlations of the clusters given by SPONGE.

Clust.	CD	Clust.	CD	Clust.	CD
27	0.5612	12	0.2270	15	0.1701
6	0.5370	9	0.2227	5	0.1602
20	0.4698	7	0.2220	4	0.1398
8	0.4546	10	0.2130	3	0.1394
18	0.3819	23	0.2045	24	0.1338
14	0.3783	28	0.1945	19	0.1295
26	0.3287	0	0.1924	22	0.1218
11	0.3191	17	0.1914	16	0.0000
25	0.2518	2	0.1804	29	0.0000

Table 1: Clusters sorted by Correlation Density, by the matrix above as an example.

With the motivation that "more strongly correlated clusters gives more confidence in future mean reversion", for each clusterization, we evaluate the densities of all clusters and drop 10 clusters with lowest densities from portfolio consideration.

2.4 Portfolio construction

2.4.1 Reproduction of portfolio construction from [2]. After winners and losers are identified, to construct the portfolio, we assign weights to individual stocks. For a given cluster C, we wish to take short positions on the winners and take long positions on the losers. To idealize the problem, [2] use 1 unit of capital to take short and long positions. This ensures market neutrality, and implies that the gross exposure for each cluster is 2 units of money. For example, if we have 5 winners and 4 losers in one cluster, then we short sell \$0.20 on each winner and bet \$0.25 on each loser. This gives rise to a cost-neutral trading strategy. Also, the strategy does not discriminate between large and small clusters, since the amount of money we invest in either of them is the same.

2.4.2 Different weight assignment strategies. Let $x_1 \geq x_2 \geq \ldots \geq x_n \geq -y_m \geq \ldots \geq -y_2 \geq -y_1$ be the deviation from the cluster mean of the winners and losers, respectively. We explore several different ways to assign capital to the stocks inside each cluster based on the signal strength, i.e. how far removed from the cluster mean a given stock's current return is:

- Linear: we short the i-th winner by $\frac{x_i}{\sum_{i=1}^n x_i}$ and long the j-th loser by $\frac{y_j}{\sum_{j=1}^m y_j}$.
- Exponential: we short the i-th winner by $\frac{\exp(x_i)}{\sum_{i=1}^n \exp(x_i)}$ and long the j-th loser by $\frac{\exp(y_j)}{\sum_{j=1}^m \exp(y_j)}$.
- Thresholding: short and long only the top/bottom 20% of winners/losers uniformly.

2.5 Trading strategy execution

2.5.1 Reproduction of trading strategy execution from [2]. Once we finish giving weights to the stocks that we wish to trade, let us recall that all we have done so far pertains to day t. We work with the weights we have at hand starting from day t, until a lookforward window of ℓ days is reached, or a win threshold q for cumulative PnL (Profit and Loss) is exceeded, whichever is earlier. Note that PnL is defined as

$$PnL_t = \frac{1}{2k} \sum_{i=1}^{n} R_{i,t} w_i,$$

where w_i is the weight assigned to every stock – with a positive (resp. negative) sign when we buy (resp. sell). The corresponding weight is zero if we do not trade a certain stock. The reason for introducing this stop-win threshold is the belief that the portfolio reaching such a PnL value is evidence for successful mean-reversion. Thus, if we do not stop trading sooner, our future PnLs might decrease.

Regardless of the exact condition, some days after t, we always stop using the existing weights. We have to reconstruct our portfolio using more recent data. We then redo everything in all previous paragraphs of this section; we continue this process until the last day when we have data available.

2.6 Evaluation criteria

2.6.1 Reproduction of evaluation criteria from [2]. The dominant metric we use for evaluating portfolio performance is PnL, as discussed in Section 2.5. We calculate the cumulative PnL over all eligible trading days in 22 years in the CRSP dataset. From the daily PnLs, we also evaluate the Sharpe ratio

$$SR = \frac{\text{mean}(PnL)}{\text{stdev}(PnL)} \times \sqrt{252},$$

where the scaling comes from 252 trading days in a year. The Sharpe ratio measures the risk-adjusted return, such that a higher value of it indicates a higher return relative to the amount of risk taken.

2.6.2 Arbitrage Success Rate. One of the underlying motivation of our statistical arbitrage method is that: as every stock follows the sub-trend in their clusters, they will evolve "around" the clusters. The ones that go higher now would go lower in the future and the ones that go lower now would go higher in the future. Although having large positive PnL is a quite strong indicator of the validity of this assumption, we are also curious about the portion among all trading periods that we actually have mean-reversion.

To do this, for each trading period, if the cumulative PnL for that trading period exceeds the winning threshold (either it terminates earlier or not), then we consider this trading period to be "successful." If not, then we say it is not "successful."

We compute the success rate of each trading execution to see whether under each trading strategy, we are actually having successful arbitrage behavior.

3 Empirical results

In this section, we report on the results from various experiments related to our modifications and other observations. We fix the parameters as in Table 2. These parameters, without any additional modifications in the pipeline, form the baseline of comparison with our experiments.

3.1 Winsorization

We conduct systematic experiments to test the effect of winsorization on metric performance. In Figure 3, we record the cumulative PnL evolution throughout 22 years using the baseline and winsorization of the raw/residual returns with $\epsilon = 1\%, 5\%$. It is evident

T	w	k (our choice)	τ^+	τ^-
60 days	5 days	40 clusters	1	1
d	P	p	ℓ	q
20 days	0.9	0	3 days	0.001

Table 2: Parameter values in our experiments.

that winsorization of the raw returns is not a good idea, as its PnL almost consistently performs worse than the baseline. At the end of the 22 years, PnL from winsorizing the raw returns is only around 0.4-0.6, which is significantly lower than around 1.0 for the baseline. On the other hand, winsorization of the residual returns does moderately help with PnL performance, boosting the final PnL value to be slightly above 1.0. Also, $\epsilon = 5\%$ seems to be a little better than $\epsilon = 1\%$.

We believe that the success of our trading strategy should rely on the idiosyncratic evolution component of individual stocks and use the regression in Section 2.1. Winsorization of the raw returns might damage some of this dynamic, as some purported outliers might instead reflect the idiosyncracies of those stocks. Residual returns, however, are already representations of such dynamic, and removing outliers from them can actually help.

We also report the quantitative results about the experiments in terms of the final cumulative PnL at the end of 2021, Sharpe ratio, and success rate in Table 3. From PnL and Sharpe ratio standpoints, we believe that we should winsorize residual returns at a threshold of 1%.

Setting	PnL	Sharpe Ratio	Success Rate
baseline	0.983313	0.935421	0.527091
winsorize raw 0.05	0.438600	0.567879	0.465858
winsorize res 0.05	1.058808	0.961939	**0.538955**
winsorize raw 0.01	0.548323	0.627381	0.494918
winsorize res 0.01	**1.086277**	**1.022078**	0.533362

Table 3: Performance of different winsorization strategies.

3.2 Cluster selection

We performed 9 runs of the result with different cluster selection combinations. We fix our baseline to be having 40 original clusters and trade for all these 40 original clusters. And we test for the following combinations:

For each experimental run, we fix our original number of clusters by SPONGE to be 40, then we decide to only keep a certain number of it by discarding the bottom ones in terms of correlation density measure. In each trading period, we are only trading with the ones that we have chosen to be remained in the "trading clusters." For each experimental run, we also perform runs on control groups, with the same number of original cluster as the number of trading clusters in the experimental group, but we do not discard any clusters from the control runs.

From both Figure 4 and 5, we see that the runs with Cluster Selection is neither significantly better nor significantly worse than the control group. The two that outperforms that are the one with

Setting #	Original Cluster #	Trading Cluster #
baseline	40	40
1	40	5
2	40	10
3	40	20
4	40	30
5	5	5
6	10	10
7	20	20
8	30	30

Table 4: The Experiment Settings for Cluster Selection. The row in blue is the run that we failed to implement since the SPONGE package constantly reports an error that we cannot fix. All other results are reported and discussed.

Run Name	Final PnL	Sharpe Ratio	Success Rate
baseline	0.983313	0.935421	0.527091
40 choose 5	0.930600	0.855173	0.541011
40 choose 10	1.013241	0.942249	**0.547154**
40 choose 20	0.957010	0.873160	0.545766
5	0.942771	0.870318	0.543015
10	0.776835	0.716692	0.546968
20	**1.049930**	**0.963611**	0.537377
30	0.973835	0.924952	0.539086

Table 5: Performance comparison of different clustering strategies in terms of Final Cumulative PnL, Sharpe Ratio, and Success Rate.

20 clusters and no selection and the one with 10 trading clusters selected from 40 clusters. Even when comparing between the corresponding experimental groups and control groups, the result is not significant enough to draw conclusions about performance of Cluster Selection.

3.3 Reweighting

Figure 5 and Table 4 record the performance of the trading strategy when the weighting scheme is modified. Both methods that put more weight on stocks with stronger signals outperform the baseline strategy where all stocks within the same cluster are treated equally. The linear weighting strategy worked especially well with regard to the PnL, obtaining triple the amount of cumulative PnL as the baseline over the 22-year period. On the other hand, the exponential strategy obtains the highest Sharpe ratio of 1.13.

Run Name	Final PnL	Sharpe Ratio	Success rate
`uniform`	0.983313	0.935421	0.527091
`linear`	**3.200949**	0.917903	**0.614686**
`exponential`	1.268054	**1.126639**	0.549596

Table 6: Performance of different weighting strategies

Figure 3: Cumulative PnL from experiments with 4 different types of winsorization compared to the baseline.

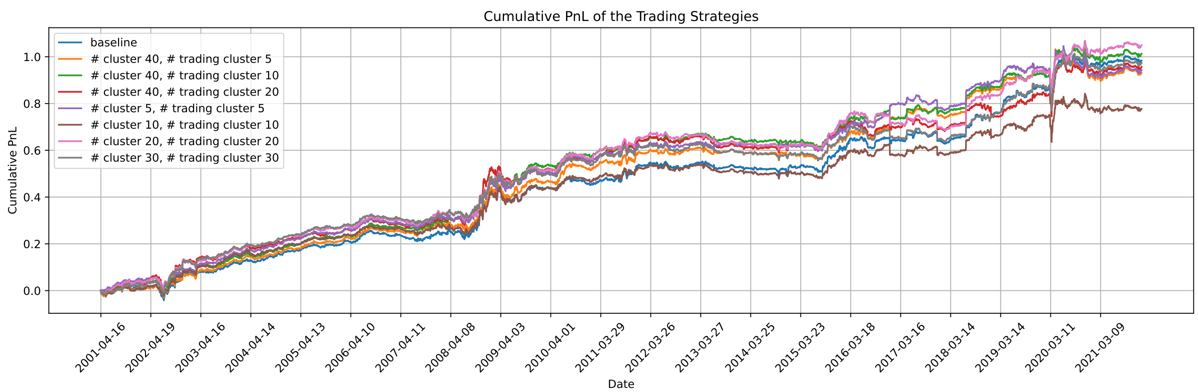

Figure 4: An overview of the experiemental result for Cluster Selection Method. We may see that the Cluster Selection Method in general is neither significantly better nor significantly worse than the non-selected ones.

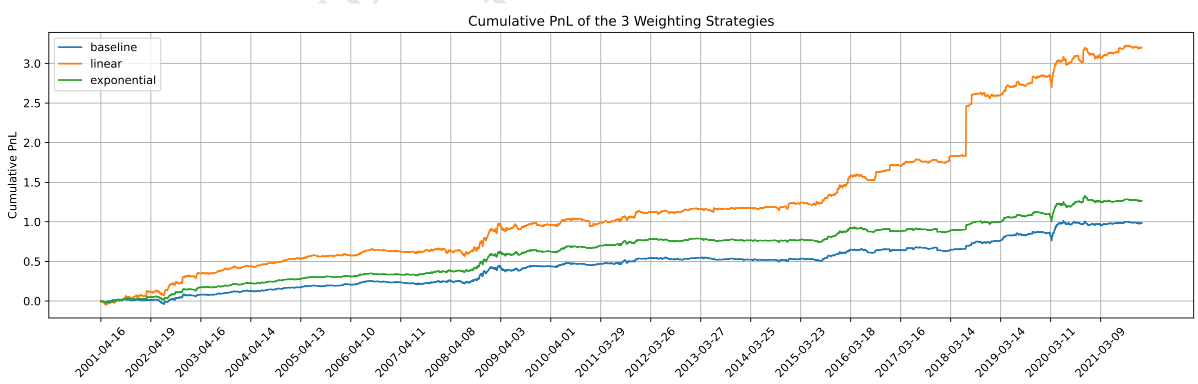

Figure 5: Culmulative PnL with 3 weighting methods: uniform, linear, and exponential. The linear weighting method strongly outperforms the others and both reweighting methods out performs the baseline.

4 Discussion & Conclusion

In this paper, we tested several motivated modifications to the statistical arbitrage pipeline via spectral clustering. Below, we have several remarks about how well each modification works.

Winsorizing residual returns can moderately improve trading performance; winsorizing raw returns gives bad results.

The experiment of Cluster Selections seems not producing conclusive results. Potential reasons are the follows: First, we just artificially set a fixed number of clusters to be dropped, instead of using a definite standard to determine whether a cluster is good enough or not and then decide whether to discard. That may result in bad clusters not being discarded and good clusters getting discarded. Second, though we have "goodness" measures about the clusters, we do not know much about how (and whether) strong intra-correlation results in good PnL performance. Potentially, simulations via Stochastic Block Model can help numerically explore that effect.

Weighting stocks based on signal strength is promising. Specifically, linear and exponential reweighting for betsize both significantly and robustly improved the profiting performance of the portfolio.

It should be pointed out that this project has suffered from numerical difficulties. For example, we have trouble in running the blue-texted experiment due to error in the `signet` package. Moreover, we should have some issues with normalizations of our PnL computation which resulted in our result goes worse even than the SPY though we have good shape against the market trend. We apologize for the numerical issues influencing our result.

There are several avenues of future work. In terms of winner/loser identification in Section 2.3, we could try to winsorize the deviations of the cumulative stock returns away from the cluster mean. Regarding cluster selection, we can try to design better criteria for deciding which clusters to keep or discard, and verify the discrepancy of trading performance between good and bad clusters. We could also consider another strategy for reweighting the stocks as in Section 2.4.2: thresholding. That is, we can specify a threshold $r > p$ such that we only assign weights to stocks whose deviation from the cluster mean exceeds r, essentially filtering out the stocks whose deviations are not sufficiently extreme.

Reproducibility

Our codes are available at
github.com/khangnguyen275/spectral_clustering_finance.git

Acknowledgments

The authors gratefully acknowledge Professor Mihai Cucuringu, for leading this seminar and offering valuable advice for our project.

References

[1] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. 2019. SPONGE: A generalized eigenproblem for clustering signed networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1088–1098.

[2] Qi Jin, Mihai Cucuringu, and Álvaro Cartea. 2023. Correlation matrix clustering for statistical arbitrage portfolios. In *Proceedings of the Fourth ACM International Conference on AI in Finance*. 557–564.

[3] Nail Khelifa, Jérôme Allier, and Mihai Cucuringu. 2024. Cluster-driven Hierarchical Representation of Large Asset Universes for Optimal Portfolio Construction. In *Proceedings of the 5th ACM International Conference on AI in Finance*. 177–185.

Received June 8, 2025