Hierarchical Risk Parity with Spectral Clustering

Model Documentation & Maintenance Guide

Emory Economic Investment Forum (EEIF) — Emory University

Quantitative Research Team

February 2026
Version 1.0

Abstract

This document describes the portfolio optimization model developed for the Emory Economic Investment Forum (EEIF). The model addresses a core challenge in portfolio construction: how to diversify across genuinely distinct risk sources rather than relying on sector labels or return forecasts that are unreliable in practice.

The approach combines two techniques. First, *spectral clustering*—a machine learning method that analyzes the eigenstructure of a similarity graph—discovers natural groupings among stocks based on how their returns co-move after removing the dominant market factor. This data-driven approach reveals structure that traditional sector classifications miss: for example, the model separates energy services from traditional energy, and groups Indian banks with European financials rather than US banks, because that is what the price data shows.

Second, *Hierarchical Risk Parity* (HRP) allocates capital across positions by recursively bisecting a hierarchically-ordered portfolio and balancing risk contributions at each split. Unlike mean-variance optimization, HRP does not require estimating expected returns—it operates entirely on the covariance structure, which is the component of portfolio construction that can actually be estimated reliably from historical data.

Applied to the current EEIF portfolio, the model identifies that approximately 64% of risk is concentrated in a single cluster (technology and growth stocks) and recommends redistributing across 9 distinct clusters, reducing annualized portfolio volatility from 21.8% to 13.3%.

This document serves as both a technical reference for the underlying mathematics and a practical maintenance guide for future EEIF members who need to update, debug, or extend the system. If you are reading this because something broke, start with Section 8.

Contents

1 Introduction

The model addresses a fundamental problem in portfolio management for EEIF: how to allocate capital across a set of holdings such that risk is diversified across genuinely distinct sources, rather than concentrated in a single factor.

Traditional approaches (Markowitz mean-variance, CAPM-based optimization) require estimating expected returns, which is notoriously unreliable. Small errors in return estimates produce wildly different optimal portfolios. Our approach avoids this entirely by focusing only on the *risk structure* of the portfolio.

The model has two stages:

1. **Spectral Clustering** (Section 3): Discovers natural groupings of stocks based on how their returns co-move, after removing the dominant market factor.
2. **Hierarchical Risk Parity** (Section 4): Allocates capital across positions such that risk contributions are balanced across the discovered clusters and within them.

1.1 Key Properties

- No return forecasts required
- Allocation is purely risk-based (inverse volatility weighting)
- Cluster structure is learned from data, not imposed by sector labels
- Model is re-runnable: update holdings, run script, get new targets
- Trained on 255 stocks across 5 markets using 5 years of daily data

2 Data Pipeline

2.1 Training Universe

The model trains on a broad universe of stocks to discover market structure. The training set is not limited to our portfolio holdings—it includes 255 stocks across:

Market	Stocks	Examples
United States	187	AAPL, JPM, JNJ, CAT, XOM
Europe	29	ASML, HSBC, LVMUY, SHEL
China	26	BABA, 0700.HK, NIO
Japan	10	TM, SONY, MUFG
India	2	HDB, IBN
Taiwan	1	TSM

Table 1: Training universe composition.

2.2 Price Data

The input file `training_data.csv` contains adjusted closing prices with the following properties:

- Date range: July 2021 to February 2026 (approximately 1,180 trading days)
- Weekdays only (weekends filtered)
- Forward-filled for missing data (holidays across markets)
- Log returns computed: $r_t = \ln(P_t/P_{t-1})$

2.3 Market Factor Removal

Raw stock correlations are dominated by a single market factor β. When the broad market rises, nearly all stocks rise; when it falls, nearly all fall. This makes the correlation matrix nearly rank-1 and prevents the clustering algorithm from finding structure beyond "energy vs. everything else."

Solution: Regress each stock's returns on the equal-weighted market return and cluster on the *residuals*.

For each stock i:

$$r_{i,t} = \alpha_i + \beta_i \cdot \bar{r}_t + \varepsilon_{i,t} \tag{1}$$

where $\bar{r}_t = \frac{1}{N}\sum_{j=1}^{N} r_{j,t}$ is the equal-weighted market proxy.

The residuals $\varepsilon_{i,t}$ capture how stock i moves *beyond* what the market explains. We compute the correlation matrix on these residuals.

Effect in practice:

- Mean raw off-diagonal correlation: ≈ 0.233
- Mean residual off-diagonal correlation: ≈ 0.006
- Variance explained by market factor: $\approx 25\%$

DO NOT MODIFY: The market factor removal step. Without it, the clustering degenerates to $k = 2$ (energy vs. everything else).

3 Spectral Clustering

Spectral clustering (von Luxburg, 2007) identifies groups of stocks that co-move in their residual returns. Unlike k-means on raw features, spectral clustering operates on a graph representation and can discover non-convex cluster shapes.

3.1 Distance Metric

We convert the residual correlation matrix ρ to a distance matrix:

$$d_{ij} = \sqrt{\frac{1}{2}(1 - \rho_{ij})} \tag{2}$$

This maps correlations of $+1$ to distance 0 and correlations of -1 to distance 1.

3.2 Graph Construction (kNN)

We build a k-nearest-neighbor similarity graph:

1. Compute Gaussian similarity: $W_{ij}^{\text{full}} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right)$
2. For each node i, keep only the k nearest neighbors
3. Symmetrize: if i is a neighbor of j OR j is a neighbor of i, connect them

We use $k = 3$ (sparse graph). Higher values of k over-connect the graph and make the algorithm unable to find fine-grained structure.

DO NOT MODIFY: The value of $k = 3$. This was calibrated specifically for this universe after testing $k \in \{2, 3, 4, 5, 6\}$.

3.3 Normalized Graph Laplacian

From the adjacency matrix W, compute the degree matrix $D = \text{diag}(W\mathbf{1})$ and the random-walk normalized Laplacian:

$$L_{\text{rw}} = I - D^{-1}W \tag{3}$$

3.4 Eigengap Heuristic for k

Compute eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ of L_{rw}.

The number of clusters k is chosen by the *eigengap heuristic*: find the largest gap $\lambda_{k+1} - \lambda_k$ in the first 20 eigenvalues. The intuition is that a graph with k connected components has exactly k zero eigenvalues, so a large gap after eigenvalue k indicates k natural clusters.

3.5 Sigma Selection (Stability Sweep)

The bandwidth parameter σ controls the scale of the Gaussian similarity. We sweep σ across 60 values from the 5th to 95th percentile of pairwise distances. For each σ, we compute the eigengap-optimal k. The most frequently occurring k is selected, and σ is set to the median of the sigmas that produced this k.

This makes the cluster count robust to the choice of σ.

3.6 k-Means in Spectral Space

Embed each stock as a point in $\mathbb{R}^k$ using the first k non-trivial eigenvectors of L_{rw}:

$$\mathbf{x}_i = (v_2(i), v_3(i), \ldots, v_{k+1}(i)) \tag{4}$$

Run k-means (30 restarts) on these embeddings to assign cluster labels.

3.7 Hierarchical Re-Clustering

The initial pass produces $k = 6$ clusters, but one cluster typically contains $> 40\%$ of stocks (the "grab-bag" of everything that isn't energy, financials, industrials, etc.). We recursively re-cluster any oversized cluster:

Algorithm 1: Hierarchical re-clustering.

while *any cluster has* $> 40\%$ *of stocks* **do**
- Select the oversized cluster C;
- Extract sub-distance matrix for stocks in C;
- Run spectral clustering on the sub-matrix (with $k = \log|C|$);
- Split C into sub-clusters;
- Re-number all labels;

end

This produces the final cluster assignment (typically $k = 15$).

4 Hierarchical Risk Parity

Hierarchical Risk Parity (Lopez de Prado, 2016) allocates capital by recursively bisecting a hierarchically-ordered set of assets and assigning weights proportional to inverse cluster variance at each split.

4.1 Portfolio Mapping

Before running HRP, each portfolio holding is assigned to one of the training clusters:

- **Direct assignment**: If the stock is in the training universe, use its cluster label.
- **Projection**: If not, compute residual correlations with all training stocks and assign to the cluster with the smallest average distance.

4.2 Quasi-Diagonalization

Compute the pairwise distance matrix of portfolio holdings and apply Ward's linkage to produce a dendrogram. Extract the leaf ordering—this places correlated assets adjacent to each other.

4.3 Recursive Bisection

Given an ordered list of assets $[1, 2, \ldots, n]$:

1. Split at the midpoint into left half L and right half R
2. Recursively allocate within L and R
3. At each split, weight the two halves by inverse variance:

$$\alpha = 1 - \frac{V_L}{V_L + V_R}, \qquad w_L = \alpha, \quad w_R = 1 - \alpha \tag{5}$$

where $V_L = \mathbf{w}_L^\top \Sigma_{LL} \mathbf{w}_L$ is the variance of the left sub-portfolio.

4.4 Weight Capping

After HRP produces raw weights, we iteratively cap any position exceeding the maximum weight (currently 15%) and redistribute the excess proportionally among uncapped positions:

$$w_i^{\text{new}} = \min(w_i, w_{\max}), \qquad \text{excess} = \sum_i \max(w_i - w_{\max}, 0) \tag{6}$$

This is repeated until convergence (typically 3–5 iterations).

5 Model Output

The model produces a JSON file (`portfolio_results.json`) containing:

- **Positions**: Current weight, HRP target weight, trade amount, cluster assignment, annualized volatility
- **Clusters**: Composition, sector breakdown, market distribution
- **Risk metrics**: Annualized volatility, diversification ratio, HHI, effective number of positions, risk contribution by cluster
- **Technical data**: Eigenvalues, sigma sweep results, correlation matrix, market factor statistics

5.1 Risk Metrics

$$\text{Portfolio volatility:} \quad \sigma_p = \sqrt{\mathbf{w}^\top \Sigma \mathbf{w}} \tag{7}$$

$$\text{Diversification ratio:} \quad \text{DR} = \frac{\sum_i w_i \sigma_i}{\sigma_p} \tag{8}$$

$$\text{HHI:} \quad \text{HHI} = \sum_i w_i^2 \tag{9}$$

$$\text{Effective positions:} \quad N_{\text{eff}} = \frac{1}{\text{HHI}} \tag{10}$$

$$\text{Risk contribution:} \quad \text{RC}_i = \frac{w_i \cdot (\Sigma \mathbf{w})_i}{\sigma_p^2} \tag{11}$$

6 Results

6.1 Cluster Structure

The model discovers 15 clusters from the 255-stock training universe. The portfolio's 28 holdings map across 9 of these clusters:

Cluster	Description	Positions	Key Holdings
C1	SaaS / Growth / Internet	11	GOOG, AMZN, META, MSFT, UBER
C2	REITs / Utilities	1	SHV
C4	Traditional Financials	3	JPM, BRK-B, MKL
C10	EU Banks / Luxury	1	HDB
C11	Energy Services	5	NE, VAL, TDW, HCC, AMR
C12	Fintech / Crypto	2	HOOD, COIN
C13	Aerospace / Industrials	2	IBKR, UNH
C14	Consumer / Healthcare	1	JNJ
C15	Mega Cap Tech / Semis	2	NVDA, TSM

Table 2: Portfolio cluster assignments.

6.2 Risk Comparison

Metric	Current	HRP	Equal Weight
Ann. Volatility	21.79%	**13.26%**	22.54%
Diversification	1.615×	**1.882**×	1.779×
Max Weight	14.46%	15.00%	3.57%
Effective Positions	17.0	15.9	28.0

Table 3: Risk metrics: current vs. HRP vs. equal weight.

The primary finding is that the current portfolio concentrates approximately 64% of risk in a single cluster (C1: tech/growth). HRP redistributes risk across 9 clusters with no single cluster exceeding 30%.

7 File Structure

File	Description
`training_data.csv`	Price data for 255 stocks, 1180 trading days. This is the input.
`portfolio_optimizer.py`	The model. Run this to produce `portfolio_results.json`.
`portfolio_results.json`	Model output. Read by the dashboard and figure generator.
`portfolio_dashboard.html`	Interactive dashboard. Open in browser after running the optimizer.
`generate_figures.py`	Generates presentation-ready PNG figures.

Table 4: Project files.

8 Maintenance Guide

This section is for future maintainers. If you are reading this because something broke, start here.

8.1 Updating Portfolio Holdings

Open `portfolio_optimizer.py` and edit the `PORTFOLIO` dictionary at the top:

```
PORTFOLIO = {
    'GOOG': ('Alphabet', 42),    # ticker: (name, shares)
    'AMZN': ('Amazon', 22),
    # ... add/remove positions here
}
```

Then run: `python portfolio_optimizer.py`

8.2 Updating the Training Universe

If you want to add new stocks to the training set:

1. Download updated price data and regenerate `training_data.csv`
2. Add ticker metadata to the `TICKER_META` dictionary in the optimizer
3. Run the optimizer

The training CSV must have dates as rows and tickers as columns.

8.3 Changing the Weight Cap

Edit `MAX_WEIGHT` at the top of `portfolio_optimizer.py`:

```
MAX_WEIGHT = 0.15   # change to desired cap (0.10 = 10%, 0.20 = 20%)
```

8.4 Common Failure Modes

Symptom	Likely Cause & Fix
$k = 2$ clusters (energy vs everything)	Market factor removal is broken or disabled. Check that the regression step is producing non-zero residuals.
One cluster has >80% of stocks	Hierarchical re-clustering failed. Check that `MAX_FRAC = 0.40` is still in the code.
Ticker not found in training data	The ticker is in `PORTFOLIO` but not in `training_data.csv`. Either add it to the CSV or the model will use projection (which is fine).
NaN in weights	A stock has zero variance (constant price). Remove it from the portfolio.
JSON file empty or missing	The optimizer crashed. Run it again and check the Python error output.
Dashboard shows stale data	You re-ran the optimizer but the browser cached the old JSON. Hard refresh (Ctrl+Shift+R).

Table 5: Troubleshooting guide.

8.5 What NOT to Change

The following components are calibrated to work together. Changing one without understanding the others will break the model:

1. **kNN value ($k = 3$)**: Controls graph sparsity. Higher values over-connect the graph and produce $k = 2$. Lower values create isolated nodes.
2. **Market factor removal**: Without this, the correlation matrix is dominated by beta and the clustering is useless.
3. **Sigma sweep range**: The 5th–95th percentile range is calibrated for the distance scale produced by the correlation-to-distance transform.
4. **Hierarchical re-clustering threshold (40%)**: This breaks up the "grab-bag" cluster. Removing it reverts to $k = 6$ with one cluster containing 174 stocks.
5. **Distance metric**: $d = \sqrt{0.5(1 - \rho)}$ is standard for correlation-based clustering. Do not substitute Euclidean distance.
6. **Ward's linkage in HRP**: This produces the quasi-diagonal ordering that HRP requires. Changing to single or complete linkage will produce different (worse) allocations.

9 Theoretical Justification

9.1 Why Not Mean-Variance?

The Markowitz mean-variance framework (Markowitz, 1952) finds the portfolio with maximum Sharpe ratio:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top \boldsymbol{\mu} - r_f}{\sqrt{\mathbf{w}^\top \Sigma \mathbf{w}}} \tag{12}$$

This requires estimating $\boldsymbol{\mu}$ (expected returns), which is the hardest problem in finance. DeMiguel et al. (2009) showed that $1/N$ (equal weight) outperforms mean-variance optimization

out-of-sample in most datasets because estimation error in $\boldsymbol{\mu}$ overwhelms the theoretical benefit of optimization.

HRP avoids this entirely by never estimating returns.

9.2 Why Spectral Clustering Over Sector Labels?

GICS sector labels are assigned by committee and updated infrequently. They reflect what a company *does*, not how its stock *behaves*. For example:

- IBKR (Interactive Brokers) is classified as a financial, but its residual returns cluster with aerospace/defense stocks
- HDB (HDFC Bank, India) clusters with European banks, not US financials
- Energy services (NE, VAL, TDW) separates cleanly from traditional energy (XOM, CVX)

Spectral clustering discovers these distinctions from price data, which is what actually matters for diversification.

9.3 Why Equal-Weighted Market Proxy?

We use the equal-weighted mean of the training universe as the market factor rather than SPY (S&P 500) because:

- It is defined by the same universe we cluster on (no external dependency)
- It captures the common factor across all markets (US, EU, China, Japan), not just US large-cap
- It avoids the cap-weighting bias of SPY (which is dominated by mega-cap tech)

10 Limitations

1. **Static model**: Clusters are computed once on the full sample. In reality, correlation regimes change over time. A rolling-window variant would address this but adds complexity.
2. **No transaction costs**: The model does not account for trading costs, taxes, or market impact.
3. **No return optimization**: HRP minimizes risk concentration but does not maximize returns. If EEIF has strong return views, these should be incorporated separately (e.g., via Black-Litterman tilts on top of HRP weights).
4. **Backward-looking**: Correlations are estimated from historical data. Structural breaks (e.g., a company pivoting its business) are not captured until sufficient new data accumulates.
5. **Small portfolio**: With 28 positions, some clusters contain only 1–2 holdings. The HRP allocation to these clusters is driven by a single stock's volatility, which makes it sensitive to that stock's recent behavior.

11 Dependencies & Environment

Package	Purpose
Python 3.8+	Runtime
NumPy	Linear algebra, array operations
Pandas	Data loading, time series handling
SciPy	Eigendecomposition, hierarchical clustering
scikit-learn	k-means clustering
Matplotlib	Figure generation (optional)

Table 6: Required Python packages.

Install: `pip install numpy pandas scipy scikit-learn matplotlib`

References

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 395–416.

Lopez de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, 42(4), 59–69.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77–91.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, 22(5), 1915–1953.