Neural Nonlinear Shrinkage of Covariance Matrices for Minimum Variance Portfolio Optimization

Liusha Yang[1], Siqi Zhao[1], Shuqi Chai[2*]
[1]School of Artificial Intelligence, Shenzhen Technology University
[2]Shenzhen Research Institute of Bia Data

***Abstract*—This paper introduces a neural network–based nonlinear shrinkage estimator of covariance matrices for the purpose of minimum variance portfolio optimization. It is a hybrid approach that integrates statistical estimation with machine learning. Starting from the Ledoit–Wolf (LW) shrinkage estimator, we decompose the LW covariance matrix into its eigenvalues and eigenvectors, and apply a lightweight Transformer-based neural network to learn a nonlinear eigenvalue shrinkage function. Trained with portfolio risk as the loss function, the resulting precision matrix (the inverse covariance matrix) estimator directly targets portfolio risk minimization. By conditioning on the concentration ratio (the asset-to-sample size ratio), the approach remains scalable across different sample sizes and asset universes. Empirical results on synthetic data and stock daily returns from Standard & Poor's 500 Index (S&P500) demonstrate that the proposed method consistently achieves lower out-of-sample realized risk than benchmark approaches. This highlights the promise of integrating structural statistical models with data-driven learning.**

***Index Terms*—global minimum variance portfolio, precision matrix estimation, nonlinear eigenvalue shrinkage**

I. Introduction

Portfolio optimization is a famous problem in finance and the classical mean-variance framework of Markowitz [1] remains a cornerstone of modern portfolio theory. However, a well-recognized limitation of this framework is its sensitivity to estimation errors in both expected returns and the covariance matrix, with the former generally considered more difficult to estimate reliably [2]. This observation has motivated a substantial body of work on improving the global minimum variance portfolio (GMVP), which bypasses return estimation and relies solely on the precision of the covariance matrix.

The sample covariance matrix (SCM) is the most widely used estimator; however, it is known to perform poorly when the sample size is comparable to the number of assets, a regime frequently encountered in financial applications [3]. To alleviate this issue, a variety of alternative estimators have been proposed, including factor-based models for structural regularization [4], [5], shrinkage-based techniques for bias-variance trade-offs [6], [7], and spectral filtering methods derived from random matrix theory (RMT) [8], [9]. Complementing these are robust estimation frameworks designed to withstand the non-Gaussian and non-stationary characteristics of financial data [10], [11], which have been integrated into robust covariance shrinkage schemes to leverage the strengths of both RMT and robust statistics [12]–[14].

Despite these advances, a fundamental gap remains: most conventional estimators are optimized for statistical loss functions (e.g., the Frobenius norm) rather than financial performance metrics. An estimator that minimizes statistical distance may yield suboptimal risk-adjusted returns in practice. For complex, task-specific performance criteria, it is often analytically intractable to derive a statistically optimal estimator. These limitations naturally motivate the use of neural networks for task-oriented optimization [15], [16].

Nevertheless, a purely neural approach to covariance estimation is fraught with challenges. The most prominent issue is that the covariance matrix's dimensions grow quadratically with the number of assets, making the direct learning of its full structure statistically demanding and computationally expensive in high-dimensional settings [17]. Alternatively, model-free neural methods that output portfolio weights directly avoid explicit covariance estimation but struggle with data efficiency [18]. By disregarding the well-established structural priors of the GMVP problem, these models require vast training datasets and sophisticated architectures to approximate what could be solved more elegantly through structural optimization [19].

In this work, we propose a hybrid GMVP framework that bridges the structural discipline of statistical estimators with the representational power of neural architectures. Instead of learning a covariance matrix from scratch, our approach utilizes the eigen-decomposition of a baseline statistical estimator. We train a lightweight Transformer-based neural network to perform nonlinear eigenvalue shrinkage optimized directly against the out-of-sample portfolio risk. By reintegrating these learned eigenvalues into the original eigen-structure, we derive a risk-optimized precision matrix, denoted by $\hat{\mathbf{C}}_{\text{risk}}^{-1}$, for GMVP construction. Importantly, we incorporate insights from RMT by conditioning the network on the concentration ratio, which enables the model to dynamically calibrate its shrinkage intensity across diverse asset universes and sample regimes. Concurrently, Bongiorno et al. [20] propose a related neural estimator for correlation matrices in the context of GMVP construction. We refer interested readers to that work for additional perspectives on neural approaches to this problem.

Empirical evaluations across synthetic and real-world stock returns' datasets demonstrate that our hybrid approach con-

* Corresponding author. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62401376, in part by the Natural Science Foundation of Top Talent of SZTU under Grant GDRC202332 and in part by the Shenzhen Science and Technology Program under Grant RCBS20231211090816034.

sistently outperforms state-of-the-art statistical and neural benchmarks. These results validate "guided" machine learning—operating in the spectral domain—as a more data-efficient and stable alternative to traditional or purely neural models for risk minimization.

II. Problem formulation

We consider a time series comprising $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^N$ logarithmic returns of N financial assets, modeled as

$$\mathbf{x}_t = \boldsymbol{\mu} + \mathbf{C}_N^{1/2}\mathbf{y}_t, \quad t = 1, 2, ..., n, \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^N$ is the mean vector of the asset returns, $\mathbf{C}_N \in \mathbb{R}^{N\times N}$ is the positive definite covariance matrix and $\mathbf{y}_t \in \mathbb{R}^N$ is a zero mean unitarily invariant random vector with $\|\mathbf{y}_t\|^2 = N$. Both $\boldsymbol{\mu}$ and $\mathbf{C}_N$ are assumed to be time-invariant over the observation period.

Let $\mathbf{h} \in \mathbb{R}^N$ denote the portfolio selection, i.e., the vector of asset holdings in units of currency normalized by the total outstanding wealth, satisfying $\mathbf{h}^T\mathbf{1}_N = 1$. In this paper, short-selling is allowed, and thus the portfolio weights may be negative. Then the portfolio variance (or risk) over the investment period of interest is defined as $\sigma^2(\mathbf{h}) = E[|\mathbf{h}^T\mathbf{x}_t|^2] = \mathbf{h}^T\mathbf{C}_N\mathbf{h}$ [1]. Accordingly, the GMVP selection problem can be formulated as the following quadratic optimization problem with a linear constraint:

$$\min_{\mathbf{h}} \quad \sigma^2(\mathbf{h}), \qquad \text{s.t.} \;\; \mathbf{h}^T\mathbf{1}_N = 1. \tag{2}$$

This has the well-known solution

$$\mathbf{h}_{\text{GMVP}} = \frac{\mathbf{C}_N^{-1}\mathbf{1}_N}{\mathbf{1}_N^T\mathbf{C}_N^{-1}\mathbf{1}_N}, \tag{3}$$

and the corresponding portfolio risk is

$$\sigma^2\left(\mathbf{h}_{\text{GMVP}}\right) = \frac{1}{\mathbf{1}_N^T\mathbf{C}_N^{-1}\mathbf{1}_N}. \tag{4}$$

Here, (4) represents the theoretical minimum portfolio risk bound, attained upon knowing the covariance matrix $\mathbf{C}_N$ exactly. In practice, $\mathbf{C}_N$ is unknown, and instead we form an estimate, denoted by $\hat{\mathbf{C}}_N$. Thus, the GMVP selection based on the plug-in estimator $\hat{\mathbf{C}}_N$ is given by

$$\hat{\mathbf{h}}_{\text{GMVP}} = \frac{\hat{\mathbf{C}}_N^{-1}\mathbf{1}_N}{\mathbf{1}_N^T\hat{\mathbf{C}}_N^{-1}\mathbf{1}_N}. \tag{5}$$

The quality of $\hat{\mathbf{h}}_{\text{GMVP}}$, implemented based on the in-sample covariance prediction $\hat{\mathbf{C}}_N$, can be measured by its achieved out-of-sample (or "realized") portfolio risk:

$$\sigma^2\left(\hat{\mathbf{h}}_{\text{GMVP}}\right) = \frac{\mathbf{1}_N^T\hat{\mathbf{C}}_N^{-1}\mathbf{C}_N\hat{\mathbf{C}}_N^{-1}\mathbf{1}_N}{(\mathbf{1}_N^T\hat{\mathbf{C}}_N^{-1}\mathbf{1}_N)^2}. \tag{6}$$

The goal is to construct a good estimator $\hat{\mathbf{C}}_N$, and consequently $\hat{\mathbf{h}}_{\text{GMVP}}$, which minimizes this quantity.

Note that, for the naive uniform diversification rule, $\mathbf{h} = \frac{1}{N}\mathbf{1}_N$. This is equivalent to setting $\hat{\mathbf{C}}_N = \mathbf{I}_N$, and yields the realized portfolio risk: $\frac{\mathbf{1}_N^T\mathbf{C}_N\mathbf{1}_N}{N^2}$. Interestingly, this extremely simple strategy has been shown in [21] to outperform numerous optimized models and will serve as a benchmark in our work.

III. Novel Precision Matrix Estimator and Portfolio Design for Risk Minimization

Designing a neural network-based estimator for covariance or precision matrices is inherently challenging. Several obstacles arise: (i) ensuring that the estimated covariance matrix is positive semidefinite—let alone strictly positive definite—is nontrivial; (ii) constructing a network architecture capable of handling matrix-valued inputs of varying dimensions (e.g., different numbers of assets N and return samples n) is not straightforward; and (iii) since the number of free parameters in the empirical covariance matrix $\mathbf{C}_N$ scales quadratically with N, the estimation problem becomes increasingly ill-posed as N grows. To overcome these challenges, we propose a simple yet effective estimation framework that integrates classical statistical techniques with data-driven neural architectures.

Instead of directly estimating the covariance matrix with a neural network, we adopt a hybrid approach that refines existing statistical estimators using neural models, with the objective of reducing portfolio risk. In particular, we build upon the Ledoit–Wolf linear shrinkage estimator [6], which not only improves estimation accuracy compared with the SCM when the asset number N is comparable to the sample size n, but also remains invertible in the more challenging regime $N > n$. In the following, we describe in detail our proposed portfolio optimization strategy, in which the overall workflow is demonstrated in Fig. 1.

A. Precision Matrix Estimation

The sample covariance matrix is defined as

$$\mathbf{S}_N = \frac{1}{n}\sum_{t=1}^{n}\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^T, \tag{7}$$

where $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. The LW linear shrinkage covariance matrix estimator is given by a convex combination of the sample covariance matrix and the identity matrix:

$$\hat{\mathbf{C}}_{\text{LW}} = \rho\mu\mathbf{I}_N + (1-\rho)\mathbf{S}_N, \tag{8}$$

with $\mu = \frac{1}{N}\text{tr}[\mathbf{S}_N]$, $\rho = \dfrac{a^2}{b^2}$, $a^2 = \|\mathbf{S}_N - \mu\mathbf{I}_N\|_{\text{F}}^2$, $b^2 = \frac{1}{n}\sum_{t=1}^{n}\|\mathbf{x}_t\mathbf{x}_t^T - \mathbf{S}_N\|_{\text{F}}^2$, where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm.

Introducing the eigen-decomposition

$$\hat{\mathbf{C}}_{\text{LW}} = \sum_{i=1}^{N}\lambda_i\mathbf{u}_i\mathbf{u}_i^\top, \tag{9}$$

where λ_i denotes the i-th largest eigenvalue and $\mathbf{u}_i$ its associated eigenvector, we seek to construct a precision matrix estimator that retains the robust eigenvectors of the LW estimator while recalibrating its spectrum, which takes the form:

$$\hat{\mathbf{C}}_{\text{risk}}^{-1} = \sum_{i=1}^{N}\eta_i\mathbf{u}_i\mathbf{u}_i^\top, \tag{10}$$

where $\eta_i \geq 0$ is a nonlinear eigenvalue shrinkage function of $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]$. The resulting $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ is optimized

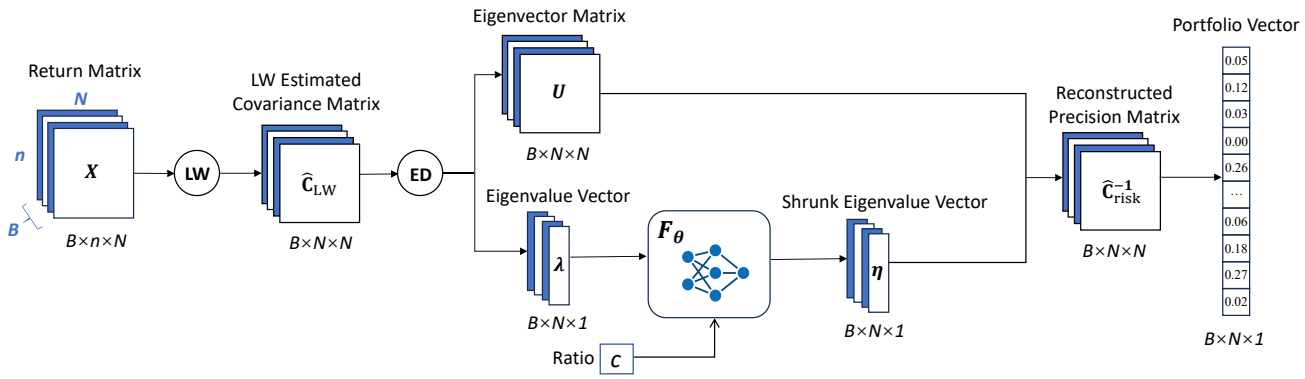

Fig. 1. Overall workflow of portfolio optimization with proposed neural network-based nonlinear precision matrix estimator. The "LW" represent Ledoit-Wolf covariance matrix estimation. The "ED" denotes eigenvalue decomposition.

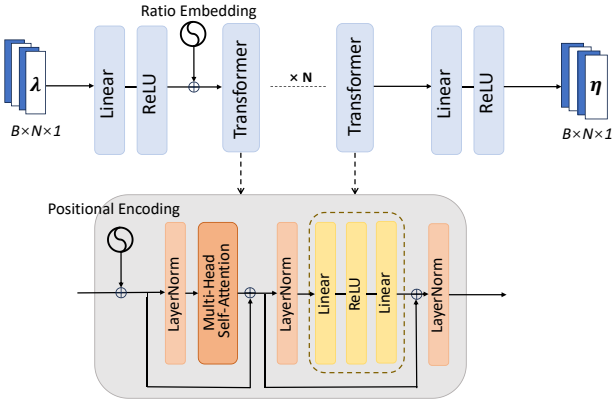

Fig. 2. Lightweight Transformer-based network architecture.

specifically to minimize the portfolio risk in (6). By fixing the eigenvectors $\{\mathbf{u}_i\}_{i=1}^N$, we effectively reduce the learning problem from estimating $O(N^2)$ parameters to learning a spectral mapping function of dimension $O(N)$, significantly mitigating the risk of overfitting. Meanwhile, the mismatch between $\mathbf{u}_i$ and the true corresponding population eigenvector is implicitly accounted for by the learned coefficient η_i.

It is important to emphasize that our goal is to estimate the precision matrix directly, as the portfolio weights in (5) depend explicitly on it. Rather than first estimating $\mathbf{C}_N$ and subsequently inverting it—which can amplify estimation errors—we construct an estimator that directly targets $\mathbf{C}_N^{-1}$.

B. Neural Eigenvalue Shrinkage

We employ a neural network F_θ to learn the nonlinear mapping

$$\boldsymbol{\eta} = F_\theta(\boldsymbol{\lambda}, c), \tag{11}$$

where $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_N]$ and $c = \frac{N}{n}$. The concentration ratio c between the number of assets and the number of samples has been shown to play a central role in designing shrinkage functions [22], [23]. Conditioned on c, the neural network flexibly adjusts the mapping from $\boldsymbol{\lambda}$ to $\boldsymbol{\eta}$ across various (N, n) configurations. This allows the method to handle return matrices $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{N\times n}$ of different shapes. Once $\boldsymbol{\eta}$ is obtained, the precision matrix estimator $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ is constructed by (10), and the corresponding portfolio weight vector is

$$\hat{\mathbf{h}}_{\text{risk}} = \frac{\hat{\mathbf{C}}_{\text{risk}}^{-1}\mathbf{1}_N}{\mathbf{1}_N^\top \hat{\mathbf{C}}_{\text{risk}}^{-1}\mathbf{1}_N}. \tag{12}$$

Learning a direct mapping from data $\mathbf{X}$ to the precision matrix $\mathbf{C}_N^{-1}$ is substantially more difficult than mapping eigenvalues $\boldsymbol{\lambda}$ to shrinkage factors $\boldsymbol{\eta}$, particularly when N and n vary. By reformulating the task as an eigenvalue shrinkage problem, we significantly reduce complexity and enable the model to generalize more effectively across different sample sizes and portfolio dimensions.

1) Network Architecture: We adopt a lightweight Transformer-based architecture, selected for its efficacy in modeling complex dependencies and interactions among eigenvalues, as shown in Fig. 2. The ratio c is embedded into the input to guide the mapping, ensuring that the learned shrinkage function adapts to diverse $\boldsymbol{\lambda}$ derived from $\hat{\mathbf{C}}_{\text{LW}}$ across different sample regimes. Finally, a ReLU activation in the output layer guarantees $\eta_i \geq 0$.

For comparison, we also benchmark an end-to-end strategy, referred to as direct weight estimation (DWE), which directly maps $\mathbf{X}$ to the portfolio weights $\mathbf{h}$ with the network architecture proposed in [24]. The architecture comprises a temporal feature extraction module based on dilated causal convolutions and an asset correlation extraction module based on self-attention. However, this approach is less effective, as shown in Section IV-B2, because it does not leverage the structural properties inherent in the optimization problem, making training considerably more challenging. Moreover, the model lacks flexibility in adapting to varying sample sizes, further limiting its practical applicability.

2) Training Objective: The neural network is trained with portfolio risk as the loss function. Given the eigenvalue mapping $\boldsymbol{\eta} = F_\theta(\boldsymbol{\lambda}, c)$, we construct $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ via (10) and compute the resulting portfolio weights as in (12). The associated

portfolio risk is

$$\mathcal{L} = \hat{\mathbf{h}}_{\text{risk}}^{\top} \mathbf{C}_N \hat{\mathbf{h}}_{\text{risk}}. \tag{13}$$

Since $\mathbf{C}_N$ is unknown in practice, we estimate risk using the out-of-sample variance of portfolio returns:

$$\mathcal{L} = \hat{\mathbf{h}}_{\text{risk}}^{\top} \left(\frac{1}{m} \sum_{t=n+1}^{n+m} \tilde{\mathbf{x}}_t \tilde{\mathbf{x}}_t^{\top} \right) \hat{\mathbf{h}}_{\text{risk}}, \tag{14}$$

where $\tilde{\mathbf{x}}_t = \mathbf{x}_t - \frac{1}{m} \sum_{i=n+1}^{n+m} \mathbf{x}_i$ denotes centered returns in the m out-of-sample periods. This decision-focused objective ensures that the network prioritizes the estimation accuracy of the eigen-components that are most consequential for risk minimization, rather than treating all directions equally. The complete training procedure of the proposed method is summarized in Algorithm 1.

Algorithm 1: Training pipeline for shrinkage function F_θ

Input: Training samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with randomly chosen (N, n); validation samples $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$; ratio c

Output: Loss value $\mathcal{L}$

1 Compute $\hat{\mathbf{C}}_{\text{LW}}$ using (8) from $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$;
2 Perform eigen-decomposition: $\hat{\mathbf{C}}_{\text{LW}} \to \boldsymbol{\lambda}, \{\mathbf{u}_i\}_{i=1}^N$;
3 Apply shrinkage function: $\boldsymbol{\eta} \leftarrow F_\theta(\boldsymbol{\lambda}, c)$;
4 Construct inverse covariance: $\hat{\mathbf{C}}_{\text{risk}}^{-1} = \sum_{i=1}^N \eta_i \mathbf{u}_i \mathbf{u}_i^T$;
5 Estimate weights: $\hat{\mathbf{h}}_{\text{risk}} = \dfrac{\hat{\mathbf{C}}_{\text{risk}}^{-1} \mathbf{1}_N}{\mathbf{1}_N^T \hat{\mathbf{C}}_{\text{risk}}^{-1} \mathbf{1}_N}$;
6 Compute loss $\mathcal{L}$ with $\{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}\}$ using (14).

IV. Experiment Results

A. Validation of loss function design

To evaluate the efficacy of the proposed risk-oriented loss function (14), we conduct a comparative analysis against the conventional Frobenius loss using synthetic data. This setup allows us to isolate the impact of the loss function under a controlled ground truth.

Synthetic Data Generation: We generate synthetic returns i.i.d. from a multivariate normal distribution. Without loss of generality, the mean vector $\boldsymbol{\mu}$ is set to zero, as centering the returns removes its influence on covariance estimation. We adopt a one-factor return structure [25] for the population covariance matrix $\mathbf{C}_N$ with dimension $N = 50$:

$$\mathbf{C}_N = \mathbf{b}\mathbf{b}^T \sigma^2 + \boldsymbol{\Sigma}, \tag{15}$$

where the factor volatility is $\sigma = 0.16$. The factor loadings $\mathbf{b} \in \mathbb{R}^N$ are evenly spaced in $[0.5, 1.5]$, and the residual variance matrix is defined as $\boldsymbol{\Sigma} = \sigma_r^2 \mathbf{I}$ with $\sigma_r = 0.2$.

Statistical vs. Decision-Aligned Loss: The prevailing paradigm in precision matrix estimation focuses on minimizing the statistical discrepancy between the estimated matrix $\hat{\mathbf{C}}_N^{-1}$ and the population matrix $\mathbf{C}_N^{-1}$. This is typically achieved via the Frobenius norm loss [13], [22]:

$$\hat{\mathbf{C}}_{\text{Fro}}^{-1} = \arg\min_{\hat{\mathbf{C}}_N^{-1}} \|\hat{\mathbf{C}}_N^{-1} - \mathbf{C}_N^{-1}\|_{\text{F}}^2. \tag{16}$$

However, the Frobenius loss is "application-agnostic"; it assigns equal weight to all entries in the covariance matrix. In the context of portfolio optimization, this is often suboptimal because the portfolio risk is a weighted quadratic aggregation of these entries. Errors in covariance elements corresponding to small portfolio weights have a negligible impact on the final objective, whereas errors in high-weight elements can significantly degrade performance.

To bridge this gap, we employ a decision-aligned risk loss that directly optimizes the downstream utility–the portfolio variance:

$$\hat{\mathbf{C}}_{\text{risk}}^{-1} = \arg\min_{\hat{\mathbf{C}}_N^{-1}} \frac{\mathbf{1}_N^T \hat{\mathbf{C}}_N^{-1} \mathbf{C}_N \hat{\mathbf{C}}_N^{-1} \mathbf{1}_N}{(\mathbf{1}_N^T \hat{\mathbf{C}}_N^{-1} \mathbf{1}_N)^2}. \tag{17}$$

By aligning the training objective with the GMVP's operational goal, this loss function guides the neural shrinker to prioritize the estimation of spectral directions that most heavily influence the realized risk. For our implementation, the neural shrinker consists of 6 Transformer layers, optimized using Adam with a learning rate of 10^{-4} and a batch size of 8.

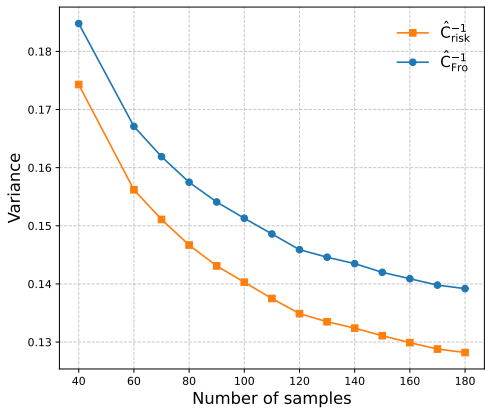

Fig. 3. Loss comparison between Frobenius norm minimization and risk minimization objectives under varying training sample sizes.

Result Analysis: Fig. 3 illustrates the realized portfolio variance for both loss functions across varying training sample sizes $n \in [40, 180]$, averaged over 200 Monte Carlo iterations. As the training sample size increases, the backtested portfolio variance decreases for both methods, indicating that longer training periods improve the model's ability to capture the underlying covariance structure. Across all sample sizes, the curve corresponding to $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ consistently lies below that of $\hat{\mathbf{C}}_{\text{Fro}}^{-1}$, demonstrating that the risk loss delivers stable improvements in realized portfolio variance. These empirical findings corroborate the use of (14) as the training objective for the neural shrinker of the covariance matrix.

B. Performance evaluation

In this section, we evaluate the performance of the proposed risk-optimized estimator, $\hat{\mathbf{C}}_{\text{risk}}^{-1}$, by benchmarking it against the following competing methods.

1) $\hat{\mathbf{C}}_{\text{SCM}}$: The standard sample covariance matrix;
2) $\hat{\mathbf{C}}_{\text{LW}}$: The Ledoit-Wolf shrinkage estimator [6];
3) $\hat{\mathbf{C}}_{\text{C}}$: The robust shrinkage estimator proposed by Chen et al. [13];
4) $\mathbf{I}_N$: The identity matrix, which yields the $1/N$ (equally weighted) portfolio [21];
5) Direct Weight Estimation (DWE): A model-free neural network [24] that directly outputs portfolio weights from historical returns, without explicit covariance estimation. The model is trained using $N = 50$ assets and a window length of $n = 60$.

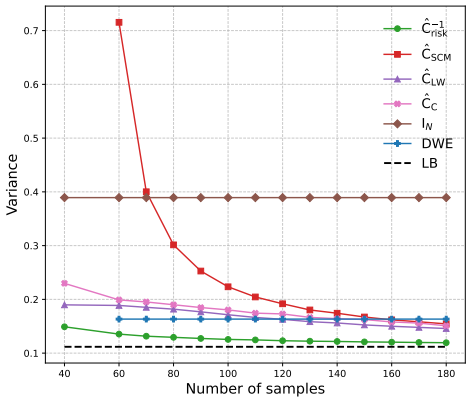

Fig. 4. Portfolio variance comparison on the synthetic dataset.

1) Results on synthetic data: We first evaluate the estimators on synthetic data generated from the one-factor model described in Section IV-A, which enables a direct comparison between the realized risk and the theoretical minimum risk (the "Oracle" bound) in (4).

Fig. 4 reports the realized portfolio variance averaged over 200 Monte Carlo simulations. The proposed estimator, $\hat{\mathbf{C}}_{\text{risk}}^{-1}$, consistently achieves the lowest variance across all sample sizes and closely approaches the theoretical lower bound. Although $\hat{\mathbf{C}}_{\text{LW}}$ and $\hat{\mathbf{C}}_{\text{C}}$ improve as n increases, they remain clearly suboptimal relative to the proposed risk-optimized neural shrinker.

Results for $\hat{\mathbf{C}}_{\text{SCM}}$ and DWE are omitted for $n < 50$ and $n < 60$, respectively. The SCM is non-invertible when $n \leq N$, so the GMVP is undefined in this under-sampled regime. DWE requires a fixed 60-day input window and therefore cannot be applied to shorter sequences. For $n > 60$, its input is truncated to length 60, which explains its unchanged performance beyond that point. Overall, the proposed method delivers the best risk control across all sample sizes in the synthetic setting.

2) Results on real stock data: We use constituents of the S&P 500 index as the empirical dataset. Dividend-adjusted daily closing prices are collected from Yahoo Finance, and continuously compounded (log) returns are computed for a randomly selected subset of 50 stocks. The sample contains 3,675 trading days from Jan. 3, 2011 to Jul. 31, 2025, and is split into 3,275 training observations (Jan. 3, 2011 to Dec. 31, 2023) and 400 test observations (Jan. 3, 2024 to Jul. 31, 2025).

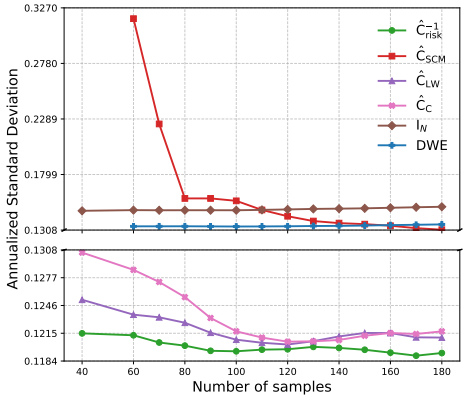

Fig. 5. Annualized out-of-sample portfolio risks over 400 trading days of 50 S&P500 stocks.

We evaluate out-of-sample portfolio performance using a rolling-window procedure. On each rebalancing date t, the previous n days are used to estimate the covariance matrix and construct the GMVP weights $\hat{\mathbf{h}}_{\text{GMVP}}$, which are then held fixed over the next 20 trading days. The window is rolled forward by 20 days until the end of the sample. Realized portfolio risk is measured by the annualized sample standard deviation of the resulting GMVP returns.

As shown in Fig. 5, the proposed estimator $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ attains the lowest realized risk for all window lengths considered, consistently outperforming all competing methods in both the under-sampled regime $n \leq N$ and the over-sampled regime $n > N$.

To provide a finer temporal comparison, we further examine the two best-performing methods, $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ and $\hat{\mathbf{C}}_{\text{LW}}$, using $n = 100$. This yields 300 out-of-sample GMVP returns. Starting from the beginning of the test period, we compute the annualized standard deviation of the most recent 40 returns and roll the window forward by one day, resulting in 261 rolling risk estimates for each method, shown in Fig. 6. We find that $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ achieves lower risk than $\hat{\mathbf{C}}_{\text{LW}}$ in 72.1% of cases, with especially clear gains during high-volatility periods such as Apr. 2024 to Jun. 2024.

Finally, we study sensitivity to the number of assets N while fixing $n = 60$. For each N, the out-of-sample risk is averaged over eight random subset selections. As shown in Fig. 7, most methods benefit from diversification, leading to gradually lower portfolio risk as the investment universe expands. By contrast, the SCM displays sharply increasing risk as N approaches n, reflecting the ill-conditioning of the sample covariance matrix. Throughout, the proposed estimator $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ maintains the lowest risk, confirming its robustness across different asset-to-sample ratios.

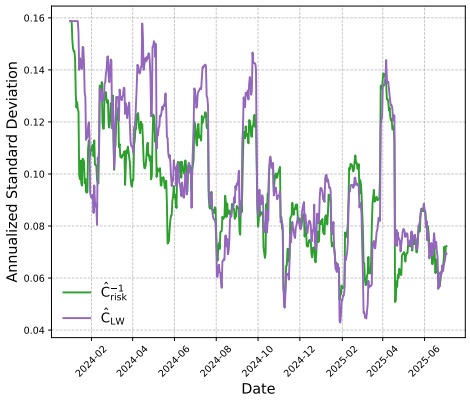

Fig. 6. Annualized rolling-window standard deviations of the most recent 40 out-of-sample GMVP log returns for $\hat{\mathbf{C}}_{\text{risk}}^{-1}$ and $\hat{\mathbf{C}}_{\text{LW}}$ ($N = 50$, $n = 100$).

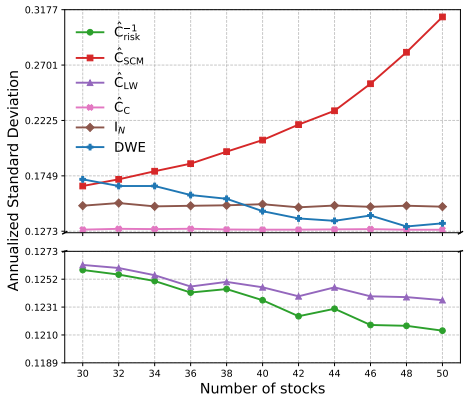

Fig. 7. Annualized out-of-sample portfolio risk as a function of the number of assets N with a fixed $n = 60$.

V. Conclusion

This paper has introduced a novel minimum-variance portfolio optimization strategy based on a nonlinear shrinkage covariance estimator, in which the shrinkage function is learned via a neural network and explicitly calibrated to minimize realized portfolio risk. By leveraging data-driven eigenvalue shrinkage, the proposed approach consistently outperforms standard benchmarks in terms of realized risk, as demonstrated on both synthetic datasets and real historical stock returns from S& P500. An important direction for future research is to extend the proposed framework to incorporate both covariance and mean-return estimation, thereby enabling optimization under alternative criteria such as Sharpe ratio maximization or the classical Markowitz mean-variance formulation. Another promising avenue is to account for temporal dependencies and the time-varying nature of covariance matrices, which may further enhance the robustness and adaptability of the proposed strategy in dynamic financial environments. These extensions are left for future work.

References

[1] Harry Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952.

[2] Ravi Jagannathan and Tongshu Ma, "Risk reduction in large portfolios: Why imposing the wrong constraints helps," *J. Finance*, vol. 58, no. 4, pp. 1651–1684, Aug. 2003.

[3] Noureddine El Karoui, "On the realized risk of high-dimensional Markowitz portfolios," *SIAM J. Finan. Math.*, vol. 4, no. 1, pp. 737–783, 2013.

[4] Louis KC Chan, Jason Karceski, and Josef Lakonishok, "On portfolio optimization: Forecasting covariances and choosing the risk model," *Rev. Finan. Studies*, vol. 12, no. 5, pp. 937–974, 1999.

[5] Jianqing Fan, Yingying Fan, and Jinchi Lv, "High dimensional covariance matrix estimation using a factor model," *J. Econometrics*, vol. 147, no. 1, pp. 186–197, Nov. 2008.

[6] Olivier Ledoit and Michael Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivar. Anal.*, vol. 88, no. 2, pp. 365–411, Feb. 2004.

[7] Olivier Ledoit and Michael Wolf, "Nonlinear shrinkage of the covariance matrix for portfolio selection: Markowitz meets Goldilocks," *Rev. Finan. Studies*, vol. 30, no. 12, pp. 4349–4388, 2017.

[8] Laurent Laloux, Pierre Cizeau, Jean-Philippe Bouchaud, and Marc Potters, "Noise dressing of financial correlation matrices," *Phys. Rev. Lett.*, vol. 83, no. 7, pp. 1467, Aug. 1999.

[9] Vasiliki Plerou, Parameswaran Gopikrishnan, Bernd Rosenow, Luis A Nunes Amaral, Thomas Guhr, and H Eugene Stanley, "Random matrix approach to cross correlations in financial data," *Phys. Rev.*, vol. 65, no. 6, pp. 066126, June 2002.

[10] Peter J Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.

[11] David E Tyler, "A distribution-free M-estimator of multivariate scatter," *Ann. Statist.*, vol. 15, no. 1, pp. 234–251, 1987.

[12] Romain Couillet, Frédéric Pascal, and Jack W Silverstein, "Robust estimates of covariance matrices in the large dimensional regime," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 7269–7278, Sept. 2014.

[13] Yilun Chen, Ami Wiesel, and Alfred O. Hero, "Robust shrinkage estimation of high-dimensional covariance matrices," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4097–4107, Sept. 2011.

[14] Romain Couillet and Matthew R McKay, "Large dimensional analysis and optimization of robust shrinkage covariance matrix estimators," *J. Mult. Anal.*, vol. 131, pp. 99–120, 2014.

[15] Shiguo Huang, Linyu Cao, Ruili Sun, Tiefeng Ma, and Shuangzhe Liu, "Enhancing portfolio optimization: A two-stage approach with deep learning and portfolio optimization," *Mathematics*, vol. 12, no. 21, pp. 3376, 2024.

[16] A Sinem Uysal, Xiaoyue Li, and John M Mulvey, "End-to-end risk budgeting portfolio optimization with neural networks," *Ann. Oper. Res.*, vol. 339, no. 1, pp. 397–426, 2024.

[17] Juchan Kim, Inwoo Tae, and Yongjae Lee, "Estimating covariance for global minimum variance portfolio: A decision-focused learning approach," in *Proceedings of the 6th ACM International Conference on AI in Finance*, 2025, pp. 105–113.

[18] Zihao Zhang, Stefan Zohren, and Stephen Roberts, "Deep learning for portfolio optimization," *J. Financ. Data Sci.*, vol. 2, no. 4, pp. 8–20, 2020.

[19] Andrew Butler and Roy H Kwon, "Integrating prediction in mean-variance portfolio optimization," *Quant. finance*, vol. 23, no. 3, pp. 429–452, 2023.

[20] Christian Bongiorno, Efstratios Manolakis, and Rosario Nunzio Mantegna, "End-to-end large portfolio optimization for variance minimization with neural networks through covariance cleaning," *J. Finance Data Sci.*, vol. 30, 2026.

[21] Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal, "Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy?," *Rev. Finan. Stud.*, vol. 22, no. 5, pp. 1915–1953, 2009.

[22] Olivier Ledoit and Michael Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivariate Anal.*, vol. 88, no. 2, pp. 365–411, 2004.

[23] David L Donoho, Matan Gavish, and Iain M Johnstone, "Optimal shrinkage of eigenvalues in the spiked covariance model," *Ann. Statist.*, vol. 46, no. 4, pp. 1742, 2018.

[24] Tianlong Zhao, Xiang Ma, Xuemei Li, and Caiming Zhang, "Asset correlation based deep reinforcement learning for the portfolio selection," *Expert Syst. Appl.*, vol. 221, pp. 119707, 2023.

[25] Zhidong Bai, Huixia Liu, and Wing-Keung Wong, "Enhancement of the applicability of Markowitz's portfolio optimization by utilizing random matrix theory," *Math. Finance*, vol. 19, no. 4, pp. 639–667, 2009.