A Novel Graph-Based Factor Model
In the Application of Markowitz Mean-Variance Portfolio Optimization

Feng Feng, Yuxiao Feng, Jiaming Liu

April 2022

1 Introduction

The factor model has provided a framework to link stock performance to their fundamental drivers in a concise, easy-to-understand, and powerful manner. The model also has its applications in portfolio allocation, in which the recomposed stock returns and covariance matrices are used for Markowitz mean-variance portfolio optimization. In particular, the returns and covariances of the stocks derived from the factors can help to decrease the impact of temporary fluctuations and increase the model robustness and reliability by linking to fundamental factors. However, there are several limitations in the application:

1. The effectiveness of the model highly depends on the wise selection and construction of the factors. If the explanatory power of the factors is low, a significant portion of the information carried in regression residuals will be lost, and the estimated returns and covariances will be largely deviated from the reality.

2. There are (possibly infinitely) many factors that drive stock price movements. However, when the number of factors grow larger, the computational complexity of multivariable regression grows cubically, and the regression faces potential problems of multicollinearity and model overfitting.

3. The fundamental factors are traditionally purely constructed from historic data, which could be out-of-date and loses predictive power. In addition, the model is not flexible enough to incorporate the investment managers' personal views, which could potentially help the portfolio to outperform the market.

The above limitations motivate us to construct a novel graph-based factor model which is presented below. Section 2 is the synthesis of the available literature regarding our topic. Section 3 introduces the construction and usage of the model. Section 4 covers backtesting experiment results that are primarily focused on the application of predicting stock covariance matrices. Section 5 summarizes the model and gives some directions of future improvement.

2 Literature Review

Multi-factor asset pricing model is one of the most basic and classic modern portfolio theories. The portfolio selection theory by Markowitz analyses how financial assets with different expected returns and risks can be optimally invested so that total risk can be reduced [Mar52]. Markowitz's portfolio selection theory is the basis of developing Sharpe theory of price formation for financial assets, well known as the Capital Asset Pricing Theory (CAPM), which started the study in factor asset pricing model [Sha64]. The asset pricing model highlights a new way to decompose stock returns into main drivers of factors, but the limitations in CAPM model are obvious:

1. It lacks explanatory power as it only attributes the stock performance to market movement, which is inadequate as asset return is affected by many drivers.

2. It lacks generalization power as the empirical study suggests that it does not work well in all markets, and thus cannot be generalized to the broader global market.

To address the first drawback, researchers have developed many new factors to add explanation to the asset return. Fama and French proposed 3-factor model in 1993 [FF93], adding size (SMB) and value (HML) factors to the traditional CAPM model. The development of factors has then started to accelerate. This includes Frazzini and Pedersen proposing volatility factors to explain the abnormal excess return of low volatility stocks due to investors' constraints [FP14], Asness proposing a model that includes quality-minus-junk (QMJ) factors which tries to capture returns between high and low-quality stocks in terms of profitability, growth, and safety [AFP19], and Fama and French proposing 5-factor model by adding profitability (RMW) and investment (CMA) factors to the original 3-factor model [FF15].

As for the second drawback, researchers have developed new models to incorporate universal factors such as macroeconomic indicators to generalize the traditional factor models. Ross first proposed the Arbitrage Pricing Theory [Ros76], which was developed as a generalization of CAPM in that the researchers selected financial and macroeconomic variables to serve as factors based on economic reasoning. It leads to two main categories of factors: macroeconomic factors, which capture broad risks across asset classes, and style factors, which help to explain returns and risk within asset classes. We mainly focus on the macroeconomic indicators in our model. Despite the extensive work on factor models, some state-of-the-art factor models is still insufficiently capable in some aspects, especially in modeling the connectivity between factors. One such paper has proposed an approach using graph-based neural networks to estimate the connectivity between firms based on their previous stock returns [SL22], which are further used to construct the risk exposure network (factor loadings) and factor network (factor values). However, the paper mainly modeled the connectivity between firms with pure statistical and machine learning techniques, still inadequate to capture the underlying macroeconomic drivers with strong fundamental and explainability support. To address these limitations as well as those proposed in Section 1, we propose our graph-based factor model as below.

3 Methodology

The following subsections detail the overview of our model (3.1), the architecture of the proposed graph-based factor model (3.2), the mechanism when posting opinions (3.3), the methods of variance / covariance prediction (3.4), and the algorithm of portfolio allocation optimization (3.5).

3.1 Model overview

Our model is designed to address the three limitations of traditional factor models listed in Section 1. To improve on the first limitation, we will incorporate the regression residuals in the estimation of the factors and stocks, such that the returns and covariances are separated into the explained part (linear output from the regression model) and the unexplained part (regression residuals). Different from the original factor model that is constructive, our model is explanatory, meaning that it does not depend on complete factor construction, but even little additional information helps to improve the model accuracy while pertaining idiosyncratic properties of the stocks. Second, we will adopt a directed-graph model of the factors, where the nodes represent factors and stocks, and one factor could explain some other factors in a layering manner. Third, we will enable subjective opinions of either returns, variances, or covariances to be injected into any of the nodes or node groups, and we will demonstrate how the opinions could be propagated downstream to ultimately impact stock returns and covariances.

Fig 1 is a summary of the procedure of our model:

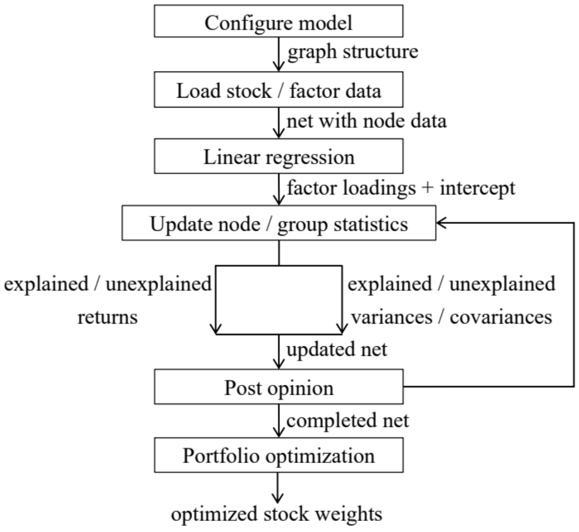

Figure 1: Procedure summary

3.2 Graph-based factor model

Since the factors included in our model are mainly macroeconomics factors, we denote the observable factor returns by f and nonobservable factor loadings (beta vectors) by β.

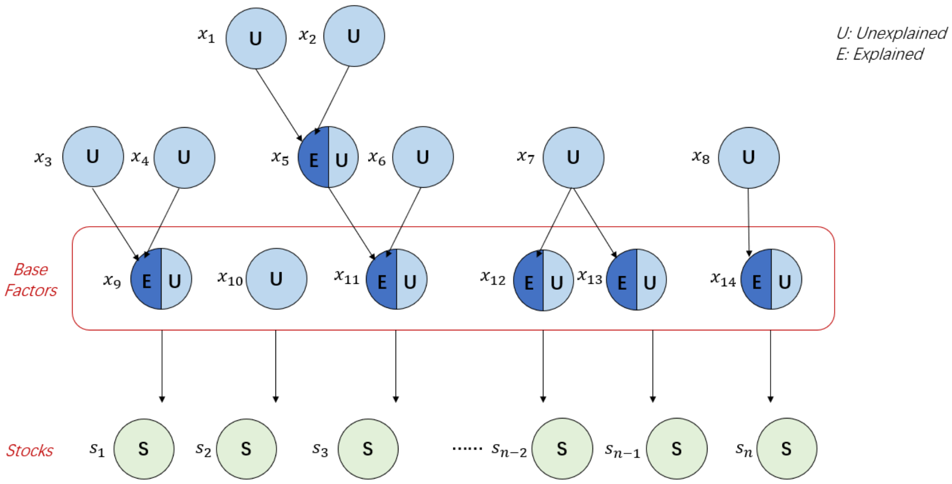

Figure 2: Net

Fig 2 shows the architecture of a basic case of the *net* of our proposed model, formed by *node*s and *group*s.

On the *node*s level, there are two kinds of nodes, the factor nodes and the stock nodes. We further separate the factor nodes into two types: the dependent ones, which are partially explained by the upstream nodes, and the independent ones, which have no upstream nodes.

Note that for a dependent node i, we call the set of all its upstream nodes a *upstream group* $g_i^{upstream}$. We also define other kinds of groups: the *base group* g^{base}, formed by the base factors on the bottom level; the *stock group* g^{stock}, containing all the stock nodes; and the *upstream common group* g_{ij}^{common}, which is a set containing all the common upstream nodes of any two dependent nodes i and j.

We also define G_i^{in} the set of all groups that node i is in, and $G_i^{downstream}$ the set of all i's downstream groups.

For instance, in Fig 2, the summary of the groups is as follows:

- Base group: $g^{base} = \{x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}\}$
- Stock group: $g^{stock} = \{s_1, s_2, s_3, ..., s_n\}$
- Common groups: $g^{common}_{x_{12}x_{13}} = \{x_7\}$, $g^{common}_{s_i s_j} = g^{base} \quad \forall i, j \in \{1, 2, ...n\}$

And taking node x_5 as an example, we have:

- x_5's upstream group: $g^{upstream}_{x5} = \{x_1, x_2\}$
- Set of all groups x_5 is in: $G^{in}_{x5} = \{\{x_5, x_6\}\}$
- Set of all x_5's downstream groups: $G^{downstream}_{x5} = \{g^{base}, g^{stocks}\}$

For a single node, we examine its return and variance. Note that the total return and variance at T are calculated by the exponentially weighted time series with weight vector w where

$$w_t = \frac{(r_{EMA})^{T-t-1}}{\sum_{s=0}^{T-1}(r_{EMA})^{T-s-1}} \tag{1}$$

where r_{EMA} is the rate of exponential decay.

The total return f_i of a node i can be divided into two parts: the explained return f_i^e and the unexplained return f_i^u. If node i is dependent, we have

$$\begin{aligned} f_i^e &= \beta_i f_i^{upstream}, \\ f_i^u &= f_i - f_i^e = \epsilon_{fi} \end{aligned} \tag{2}$$

where $f_i^{upstream} = \langle f_k \rangle$ are the return vector of all the factors $k \in g_i^{upstream}$ and β_i is the beta vector estimated by linear regression with sample weight w. If node i is independent, then $f_i^e = 0$ and $f_i^u = f_i$.

Similarly, the total variance σ_i^2 of node i can also be divided into explained and unexplained. If node i is dependent, we have

$$\begin{aligned}(\sigma_i^e)^2 &= \beta_i \Sigma_i^{upstream} \beta_i^T, \\ (\sigma_i^u)^2 &= \sigma_i^2 - (\sigma_i^e)^2 = \epsilon_{\sigma^2 i}\end{aligned} \tag{3}$$

and $\Sigma_i^{upstream}$ is the covariance matrix of $g_i^{upstream}$. If node i is independent, then $\sigma_i^e = 0$ and $\sigma_i^u = \sigma_i$.

For the covariance between two nodes i and j, there are two cases:

Case 1. Node i and node j have no common upstream node

Since there are no common upstream node, the explained part $\sigma_{ij}^e = 0$ and the unexplained part $\sigma_{ij}^u = \sigma_{ij}$.

Case 2. Node i and node j have common upstream nodes

Since there are common upstream nodes, the covariance can be partially explained by the covariance of the common upstream nodes. We have

$$\begin{aligned}\sigma_{ij}^e &= \beta_i^{'} \Sigma_{ij}^{common} \beta_j^{'T}, \\ \sigma_{ij}^u &= \sigma_{ij} - \sigma_{ij}^e\end{aligned} \tag{4}$$

where Σ_{ij}^{common} is the covariance matrix of g_{ij}^{common}, and $\beta_i^{'}$ and $\beta_j^{'}$ are the beta vectors estimated by linear regression with sample weight w using the common upstream nodes only. Then the correlation between nodes i and j is $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$.

For a group g, we examine its covariance and correlation matrix, which are formed by the covariance and correlation of all pairs of nodes i, $j \in g$, where

$$\begin{aligned}\Sigma_{ij}^g &= \sigma_{ij}, \\ \Sigma_{ij}^{ge} &= \sigma_{ij}^e, \\ \Sigma_{ij}^{gu} &= \sigma_{ij}^u, \\ P_{ij}^g &= \rho_{ij}\end{aligned} \tag{5}$$

If all nodes in g have the same upstream groups, we say that g is multivariate-separable as all the residuals of the linear regression with sample weight w on the time series (the unexplained part) in g are multivariate distributed; and otherwise multivariate-unseparable. This property will be of great use in prediction of unexplained covariance matrix discussed in 3.4.

3.3 Posting opinions

Based on the proposed model, we can post opinions to see the impact. According to the object of impacts, there are five types of opinions: (1) return, (2) variance, (3) unexplained variance, (4) correlation, and (5) unexplained covariance matrix. Note that for return, variance and unexplained variance, there is only one node with opinion, for correlation there are two nodes with opinion and these two nodes must in the same groups, and for unexplained covariance matrix the object with opinion is a group.

Also, based on the mode of impact, an opinion can either be overwriting the original, scaling, or simply adding or subtracting. As sanity checks, we require: for opinions on variance and unexplained variance of a node, the new total variance and unexplained variance should be greater than 0; for an opinion on the correlation of two nodes i and j in the same groups, for any node k in any group that i and j are both in [Olk81]

$$\rho_{ik}\rho_{jk} - \sqrt{(1-\rho_{ik}^2)(1-\rho_{jk}^2)} \leq \rho_{ij}^* \leq \rho_{ik}\rho_{jk} + \sqrt{(1-\rho_{ik}^2)(1-\rho_{jk}^2)} \tag{6}$$

In our experiement, there were several instances of violations for some of the checks. As we did not have an unbiased fix to them, we decided to ignore the occasional violations. However, if there are too many instances of violations, there might be some issue with the model configuration, which needs to be investigated. One of the potential causes and the corresponding solution for the breach will be discussed in Conclusion section.

Fig 3 shows a simple case of posting an opinion on a single node's return (or variance). We assume that all the upstream nodes and groups and all the attributes of the node other than those specified in the opinion remain unchanged.

The following is the detailed description of posting the different types of opinions:

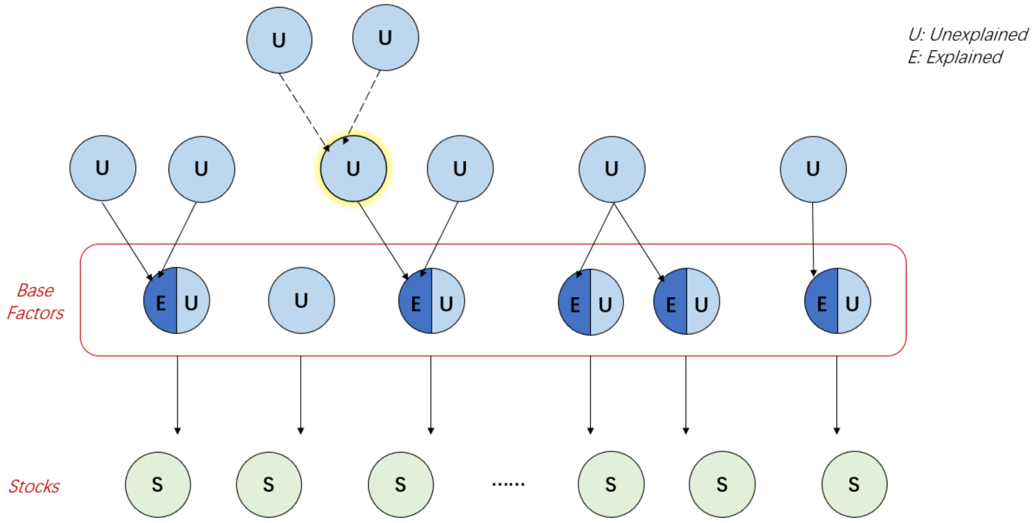

Figure 3: Posting an opinion on a node's return

- *Opinion on return:*

  Let us post an opinion to the return of node k, and assume the new total return of k becomes $f_k = f_k^*$. Since now the information completely comes from the opinion, we have $f_k^e = 0$ and $f_k^u = f_k^*$.

  Since the return of node k is changed, the explained part of the returns of all k's downstream nodes should be recalculated, while the unexplained part remains unchanged. Thus, for any node p in the downstream of k

  $$f_p = f_p^{*e} + f_p^u \tag{7}$$

  where f_p^{*e} is the recalculated explained part of node p.

- *Opinion on variance:*

  Now, if the opinion is on the variance of k, the new total variance of k becomes $\sigma_k^2 = (\sigma_k^*)^2$. Similar to return, $\sigma_k^e = 0$ and $\sigma_k^u = \sigma_k^*$.

  Since the variance of node k is changed, all the covariance matrices of groups that k involved in are updated, for $g \in G_k^{in}$ we have

  $$\begin{aligned} \Sigma_{kk}^g = \Sigma_{kk}^{gu} = (\sigma_k^*)^2, \Sigma_{kk}^{ge} = 0 \\ \Sigma_{jk}^g = \Sigma_{kj}^g = \Sigma_{jk}^{gu} = \Sigma_{kj}^{gu} = \rho_{ij}\sigma_j\sigma_k^* \\ \Sigma_{jk}^{ge} = \Sigma_{kj}^{ge} = 0 \quad \forall j \in g, j \neq k \end{aligned} \tag{8}$$

  After that, similar to returns, the explained part of the variances of all k's downstream nodes will be recalculated while the unexplained part remains unchanged. Also, the covariance matrices of the groups these downstream nodes involved in are also updated.

- *Opinion on unexplained variance:*

  If the opinion is on the unexplained variance of k and we assume the new unexplained variance of k is σ_k^{*u}, then we have $\sigma_k^u = \sigma_k^{*u}$, σ_k^e remains unchanged, and $\sigma_k^2 = (\sigma_k^e)^2 + (\sigma_k^{*u})^2$.

  For $g \in G_k^{in}$ we have

  $$\begin{aligned} \Sigma_{kk}^g = (\sigma_k^{*u})^2 + \sigma_k^2, \\ \Sigma_{kk}^{ge} = (\sigma_k^e)^2, \\ \Sigma_{kk}^{gu} = (\sigma_k^{*u})^2 \end{aligned} \tag{9}$$

For k and $j \in g$ ($j \neq k$), note that the explained covariance σ^e_{kj} remains unchanged. Our key assumption here is that the correlation between the unexplained portions remains unchanged, such that

$$\rho^u_{kj} \equiv \frac{\sigma^u_{kj}}{\sigma^u_k \sigma^u_j} \tag{10}$$

Hence the new unexplained and total covariance can be calculated by

$$\begin{aligned} \sigma^u_{kj} = \sigma^{*u}_{kj} = \rho^u_{kj}\sigma^u_j\sigma^{*u}_k, \\ \sigma_{kj} = \sigma^{*u}_{kj} + \sigma^e_{kj} \end{aligned} \tag{11}$$

Note that the correlation also needs to be updated. We denote the updated covariance matrix by Σ^{*g}, and then the correlation between k and j becames

$$\rho_{kj} = \frac{\Sigma^{*g}_{kj}}{\sqrt{\Sigma^{*g}_{kk}\Sigma^{*g}_{jj}}} \tag{12}$$

Similar to opinions on variance, we then update the downstream groups and nodes.

- *Opinion on correlation:*

  For the case where the opinion is on the correlation of two nodes k and l in the same groups, covariance matrices of all the groups both k and l are in are updated. For any group g both k and l are in, denote the new correlation of k and l by ρ^*_{kl}, such that

$$\begin{aligned} \Sigma^g_{kl} = \Sigma^g_{lk} = \Sigma^{gu}_{kl} = \Sigma^{gu}_{lk} = \rho^*_{kl}\sigma_k\sigma_l, \\ \Sigma^{ge}_{kl} = \Sigma^{ge}_{lk} = 0 \end{aligned} \tag{13}$$

  Thus, the variances of the downstream nodes and the covariance / correlation matrices of the downstream groups are also updated.

- *Opinion on unexplained covariance matrix:*

  Our last case is to post an opinion on the unexplained covariance matrix of a group g. In this case, assume the new unexplained covariance matrix of g is $\Sigma^{gu} = \Sigma^{*gu}$, then we have Σ^{ge} unchanged, and the total covariance matrix becomes $\Sigma^g = \Sigma^{*g} = \Sigma^{ge} + \Sigma^{*gu}$. Note that the correlation matrix also needs to be updated:

$$\rho_{ij} = \frac{\Sigma^{*g}_{ij}}{\sqrt{\Sigma^{*g}_{ii}\Sigma^{*g}_{jj}}} \qquad \forall i, j \in g \tag{14}$$

  Since the covariance matrix is updated, the variances of all the nodes in g also need to be changed. For $k \in g$,

$$(\sigma^u_k)^2 = \Sigma^{gu}_{kk}, \sigma^2_k = \Sigma^{gu}_{kk} + (\sigma^e_k)^2 \tag{15}$$

  Since the variance of k is updated, covariance matrix of all groups in G^{in}_k should also be updated. Furthermore, we update all the variances of downstream nodes, the covariance matrices of downstream groups of g, and all other groups containing the nodes in g.

Rather than posting only one opinion, we can also post sequential opinions in the same net. In order to avoid the situation that the opinion impact on the downstream is overwritten by an upstream impact in case an opinion on an upstream node or group was posted after an opinion on a downstream one, we apply them in the spanning order, which is the order of the directed graph.

For the validity checking in the situation of posting more than one opinion, we only require the checking regarding correlation and covariance to hold in the final state, as in the middle of execution of these opinions, there may be intermediate violations such as correlation infeasibility or negative covariance while the end result may still be valid.

3.4 Variance / covariance prediction

In this subsection, we discuss the prediction of variances, covariances, and correlations without GARCH, with univariate GARCH in nodes' unexplained variance prediction, and with multivariate GARCH in groups' unexplained covariance matrix prediction.

For the net without GARCH, the prediction is simply done by rolling the regression window and recalculating. However, GARCH can be applied to increase the prediction accuracy.

In the simpler case, nets with univariate GARCH, we post an overwrite opinion on the unexplained variance of a node. We conduct univariate GARCH on the unexplained part of the node's time series and form the opinion using the univariate GARCH prediction.

Recall the multivariate-(un)separable property of groups defined in 3.2. If a group g is multivariate-separable, the unexplained part of g is considered multivariate. Thus, for the net with multivariate GARCH, we conduct multivariate GARCH on the groups that are multivariate-separable and post overwrite opinion on the unexplained covariance matrix. For the groups that are multivariate-unseparable, we go through the same process as univariate GARCH.

3.5 Portfolio allocation optimization

In solving the optimization problem of portfolio allocation, we use python *cvxpy* package. The optimization problem is

$$\begin{aligned} max_x \quad & \mu^T x \\ s.t. \quad & x^T V x \leq \sigma^2 \\ & e^T x = 1 \end{aligned} \tag{16}$$

where we change μ to f and change V to Σ to be consistent with our paper, and x is the vector of stock position weights.

To speed up the searching process while pertaining adequate solution accuracy, we adopt the following algorithm. We first try to solve the problem with a target volatility that is some constant $c < 1$ multiplied by the lowest single stock volatility. If the constraint is infeasible, the target volatility is multiplied by a constant $\lambda > 1$, and the process is repeated until a feasible solution is found, which is the minimum variance portfolio. If the constraint is feasible at the beginning, we divide the target volatility by λ to set a lower target and repeat the process until the constraint is infeasible, in which case the last feasible solution represents the minimum variance portfolio. Once the minimum variance portfolio is found, repeat multiplying λ to the target volatility and record the portfolio Sharpe ratio. When the Sharpe ratio just starts to decrease, the last portfolio with the largest Sharpe ratio represents the tangency portfolio.

The following pseudo code summarizes the algorithm [Dia]:

Algorithm 1 Optimize Portfolios Pseudocode

```
Input: $\mathrm{f}_{\mathrm{stocks\_predict}}, \Sigma_{\mathrm{stocks\_predict}}, r_{\mathrm{risk\_free\_rate}}$
flag_infeasible_searched = False                    ▷ whether an infeasible point is searched
flag_feasible = False    ▷ whether the feasible region is reached after an infeasible point is searched
$\sigma^2_{\mathrm{target}} = \mathrm{c} * \min(\sigma^2_{\mathrm{stock\_predict}})$                    ▷ initialize target volatility
while True do
    solution = cvx_optimize ($\mathrm{f}_{\mathrm{stocks\_predict}}, \Sigma_{\mathrm{stocks\_predict}}$)
    if solution is None then                        ▷ infeasible point searched
        flag_infeasible_searched = True
        $\sigma^2_{\mathrm{target}} = \sigma^2_{\mathrm{target}} * \lambda$                    ▷ increase target volatility
        continue
    if not flag_infeasible_searched then            ▷ no infeasible point searched yet
        $\sigma^2_{\mathrm{target}} = \sigma^2_{\mathrm{target}} / \lambda$                    ▷ decrease target volatility
        continue    ▷ an infeasible point has been reached now, and current iteration represents a feasible solution
    $\mathrm{Sharpe}_{\mathrm{solution}} = (\mathrm{f}_{\mathrm{solution}} - \mathrm{r}_{\mathrm{risk\_free\_rate}}) / \sigma_{\mathrm{solution}}$                    ▷ calculate Sharpe ratio
                            ▷ save the first feasible solution to minimum variance portfolios
    if not flag_feasible then
        $\mathrm{portfolio}_{\mathrm{minvar}}$ = solution
        $\mathrm{Sharpe}_{\mathrm{max}} = \mathrm{Sharpe}_{\mathrm{solution}}$                    ▷ initialize maximum Sharpe ratio
        flag_feasible = True                        ▷ update the flag that feasible region is reached
        $\sigma^2_{\mathrm{target}} = \sigma^2_{\mathrm{target}} * \lambda$
        continue
        ▷ minimum variance portfolio has been found at this point, and currently searching for tangency portfolio
    if $\text{Sharpe}_{\text{solution}}$ then $> \text{Sharpe}_{\text{max}}$        ▷ Sharpe ratio still increasing
        $\text{Sharpe}_{\text{max}} = \text{Sharpe}_{\text{solution}}$        ▷ update maximum Sharpe ratio
        $\sigma^2_{\text{target}} = \sigma^2_{\text{target}} * \lambda$
    else        ▷ Sharpe ratio starts to decrease
        $\text{portfolio}_{\text{tangency}} = \text{solution}_{\text{previous}}$        ▷ save the previous solution to tangency portfolio
        return $\text{portfolio}_{\text{minvar}}, \text{portfolio}_{\text{tangency}}$
```

4 Experiment

In this section, we will discuss the details of our empirical experiment, with the focus on testing the volatility prediction with the proposed graph-based factor model with GARCH.

4.1 Data and configurations

We obtained the daily return data of factors and stocks from January 2014 to December 2021 detailed in Table 1 and 2 from *Yahoo Finance*.

For the weighted regression, as discussed in section 3, we set the exponential discount rate to be 0.99, and the length of our training window is 63 trading days (3 months).

In our experiment, we used python *arch* package [She] for univariate GARCH fitting with a configuration of GARCH(1, 0, 1) under t-distribution. We used python *mgarch* package [Sri] for multivariate GARCH fitting with a configuration of DCC-GARCH(1, 1) under t-distribution.

In our cvx optimization algorithm, we chose c to be 0.8 and λ to be 1.02.

Factor	Ticker	Description
Market	ˆGSPC	S&P500
Finance Sector	XLF	Financial Select Sector SPDR Fund
Technology Sector	IYW	iShares U.S. Technology ETF
Industry Sector	XLI	Industrial Select Sector SPDR Fund
Consumer Sector	IYC	iShares US Consumer Discretionary ETF

Table 1: Factors selected

Company	Ticker	Sector
Apple Inc.	AAPL	Technology
Alphabet Inc.	GOOG	Technology
General Electric Company	GE	Industrials
Caterpillar Inc.	CAT	Industrials
McDonald's Corporation	MCD	Consumer
Walmart Inc.	WMT	Consumer
The Goldman Sachs Group, Inc.	GS	Finance
Citigroup Inc.	C	Finance

Table 2: Stocks selected

4.2 Model setups

In order to assess the performance of our proposed methodology (denote by Mp), we compared portfolios constructed by the graph-based model shown in Fig 4 to the portfolios constructed by several benchmark models. The following is the whole list of the models we studied:

- **M1: Model of stocks only without GARCH prediction**

  The base model, used as the benchmark for relative performance analysis.

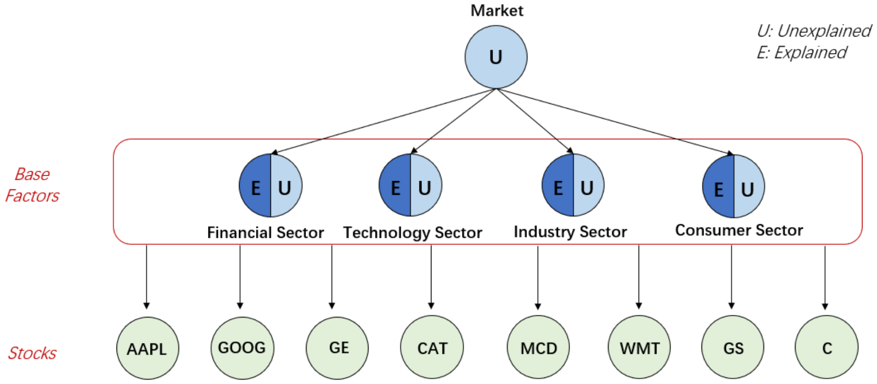

Figure 4: Proposed graph-based factor model (Mp)

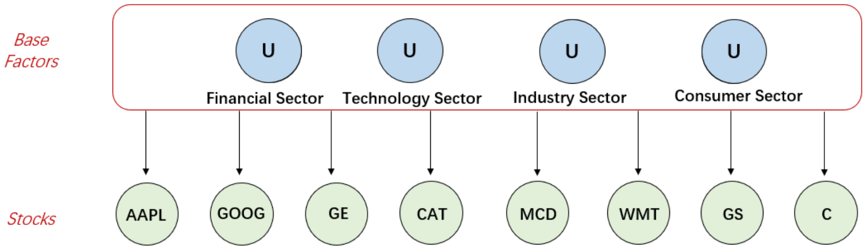

Figure 5: Factor model with base group only (M4)

- **M2/M3: Model of stocks only with uni/multivariate GARCH**

  To test the effect of uni/multivariate GARCH prediction.

- **M4: Model of stocks and base factors with univariate GARCH on base group**

  To test the effect of factors, structured as in Fig 5. And by comparing Mp to M4, we can test the effect of the graph structure.

- **M5: graph-based model of stocks and factors with univariate GARCH on all stocks and factors**

  M4 share the same structure as Mp as shown in Fig 4, different in that it also conduct GARCH prediction on stock nodes.

- **Mp: graph-based model of stocks and factors with univariate GARCH on factors**

  The proposed model, structured as in Fig 4.

4.3 Empirical results

In this subsection, we report the tangency/minimum variance portfolio performance by providing the both independent and relative results (with M1 as the benchmark model), detailed in Table 3-6 and Fig 6-9. We also calculated each model's correlation coefficient between the daily rank of predicted stock volatilities and the rank of absolute value of actual stock returns (volatility rank IC) and computational complexity, reported in Table 7-8. Moreover, in order to ensure that the good performance of the proposed portfolio is not owing to a single stock, we analyzed the portfolios' leverage and stock concentration, detailed in Table 9-12. The best one of each statistic is in red, and the second best one is in blue.

Table 3 and 4 highlights that the tangency portfolio constructed by the proposed graph-based model performed well on both independent and relative basis. With regards to tangency portfolio, Mp outperforms all the benchmark models in total return (Fig 6 also indicates that the outperformance is consistent), annualized return, Sharpe ratio and maximum drawdown. In relative analysis, tangency portfolio constructed by Mp also obtained the highest total active return, annualized active return, and information ratio.

Table 5 shows that the minimum variance portfolio constructed by Mp performs reasonably well. This portfolio obtained the second lowest volatility (and the gap between it and the one with the lowest volatility (M1) is not large)

Model	Total return	Annualized return	Volatility	Sharpe	Max drawdown
M1	229.33%	16.65%	21.58%	0.74	-32.83%
M2	241.21%	17.18%	20.53%	0.80	-25.18%
M3	241.20%	17.18%	18.73%	0.88	-32.27%
M4	318.05%	20.30%	22.51%	0.87	-27.26%
M5	206.29%	15.56%	21.61%	0.69	-23.03%
Mp	358.93%	21.76%	22.13%	0.95	-20.02%

Table 3: Tangency portfolio independent performance

Model	Total active return	Annualized active return	Information Ratio	β	Annualized α
M2	11.88%	0.54%	0.04	0.78	4.11%
M3	11.87%	0.53%	0.03	0.55	7.71%
M4	88.72%	3.65%	0.36	0.93	4.74%
M5	-23.05%	-1.09%	-0.10	0.87	0.99%
Mp	129.59%	5.11%	0.52	0.92	6.37%

Table 4: Tangency portfolio relative performance

Model	Total return	Annualized return	Volatility	Sharpe	Max drawdown
M1	169.97%	13.69%	17.20%	0.75	-19.84%
M2	250.19 %	17.58%	17.78%	0.95	-21.63%
M3	191.90%	14.84%	17.98%	0.79	-32.29%
M4	242.51%	17.24%	18.07%	0.91	-21.32%
M5	216.36%	16.04%	17.80%	0.86	-21.83%
Mp	244.86%	17.34%	17.70%	0.94	-19.78%

Table 5: Minimum variance portfolio independent performance

Model	Total active return	Annualized active return	Information Ratio	β	Annualized α
M2	88.22%	3.89%	0.41	0.88	5.41%
M3	21.92%	1.15%	0.09	0.78	4.07%
M4	72.54%	3.55%	0.48	0.96	4.06%
M5	46.38%	2.35%	0.35	0.96	2.86%
Mp	74.89%	3.65%	0.61	0.97	4.06%

Table 6: Minimum variance portfolio relative performance

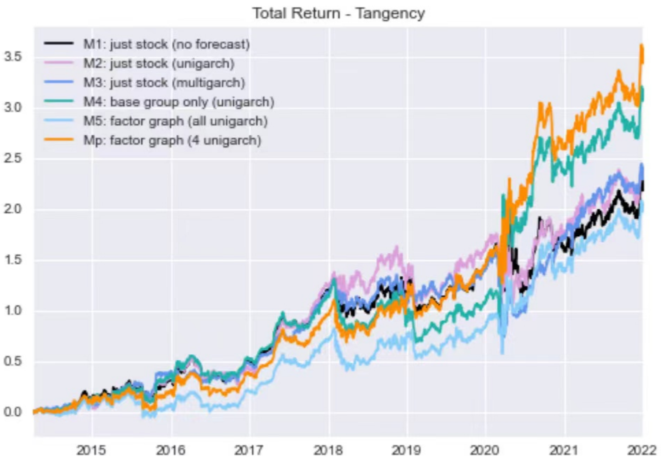

Figure 6: Total return of tangency portfolios

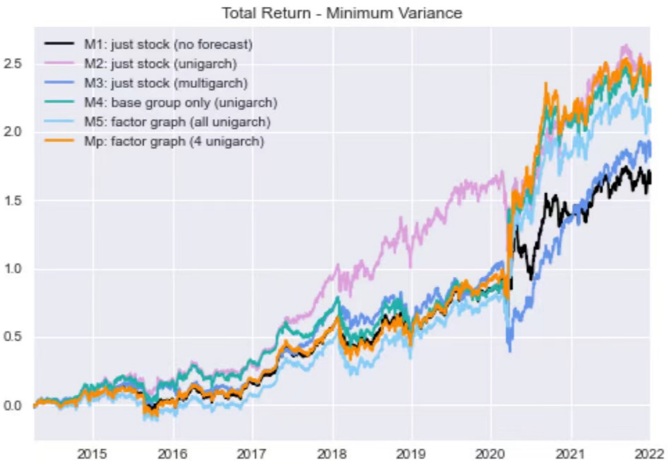

Figure 7: Total return of minimum variance portfolios

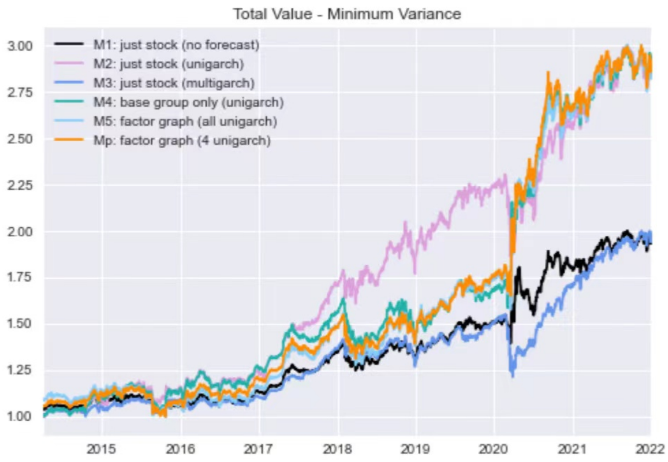

Figure 8: Total value of tangency portfolios

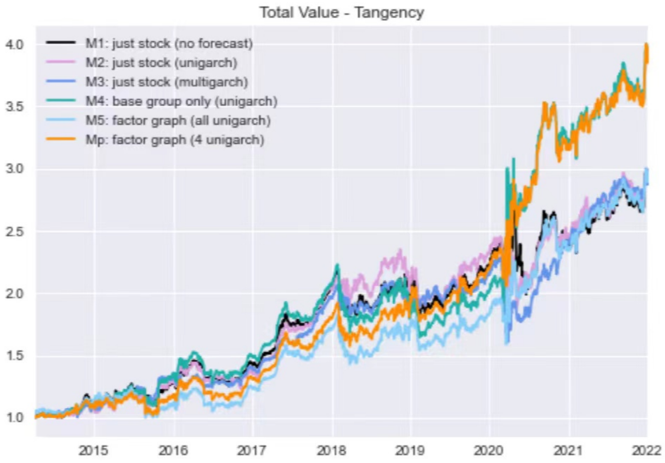

Figure 9: Total value of minimum variance portfolios

Model	M1	M2	M3	M4	M5	Mp
Volatility Rank IC	0.03761	0.03871	0.04515	0.028319	0.04382	0.041104

Table 7: Volatility rank IC

and the smallest maximum drawdown among all the models. Volatility rank IC as in Table 7 also demonstrated the ability of Mp to outperform the benchmark M1 in terms of stock volatility predictions.

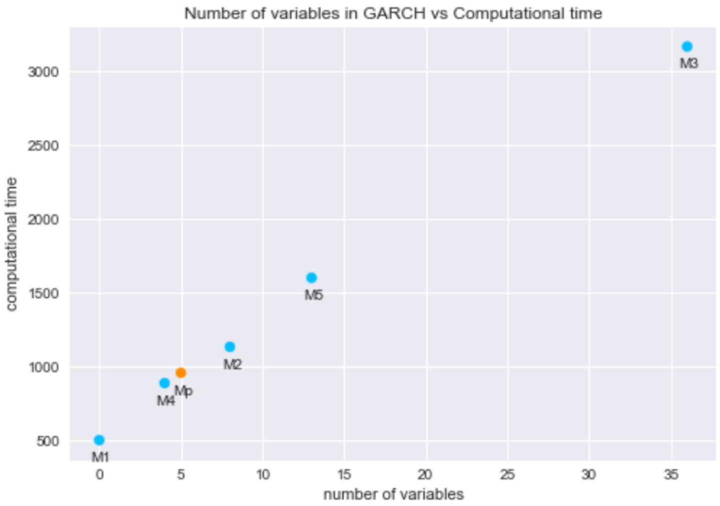

Figure 10: Number of variables in GARCH vs Computational time

Model	M1	M2	M3	M4	M5	Mp
# of nodes with GARCH	0	8	8 (multi, 36 variables)	4	13	5
Computation time (s)	501.79246	1133.95895	3168.23592	888.26300	1601.43049	958.14967

Table 8: Computational complexity

Note that from Table 5-6, Mp's performance is sometimes surpassed by M2. But when computational cost is taken into consideration (as shown in Table 8 and Fig 10), the proposed model Mp achieves a better balance between performance and complexity.

From Table 9-12, we can see that both the leverage and stock concentration of our proposed model are within the

Model	M1	M2	M3	M4	M5	Mp
Leverage	1.86405	1.81014	1.14760	1.81342	1.86131	1.82655

Table 9: Average leverage (total asset over equity) of tangency portfolios

Model	M1	M2	M3	M4	M5	Mp
Leverage	1.42686	1.44056	1.02124	1.40299	1.44339	1.42883

Table 10: Average leverage of minimum variance portfolios

model/stock	AAPL	GOOG	GE	CAT	MCD	WMT	GS	C
M1	0.21133	0.20111	0.16442	0.17335	0.39871	0.29238	0.20901	0.21370
M2	0.20790	0.19588	0.18655	0.15601	0.36876	0.27059	0.21792	0.20655
M3	0.13614	0.12959	0.10475	0.10102	0.25868	0.21923	0.10523	0.09293
M4	0.21796	0.21479	0.15429	0.16811	0.37444	0.27495	0.20047	0.20839
M5	0.22106	0.20088	0.14975	0.17510	0.39469	0.30278	0.21044	0.20659
Mp	0.21329	0.19858	0.15253	0.16939	0.39750	0.28477	0.20672	0.20376

Table 11: Stock concentration (mean absolute weight) of tangency portfolios

model/stock	AAPL	GOOG	GE	CAT	MCD	WMT	GS	C
M1	0.11569	0.13652	0.12829	0.10193	0.38159	0.30631	0.11443	0.14212
M2	0.12676	0.13786	0.15940	0.09535	0.34595	0.27675	0.14570	0.15279
M3	0.09727	0.10736	0.10216	0.07591	0.24903	0.22175	0.08539	0.08236
M4	0.12435	0.14901	0.12710	0.10501	0.35724	0.28363	0.11574	0.14089
M5	0.12370	0.13635	0.12884	0.10557	0.37125	0.31476	0.12119	0.14173
Mp	0.12187	0.14201	0.12821	0.10433	0.37794	0.29594	0.12165	0.13689

Table 12: Stock concentration of minimum variance portfolios

reasonable range, thus we can safely conclude that the good performance of Mp is not because of sticking to a stock that did particularly well.

Regarding the tests stated in 4.3, we find that M2/M3 consistently outperforms M1, thereby yielding the effectiveness of uni/multivariate GARCH prediction. M4 gives better performance than M2/3 (occasionally overtaken by M2 but overall better considering the complexity), demonstrating the usefulness of the factor model. Comparing Mp and M5, we can see that Mp is always better, showing that conducting GARCH prediction only on factors is sufficient, and the performance would be better. M5 may have faced the overfitting issue. Through this analysis, we are able to state that the proposed graph-based factor model is effective and efficient.

5 Conclusion

In the paper, we have presented the graph-based factor model in the application of portfolio optimization. Extended from the ordinary factor model, our model focuses on not only the relationship between factors and stocks, but also the interactions among the factors themselves. In the model, upstream nodes represent the driving factors, and the downstream nodes represent the dependent variables that behave accordingly. The model can be treated not only as a statistical tool but also as a highly flexible framework describing the broader financial and economic activities, where subjective discretion can be involved in model configuration and posting opinions, emphasizing the importance of fundamental reasoning in the portfolio allocation process. The experiment results have demonstrated the efficacy of the model.

The model can be further explored in the following aspects:

1. In the model, the factor loadings are from weighted historic time series regression. While this is hysteretic, the returns and covariance matrices with opinions are for future predictions, creating an imbalance in time horizon. Although the factor loadings are considered relatively stable in the long run, the assumption may not hold in some circumstances such as extreme market events or fundamental changes in the business nature of a company. This could be one of the reasons for the check breaches discussed in Section 3.3. There could be two potential

ways to tackle this. The first is to construct a conditional model and enable opinions on the factor loadings. The challenge is that unlike returns or covariances, factor loadings are more difficult to estimate with methods other than regression, especially in multivariable linear models. The second solution is to use time series of higher frequency for regression to calculate the factor loadings only. The benefit is that factor loadings can be estimated based on more recent data, but the potential problem could be that the mismatch between the factor frequency and coefficient frequency could lead to unreliable results, as well as that data may be unavailable for the factors of higher frequencies. Nevertheless, it is worth studying the methods.

2. As a hyperparameter, the factor graph design is currently entirely discretionary. It is crucial to study the principles to increase the explanatory and predictive power of a graph design. For example, some statistical methods such as Gaussian graph-based model can be used to determine the conditional dependence among the nodes in model configuration [Nit19].

3. Although some nonlinearity can be involved in designing the individual factors, the model is still primarily linear. To capture the fundamental principles more precisely, the relationships between nodes can be generalized to include nonlinearity, in which the related operations also need to be redefined.

4. The causation between the upstream and downstream nodes implies time dependence of the changes, which is consistent with how the economy and markets evolve. Therefore, it is interesting to integrate time series into the framework. By doing so, cycles can be introduced into the directed graph to represent the mutual influences of the factors and the stocks, which is one step closer to the reality. =D

References

[AFP19] Cliff Asness, Andrea Frazzini, and Lasse Pedersen. “Quality Minus Junk”. In: *Rev Account Stud* 24 (2019), pp. 34–112.

[Dia] Steven Diamond. *CVXPY 1.2.* `https://www.cvxpy.org/`. Accessed: 2022-03-30.

[FF15] Eugene Fama and Kenneth French. “A Five-Factor Asset Pricing Model”. In: *Journal of Financial Economics* 116 (2015), pp. 1–22.

[FF93] Eugene Fama and Kenneth French. “Common risk factors in the returns on stocks and bonds”. In: *Journal of Financial Economics* 33 (1993), pp. 3–56.

[FP14] Andrea Frazzini and Lasse Pedersen. “Betting Against Beta”. In: *Journal of Financial Economics* 111.1 (2014), pp. 1–25.

[Mar52] Harry Markowitz. “Portfolio Selection”. In: *The Journal of Finance* 7.1 (1952), pp. 77–91.

[Nit19] Bhushan Nitin. “Using a Gaussian Graphical Model to Explore Relationships Between Items and Variables in Environmental Psychology Research”. In: *Frontiers in Psychology* 10 (2019).

[Olk81] Ingram Olkin. “Range restrictions for product-moment correlation matrices”. In: *Psychometrika* 46.4 (1981), pp. 469–472.

[Ros76] Stephen Ross. “The Arbitrage Theory of Capital Asset Pricing”. In: *Journal of Economic Theory* 13 (1976), pp. 341–360.

[Sha64] William Sharpe. “CAPITAL ASSET PRICES: A THEORY OF MARKET EQUILIBRIUM UNDER CONDITIONS OF RISK”. In: *The Journal of Finance* 19.3 (1964), pp. 425–442.

[She] Kevin Sheppard. *Introduction to ARCH Models.* `https://arch.readthedocs.io/en/latest/univariate/introduction.html`. Accessed: 2022-03-30.

[SL22] Bumho Son and Jaewook Lee. “Graph-based multi-factor asset pricing model”. In: *Finance Research Letters* 44 (2022).

[Sri] Prashant Srivastava. *mgarch 0.2.0.* `https://pypi.org/project/mgarch/`. Accessed: 2022-03-30.