

# Combining Forecasts under Structural Breaks Using Graphical LASSO

Tae-Hwy Lee\* and Ekaterina Seregina†

September 23, 2023

## Abstract

In this paper we develop a novel method of combining many forecasts based on a machine learning algorithm called Graphical LASSO (GL). We visualize forecast errors from different forecasters as a network of interacting entities and generalize network inference in the presence of common factor structure and structural breaks. First, we note that forecasters often use common information and hence make common mistakes, which makes the forecast errors exhibit common factor structures. We use the Factor Graphical LASSO (FGL, Lee and Seregina (2023)) to separate common forecast errors from the idiosyncratic errors and exploit sparsity of the precision matrix of the latter. Second, since the network of experts changes over time as a response to unstable environments such as recessions, it is unreasonable to assume constant forecast combination weights. Hence, we propose Regime-Dependent Factor Graphical LASSO (RD-FGL) that allows factor loadings and idiosyncratic precision matrix to be regime-dependent. We develop its scalable implementation using the Alternating Direction Method of Multipliers (ADMM) to estimate regime-dependent forecast combination weights. The empirical application to forecasting macroeconomic series using the data of the European Central Bank's Survey of Professional Forecasters (ECB SPF) demonstrates superior performance of a combined forecast using FGL and RD-FGL.

*Keywords:* Common Forecast Errors, Regime Dependent Forecast Combination, Sparse Precision Matrix of Idiosyncratic Errors, Structural Breaks.

*JEL Classifications:* C13, C38, C55

---

\*Department of Economics, University of California Riverside. Email: tae.lee@ucr.edu.

†Department of Economics, Colby College. Email: eseregin@colby.edu.

# 1 Introduction

A search for the best forecast combination has been an important on-going research question in economics. [Clemen \(1989\)](#) pointed out that combining forecasts is “practical, economical and useful. Many empirical tests have demonstrated the value of composite forecasting. We no longer need to justify that methodology”. However, as demonstrated by [Diebold and Shin \(2019\)](#), there are still some unresolved issues. Despite the findings based on the theoretical grounds, equal-weighted forecasts have proved surprisingly difficult to beat. Many methodologies that seek for the best forecast combination use equal weights as a benchmark: for instance, [Diebold and Shin \(2019\)](#) develop “partially egalitarian LASSO”.

The success of equal weights is partly due to the fact that the forecasters use the same set of public information to make forecasts, hence, they tend to make common mistakes. For example, in the ECB SPF of Euro-area real GDP growth, the forecasters tend to *jointly* understate or overstate GDP growth. Therefore, we stipulate that the forecast errors include common and idiosyncratic components, which allows the forecast errors to move together due to the common error component. Our paper provides a simple framework to learn from analyzing forecast errors: we separate unique errors from the common errors to improve the accuracy of the combined forecast.

Dating back to [Bates and Granger \(1969\)](#), the well-known expression for the optimal forecast combination weights requires an estimator of inverse covariance (precision) matrix. Precision matrix represents a network of interacting entities, such as corporations or genes. When the data is Gaussian, the sparsity in the precision matrix encodes the conditional independence graph - two variables are conditionally independent given the rest if and only if the entry corresponding to these variables in the precision matrix is equal to zero. Graph-

ical models are a powerful tool to directly estimate precision matrix, avoiding the step of obtaining an estimator of covariance matrix to be inverted. Prominent examples of graphical models include GL (Friedman et al. (2008)) and nodewise regression (Meinshausen and Bühlmann (2006)). Despite using different strategies for estimating precision matrix, all graphical models assume that precision matrix is sparse: many entries of precision matrix are zero, which is a necessary condition to consistently estimate inverse covariance. Our paper demonstrates that such assumption contradicts the stylized fact that experts tend to make common mistakes and hence the forecast errors move together through common factors. Lee and Seregina (2023) show that graphical models fail to recover the entries of a nonsparse precision matrix under the factor structure and propose FGL that combines the benefits of graphical models and factor models.

At the same time, the network of experts changes over time, that is, the relationships between forecasts produced by different experts or models can change either smoothly or abruptly (e.g., as a response to an unexpected policy shock, or in the times of economic downturns). Such changes give rise to different regimes and it is important to account for changes in optimal forecast combination weights induced by structural breaks. This paper augments Lee and Seregina (2023) and develops a unified framework to generalize network inference in the presence of structural breaks. As a first extension, we model structural changes in factor loadings. As a second extension, we model structural changes in the precision matrix of the idiosyncratic component after removing common factors. We estimate regime-dependent precision matrix for forecast combination using both pre- and post-break data when forecast errors are driven by common factors. We call the proposed algorithm *Regime-Dependent Factor Graphical LASSO* (RD-FGL) and develop its scalable implementation using the Alternating Direction Method of Multipliers (ADMM).

Our paper makes several contributions. First, we allow the forecast errors to be highly correlated due to the common component which is motivated by the stylized fact that the forecasters tend to jointly understate or overstate the predicted series of interest. [Second](#), to tackle changing relationships between forecasts produced by different experts or models as a response to unstable environments, we develop a unified framework to generalize network inference in the presence of structural breaks. We propose RD-FGL that models structural changes in factor loadings and idiosyncratic precision matrix. We develop scalable implementation of RD-FGL using ADMM to estimate regime-dependent forecast combination weights. Third, an empirical application to forecasting macroeconomic series using the data of the ECB SPF shows that incorporating (i) factor structure in the forecast errors together with (ii) sparsity in the precision matrix of the idiosyncratic components and (iii) regime-dependent combination weights improves the performance of a combined forecast over forecast combinations using equal weights.

We emphasize that in this paper our goal is to develop a framework for forecast combinations that incorporates structural breaks which have already occurred in the past. We neither consider the possibility of breaks over the forecast horizon (as explored in [Pesaran et al. \(2006\)](#)), nor study the case of the out-of-sample breaks. These scenarios are interesting extensions of this paper, however they lie outside the scope of this paper. We also emphasize that we take the individual forecasts to be combined as given and do not discuss how the forecasts are generated.

The paper is structured as follows. Section 2 studies the approximate factor model for the forecast errors. Section 3 reviews FGL and contains theoretical results on the consistency of the FGL estimator for forecast combinations. Section 4 introduces Regime-Dependent graphical model and discusses its implementation using ADMM. Section 5 validates theoret-

ical results using simulations. Section 6 studies an empirical application for macroeconomic time-series forecasting. Section 7 concludes.

**Notation.** For the convenience of the reader, we summarize the notation to be used throughout the paper. Let  $\mathcal{S}_p$  denote the set of all  $p \times p$  symmetric matrices. For any matrix  $\mathbf{C}$ , its  $(i, j)$ -th element is denoted as  $c_{ij}$ . Given a vector  $\mathbf{u} \in \mathbb{R}^d$  and a parameter  $a \in [1, \infty)$ , let  $\|\mathbf{u}\|_a$  denote  $\ell_a$ -norm. Given a matrix  $\mathbf{U} \in \mathcal{S}_p$ , let  $\lambda_{\max}(\mathbf{U}) \equiv \lambda_1(\mathbf{U}) \geq \lambda_2(\mathbf{U}) \geq \dots \geq \lambda_{\min}(\mathbf{U}) \equiv \lambda_p(\mathbf{U})$  be the eigenvalues of  $\mathbf{U}$ . Given a matrix  $\mathbf{U} \in \mathbb{R}^{p \times p}$  and parameters  $a, b \in [1, \infty)$ , let  $\|\|\mathbf{U}\|\|_{a,b} \equiv \max_{\|\mathbf{y}\|_a=1} \|\mathbf{U}\mathbf{y}\|_b$  denote the induced matrix-operator norm. The special cases are  $\|\|\mathbf{U}\|\|_1 \equiv \max_{1 \leq j \leq p} \sum_{i=1}^p |u_{ij}|$  for the  $\ell_1/\ell_1$ -operator norm; the operator norm ( $\ell_2$ -matrix norm)  $\|\|\mathbf{U}\|\|_2^2 \equiv \lambda_{\max}(\mathbf{U}\mathbf{U}')$  is equal to the maximal singular value of  $\mathbf{U}$ . Finally,  $\|\mathbf{U}\|_{\max} \equiv \max_{i,j} |u_{ij}|$  denotes the element-wise maximum. For two sequences  $a_{T,p}$  and  $b_{T,p}$ , we denote  $a_{T,p} \asymp b_{T,p}$  if there exist constants  $c_1, c_2 > 0$  such that  $c_1 a_{T,p} \leq b_{T,p} \leq c_2 a_{T,p}$ .

## 2 Approximate Factor Models for Forecast Errors

The approximate factor models for the forecasts were first considered by [Chan et al. \(1999\)](#). They modeled a panel of ex-ante forecasts of a single time-series as a dynamic factor model and found out that the combined forecasts improved on individual ones when all forecasts have the same information set (up to difference in lags). This result emphasizes the benefit of forecast combination even when the individual forecasts are not based on different information and, therefore, do not broaden the information set used by any one forecaster.

In this paper, we are interested in finding the combination of forecasts which yields the best out-of-sample performance in terms of the mean-squared forecast error. We claim that the forecasters use the same set of public information to make forecasts and hence they tend

to make common mistakes. Figure 1 illustrates this statement: it shows quarterly forecasts of Euro-area real GDP growth produced by the ECB SPF from 1999Q3 to 2019Q3. As described in Diebold and Shin (2019), forecasts are solicited for one year ahead of the latest available outcome: e.g., the 2007Q1 survey asked the respondents to forecast the GDP growth over 2006Q3-2007Q3. As evidenced from Figure 1, forecasters tend to jointly understate or overstate GDP growth, meaning that their forecast errors include common and idiosyncratic parts. Therefore, we can model the tendency of the forecast errors to move together via factor decomposition.

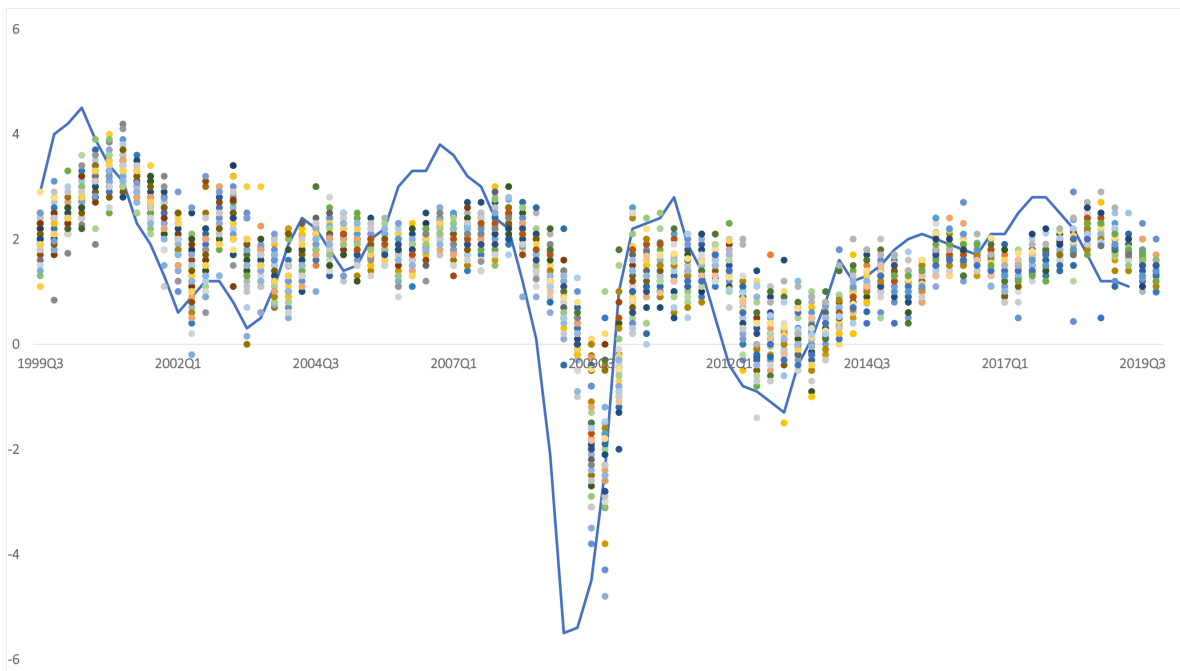


Figure 1: **The European Central Bank’s (ECB) Survey of Professional Forecasters (SPF)**. Each circle denotes the forecast of each professional forecaster in the SPF for the quarterly 1-year-ahead forecasts of Euro-area real GDP growth, year-on-year percentage change. Actual series is the blue line. *Source: European Central Bank.*

Suppose we have  $p$  competing forecasts of the univariate series  $y_t$ ,  $t = 1, \dots, T$  and  $\tilde{\mathbf{e}}_t = (\tilde{e}_{1t}, \dots, \tilde{e}_{pt})' \sim \mathcal{N}(\mathbf{m}, \Sigma)$  is a  $p \times 1$  vector of forecast errors. Note that we allow bias

in the forecasts. In fact, Figure 1 demonstrates that the individual forecasts are indeed biased. Assume that the generating process for the forecast errors follows a  $q$ -factor model:  $\tilde{\mathbf{e}}_t = \mathbf{m} + \mathbf{B}\mathbf{f}_t + \boldsymbol{\varepsilon}_t$ , where  $\mathbf{f}_t = (f_{1t}, \dots, f_{qt})'$  are the common factors of the forecast errors for  $p$  models,  $\mathbf{B}$  is a  $p \times q$  matrix of factor loadings, and  $\boldsymbol{\varepsilon}_t$  is the idiosyncratic component that cannot be explained by the common factors. Define demeaned forecast errors as  $\mathbf{e}_t \equiv \tilde{\mathbf{e}}_t - \mathbf{m}$  such that:

$$\underbrace{\mathbf{e}_t}_{p \times 1} = \mathbf{B} \underbrace{\mathbf{f}_t}_{q \times 1} + \boldsymbol{\varepsilon}_t, \quad t = 1, \dots, T, \quad (2.1)$$

Unobservable factors,  $\mathbf{f}_t$ , and loadings,  $\mathbf{B}$ , are usually estimated by the principal component analysis (PCA), studied in Bai and Ng (2002); Stock and Watson (2002). Strict factor structure assumes that the idiosyncratic forecast error terms,  $\boldsymbol{\varepsilon}_t$ , are uncorrelated with each other, whereas approximate factor structure allows correlation of the idiosyncratic components (Chamberlain and Rothschild (1983)).

We use the following notations:  $\mathbb{E}[\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t'] = \boldsymbol{\Sigma}_\varepsilon$ ,  $\mathbb{E}[\mathbf{f}_t \mathbf{f}_t'] = \boldsymbol{\Sigma}_f$ , and  $\mathbb{E}[\mathbf{e}_t \mathbf{e}_t'] = \boldsymbol{\Sigma} = \mathbf{B}\boldsymbol{\Sigma}_f \mathbf{B}' + \boldsymbol{\Sigma}_\varepsilon$ . Let  $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ ,  $\boldsymbol{\Theta}_\varepsilon = \boldsymbol{\Sigma}_\varepsilon^{-1}$  and  $\boldsymbol{\Theta}_f = \boldsymbol{\Sigma}_f^{-1}$  be the precision matrices of forecast errors, idiosyncratic and common components respectively. The objective function to recover factors and loadings from (2.1) is:

$$\min_{\mathbf{f}_1, \dots, \mathbf{f}_T, \mathbf{B}} \frac{1}{T} \sum_{t=1}^T (\mathbf{e}_t - \mathbf{B}\mathbf{f}_t)' (\mathbf{e}_t - \mathbf{B}\mathbf{f}_t), \quad \text{s.t. } \mathbf{B}'\mathbf{B} = \mathbf{I}_q, \quad (2.2)$$

where the constraint is necessary for the unique identification of factors. Fixing the value of  $\mathbf{B}$ , we can project forecast errors  $\mathbf{e}_t$  into the space spanned by  $\mathbf{B}$ :  $\mathbf{f}_t = (\mathbf{B}'\mathbf{B})^{-1} \mathbf{B}'\mathbf{e}_t = \mathbf{B}'\mathbf{e}_t$ .

When combined with (2.2), this yields a concentrated objective function for  $\mathbf{B}$ :

$$\max_{\mathbf{B}} \operatorname{tr} \left[ \mathbf{B}' \left( \frac{1}{T} \sum_{t=1}^T \mathbf{e}_t \mathbf{e}_t' \right) \mathbf{B} \right]. \quad (2.3)$$

It is well-known (see [Stock and Watson \(2002\)](#) among others) that  $\widehat{\mathbf{B}}$  estimated from the first  $q$  eigenvectors of  $\frac{1}{T} \sum_{t=1}^T \mathbf{e}_t \mathbf{e}_t'$  is the solution to (2.3). Given a sample of the estimated residuals  $\{\widehat{\boldsymbol{\varepsilon}}_t = \mathbf{e}_t - \widehat{\mathbf{B}}\widehat{\mathbf{f}}_t\}_{t=1}^T$  and the estimated factors  $\{\widehat{\mathbf{f}}_t\}_{t=1}^T$ , let  $\widehat{\boldsymbol{\Sigma}}_\varepsilon = (1/T) \sum_{t=1}^T \widehat{\boldsymbol{\varepsilon}}_t \widehat{\boldsymbol{\varepsilon}}_t'$  and  $\widehat{\boldsymbol{\Sigma}}_f = (1/T) \sum_{t=1}^T \widehat{\mathbf{f}}_t \widehat{\mathbf{f}}_t'$  be the sample counterparts of the covariance matrices.

Moving forward to the forecast combination exercise, suppose we have  $p$  competing forecasts,  $\widehat{\mathbf{y}}_t = (\widehat{y}_{1,t}, \dots, \widehat{y}_{p,t})'$ , of the variable  $y_t$ ,  $t = 1, \dots, T$ . The forecast combination is defined as  $\widehat{y}_t^c = \mathbf{w}'\widehat{\mathbf{y}}_t$ , where  $\mathbf{w}$  is a  $p \times 1$  vector of weights. Define a measure of risk  $\text{MSFE}(\mathbf{w}, \boldsymbol{\Sigma}) = \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}$ . As shown in [Bates and Granger \(1969\)](#), the *optimal* forecast combination minimizes the MSFE of the combined forecast error:

$$\min_{\mathbf{w}} \text{MSFE} = \min_{\mathbf{w}} \mathbb{E} \left[ \mathbf{w}' \mathbf{e}_t \mathbf{e}_t' \mathbf{w} \right] = \min_{\mathbf{w}} \mathbf{w}' \boldsymbol{\Sigma} \mathbf{w}, \text{ s.t. } \mathbf{w}' \boldsymbol{\iota}_p = 1, \quad (2.4)$$

where  $\boldsymbol{\iota}_p$  is a  $p \times 1$  vector of ones.<sup>1</sup> The solution to (2.4) yields a  $p \times 1$  vector of the optimal forecast combination weights:

$$\mathbf{w} = \frac{\boldsymbol{\Theta} \boldsymbol{\iota}_p}{\boldsymbol{\iota}_p' \boldsymbol{\Theta} \boldsymbol{\iota}_p}. \quad (2.5)$$

If the true precision matrix is known, the equation (2.5) guarantees to yield the optimal forecast combination. In reality, one has to estimate  $\boldsymbol{\Theta}$ . As pointed out by [Smith and Wallis \(2009\)](#), when the estimation uncertainty of the weights is taken into account, there is no

---

<sup>1</sup>As noted in [Timmermann \(2006\)](#), global mean-variance optimization problem in finance is similar to the forecast combination problem. Combined forecasts can be viewed as the portfolio and the source of risk reflects incomplete information about the target variable and model misspecification possibly due to non-stationarities in the underlying data generating process.



guarantee that the “optimal” forecast combination will be better than the equal weights or even improve the individual forecasts. Define  $a = \boldsymbol{\nu}'_p \boldsymbol{\Theta} \boldsymbol{\nu}_p / p$  and  $\hat{a} = \boldsymbol{\nu}'_p \hat{\boldsymbol{\Theta}} \boldsymbol{\nu}_p / p$ . We can write  $\left| \frac{\text{MSFE}(\hat{\mathbf{w}}, \hat{\boldsymbol{\Sigma}})}{\text{MSFE}(\mathbf{w}, \boldsymbol{\Sigma})} - 1 \right| = \left| \frac{\hat{a}^{-1}}{a^{-1}} - 1 \right| = \frac{|a - \hat{a}|}{|\hat{a}|}$  and  $\|\hat{\mathbf{w}} - \mathbf{w}\|_1 = \left[ (a \hat{\boldsymbol{\Theta}} \boldsymbol{\nu}_p) - (a \boldsymbol{\Theta} \boldsymbol{\nu}_p) + (a \boldsymbol{\Theta} \boldsymbol{\nu}_p) - (\hat{a} \boldsymbol{\Theta} \boldsymbol{\nu}_p) \right] / p \cdot (\hat{a} a)$ . Therefore, in order to control the estimation uncertainty in the MSFE and combination weights, one needs to obtain a consistent estimator of the precision matrix  $\boldsymbol{\Theta}$ . More details are discussed in Section 3 and Theorem 1.

### 3 Factor Graphical LASSO for Forecast Errors

Since our interest is in constructing weights for the forecast combination, our goal is to estimate a precision matrix of the forecast errors. This brings us to consider a family of graphical models, which have evolved from the connection between partial correlations and the entries of an adjacency matrix. The adjacency matrix has zero or one in its entries, with a zero entry indicating that two variables are independent conditional on the rest. The adjacency matrix is sometimes referred to as a “graph”. In graphical models, each vertex represents a random variable, and the graph visualizes the joint distribution of the entire set of random variables. *Sparse graphs* have a relatively small number of edges. [Weighted Graphical Lasso \(GL\) procedure \(Friedman et al. \(2008\)\)](#) described in [Supplemental Appendix A](#) is a representative member of graphical models family.

Before estimating precision matrix of forecast errors,  $\boldsymbol{\Theta}$ , we first obtain estimates of factors,  $\hat{\mathbf{f}}_t$ , and factor loadings,  $\hat{\mathbf{B}}$ , using PCA. Second, we obtain  $\hat{\boldsymbol{\Sigma}}_f = \frac{1}{T} \sum_{t=1}^T \hat{\mathbf{f}}_t \hat{\mathbf{f}}_t'$ ,  $\hat{\boldsymbol{\Theta}}_f = \hat{\boldsymbol{\Sigma}}_f^{-1}$ ,  $\hat{\boldsymbol{\epsilon}}_t = \mathbf{e}_t - \hat{\mathbf{B}} \hat{\mathbf{f}}_t$ , and  $\hat{\boldsymbol{\Sigma}}_\epsilon = \frac{1}{T} \sum_{t=1}^T \hat{\boldsymbol{\epsilon}}_t \hat{\boldsymbol{\epsilon}}_t'$ . Third, we note that when common factors are present across the forecast errors, the precision of forecast errors,  $\boldsymbol{\Theta}$ , cannot be sparse because all pairs of the forecast errors are partially correlated given other forecast errors

through the common factors. Therefore, instead of imposing sparsity assumption on  $\Theta$  we require sparsity of the precision matrix of the idiosyncratic errors,  $\Theta_\varepsilon$ .

Once we condition on the common components, it is sensible to assume that many remaining partial correlations of  $\varepsilon_t$  will be negligible and thus  $\Theta_\varepsilon$  is sparse. Hence, it is estimated with the Weighted Graphical Lasso penalty:

$$\widehat{\Theta}_{\varepsilon,\tau} = \arg \min_{\Theta_\varepsilon = \Theta_\varepsilon'} \text{tr}(\mathbf{W}_\varepsilon \Theta_\varepsilon) - \log \det(\Theta_\varepsilon) + \tau \sum_{i \neq j} \widehat{\gamma}_{\varepsilon,ii} \widehat{\gamma}_{\varepsilon,jj} |\theta_{\varepsilon,ij}|, \quad (3.1)$$

initialized with  $\mathbf{W}_\varepsilon = \widehat{\Sigma}_\varepsilon + \tau \mathbf{I}$ , where  $\widehat{\gamma}_{\varepsilon,ii}$  is the  $(i, i)$ -th element of  $\widehat{\Gamma}_\varepsilon^2 \equiv \text{diag}(\mathbf{W}_\varepsilon)$ . The subscript  $\tau$  in  $\widehat{\Theta}_{\varepsilon,\tau}$  means that the solution of the optimization problem in (3.1) will depend upon the choice of the tuning parameter  $\tau$ . In order to simplify notation, we will omit the subscript  $\tau$ .

Finally we put all estimates together using the Sherman-Morrison-Woodbury formula to estimate the precision of forecast errors:

$$\widehat{\Theta} = \widehat{\Theta}_\varepsilon - \widehat{\Theta}_\varepsilon \widehat{\mathbf{B}} [\widehat{\Theta}_f + \widehat{\mathbf{B}}' \widehat{\Theta}_\varepsilon \widehat{\mathbf{B}}]^{-1} \widehat{\mathbf{B}}' \widehat{\Theta}_\varepsilon. \quad (3.2)$$

The aforementioned procedure introduced in Lee and Seregina (2023) is called Factor Graphical LASSO (FGL).<sup>2</sup> It is summarized in Supplemental Appendix B, where we also discuss the choice of the tuning parameter  $\tau$  in (3.1).

We can use  $\widehat{\Theta}$  to estimate the forecast combination weights  $\widehat{\mathbf{w}} = \Theta \boldsymbol{\nu}_p / \boldsymbol{\nu}_p' \widehat{\Theta} \boldsymbol{\nu}_p$ . This approach allows to extract the benefits of modeling common movements in forecast errors,

---

<sup>2</sup>Instead of decomposing covariance matrix into low-rank and idiosyncratic components,  $\mathbb{E}[\mathbf{e}_t \mathbf{e}_t'] = \boldsymbol{\Sigma} = \mathbf{B} \boldsymbol{\Sigma}_f \mathbf{B}' + \boldsymbol{\Sigma}_\varepsilon$ , Shi et al. (2022) regularize  $\boldsymbol{\Sigma}$  directly with the factor structure (more broadly, latent group structure) in mind. FGL approach is different since it focuses on estimating precision matrix directly which is the main focus of the theoretical analysis in Lee and Seregina (2023).

captured by a factor model, and the benefits of using many competing forecasting models that give rise to a high-dimensional precision matrix, captured by a graphical model.

Let us now examine the asymptotic properties of FGL. We first introduce some terminology and notations. Let  $A \in \mathcal{S}_p$ . Define the following set for  $j = 1, \dots, p$ :

$$D_j(A) \equiv \{i : A_{ij} \neq 0, i \neq j\}, \quad d_j(A) \equiv \text{card}(D_j(A)), \quad d(A) \equiv \max_{j=1, \dots, p} d_j(A), \quad (3.3)$$

where  $d_j(A)$  is the number of edges adjacent to the vertex  $j$  (i.e., the *degree* of vertex  $j$ ), and  $d(A)$  measures the maximum vertex degree. Define  $S(A) \equiv \bigcup_{j=1}^p D_j(A)$  to be the overall off-diagonal sparsity pattern, and  $s(A) \equiv \sum_{j=1}^p d_j(A)$  is the overall number of edges contained in the graph. Note that  $\text{card}(S(A)) \leq s(A)$ : when  $s(A) = p(p-1)/2$  this would give a fully connected graph. We now list the assumptions on the model (2.1):

**(A.1)** (Spiked covariance model) Assume that (i) As  $p \rightarrow \infty$ ,  $\lambda_1(\boldsymbol{\Sigma}) > \lambda_2(\boldsymbol{\Sigma}) > \dots > \lambda_q(\boldsymbol{\Sigma}) \gg \lambda_{q+1}(\boldsymbol{\Sigma}) \geq \dots \geq \lambda_p(\boldsymbol{\Sigma}) > 0$ , where  $\lambda_j(\boldsymbol{\Sigma}) = \mathcal{O}(p)$  for  $j \leq q$ , while the non-spiked eigenvalues are bounded, that is,  $c_0 \leq \lambda_j(\boldsymbol{\Sigma}) \leq C_0$ ,  $j > q$  for constants  $c_0, C_0 > 0$ . And assume that (ii)  $\boldsymbol{\nu}'_p \boldsymbol{\Theta} \boldsymbol{\nu}_p / p \geq c > 0$ , where  $c > 0$  is a positive constant.

**(A.2)** (Pervasive factors) There exists a positive definite  $q \times q$  matrix  $\check{\mathbf{B}}$  such that

$$\left\| p^{-1} \mathbf{B}' \mathbf{B} - \check{\mathbf{B}} \right\|_2 \rightarrow 0 \text{ and } \lambda_{\min}(\check{\mathbf{B}})^{-1} = \mathcal{O}(1) \text{ as } p \rightarrow \infty.$$

We also impose strong mixing condition. Let  $\mathcal{F}_{-\infty}^0$  and  $\mathcal{F}_T^\infty$  denote the  $\sigma$ -algebras that are generated by  $\{(\mathbf{f}_t, \boldsymbol{\varepsilon}_t) : t \leq 0\}$  and  $\{(\mathbf{f}_t, \boldsymbol{\varepsilon}_t) : t \geq T\}$  respectively. Define the mixing coefficient  $\alpha(T) = \sup_{A \in \mathcal{F}_{-\infty}^0, B \in \mathcal{F}_T^\infty} |\Pr A \Pr B - \Pr AB|$ .

**(A.3)** (Strong mixing) There exists  $r_3 > 0$  such that  $3r_1^{-1} + 1.5r_2^{-1} + 3r_3^{-1} > 1$ , and  $C > 0$  satisfying, for all  $T \in \mathbb{Z}^+$ ,  $\alpha(T) \leq \exp(-CT^{r_3})$ .

Assumption **(A.1)** divides the eigenvalues into the diverging and bounded ones. This assumption is satisfied by the factor model with pervasive factors, which is stated in Assumption **(A.2)**. We say that a factor is pervasive in the sense that it has non-negligible effect on a non-vanishing proportion of individual time-series. Part (ii) of Assumption **(A.1)** is needed for consistent estimation of the optimal forecast combination weights. Assumptions **(A.1)**-**(A.2)** are crucial for estimating a high-dimensional factor model: they ensure that the space spanned by the principal components in the population level  $\Sigma$  is close to the space spanned by the columns of the factor loading matrix  $\mathbf{B}$ . Assumption **(A.3)** is a technical condition which is needed to consistently estimate the factors and loadings. Assumptions **(A.1)**(i), **(A.2)**, and **(A.3)** are standard assumptions and are used in [Fan et al. \(2013\)](#).

Let  $\Lambda_q = \text{diag}(\lambda_1, \dots, \lambda_q)$  be a matrix of  $q$  leading eigenvalues of  $\Sigma$ , and  $\mathbf{V}_q = (\mathbf{v}_1, \dots, \mathbf{v}_q)$  is a  $p \times q$  matrix of their corresponding leading eigenvectors. Define  $\hat{\Sigma}, \hat{\Lambda}_q, \hat{\mathbf{V}}_q$  to be the estimators of  $\Sigma, \Lambda_q, \mathbf{V}_q$ . We further let  $\hat{\Lambda}_q = \text{diag}(\hat{\lambda}_1, \dots, \hat{\lambda}_q)$  and  $\hat{\mathbf{V}}_q = (\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_q)$  to be constructed by the first  $q$  leading empirical eigenvalues and the corresponding eigenvectors of  $\hat{\Sigma}$  and  $\hat{\mathbf{B}}\hat{\mathbf{B}}' = \hat{\mathbf{V}}_q\hat{\Lambda}_q\hat{\mathbf{V}}_q'$ . Similarly to [Fan et al. \(2018\)](#), we require the following bounds on the componentwise maximums of the estimators:

$$\text{(B.1)} \quad \left\| \hat{\Sigma} - \Sigma \right\|_{\max} = \mathcal{O}_P(\sqrt{\log p/T}),$$

$$\text{(B.2)} \quad \left\| (\hat{\Lambda}_q - \Lambda_q)\Lambda_q^{-1} \right\|_{\max} = \mathcal{O}_P(\sqrt{\log p/T}),$$

$$\text{(B.3)} \quad \left\| \hat{\mathbf{V}}_q - \mathbf{V}_q \right\|_{\max} = \mathcal{O}_P(\sqrt{\log p/(Tp)}).$$

Assumptions **(B.1)**-**(B.3)** are needed in order to ensure that the first  $q$  principal components are approximately the same as the columns of the factor loadings. The estimator  $\hat{\Sigma}$  can be thought of as any “pilot” estimator that satisfies **(B.1)**. For sub-Gaussian distributions, sample covariance matrix, its eigenvectors and eigenvalues satisfy **(B.1)**-**(B.3)**.

In addition, the following structural assumption on the model is imposed:

$$(C.1) \quad \|\boldsymbol{\Sigma}\|_{\max} = \mathcal{O}(1) \text{ and } \|\mathbf{B}\|_{\max} = \mathcal{O}(1).$$

Note that Assumptions (B.1)-(B.3) and (C.1) are standard assumptions and are used in Fan et al. (2018).

To study the properties of the combination weights and MSFE, we first need to establish the convergence properties of precision matrix produced by Algorithm B.2. Let  $\omega_T \equiv \sqrt{\log p/T} + 1/\sqrt{p}$ . Also, let  $s(\boldsymbol{\Theta}_\varepsilon) = \mathcal{O}_P(s_T)$  for some sequence  $s_T \in (0, \infty)$  and  $d(\boldsymbol{\Theta}_\varepsilon) = \mathcal{O}_P(d_T)$  for some sequence  $d_T \in (0, \infty)$ . The deterministic sequences  $s_T$  and  $d_T$  will control the sparsity  $\boldsymbol{\Theta}_\varepsilon$  for FGL. Note that  $d_T$  can be smaller than or equal to  $s_T$ .

Let  $\varrho_T$  be a sequence of positive-valued random variables such that  $\varrho_T^{-1}\omega_T \xrightarrow{P} 0$  and  $\varrho_T d_T s_T \xrightarrow{P} 0$ , with  $\tau \asymp \omega_T$  (where  $\tau$  is the tuning parameter for the FGL in (B.1)). Lee and Seregina (2023) show that under the Assumptions (A.1)-(A.3), (B.1)-(B.3) and (C.1),  $\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_1 = \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1)$  and  $\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}\|_2 = \mathcal{O}_P(\varrho_T s_T) = o_P(1)$  for FGL.

Having established the convergence rates for precision matrix, we now study the properties of the combination weights and the resulted MSFE.

**Theorem 1.** *Assume (A.1)-(A.3), (B.1)-(B.3), and (C.1) hold. FGL consistently estimates forecast combination weights and  $MSFE(\widehat{\mathbf{w}}, \widehat{\boldsymbol{\Sigma}})$ :*

- (i) *If  $\varrho_T d_T^2 s_T \xrightarrow{P} 0$ ,  $\|\widehat{\mathbf{w}} - \mathbf{w}\|_1 = \mathcal{O}_P(\varrho_T d_T^2 s_T) = o_P(1)$ .*
- (ii) *If  $\varrho_T d_T s_T \xrightarrow{P} 0$ ,  $\left| \frac{MSFE(\widehat{\mathbf{w}}, \widehat{\boldsymbol{\Sigma}})}{MSFE(\mathbf{w}, \boldsymbol{\Sigma})} - 1 \right| = \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1)$ .*

The proof of Theorem 1 can be found in Supplementary Appendix C. Note that the rates of convergence for MSFE and precision matrix  $\boldsymbol{\Theta}$ , which was derived in Lee and Seregina (2023), are the same and both are faster than the combination weight rates. In contrast

to the classical graphical model in Algorithm A.1, the convergence properties of which were examined by Janková and van de Geer (2018) among others, the rates in Theorem 1 depend on the sparsity of  $\Theta_\varepsilon$  rather than of  $\Theta$ . This means that instead of assuming that many partial correlations of forecast errors  $\mathbf{e}_t$  are negligible, which is not realistic under the factor structure, we impose a milder restriction requiring many partial correlations of  $\varepsilon_t$  to be negligible once the common components have been taken into account.

## 4 RD-FGL for Forecast Errors

There are two streams of literature that study time-varying networks. The first one models dynamics in the precision matrix locally. Zhou et al. (2010) develop a nonparametric method for estimating time-varying graphical structure for multivariate Gaussian distributions using an  $\ell_1$ -penalized log-likelihood. They find out that if the covariances change smoothly over time, the covariance matrix can be estimated well in terms of predictive risk even in high-dimensional problems. Lu et al. (2015) introduce nonparanormal graphical models that allow to model high-dimensional heavy-tailed systems and the evolution of their network structure. They show that the estimator consistently estimates the latent inverse Pearson correlation matrix. The second stream of literature allows the network to vary with time by introducing two different frequencies. Hallac et al. (2017) study time-varying Graphical LASSO with smoothing evolutionary penalty.

We augment the framework in Section 3 to account for regime switching by modeling the change in precision matrix due to  $N$  structural breaks. Define  $n_j \equiv t_j - t_{j-1}$  to be the sample between the  $j$ -th and  $(j-1)$ -th break points, where  $j = 1, \dots, N+1$ ,  $\sum_{j=1}^{N+1} n_j = T$ ,  $t_0 = 0$ ,  $N \leq T$ .

## 4.1 Regime-Dependent Factor Loadings

Macroeconomic and financial datasets typically span a long time period, hence, the assumption of time-invariant factor loadings is restrictive. As a first extension to FGL, we model structural changes in factor loadings using a framework similar to [Su and Wang \(2017\)](#). For now assume a single known break  $N = 1$  which occurs at  $T_1$ .<sup>3</sup> Write equation (2.1) as:

$$e_{it} = \underbrace{\mathbf{b}'_i}_{1 \times q} \underbrace{\mathbf{f}_t}_{q \times 1} + \varepsilon_{it}, \quad t = 1, \dots, T, \quad i = 1, \dots, p. \quad (4.1)$$

To estimate  $\{\mathbf{b}_i\}_{i=1}^p$  and  $\{\mathbf{f}_t\}_{t=1}^T$ , we can consider the following weighted least squares problem:

$$\min_{\{\mathbf{b}_i\}_{i=1}^p, \{\mathbf{f}_t\}_{t=1}^T} (pT)^{-1} \sum_{i=1}^p \sum_{t=1}^T [e_{it} - \mathbf{b}'_i \mathbf{f}_t]^2 K_{\gamma t}, \quad (4.2)$$

subject to certain identification restrictions to be specified later on. Here,  $K_{\gamma t} = \gamma \mathbb{1}[t \leq T_1] + \mathbb{1}[t > T_1]$  is a discrete kernel as in [Li et al. \(2013\)](#) with  $\gamma \in [0, 1]$ . Since more recent information is usually more relevant to forecasting, such kernel-weight estimator gives weight 1 to post-break observations and weight  $\gamma$  to pre-break observations.<sup>4</sup>

Define the  $T \times p$  matrices  $\mathbf{E}(\gamma) = (\mathbf{e}_1(\gamma), \dots, \mathbf{e}_p(\gamma))$ ,  $\mathcal{E}(\gamma) = (\boldsymbol{\varepsilon}_1(\gamma), \dots, \boldsymbol{\varepsilon}_p(\gamma))$ , where  $\mathbf{e}_i(\gamma) = (K_{\gamma 1}^{1/2} e_{i1}, \dots, K_{\gamma T}^{1/2} e_{iT})'$  and  $\boldsymbol{\varepsilon}_i(\gamma) = (K_{\gamma 1}^{1/2} \varepsilon_{i1}, \dots, K_{\gamma T}^{1/2} \varepsilon_{iT})'$ . Also, let  $\mathbf{F}(\gamma) = (K_{\gamma 1}^{1/2} \mathbf{f}_1, \dots, K_{\gamma T}^{1/2} \mathbf{f}_T)'$  be a  $T \times q$  matrix collecting factors. In matrix notation, the

<sup>3</sup>The possible presence of break over the forecast horizon, as explored in [Pesaran et al. \(2006\)](#), would result in common forecast errors of all forecasting models, which can be captured by using the factor model. This would be another scenario of common forecast errors as motivated by [Figure 1](#).

<sup>4</sup>Adjusting the parameter to control the degree to which pre-break data are discounted by the model is useful for applied researchers. Bayesian framework with hierarchical priors provides an alternative way to combine the sample information contained in the objective function with prior information about the values of the model parameters and the relations among them (see [Pástor and Stambaugh \(2001\)](#); [Pesaran et al. \(2006\)](#)).

transformed model in (4.1) can be written as  $\mathbf{E}(\gamma) = \mathbf{F}(\gamma)\mathbf{B}' + \mathcal{E}(\gamma)$ , where  $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_p)'$  is a  $p \times q$  matrix of factor loadings.

As shown in Su and Wang (2017) for the continuous kernel, the minimization problem in (4.2) reduces to:

$$\begin{aligned} \min_{\mathbf{F}(\gamma), \mathbf{B}} \operatorname{tr} \left[ \left( \mathbf{E}(\gamma) - \mathbf{F}(\gamma)\mathbf{B}' \right) \left( \mathbf{E}(\gamma) - \mathbf{F}(\gamma)\mathbf{B}' \right)' \right] \\ \text{s.t. } \mathbf{F}'(\gamma)\mathbf{F}(\gamma)/T = \mathbf{I}_q \text{ and } \mathbf{B}'\mathbf{B} = \text{diagonal matrix.} \end{aligned} \quad (4.3)$$

The problem in (4.3) is the conventional PCA problem. The estimated factor matrix  $\widehat{\mathbf{F}}(\gamma) = \left( K_{\gamma_1}^{1/2}\widehat{\mathbf{f}}_1, \dots, K_{\gamma_T}^{1/2}\widehat{\mathbf{f}}_T \right)'$  is  $\sqrt{T}$  times eigenvectors corresponding to the  $q$  largest eigenvalues of  $\mathbf{E}(\gamma)\mathbf{E}'(\gamma)$ , arranged in descending order, and  $\widehat{\mathbf{B}}'(\gamma) = \left( \widehat{\mathbf{F}}(\gamma)\widehat{\mathbf{F}}'(\gamma) \right)^{-1}\widehat{\mathbf{F}}'(\gamma)\mathbf{E}(\gamma) = \widehat{\mathbf{F}}'(\gamma)\mathbf{E}(\gamma)/T$  are the estimators of the corresponding time-varying factor loadings, where  $\widehat{\mathbf{B}}(\gamma) = \left( \widehat{\mathbf{b}}_1(\gamma), \dots, \widehat{\mathbf{b}}_p(\gamma) \right)'$  is  $p \times q$ .

Since the estimator  $\widehat{\mathbf{F}}(\gamma)$  is only consistent up to a rotation, we use a two-stage estimation procedure to obtain a consistent estimator (Su and Wang (2017)). Based on the consistent estimators of  $\mathbf{b}_i$ 's obtained from the first stage, consistent estimators of  $\mathbf{f}_t(\gamma)$  can be obtained by considering the following least squares problem  $\widehat{\mathbf{f}}_t(\gamma) = \operatorname{argmin}_{\mathbf{f}_t} \sum_{i=1}^p \left[ e_{it} - \widehat{\mathbf{b}}_i'(\gamma)\mathbf{f}_t \right]^2$  which yields the solution  $\widehat{\mathbf{f}}_t(\gamma) = \left( \sum_{i=1}^p \widehat{\mathbf{b}}_i(\gamma)\widehat{\mathbf{b}}_i'(\gamma) \right)^{-1} \left( \sum_{i=1}^p \widehat{\mathbf{b}}_i(\gamma)e_{it} \right)$ .

As in Su and Wang (2017), we assume that  $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t']$  is homogeneous over  $t$ . This assumption is not restrictive, since if  $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t'] = \boldsymbol{\Sigma}_{f,t}$ , we can rewrite the common component as  $\mathbf{b}_i'\mathbf{f}_t = \left( \boldsymbol{\Sigma}_f^{-1/2}\boldsymbol{\Sigma}_{f,t}^{1/2}\mathbf{b}_i \right)' \boldsymbol{\Sigma}_f^{1/2}\boldsymbol{\Sigma}_{f,t}^{-1/2}\mathbf{f}_t = \mathbf{b}_i^*\mathbf{f}_t^*$ , where  $\mathbf{b}_i^* = \boldsymbol{\Sigma}_f^{-1/2}\boldsymbol{\Sigma}_{f,t}^{1/2}\mathbf{b}_i$ , and  $\mathbf{f}_t^* = \boldsymbol{\Sigma}_f^{1/2}\boldsymbol{\Sigma}_{f,t}^{-1/2}\mathbf{f}_t$  satisfies  $\mathbb{E}[\mathbf{f}_t^*\mathbf{f}_t^{*'}] = \boldsymbol{\Sigma}_f$  for each  $t$ .

To choose the optimal tuning parameter  $\gamma$  in (4.2), we use the cross-validation and solve



the following minimization problem:

$$\min_{\gamma} \text{CV}(\gamma) = \frac{1}{p(T - T_1)} \sum_{i=1}^p \sum_{s=T_1+1}^T [e_{is} - \widehat{\mathbf{b}}_i^{(-s)}(\gamma) \widehat{\mathbf{f}}_s^{(-s)}(\gamma)]^2, \quad (4.4)$$

where  $\widehat{\mathbf{b}}_i^{(-s)}(\gamma)$  and  $\widehat{\mathbf{f}}_s^{(-s)}(\gamma)$  are estimated by leaving the  $s$ -th time series observation out of the PCA procedure.

**Remark 1.** *The procedure for estimating regime-dependent factor loadings can be easily extended to the case when the number of breaks is greater than 1 ( $N > 1$ ). The kernel in (4.2) would be adjusted accordingly  $K_{\gamma_j t} = \gamma_j \mathbf{1}[t \leq T_j] + \mathbf{1}[t > T_N]$ , where  $j = 1, \dots, N + 1$ . To estimate  $\{\gamma_j\}_{j=1}^{N+1}$  we use cross-validation as in (4.4) consequently applied to each two periods separated by a break.*

## 4.2 Regime-Dependent Idiosyncratic Precision Matrix

As a second extension to FGL, we model structural changes in the precision matrix of the idiosyncratic component. Let  $\Sigma_{\varepsilon,j}$  and  $\Sigma_j$  be covariance matrices of idiosyncratic part and forecast errors in regime  $j$ . Define the corresponding precision matrices to be  $\Theta_{\varepsilon,j} \equiv \Sigma_{\varepsilon,j}^{-1}$  and  $\Theta_j \equiv \Sigma_j^{-1}$ . Similarly to the previous subsection, without loss of generality we assume  $\Sigma_{f_j} = \Sigma_f$  for all regimes  $j$ .

Let  $\widehat{\Sigma}_{\varepsilon,j} = \frac{1}{n_j} \sum_{k=1}^{n_j} \widehat{\varepsilon}_{j,k} \widehat{\varepsilon}_{j,k}'$ . To model dynamics in  $\{\Theta_{\varepsilon,j}\}_{j=1}^{N+1}$  we use the following optimization problem:

$$\min_{\{\Theta_{\varepsilon,j}\}_{j=1}^{N+1}} \sum_{j=1}^{N+1} n_j \left[ \text{tr} \left( \widehat{\Sigma}_{\varepsilon,j} \Theta_{\varepsilon,j} \right) - \log \det \Theta_{\varepsilon,j} \right] + \alpha \|\Theta_{\varepsilon,j}\|_{\text{od},1} + \beta \sum_{j=2}^{N+1} \psi(\Theta_{\varepsilon,j} - \Theta_{\varepsilon,j-1}), \quad (4.5)$$

where the penalty for the off-diagonal (od) elements is  $\|\Theta_{\varepsilon,j}\|_{\text{od},1} = \sum_{l \neq q} \widehat{\gamma}_{\varepsilon,l,j} \widehat{\gamma}_{\varepsilon,qq,j} |\theta_{\varepsilon,lq,j}|$ ,

$\widehat{\gamma}_{\varepsilon, ll, j}$  is the  $(l, l)$ -th element of  $\widehat{\Gamma}_{\varepsilon, j}^2 \equiv \text{diag}(\widehat{\Sigma}_{\varepsilon, j})$  and  $\theta_{\varepsilon, lq, j}$  is the  $lq$ -th element of matrix  $\Theta_{\varepsilon, j}$ .

Figure 2 visualizes dynamics of the precision matrix.

$$\begin{array}{ccc}
 \begin{array}{l} \text{tr}(\widehat{\Sigma}_{\varepsilon, 1} \Theta_{\varepsilon, 1}) \\ - \log \det \Theta_{\varepsilon, 1} \\ + \alpha \|\Theta_{\varepsilon, 1}\|_{\text{od}, 1} \end{array} & 
 \begin{array}{l} \text{tr}(\widehat{\Sigma}_{\varepsilon, 2} \Theta_{\varepsilon, 2}) \\ - \log \det \Theta_{\varepsilon, 2} \\ + \alpha \|\Theta_{\varepsilon, 2}\|_{\text{od}, 1} \end{array} & 
 \begin{array}{l} \text{tr}(\widehat{\Sigma}_{\varepsilon, N+1} \Theta_{\varepsilon, N+1}) \\ - \log \det \Theta_{\varepsilon, N+1} \\ + \alpha \|\Theta_{\varepsilon, N+1}\|_{\text{od}, 1} \end{array} \\
 \hline
 \begin{array}{ccc}
 | & | & | \\
 t_1 & t_2 & t_{N+1} \\
 \beta\psi(\Theta_{\varepsilon, 2} - \Theta_{\varepsilon, 1}) & \beta\psi(\Theta_{\varepsilon, 3} - \Theta_{\varepsilon, 2}) & \beta\psi(\Theta_{\varepsilon, N+1} - \Theta_{\varepsilon, N})
 \end{array}
 \end{array}$$

Figure 2: **Change of precision matrix over time:**  $\beta$  is the penalty that enforces temporal consistency and  $\psi$  is a convex penalty function.

The optimization problem in (4.5) has two tuning parameters:  $\alpha$ , which determines the sparsity level of the network, and  $\beta$ , which controls the strength of resemblance between two neighboring precision estimators.<sup>5</sup> In simulations and the empirical application we use the following procedure for tuning  $\alpha$  and  $\beta$ : first, we set a grid of values  $(\alpha, \beta) \in \{0, 0.25, 0.5, 1, 10, 30\}$ . Second, we use the first 2/3 of the training data to estimate forecast combination weights and jointly tune  $\alpha$  and  $\beta$  in the remaining 1/3 to yield the smallest value of the objective function, which is chosen to be either  $\|\cdot\|_2$ -loss of precision matrix for simulations in Subsection 5.1, or MSFE for simulations in Subsection 5.2 and the empirical application. Note that when  $\beta = 0$ , the optimization in (4.5) reduces to estimating  $\Theta_{\varepsilon, i}$  using Algorithm B.2 in each regime separately. Naturally, this incorporates the case when the structural break is strong and only the post-break data is used for producing forecast combination weights. When  $\beta$  is large, there are weak structural breaks in  $\Theta_{\varepsilon, j}$ , and  $\Theta_{\varepsilon, j}$ 's are estimated by using the data across different regimes. Section 6 provides

<sup>5</sup>Pesaran et al. (2006) use Bayesian framework to handle parameter instability using hierarchical priors. As pointed out in their paper, “intuition for the use of hierarchical priors comes from the shrinkage literature, since one can think of the parameters within the individual regimes as being shrunk towards a set of the so-called hyperparameters that characterize the “top” layer of the hierarchy” (p. 1059).

more discussion on this in the context of our empirical application.

The smoothing function  $\psi(\cdot)$  in (4.5) can be LASSO ( $\psi = \sum_{l,q} |\cdot|$ ), Group LASSO ( $\psi = \sum_q \|\cdot\|_2$ ), or Ridge ( $\psi = \sum_{l,q} (\cdot)_{lq}^2$ ). LASSO penalty encourages small changes in the precision matrix over time: when the  $lq$ -th element changes at two consecutive times, the penalty forces the rest of the elements of the precision to remain the same. Group LASSO penalty allows the entire graph to restructure at some time points. This penalty is useful for anomaly detection, since it can identify structural changes in the network structure. Ridge penalty allows the network to change smoothly over time. This penalty is less strict than the LASSO penalty: instead of encouraging the graphs to be exactly the same, it allows smooth transitions. In our empirical application we use Ridge penalty to accommodate smooth transitions of precision over time.

To estimate (4.5) we use the ADMM algorithm described in details in Supplementary Appendix D. Once  $\Theta_{\varepsilon,i}$  is estimated, we combine estimated factors, loadings and precision matrix of the idiosyncratic components using Sherman-Morrison-Woodbury formula to estimate the final precision matrix of forecast errors and use it to compute optimal forecast combination weights. We call the aforementioned procedure RD-FGL and summarize it in Algorithm 1.

---

**Algorithm 1** RD-FGL
 

---

- 1: Estimate  $\{\mathbf{b}_i\}_{i=1}^p$  and  $\{\mathbf{f}_t(\gamma_j)\}_{t=1}^T$  in (4.1) using the weighted least squares problem in (4.2). Get  $\widehat{\Sigma}_f$ ,  $\widehat{\Theta}_f$  and  $\widehat{\boldsymbol{\varepsilon}}_t(\gamma_j) = \mathbf{e}_t - \widehat{\mathbf{B}}(\gamma_j)\widehat{\mathbf{f}}_t(\gamma_j)$ .
  - 2: Solve (4.5) using ADMM to get  $\widehat{\Theta}_{\varepsilon,j}$ .
  - 3: Use  $\widehat{\Theta}_{\varepsilon,j}$ ,  $\widehat{\Theta}_f$  and  $\widehat{\mathbf{B}}(\gamma_j)$  from Steps 1-2 to get  $\widehat{\Theta}_j(\gamma_j) = \widehat{\Theta}_{\varepsilon,j} - \widehat{\Theta}_{\varepsilon,j}\widehat{\mathbf{B}}(\gamma_j)[\widehat{\Theta}_f + \widehat{\mathbf{B}}'(\gamma_j)\widehat{\Theta}_{\varepsilon,j}\widehat{\mathbf{B}}(\gamma_j)]^{-1}\widehat{\mathbf{B}}'(\gamma_j)\widehat{\Theta}_{\varepsilon,j}$ .
  - 4: Use  $\widehat{\Theta}_j(\gamma_j)$  to get forecast combination weights  $\widehat{\mathbf{w}}_j(\gamma_j) = \frac{\widehat{\Theta}_j(\gamma_j)\boldsymbol{\iota}_p}{\boldsymbol{\iota}_p'\widehat{\Theta}_j(\gamma_j)\boldsymbol{\iota}_p}$ .
- 

We develop a scalable implementation of (4.5) for the RD-FGL in Algorithm 1 through ADMM, which is extensively discussed in Supplementary Appendix D. ADMM is a distributed convex optimization approach (Parikh and Boyd (2014)) that allows us to split the optimization problem in (4.5) into a series of subproblems. As pointed out in Hallac et al. (2017), the scalability of ADMM comes from the improved runtime: to estimate a  $p \times p$  matrix, the cost per iteration of ADMM is  $\mathcal{O}(p^3)$  (which is the cost of an eigendecomposition of the  $\Theta$  step in Supplemental Appendix D). In contrast, the runtime of general interior-point methods is  $\mathcal{O}(p^6)$  (Mohan et al. (2014)).

**Remark 2.** *Let us comment on the theoretical properties of RD-FGL. First, as shown in Su and Wang (2017), introducing time-varying factors does not change the main assumptions (A.1)-(A.3) on the errors, factors, factor loadings, and their interactions. This is because, as shown in (4.3), the formulation with time-varying loadings can be reduced to the conventional PCA problem. Additional assumption that we need to impose is that  $\mathbb{E}[\mathbf{f}_t\mathbf{f}_t']$  is homogeneous over  $t$ . As discussed in Subsection 4.1, this assumption is not restrictive. Second, we assume that the number of factors,  $q$ , and the number of forecasts,  $p$ , are not af-*

affected by the structural changes in loadings or idiosyncratic precision matrix. Allowing  $p$  and  $q$  to change is a straightforward extension and is left for future research. Third, assumptions **(B.1)**-**(B.3)** and assumption **(C.1)** are required to hold for each regime  $j = 1, \dots, N + 1$ . Finally, we allow  $s(\Theta_{\varepsilon,j}) = \mathcal{O}_P(s_{n_j})$  and  $d(\Theta_{\varepsilon,j}) = \mathcal{O}_P(d_{n_j})$  to change for  $j = 1, \dots, N + 1$ . Let  $\omega_{n_j} \equiv \sqrt{\log p/n_j + 1}/\sqrt{p}$ . As long as  $\varrho_{n_j}^{-1}\omega_{n_j} \xrightarrow{P} 0$  and  $\varrho_{n_j}d_{n_j}s_{n_j} \xrightarrow{P} 0$  for each  $j$ , RD-FGL achieves the same rate as FGL in each regime:

- (i) If  $\varrho_{n_j}d_{n_j}^2s_{n_j} \xrightarrow{P} 0$ , RD-FGL consistently estimates forecast combination weights  $\widehat{\mathbf{w}}_j(\gamma_j)$  in Algorithm 1:  $\|\widehat{\mathbf{w}}_j(\gamma_j) - \mathbf{w}_j\|_1 = \mathcal{O}_P(\varrho_{n_j}d_{n_j}^2s_{n_j}) = o_P(1)$ .
- (ii) If  $\varrho_{n_j}d_{n_j}s_{n_j} \xrightarrow{P} 0$ , FGL consistently estimates MSFE( $\mathbf{w}_j, \Sigma_j$ ):  $\left| \frac{MSFE(\widehat{\mathbf{w}}_j(\gamma), \widehat{\Sigma}_j)}{MSFE(\mathbf{w}_j, \Sigma_j)} - 1 \right| = \mathcal{O}_P(\varrho_{n_j}d_{n_j}s_{n_j}) = o_P(1)$ .

### 4.3 Unknown Break Time and Number of Breaks

The previous two subsections assumed that the number and location of breaks are known. We now relax these assumptions. First, assume that the number of breaks in factor loadings,  $N_B$ , and the number of breaks in idiosyncratic precision,  $N_{\Theta}$ , are known and  $N_B = N_{\Theta} = 1$ , but their locations are unknown and might differ from each other.

To estimate the location of the break in factor loadings, we adapt the procedure in Bai et al. (2020). For a given break point in loadings,  $T_1$ , define the sum of squared residuals (SSR) as in (4.2):

$$\text{SSR}(T_1) = (pT)^{-1} \sum_{i=1}^p \sum_{t=1}^T [e_{it} - \mathbf{b}'_{it}\mathbf{f}_t]^2 K_{\gamma t}, \quad (4.6)$$

where  $K_{\gamma t} = \gamma \mathbb{1}[t \leq T_1] + \mathbb{1}[t > T_1]$  is a discrete kernel. The estimated break date is given by  $\widehat{T}_1 = \text{argmin}_{1 \leq T_1 \leq T-1} \text{SSR}(T_1)$ .

To estimate the location of the break in  $\Theta_\varepsilon$  we use the procedure similar to Bai (2010). Define  $t_1$  to be a break point in  $\Theta_\varepsilon$ . Recall,  $n_j = t_j - t_{j-1}$ , where  $j = 1, 2$ . Note that the number of observations in each regime depends on  $t_1$ :  $n_1 = t_1 - t_0$  and  $n_2 = t_2 - t_1$ . For a given break point in idiosyncratic precision  $t_1$ , define the following objective function as in (4.5):

$$L(t_1) = \sum_{j=1}^2 n_j \left[ \text{tr} \left( \widehat{\Sigma}_{\varepsilon,j} \Theta_{\varepsilon,j} \right) - \log \det \Theta_{\varepsilon,j} \right] + \alpha \|\Theta_{\varepsilon,j}\|_{\text{od},1} + \beta \psi(\Theta_{\varepsilon,2} - \Theta_{\varepsilon,1}). \quad (4.7)$$

The estimated break date is given by  $\widehat{t}_1 = \text{argmin}_{1 \leq t_1 \leq T-1} L(t_1)$ .<sup>6</sup>

When the number of breaks in either loadings is known and greater than 1 ( $N_B > 1$ ) and/or  $N_\Theta > 1$ , we can use the one-at-a-time approach as in Bai (2010): the objective functions are identical to (4.6) and (4.7). The breaks are estimated sequentially. Once the first break is obtained, we split the sample at the estimated break point, resulting in two subsamples. A single break point in each subsample is estimated, but only one that achieves the smallest objective function ((4.6) or (4.7)) is retained. If the number of breaks is equal to two, the procedure is stopped. Otherwise, we continue splitting into subsamples until all breaks are estimated.

If the number of breaks is unknown, we proceed as suggested in Bai (2010): in the aforementioned one-at-a-time approach apply the test for existence of break point (Bai and Perron (2003)) to each subsample before estimating a break point.

---

<sup>6</sup>As noted by Bai and Perron (1998), it is difficult to detect the break near the end of the sample. It is reasonable to expect that if break magnitude is reduced, the procedures in (4.6) and (4.7) might not detect the break or detect it with a delay. Smith and Timmermann (2021) point out that in such situations only few observations from the current regime are available to estimate the model parameters, leading to volatile and inaccurate forecasts. To address slow detection of breaks, Smith and Timmermann (2021) exploit information in the cross-section to detect breaks more rapidly in real time. To address small sample size problem, Smith and Timmermann (2021) adopt a Bayesian approach that uses economically motivated priors to shrink the parameters towards sensible values that rule out economically implausible values. As we discussed in Footnote 4, the Bayesian framework is alternative to using kernel-weighted observations.

## 5 Monte Carlo

We divide the simulation results into two subsections. In the first subsection we study the consistency of the FGL and RD-FGL for estimating precision matrix and the combination weights. In the second subsection we evaluate the out-of-sample forecasting performance of combined forecasts in terms of MSFE. We compare the performance of forecast combinations based on the factor models in Algorithms 1, B.2 with equal-weighted (EW) forecast combination<sup>7</sup>, and combinations that use GL without factor structure (Algorithm A.1). We examine the performance of RD-FGL for different specifications of the smoothing function  $\psi(\cdot)$  as described in Subsection 4.2. LASSO penalty is denoted as  $\ell_1$ , Group LASSO as  $\ell_g$ , and Ridge as  $\ell_2$ . Similarly to the literature on graphical models, all exercises use 100 Monte Carlo simulations. We present simulation results with a structural break in both  $\mathbf{B}$  and  $\Theta_\varepsilon$ . The results without a break, with break only in  $\Theta_\varepsilon$ , and with multiple breaks can be found in Supplemental Appendix E.

### 5.1 Consistent Estimation of Forecast Combination Weights

We consider sparse Gaussian graphical models which may be fully specified by a precision matrix  $\Theta_0$ . Therefore, the random sample is distributed as  $\mathbf{e}_t = (e_{1t}, \dots, e_{pt})' \sim \mathcal{N}(0, \Sigma_0)$ , where  $\Theta_0 = (\Sigma_0)^{-1}$  for  $t = 1, \dots, T$ ,  $i = 1, \dots, p$ . Let  $\hat{\Theta}$  be the precision matrix estimator. We show consistency of the FGL in (i) the operator norm,  $\left\| \hat{\Theta} - \Theta_0 \right\|_2$ , and (ii) in  $\ell_1$ -vector norm for the combination weights,  $\|\hat{\mathbf{w}} - \mathbf{w}\|_1$ , where  $\mathbf{w}$  is given by (2.5).

---

<sup>7</sup>As pointed out by the referee, EW arises when the forecast errors follow a factor structure (one factor, homogeneous idiosyncratic variance). It can be viewed as one of “factor-based” methods.

The forecast errors are assumed to have the following structure:

$$\underbrace{\mathbf{e}_t}_{p \times 1} = \mathbf{B} \underbrace{\mathbf{f}_t}_{q \times 1} + \boldsymbol{\varepsilon}_t, \quad \mathbf{f}_t = \phi_f \mathbf{f}_{t-1} + \boldsymbol{\zeta}_t, \quad t = 1, \dots, T \quad (5.1)$$

where  $\mathbf{e}_t$  is a  $p \times 1$  vector of forecast errors following  $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ ,  $\mathbf{f}_t$  is a  $q \times 1$  vector of factors,  $\mathbf{B}$  is a  $p \times q$  matrix of factor loadings,  $\phi_f$  is an autoregressive parameter in the factors which is a scalar for simplicity,  $\boldsymbol{\zeta}_t$  is a  $q \times 1$  random vector with each component independently following  $\mathcal{N}(0, \sigma_\zeta^2)$ ,  $\boldsymbol{\varepsilon}_t$  is a  $p \times 1$  random vector following  $\mathcal{N}(0, \boldsymbol{\Sigma}_\varepsilon)$ , with sparse  $\boldsymbol{\Theta}_\varepsilon$  that has a random graph structure described below. To create  $\mathbf{B}$  in (5.1) we take the first  $q$  columns of an upper triangular matrix from a Cholesky decomposition of the  $p \times p$  Toeplitz matrix parameterized by  $\rho$ : that is,  $\mathbf{B} = (b)_{lm}$ , where  $(b)_{lm} = \rho^{|l-m|}$ ,  $l, m \in \{1, \dots, p\}$ . We set  $\rho = 0.2$ ,  $\phi_f = 0.2$  and  $\sigma_\zeta^2 = 1$ . The specification in (5.1) leads to the low-rank plus sparse decomposition of the covariance matrix  $\mathbb{E}[\mathbf{e}_t \mathbf{e}_t'] = \boldsymbol{\Sigma} = \mathbf{B} \boldsymbol{\Sigma}_f \mathbf{B}' + \boldsymbol{\Sigma}_\varepsilon$ . When  $\boldsymbol{\Sigma}_\varepsilon$  has a sparse inverse  $\boldsymbol{\Theta}_\varepsilon$ , it leads to the low-rank plus sparse decomposition of the precision matrix  $\boldsymbol{\Theta}$ , such that  $\boldsymbol{\Theta}$  can be expressed as a function of the low-rank  $\boldsymbol{\Theta}_f$  plus sparse  $\boldsymbol{\Theta}_\varepsilon$ .

We consider the following setup: let  $p = T^\delta$ ,  $\delta = 0.85$ ,  $q = 2(\log(T))^{0.5}$  and  $T = \lceil 2^\kappa \rceil$ , for  $\kappa = 7, 7.5, 8, \dots, 9.5$ . Our setup allows the number of individual forecasts,  $p$ , and the number of common factors in the forecast errors,  $q$ , to increase with the sample size,  $T$ .

A sparse precision matrix of the idiosyncratic components  $\boldsymbol{\Theta}_\varepsilon$  is constructed as follows: we first generate the adjacency matrix using a random graph structure. Define a  $p \times p$  adjacency matrix  $\mathbf{A}_\varepsilon$  which represents the structure of the graph with  $a_{\varepsilon,lm}$  being the  $l, m$ -th element of the adjacency matrix  $\mathbf{A}_\varepsilon$ . We set  $a_{\varepsilon,lm} = a_{\varepsilon,ml} = 1$ , for  $l \neq m$  with probability  $\pi$ , and 0 otherwise. Such structure results in  $s_T = p(p-1)\pi/2$  edges in the graph. To control sparsity, we set  $\pi = 500/(pT^{0.8})$ , which makes  $s_T = \mathcal{O}(T^{0.05})$ . The adjacency matrix has all



diagonal elements equal to zero. To generate a sparse symmetric positive-definite precision matrix we use Scikit-Learn datasets package in Python (Pedregosa et al. (2011)). To control the magnitude of partial correlations, the value of the smallest coefficient is set to 0.1 and the value of the largest coefficient is set to 0.3.

To incorporate structural breaks in  $\Theta_\varepsilon$  and factor loadings  $\mathbf{B}$ , we proceed as follows. We fix a single break point in the middle of the sample size,  $T/2$ : in the precision matrix of the idiosyncratic errors before the break, referred to as  $\Theta_{\varepsilon,1}$ , the value of the largest coefficient is set to 0.4; whereas in the precision matrix of the idiosyncratic errors after the break,  $\Theta_{\varepsilon,2}$ , the value of the largest coefficient is set to 0.6. As a consequence, even though both matrices are still sparse,  $\Theta_{\varepsilon,2}$  has larger partial correlations. We use  $\Theta_{\varepsilon,1}$  and  $\Theta_{\varepsilon,2}$  to generate  $\varepsilon_t$  in (5.1). For the structural break in factor loadings (which is assumed to happen at the same time as the structural change in  $\Theta_\varepsilon$ ), before the break we set  $\rho_1 = 0.2$  in the Toeplitz matrix used to generate  $\mathbf{B}$  (i.e.,  $\mathbf{B} = (b)_{lm}$ , where  $(b)_{lm} = \rho^{|l-m|}$ ), and after the break we set  $\rho_2 = 0.6$ .

Figure 3 shows the averaged (over Monte Carlo simulations) errors of the estimators of the precision matrix  $\Theta$  and the optimal combination weight versus the sample size  $T$  in the logarithmic scale (base 2). The estimate of the precision matrix of the EW forecast combination is obtained using the fact that diagonal covariance and precision matrices imply equal weights. To determine the values of the diagonal elements we use the shrinkage intensity coefficient calculated as the average of the eigenvalues of the sample covariance matrix of the forecast errors (see Ledoit and Wolf (2004)).

Figure 3 examines the performance when there are breaks in both  $\Theta_\varepsilon$  and  $\mathbf{B}$ : accounting for the break significantly reduces the estimation error of precision matrix and combination weights. We report the results for the case when  $\gamma$  is estimated using cross-validation ( $\gamma = \hat{\gamma}$ )

(as discussed in Section 4). Supplemental Appendix E presents the results for the case when the break is only in  $\Theta_\epsilon$ .

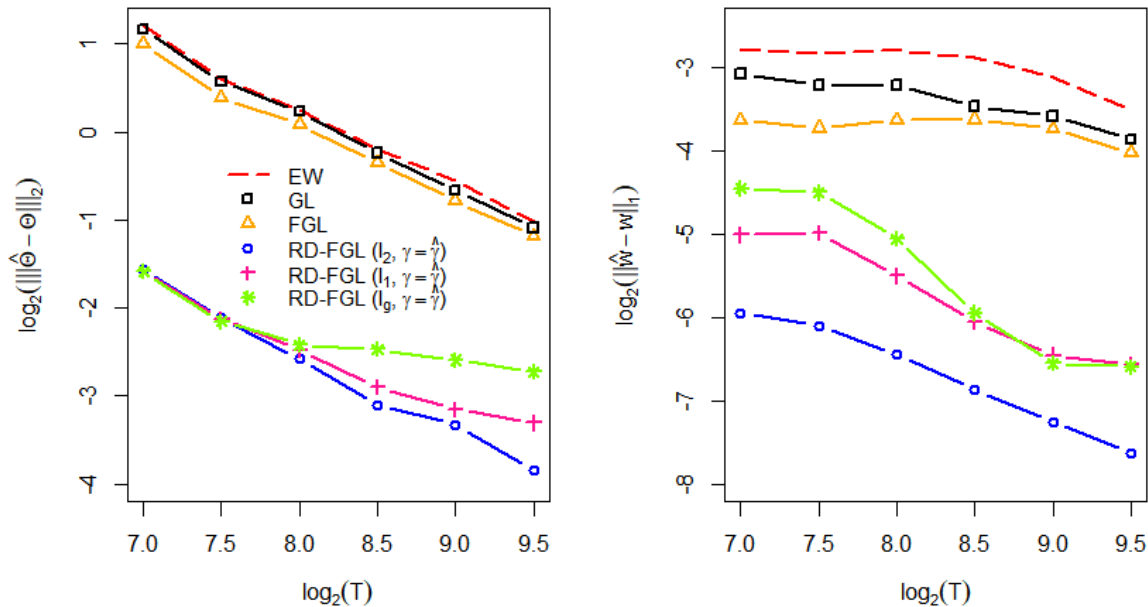


Figure 3: Averaged errors of the estimators of  $\Theta$  (left) and  $w$  on logarithmic scale (base 2): break in  $\Theta_\epsilon$  and factor loadings  $B$ .  $p = T^{0.85}$ ,  $q = 2(\log(T))^{0.5}$ ,  $s_T = \mathcal{O}(T^{0.05})$ .

## 5.2 Comparing Performance of Forecast Combinations

We consider the standard forecasting model in the literature (e.g., [Stock and Watson \(2002\)](#)), which uses the factor structure of the high dimensional predictors. Suppose the data is generated from the following data generating process (DGP):

$$\mathbf{x}_t = \Lambda \mathbf{g}_t + \mathbf{v}_t, \quad \mathbf{g}_t = \phi \mathbf{g}_{t-1} + \boldsymbol{\xi}_t, \quad y_{t+1} = \mathbf{g}'_t \boldsymbol{\alpha} + \sum_{s=1}^{\infty} \theta_s \epsilon_{t+1-s} + \epsilon_{t+1}, \quad (5.2)$$

where  $y_{t+1}$  is a univariate series of our interest in forecasting,  $\mathbf{x}_t$  is an  $M \times 1$  vector of regressors (predictors),  $\boldsymbol{\alpha}$  is an  $M \times 1$  parameter vector,  $\mathbf{g}_t$  is an  $r \times 1$  vector of factors,  $\boldsymbol{\Lambda}$  is an  $M \times r$  matrix of factor loadings,  $\mathbf{v}_t$  is an  $M \times 1$  random vector following  $\mathcal{N}(0, \sigma_v^2 \mathbf{I}_M)$ ,  $\phi$  is an autoregressive parameter in the factors which is a scalar for simplicity,  $\boldsymbol{\xi}_t$  is an  $M \times 1$  random vector with each component independently following  $\mathcal{N}(0, \sigma_\xi^2)$ ,  $\epsilon_{t+1}$  is a random error following  $\mathcal{N}(0, \sigma_\epsilon^2)$ , and  $\boldsymbol{\alpha}$  is an  $r \times 1$  parameter vector which is drawn randomly from  $\mathcal{N}(1, 1)$ . We set  $\sigma_\epsilon = 1$ . The coefficients  $\theta_s$  are set according to the rule  $\theta_s = (1 + s)^{c_1} c_2^s$  as in Hansen (2008). We set  $c_1 = 0.75$ . We set  $M = 100$  and generate  $r = 5$  factors. To create  $\boldsymbol{\Lambda}$  in (5.2) we take the first  $r$  rows of an upper triangular matrix from a Cholesky decomposition of the  $M \times M$  Toeplitz matrix parameterized by  $\rho = 0.9$ . The ranking of competing models was not very sensitive to varying values of  $\phi$ ,  $\rho$ ,  $c_2$ , and  $r$ .

One-step ahead forecasts are estimated from the factor-augmented autoregressive (FAR) models of orders  $k, l$ , denoted as FAR( $k, l$ ):

$$\hat{y}_{t+1} = \hat{\mu} + \hat{\kappa}_1 \hat{g}_{1,t} + \cdots + \hat{\kappa}_k \hat{g}_{k,t} + \hat{\psi}_1 y_t + \cdots + \hat{\psi}_l y_{t+1-l}, \quad (5.3)$$

where the factors  $(\hat{g}_{1,t}, \dots, \hat{g}_{k,t})$  are estimated from equation (5.2). We consider the FAR models of various orders, with  $k = 1, \dots, K$  and  $l = 1, \dots, L$ . We also consider the models without any lagged  $y$  or any factors. Therefore, the total number of forecasting models is  $p \equiv (1 + K) \times (1 + L)$ , which includes the forecasting models using naive average or no factors. We set  $K = 2$  and  $L = 7$ .

The total number of observations is  $m$ . The period for training the models is set to be  $m_1 = m/2$  – this is used to train competing FAR models in (5.3). The remaining part of the sample,  $m_2 = m - m_1$  is split as follows: the estimation window for training competing

models (that is, EW, GL, FGL, and RD-FGL) is set to be of size =  $m_2/2$ . We roll the estimation window over the the test sample of the size  $m_2/2$  to update all the estimates in each point of time. Recall that  $q$  denotes the number of factors in the forecast errors as in equation (2.1).

To incorporate structural break we proceed as follows. The period for training the models is set to be  $m_1 = m/3$  – this is used to train competing FAR models in (5.3). The remaining part of the sample,  $m_2 = m - m_1$  is split as follows: the estimation window for training competing models is set to be of size =  $m_2/2$ . We roll the estimation window over the test sample of the size  $m_2/2$ . The break point is fixed at 1/2 of the first estimation window. Before the break, when generating  $\theta_s$  we set  $c_2 = 0.3$ , and after the break  $c_2 = 0.9$ . All other parameters stay unchanged. Notice that the break in  $c_2$  can propagate into both a break in precision matrix and factor loadings.

Similarly to the previous subsection, we include different specifications of the smoothing function  $\psi(\cdot)$ . Figure 4 shows the performance of all models including RD-FGL with  $\gamma$  estimated using cross-validation: similarly to the conclusions in the previous subsection, accounting for the break significantly reduces MSFE of the combined forecast.

## 6 Application to Combining ECB SPF Forecasts

We use quarterly forecasts on the expected rates of inflation, real GDP growth and unemployment rate in the Euro area published by the ECB. The raw data records 119 forecasters in total, but the panel is highly unbalanced with many missing values due to entry and exit in the long span. We follow Shi et al. (2020) to obtain most qualified forecasters: first, we filter out irregular respondents if they missed more than 45% of the observations;

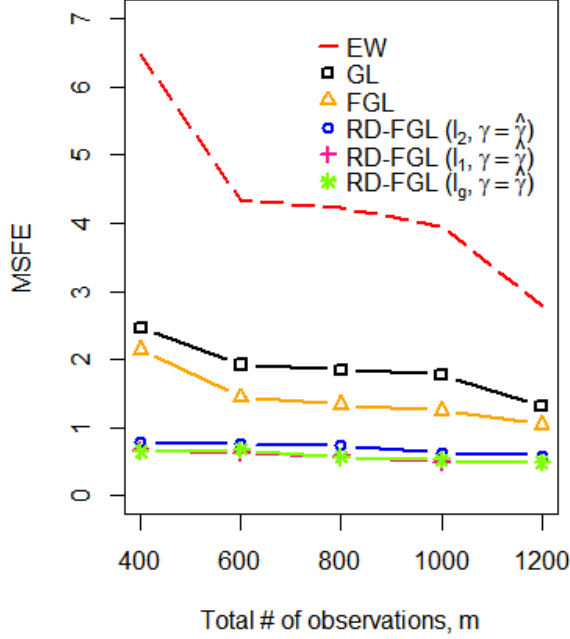


Figure 4: **Plots of the MSFE over the total number of observations  $m$ .**  $c_1 = 0.75$ ,  $c_2 = 0.3$  (before the break),  $c_2 = 0.9$  (after the break),  $M = 100$ ,  $r = 5$ ,  $\sigma_\xi = 1$ ,  $L = 7$ ,  $K = 2$ ,  $p = 24$ ,  $q = 3$ ,  $\rho = 0.9$ ,  $\phi = 0.8$ .

second, we use a random forest imputation algorithm (Stekhoven (2022); Stekhoven and Buhlmann (2012)) to interpolate the remaining missing values. We consider the forecasts of three main economic indicators: (1) Real GDP growth defined as the year-on-year (YoY) percentage change of real GDP, based on standardized European System of National and Regional Accounts (ESA) definition. The time period under consideration is 1999:Q3-2023Q1 (which yields the total number of observations equal to 95), the final number of forecasters is  $p = 59$ , and the prediction horizon is 2-quarters ahead. (2) Inflation which is defined as the YoY percentage change of the Harmonised Index of Consumer Prices (HICP) published by Eurostat. The time period under consideration is 1999:Q4-2023Q1 (which yields the total number of observations equal to 95), the final number of forecasters is  $p = 56$ , and the

prediction horizon is 2-quarters ahead. (3) Unemployment rate which refers to Eurostat’s definition and it is calculated as percentage of the labor force. The time period under consideration is 1999:Q3-2022Q4 (which yields the total number of observations equal to 94), the final number of forecasters is  $p = 45$ , and the prediction horizon is 2-quarters ahead.

We consider four choices of the training sample:  $R \in \{20, 30, 40, 50\}$ , the estimation window is rolled over the test sample to update the estimates in each point of time. The optimal number of factors in the forecast errors (denoted as  $q$  in equation (2.1)) is chosen using the standard data-driven method that uses the information criterion IC1 described in Bai and Ng (2002). In the majority of the cases the optimal number of factors was estimated to be equal to 1. To explore the benefits of using FGL and RD-FGL for forecast error quantification, we consider several alternative estimators of covariance/precision matrix of the idiosyncratic component in (3.2): (1) linear shrinkage estimator of covariance developed by Ledoit and Wolf (2004) further referred to as Factor LW (FLW); (2) nonlinear shrinkage estimator of covariance by Ledoit and Wolf (2017) (Factor NLW or FNLW); (3) POET (Fan et al. (2013)); (4) constrained  $\ell_1$ -minimization for inverse matrix estimator, CLIME (Cai et al. (2011)) (Factor CLIME or FCLIME); (5) nodewise regression developed by Meinshausen and Bühlmann (2006) (Factor MB or FMB). To examine the benefits of imposing sparsity on  $\Theta_\varepsilon$  we also include the factor model without sparsity assumption on the idiosyncratic error precision matrix (referred to as Not Sparse) – this corresponds to imposing  $\tau = 0$  in (B.1). To examine the benefits of using factor structure, we include several counterparts of the aforementioned models that directly estimate precision of the forecast errors without estimating factors and loadings: GL, LW, NLW, CLIME, and MB. For RD-FGL, similarly to the simulations, we include different specifications of the smoothing function  $\psi(\cdot)$  and report the results for the case when the break parameter in factor loadings is estimated using

cross-valuation ( $\gamma = \hat{\gamma}$ ).

Our benchmark is the simple average with equal weights on all forecasters (referred to as EW). Going back to the discussion in Section 4 regarding setting  $\beta = 0$  in equation (4.5): as we pointed out, this corresponds to using only post-break sample for estimation which is suboptimal since the value of  $\beta$  is already chosen optimally from the grid that includes  $\beta = 0$  to minimize the MSFE. Hence, by construction, RD-FGL is superior to using only post-break data.

For RD-FGL the number of breaks for loadings and precision is estimated using the test for existence of break point (Bai and Perron (2003)): using their sequential procedure we search for up to three breaks and set the trimming parameter to 10% of the total number of observations, and the significance level at 5%. The location of the break points for each series is estimated using the one-at-a-time approach described in Subsection 4.3.

Table 1 compares the performance of FGL and RD-FGL with the competitors for predicting three macroeconomic indicators for Euro-area using a combination of ECB SPF forecasts. It reports the ratios of MSFE of each method to the MSFE of the EW combined forecast. Using the Model Confidence Set (MCS) of Hansen et al. (2011), we identify the set of superior models (SSM) for each series and horizon at 90% confidence level. Once SSM is identified, we rank these models according to the relative sample loss of the  $i$ -th model relative to the average across models in SSM, and report the ranking in Table 1.

There are three main findings that we learn from analyzing Table 1: **(1)** for all series factor-based models outperform non-factor ones. This means that incorporating the factor structure in the forecast errors improves forecasting performance. **(2)** for all series the Not Sparse model provides one of the worst performances. This means that the factor structure per se is not sufficient to achieve performance gains over EW, hence, it is necessary to impose

	GL	LW	NIW	CLIME	MB	POET	Not Sparse	FGL	FLW	FNIW	FCLIME	FMB	RD-FGL ( $\ell_2, \gamma = \hat{\gamma}$ )	RD-FGL ( $\ell_1, \gamma = \hat{\gamma}$ )	RD-FGL ( $\ell_g, \gamma = \hat{\gamma}$ )
<b>Real GDP growth</b>															
$R=20$	3.3820	4.7845	1.5768	7.3391	1.1164	3.8275	7.0076	1.3178	0.9961	0.8719	3.4197	0.9962	<b>0.4092</b>	0.4142	0.4128
Ranking										<sup>4</sup>			<sup>1</sup>	<sup>2</sup>	<sup>3</sup>
$R=30$	1.8984	1.2855	3.9910	5.9043	56.2033	0.9806	6.6966	0.9836	0.9372	3.5575	0.9865	0.8954	<b>0.4163</b>	0.4221	0.4204
Ranking													<sup>1</sup>	<sup>2</sup>	<sup>3</sup>
$R=40$	2.1372	1.7982	4.9564	7.8310	0.9619	1.0327	18.1483	0.9353	0.9467	2.9197	0.9732	1.4725	<b>0.4796</b>	0.4868	0.4863
Ranking													<sup>1</sup>	<sup>2</sup>	<sup>3</sup>
$R=50$	1.1706	1.4685	2.6014	3.9301	0.9594	1.0420	32.5874	0.9133	0.9399	2.5874	0.9860	0.9231	<b>0.4296</b>	0.4330	0.4329
Ranking												<sup>1</sup>	<sup>2</sup>	<sup>3</sup>	
<b>Inflation</b>															
$R=20$	0.9943	0.7184	0.5277	1.0063	2.9301	0.5431	1.7126	0.8970	0.6740	0.6617	0.5182	0.7917	0.3682	0.3711	<b>0.3678</b>
Ranking													<sup>1</sup>	<sup>3</sup>	<sup>2</sup>
$R=30$	0.9659	0.7403	0.7628	1.0158	0.9019	0.5582	1.0544	0.9493	0.7301	0.5959	0.6970	0.7995	<b>0.4326</b>	0.4353	0.4832
Ranking						<sup>4</sup>							<sup>1</sup>	<sup>3</sup>	<sup>2</sup>
$R=40$	0.8584	0.7780	0.5999	1.0122	0.9093	0.9463	1.0360	0.6641	0.5910	0.5692	0.6831	0.9416	0.3145	0.3355	<b>0.3088</b>
Ranking													<sup>1</sup>	<sup>2</sup>	<sup>3</sup>
$R=50$	0.9014	0.7694	0.5576	0.6314	0.9375	1.0170	3.9136	0.5828	0.5903	0.5336	0.9573	0.7392	<b>0.4184</b>	0.4180	0.4314
Ranking													<sup>1</sup>	<sup>2</sup>	<sup>3</sup>
<b>Unemployment rate</b>															
$R=20$	0.8731	0.9956	0.8496	0.9884	19.8034	0.9823	1.3067	0.9178	0.8732	<b>0.7557</b>	0.9247	1.6464	0.8951	0.9042	0.9054
Ranking	<sup>4</sup>				<sup>6</sup>			<sup>5</sup>	<sup>3</sup>	<sup>2</sup>	<sup>1</sup>				
$R=30$	0.8185	0.8763	0.8358	0.9713	42.1746	0.9815	1.7945	<b>0.8031</b>	0.8599	0.8320	0.9647	3.8414	1.0769	1.1357	1.1237
Ranking	<sup>3</sup>	<sup>2</sup>	<sup>5</sup>	<sup>1</sup>						<sup>4</sup>					
$R=40$	1.3733	1.3748	1.3526	1.4163	1.7992	0.9793	11.8839	<b>0.9333</b>	1.2844	1.2652	0.9941	1.1644	1.3359	1.3430	1.3383
Ranking*						<sup>3</sup>		<sup>1</sup>			<sup>2</sup>				
$R=50$	1.0145	1.4473	1.7531	1.2623	59.6932	1.4259	7.0196	<b>0.8419</b>	1.1558	1.1355	0.9896	2.7930	1.3761	1.4660	1.4607
Ranking	<sup>3</sup>							<sup>1</sup>			<sup>2</sup>				

Table 1: **Prediction of Quarterly Macroeconomic Variables for Euro-area Using ECB SPF Forecasts.** MSFEs of competing methods are reported for each value of  $R$ , where  $R$  indicates the length of the training window. Ratio indicates the ratio to MSFE of the Equal-Weighted combined forecast. Models with the lowest ratio are in bold. Models that belong to the SSM according to MCS test are ranked according to the relative sample loss of the  $i$ -th model relative to the average across models in SSM (at 90% confidence level). For unemployment rate ( $R=40$ ), the ranking marked with a star means that EW is included in the SSM (EW is ranked the 4th).



sparsity on the precision matrix of the idiosyncratic components. **(3)** For real GDP growth and inflation series RD-FGL is always included in the SSM. For the unemployment rate, FGL outperforms RD-FGL. This result is supported by the behavior observed in the actual series: real GDP growth and inflation exhibit strong breaks following the global financial crisis and Covid pandemic, however this is not the case for the unemployment rate series that did not have strong breaks throughout the whole sample period.

## 7 Conclusions

In this paper we develop a unified framework to generalize network inference under a factor structure in the presence of structural breaks. We overcome the challenge of using graphical models under the factor structure and provide a simple approach that allows practitioners to combine a large number of forecasts when experts tend to make common mistakes. Using pre- and post-break data, our new approach to forecast combinations breaks down forecast errors into common and unique parts which improves the accuracy of the combined forecast. **We allow the structural breaks to affect factor loadings and idiosyncratic precision matrix.** For the ease of practical use we develop a scalable optimization procedure for RD-FGL, based on the ADMM. The empirical application to forecasting macroeconomic series using the data of the ECB Survey of Professional Forecasters shows that incorporating (i) factor structure in the forecast errors together with (ii) sparsity in the precision matrix of the idiosyncratic components and (iii) regime-dependent combination weights improves the performance of a combined forecast.

## References

- Bai, J. (2010). Common breaks in means and variances for panel data. *Journal of Econometrics*, 157(1):78–92. 21
- Bai, J., Han, X., and Shi, Y. (2020). Estimation and inference of change points in high-dimensional factor models. *Journal of Econometrics*, 219(1):66–100. 20
- Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221. 6, 29
- Bai, J. and Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, pages 47–78. 21
- Bai, J. and Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of applied econometrics*, 18(1):1–22. 21, 30
- Bates, J. M. and Granger, C. W. J. (1969). The combination of forecasts. *Operations Research*, 20(4):451–468. 1, 7
- Cai, T., Liu, W., and Luo, X. (2011). A constrained l1-minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607. 29
- Chamberlain, G. and Rothschild, M. (1983). Arbitrage, factor structure, and mean-variance analysis on large asset markets. *Econometrica*, 51(5):1281–1304. 6
- Chan, Y. L., Stock, J. H., and Watson, M. W. (1999). A dynamic factor model framework for forecast combination. *Spanish Economic Review*, 1(2):91–121. 4
- Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5(4):559–583. 1
- Diebold, F. and Shin, M. (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting*, 35(4):1679–1691. 1, 5
- Fan, J., Liao, Y., and Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B*, 75(4):603–680. 11, 29
- Fan, J., Liu, H., and Wang, W. (2018). Large covariance estimation through elliptical factor models. *The Annals of Statistics*, 46(4):1383–1414. 11, 12
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the Graphical Lasso. *Biostatistics*, 9(3):432–441. 2, 8, 37, 38
- Hallac, D., Park, Y., Boyd, S., and Leskovec, J. (2017). Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’17, pages 205–213, New York, NY, USA. ACM. 13, 19

- Hansen, B. E. (2008). Least-squares forecast averaging. *Journal of Econometrics*, 146(2):342–350. [26](#)
- Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497. [30](#)
- Janková, J. and van de Geer, S. (2018). Inference in high-dimensional graphical models. *Handbook of Graphical Models*, Chapter 14, pages 325–351. CRC Press. [13](#), [38](#)
- Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411. [24](#), [29](#), [49](#)
- Ledoit, O. and Wolf, M. (2017). Nonlinear shrinkage of the covariance matrix for portfolio selection: Markowitz meets goldilocks. *The Review of Financial Studies*, 30(12):4349–4388. [29](#)
- Lee, T.-H. and Seregina, E. (2023). Optimal Portfolio Using Factor Graphical Lasso. *Journal of Financial Econometrics*. [1](#), [2](#), [9](#), [12](#)
- Li, Q., Ouyang, D., and Racine, J. S. (2013). Categorical semiparametric varying-coefficient models. *Journal of Applied Econometrics*, 28(4):551–579. [14](#)
- Lu, J., Kolar, M., and Liu, H. (2015). Post-regularization inference for time-varying non-paranormal graphical models. *Journal of Machine Learning Research*, 18:203:1–203:78. [13](#)
- Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462. [2](#), [29](#)
- Mohan, K., London, P., Fazel, M., Witten, D., and Lee, S.-I. (2014). Node-based learning of multiple gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488. [19](#)
- Parikh, N. and Boyd, S. (2014). Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239. [19](#), [45](#)
- Pástor, L. and Stambaugh, R. F. (2001). The equity premium and structural breaks. *The Journal of Finance*, 56(4):1207–1239. [14](#)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. [24](#)
- Pesaran, M. H., Pettenuzzo, D., and Timmermann, A. (2006). Forecasting time series subject to multiple structural breaks. *The Review of Economic Studies*, 73(4):1057–1084. [3](#), [14](#), [17](#)
- Ravikumar, P., J. Wainwright, M., Raskutti, G., and Yu, B. (2011). High-dimensional covariance estimation by minimizing  $\ell_1$ -penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980. [36](#)

- Shi, Z., Su, L., and Xie, T. (2020). High dimensional forecast combinations under latent structures. *arXiv:2010.09477*. 27
- Shi, Z., Su, L., and Xie, T. (2022). l2-Relaxation: With Applications to Forecast Combination and Portfolio Analysis. *The Review of Economics and Statistics*, pages 1–44. 9
- Smith, J. and Wallis, K. F. (2009). A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics*, 71(3):331–355. 7
- Smith, S. C. and Timmermann, A. (2021). Break risk. *The Review of Financial Studies*, 34(4):2045–2100. 21
- Stekhoven, D. J. (2022). *missForest: Nonparametric Missing Value Imputation using Random Forest*. R package version 1.5. 28
- Stekhoven, D. J. and Buhlmann, P. (2012). Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118. 28
- Stock, J. H. and Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179. 6, 7, 25
- Su, L. and Wang, X. (2017). On time-varying factor models: Estimation and testing. *Journal of Econometrics*, 198(1):84–101. 14, 15, 19
- Timmermann, A. (2006). Chapter 4: Forecast Combinations. volume 1 of *Handbook of Economic Forecasting*, pages 135–196. Elsevier. 7
- Zhou, S., Lafferty, J., and Wasserman, L. (2010). Time varying undirected graphs. *Machine Learning*, 80(2):295–319. 13

SUPPLEMENTAL APPENDIX TO  
“COMBINING FORECASTS UNDER STRUCTURAL BREAKS  
USING GRAPHICAL LASSO”

## Appendix A Graphical Lasso Algorithm

Recall that we have  $p$  competing forecasts of the univariate series  $y_t$ ,  $t = 1, \dots, T$ . Let  $\mathbf{e}_t = (e_{1t}, \dots, e_{pt})' \sim \mathcal{N}(\mathbf{0}, \Sigma)$  be a  $p \times 1$  vector of forecast errors. Assume they follow a Gaussian distribution. The precision matrix  $\Sigma^{-1} \equiv \Theta$  contains information about partial covariances between the variables. For instance, if  $\theta_{ij}$ , which is the  $ij$ -th element of the precision matrix, is zero, then the variables  $i$  and  $j$  are conditionally independent, given the other variables.

Let  $\mathbf{W}$  be the estimate of  $\Sigma$ . Given a sample  $\{\mathbf{e}_t\}_{t=1}^T$ , let  $\mathbf{S} = (1/T) \sum_{t=1}^T (\mathbf{e}_t)(\mathbf{e}_t)'$  denote the sample covariance matrix, which can be used as a choice for  $\mathbf{W}$ . Also, let  $\widehat{\Gamma}^2 \equiv \text{diag}(\mathbf{W})$  and its  $(i, j)$ -th element is denoted as  $\widehat{\gamma}_{ij}$ . We can write down truncated Gaussian log-likelihood (up to constants)  $l(\Theta) = \log \det(\Theta) - \text{tr}(\mathbf{W}\Theta)$ . When  $\mathbf{W} = \mathbf{S}$ , the maximum likelihood estimator of  $\Theta$  is  $\widehat{\Theta} = \mathbf{S}^{-1}$ . The objective function associated with truncated Gaussian log-likelihood is also known as Bregman divergence and was shown to be applicable for non-Gaussian distributions (Ravikumar et al. (2011)).

In the high-dimensional settings it is necessary to regularize the precision matrix, which means that some edges will be zero. A natural way to induce sparsity in the estimation of precision matrix is to add penalty to the maximum likelihood and use the connection between the precision matrix and regression coefficients to maximize the following penalized log-likelihood that weighs the variables by their scale:

$$\widehat{\Theta}_\tau = \arg \min_{\Theta = \Theta'} \text{tr}(\mathbf{W}\Theta) - \log \det(\Theta) + \tau \sum_{i \neq j} \widehat{\gamma}_{ii} \widehat{\gamma}_{jj} |\theta_{ij}|, \quad (\text{A.1})$$

over positive definite symmetric matrices, where  $\tau \geq 0$  is a penalty parameter for the off-diagonal elements. We refer to the objective function in (A.1) as a “weighted penalized log-likelihood”. The subscript  $\tau$  in  $\widehat{\Theta}_\tau$  means that the solution of the optimization problem in (A.1) will depend upon the choice of the tuning parameter. In order to simplify notation, we will omit the subscript.

Define the following partitions of  $\mathbf{W}$ ,  $\mathbf{S}$  and  $\Theta$ :

$$\mathbf{W} = \begin{pmatrix} \underbrace{\mathbf{W}_{11}}_{(p-1) \times (p-1)} & \underbrace{\mathbf{w}_{12}}_{(p-1) \times 1} \\ \mathbf{w}'_{12} & w_{22} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} \underbrace{\mathbf{S}_{11}}_{(p-1) \times (p-1)} & \underbrace{\mathbf{s}_{12}}_{(p-1) \times 1} \\ \mathbf{s}'_{12} & s_{22} \end{pmatrix}, \Theta = \begin{pmatrix} \underbrace{\Theta_{11}}_{(p-1) \times (p-1)} & \underbrace{\theta_{12}}_{(p-1) \times 1} \\ \theta'_{12} & \theta_{22} \end{pmatrix}. \quad (\text{A.2})$$

Let  $\beta \equiv -\theta_{12}/\theta_{22}$ . The idea of GL is to set  $\mathbf{W} = \mathbf{S} + \tau \mathbf{I}$  in (A.1) and combine the gradient of (A.1) with the formula for partitioned inverses to obtain the following  $\ell_1$ -regularized quadratic program

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{p-1}} \left\{ \frac{1}{2} \beta' \mathbf{W}_{11} \beta - \beta' \mathbf{s}_{12} + \sum_{i \neq j} \tau \hat{\gamma}_{ii} \hat{\gamma}_{jj} \|\beta\|_1 \right\}. \quad (\text{A.3})$$

As shown by Friedman et al. (2008), (A.3) can be viewed as a LASSO regression, where the LASSO estimates are functions of the inner products of  $\mathbf{W}_{11}$  and  $\mathbf{s}_{12}$ . Hence, (A.1) is equivalent to  $p$  coupled LASSO problems. Once we obtain  $\hat{\beta}$ , we can estimate the entries  $\Theta$  using the formula for partitioned inverses. The weighted GL procedure is summarized in Algorithm A.1.

---

**Algorithm A.1** Weighted Graphical LASSO

---

- 1: Initialize  $\mathbf{W} = \mathbf{S} + \tau \mathbf{I}$ , with  $w_{ii} = s_{ii}$ . The diagonal of  $\mathbf{W}$  remains the same in what follows.
- 2: Estimate a sparse  $\Theta$  using the following weighted Graphical LASSO objective function:

$$\hat{\Theta}_\tau = \arg \min_{\Theta = \Theta'} \text{tr}(\mathbf{W}\Theta) - \log \det(\Theta) + \tau \sum_{i \neq j} \hat{\gamma}_{ii} \hat{\gamma}_{jj} |\theta_{ij}|,$$

over positive definite symmetric matrices.

- 3: Repeat for  $j = 1, \dots, p, 1, \dots, p, \dots$  until convergence:
  - Partition  $\mathbf{W}$  into part 1: all but the  $j$ -th row and column, and part 2: the  $j$ -th row and column.
  - Solve the score equations using the cyclical coordinate descent:

$$\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \tau \hat{\gamma}_{ii} \hat{\gamma}_{jj} \cdot \text{Sign}(\boldsymbol{\beta}) = \mathbf{0}.$$

This gives a  $(p-1) \times 1$  vector solution  $\hat{\boldsymbol{\beta}}$ .

- Update  $\hat{\mathbf{w}}_{12} = \mathbf{W}_{11}\hat{\boldsymbol{\beta}}$ .

- 4: In the final cycle (for  $i = 1, \dots, p$ ) solve for

$$\frac{1}{\hat{\theta}_{22}} = w_{22} - \hat{\boldsymbol{\beta}}' \hat{\mathbf{w}}_{12}, \quad \hat{\boldsymbol{\theta}}_{12} = -\hat{\theta}_{22} \hat{\boldsymbol{\beta}}.$$

---

As was shown in Friedman et al. (2008), the estimator produced by Algorithm A.1 is guaranteed to be positive definite. Furthermore, Janková and van de Geer (2018) showed that Algorithm A.1 is guaranteed to converge and produces consistent estimator of precision matrix under certain sparsity conditions.

## Appendix B Factor Graphical LASSO

---

**Algorithm B.2** Factor Graphical LASSO (FGL)

---

- 1: Estimate factors,  $\widehat{\mathbf{f}}_t$ , and factor loadings,  $\widehat{\mathbf{B}}$ , using PCA. Obtain  $\widehat{\Sigma}_f = \frac{1}{T} \sum_{t=1}^T \widehat{\mathbf{f}}_t \widehat{\mathbf{f}}_t'$ ,  $\widehat{\Theta}_f = \widehat{\Sigma}_f^{-1}$ ,  $\widehat{\boldsymbol{\varepsilon}}_t = \mathbf{e}_t - \widehat{\mathbf{B}} \widehat{\mathbf{f}}_t$ , and  $\widehat{\Sigma}_\varepsilon = \frac{1}{T} \sum_{t=1}^T \widehat{\boldsymbol{\varepsilon}}_t \widehat{\boldsymbol{\varepsilon}}_t'$ .
- 2: Estimate a sparse  $\Theta_\varepsilon$  using the weighted Graphical LASSO in initialized with  $\mathbf{W}_\varepsilon = \widehat{\Sigma}_\varepsilon + \tau \mathbf{I}$ :

$$\widehat{\Theta}_{\varepsilon,\tau} = \arg \min_{\Theta_\varepsilon = \Theta_\varepsilon'} \text{tr}(\mathbf{W}_\varepsilon \Theta_\varepsilon) - \log \det(\Theta_\varepsilon) + \tau \sum_{i \neq j} \widehat{\gamma}_{\varepsilon,ii} \widehat{\gamma}_{\varepsilon,jj} |\theta_{\varepsilon,ij}|. \quad (\text{B.1})$$

where  $\widehat{\gamma}_{\varepsilon,ii}$  is the  $(i, i)$ -th element of  $\widehat{\Gamma}_\varepsilon^2 \equiv \text{diag}(\mathbf{W}_\varepsilon)$ .

- 3: Use  $\widehat{\Theta}_f$  from Step 1 and  $\widehat{\Theta}_\varepsilon$  from Step 2 to estimate  $\Theta$  using the Sherman-Morrison-Woodbury formula:

$$\widehat{\Theta} = \widehat{\Theta}_\varepsilon - \widehat{\Theta}_\varepsilon \widehat{\mathbf{B}} [\widehat{\Theta}_f + \widehat{\mathbf{B}}' \widehat{\Theta}_\varepsilon \widehat{\mathbf{B}}]^{-1} \widehat{\mathbf{B}}' \widehat{\Theta}_\varepsilon. \quad (\text{B.2})$$


---

Let  $\widehat{\Theta}_{\varepsilon,\tau}$  be the solution to (B.1) for a fixed  $\tau$ . To choose the optimal shrinkage intensity coefficient, we minimize the following Bayesian Information Criterion (BIC) using grid search:

$$\text{BIC}(\tau) \equiv T \left[ \text{tr}(\widehat{\Theta}_{\varepsilon,\tau} \widehat{\Sigma}_\varepsilon) - \log \det(\widehat{\Theta}_{\varepsilon,\tau}) \right] + (\log T) \sum_{i \leq j} \mathbf{1} \left[ \widehat{\theta}_{\varepsilon,\tau,ij} \neq 0 \right]. \quad (\text{B.3})$$

The grid  $\mathcal{G} \equiv \{\tau_1, \dots, \tau_M\}$  is constructed as follows: the maximum value in the grid,  $\tau_M$ , is set to be the smallest value for which all the off-diagonal entries of  $\widehat{\Theta}_{\varepsilon,\tau_M}$  are zero, that is, the maximum modulus of the off-diagonal entries of  $\widehat{\Sigma}_\varepsilon$ . The smallest value of the grid,  $\tau_1 \in \mathcal{G}$ , is determined as  $\tau_1 \equiv \vartheta \tau_M$  for a constant  $0 < \vartheta < 1$ . The remaining grid values  $\tau_1, \dots, \tau_M$  are constructed in the ascending order from  $\tau_1$  to  $\tau_M$  on the log scale:

$$\tau_i = \exp \left( \log(\tau_1) + \frac{i-1}{M-1} \log(\tau_M/\tau_1) \right), \quad i = 2, \dots, M-1.$$

We use  $\vartheta = \sqrt{\log p/T} + 1/\sqrt{p}$  (motivated by the convergence rate from Theorem 1) and  $M = 10$  in the simulations and the empirical exercise.



## Appendix C Proof of Theorem 1

We first present a lemma which is used in the proof.

**Lemma 1.**

- (a)  $\|\Theta\|_1 = \mathcal{O}(d_T)$ .
- (b)  $a \geq c > 0$ , where  $a$  was defined in Section 3 and  $c$  was defined in Assumption (A.1) (ii).
- (c)  $|\hat{a} - a| = \mathcal{O}_P(\varrho_T d_T s_T)$ , where  $\hat{a}$  was defined in Section 3.

*Proof.*

- (a) To prove part (a) we use the following matrix inequality which holds for any  $\mathbf{A} \in \mathcal{S}_p$ :

$$\|\mathbf{A}\|_1 = \|\mathbf{A}\|_\infty \leq \sqrt{d(\mathbf{A})} \|\mathbf{A}\|_2, \quad (\text{C.1})$$

where  $d(\mathbf{A})$  was defined in Section 4. The proof of (C.1) is a straightforward consequence of the Schwarz inequality.

Sherman-Morrison-Woodbury formula together with (C.1) and Assumptions (B.1)-(B.3) yield:

$$\begin{aligned} \|\Theta\|_1 &\leq \|\Theta_\varepsilon\|_1 + \|\Theta_\varepsilon \mathbf{B} [\Theta_f + \mathbf{B}' \Theta_\varepsilon \mathbf{B}]^{-1} \mathbf{B}' \Theta_\varepsilon\|_1 \\ &= \mathcal{O}(\sqrt{d_T}) + \mathcal{O}\left(\sqrt{d_T} \cdot p \cdot \frac{1}{p} \cdot \sqrt{d_T}\right) = \mathcal{O}(d_T). \end{aligned} \quad (\text{C.2})$$

- (b) Under Assumption (A.1):

$$a = \boldsymbol{\nu}'_p \Theta \boldsymbol{\nu}_p / p \geq c > 0.$$

- (c) Using the Hölders inequality, we have

$$\begin{aligned} |\hat{a} - a| &= \left| \frac{\boldsymbol{\nu}'_p (\hat{\Theta} - \Theta) \boldsymbol{\nu}_p}{p} \right| \leq \frac{\|(\hat{\Theta} - \Theta) \boldsymbol{\nu}_p\|_1 \|\boldsymbol{\nu}_p\|_\infty}{p} \leq \|\hat{\Theta} - \Theta\|_1 \\ &= \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1), \end{aligned}$$

where the last rate is obtained using the assumptions of Theorem 1.

□

## C.1 Proof of Theorem 1

First, note that the forecast combination weight can be written as

$$\begin{aligned}\widehat{\mathbf{w}} - \mathbf{w} &= \frac{\left( (a\widehat{\Theta}\boldsymbol{\iota}_p) - (\widehat{a}\Theta\boldsymbol{\iota}_p) \right) / p}{\widehat{a}a} \\ &= \frac{\left( (a\widehat{\Theta}\boldsymbol{\iota}_p) - (a\Theta\boldsymbol{\iota}_p) + (a\Theta\boldsymbol{\iota}_p) - (\widehat{a}\Theta\boldsymbol{\iota}_p) \right) / p}{\widehat{a}a}.\end{aligned}$$

As shown in Callot et al. (2019), the above can be rewritten as

$$\|\widehat{\mathbf{w}} - \mathbf{w}\|_1 \leq \frac{a \frac{\|(\widehat{\Theta} - \Theta)\boldsymbol{\iota}_p\|_1}{p} + |a - \widehat{a}| \frac{\|\Theta\boldsymbol{\iota}_p\|_1}{p}}{|\widehat{a}|a}. \quad (\text{C.3})$$

Prior to bounding the terms in (C.3), we first present an inequality which is used in the derivations. Let  $\mathbf{A} \in \mathbb{R}^{p \times p}$  and  $\mathbf{v} \in \mathbb{R}^{p \times 1}$ . Also, let  $\mathbf{A}_j$  and  $\mathbf{A}'_j$  be a  $p \times 1$  and  $1 \times p$  row and column vectors in  $\mathbf{A}$ , respectively.

$$\begin{aligned}\|\mathbf{A}\mathbf{v}\|_1 &= |\mathbf{A}'_1\mathbf{v}| + \dots + |\mathbf{A}'_p\mathbf{v}| \leq \|\mathbf{A}_1\|_1 \|\mathbf{v}\|_\infty + \dots + \|\mathbf{A}_p\|_1 \|\mathbf{v}\|_\infty \\ &= \left( \sum_{j=1}^p \|\mathbf{A}_j\|_1 \right) \|\mathbf{v}\|_\infty \leq p \max_j \|\mathbf{A}_j\|_1 \|\mathbf{v}\|_\infty.\end{aligned} \quad (\text{C.4})$$

Hölder's inequality was used to obtain each inequality in (C.4). If  $\mathbf{A} \in \mathcal{S}_p$ , then the last expression can be further reduced to  $p\|\mathbf{A}\|_1 \|\mathbf{v}\|_\infty$ .

Let us now bound the right-hand side of (C.3). In the numerator we have:

$$\left\| \frac{(\widehat{\Theta} - \Theta)\boldsymbol{\iota}_p}{p} \right\|_1 \leq \|\Theta\|_1 = \mathcal{O}_P(\varrho_T d_T s_T), \quad (\text{C.5})$$

the rates was derived in Lee and Seregina (2020), and the inequality follows from (C.4).

$$\frac{\|\Theta\boldsymbol{\iota}_p\|_1}{p} \leq \|\Theta\|_1 = \mathcal{O}(d_T), \quad (\text{C.6})$$

where the rate follows from Lemma 1 (a) and the inequality is obtained from (C.4). Com-

binning (C.5), (C.6), and Lemma 1 (c) we get:

$$\begin{aligned} a \frac{\|(\widehat{\Theta} - \Theta)\iota_p\|_1}{p} + |a - \widehat{a}| \frac{\|\Theta\iota_p\|_1}{p} &= \mathcal{O}(1) \cdot \mathcal{O}_P(\varrho_T d_T s_T) + \mathcal{O}_P(\varrho_T d_T s_T) \cdot \mathcal{O}(d_T) \\ &= \mathcal{O}_P(\varrho_T d_T^2 s_T) = o_P(1), \end{aligned} \tag{C.7}$$

where the last equality holds under the assumptions of Theorem 1.

For the denominator of (C.3) it easy to see that  $|\widehat{a}|a = \mathcal{O}_P(1)$  using the results of Lemma 1 (b).

For the MSFE part of Theorem 1, using Lemma 1 (b)-(c), we get

$$\left| \frac{\widehat{a}^{-1}}{a^{-1}} - 1 \right| = \frac{|a - \widehat{a}|}{|\widehat{a}|} = \mathcal{O}_P(\varrho_T d_T s_T) = o_P(1),$$

where the last rate is obtained using the assumptions of Theorem 1.

## Appendix D Implementation via ADMM Algorithm

To enable practical implementation of the RD-FGL, we develop an optimization procedure using ADMM algorithm to solve the convex optimization problem in (4.5).

First, we need to reformulate the unconstrained problem in (4.5) as a constrained problem which can be solved using ADMM:

$$\{\widehat{\Theta}_{\varepsilon,j}\}_{j=1}^{N+1} = \arg \min_{\{\Theta_{\varepsilon,j}\}_{j=1}^{N+1}} \sum_{j=1}^{N+1} n_j \left[ \text{tr} \left( \widehat{\Sigma}_{\varepsilon,j} \Theta_{\varepsilon,j} \right) - \log \det \Theta_{\varepsilon,j} \right] + \alpha \|\Theta_{\varepsilon,j}\|_{\text{od},1} \quad (\text{D.1})$$

$$+ \beta \sum_{j=2}^{N+1} \psi(\Theta_{\varepsilon,j} - \Theta_{\varepsilon,j-1})$$

$$\text{s.t. } \mathbf{Z}_{j,0} = \Theta_{\varepsilon,j}, \text{ for } j = 1, \dots, N+1 \quad (\text{D.2})$$

$$\left( \mathbf{Z}_{j-1,1}, \mathbf{Z}_{j,2} \right) = \left( \Theta_{\varepsilon,j-1}, \Theta_{\varepsilon,j} \right), \text{ for } j = 2, \dots, N+1. \quad (\text{D.3})$$

Let  $\mathbf{Z} = \{\mathbf{Z}_0, \mathbf{Z}_1, \mathbf{Z}_2\} = \left\{ \left( \mathbf{Z}_{1,0}, \dots, \mathbf{Z}_{N+1,0} \right), \left( \mathbf{Z}_{1,1}, \dots, \mathbf{Z}_{N,1} \right), \left( \mathbf{Z}_{2,2}, \dots, \mathbf{Z}_{N+1,2} \right) \right\}$ .

Let  $\mathbf{U} = \{\mathbf{U}_0, \mathbf{U}_1, \mathbf{U}_2\} = \left\{ \left( \mathbf{U}_{1,0}, \dots, \mathbf{U}_{N+1,0} \right), \left( \mathbf{U}_{1,1}, \dots, \mathbf{U}_{N,1} \right), \left( \mathbf{U}_{2,2}, \dots, \mathbf{U}_{N,2} \right) \right\}$  be the scaled dual variable and  $\rho > 0$  is the ADMM penalty parameter. Now we can use scaled ADMM to write down the augmented Lagrangian:

$$\begin{aligned} \mathcal{L}_\rho(\Theta_\varepsilon, \mathbf{Z}, \mathbf{U}) &= \sum_{j=1}^{N+1} n_j \left[ \text{tr} \left( \widehat{\Sigma}_{\varepsilon,j} \Theta_{\varepsilon,j} \right) - \log \det \Theta_{\varepsilon,j} \right] + \alpha \|\mathbf{Z}_{j,0}\|_{\text{od},1} \quad (\text{D.4}) \\ &+ \beta \sum_{j=2}^{N+1} \psi(\mathbf{Z}_{j,2} - \mathbf{Z}_{j-1,1}) \\ &+ \left( \frac{\rho}{2} \right) \sum_{j=1}^{N+1} \left( \|\Theta_{\varepsilon,j} - \mathbf{Z}_{j,0} + \mathbf{U}_{j,0}\|_F^2 - \|\mathbf{U}_{j,0}\|_F^2 \right) \\ &+ \left( \frac{\rho}{2} \right) \sum_{j=2}^{N+1} \left( \|\Theta_{\varepsilon,j-1} - \mathbf{Z}_{j-1,1} + \mathbf{U}_{j-1,1}\|_F^2 - \frac{\rho}{2} \|\mathbf{U}_{j-1,1}\|_F^2 \right. \\ &\left. + \|\Theta_{\varepsilon,j} - \mathbf{Z}_{j,2} + \mathbf{U}_{j,2}\|_F^2 - \|\mathbf{U}_{j,2}\|_F^2 \right). \end{aligned}$$

Let  $k$  denote the iteration number, then ADMM consists of the following iterative updates:

$$\Theta_{\varepsilon,j}^{k+1} \equiv \arg \min_{\Theta_{\varepsilon} > 0} \mathcal{L}_{\rho}(\Theta_{\varepsilon}, \mathbf{Z}^k, \mathbf{U}^k), \quad (\text{D.5})$$

$$\mathbf{Z}^{k+1} = \begin{bmatrix} \mathbf{Z}_0^{k+1} \\ \mathbf{Z}_1^{k+1} \\ \mathbf{Z}_2^{k+1} \end{bmatrix} \equiv \arg \min_{\mathbf{Z}_0, \mathbf{Z}_1, \mathbf{Z}_2} \mathcal{L}_{\rho}(\Theta_{\varepsilon}^{k+1}, \mathbf{Z}, \mathbf{U}^k), \quad (\text{D.6})$$

$$\mathbf{U}^{k+1} = \begin{bmatrix} \mathbf{U}_0^{k+1} \\ \mathbf{U}_1^{k+1} \\ \mathbf{U}_2^{k+1} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{U}_0^k \\ \mathbf{U}_1^k \\ \mathbf{U}_2^k \end{bmatrix} + \begin{bmatrix} \Theta_{\varepsilon}^{k+1} - \mathbf{Z}_0^{k+1} \\ (\Theta_{\varepsilon,1}^{k+1}, \dots, \Theta_{\varepsilon,N}^{k+1}) - \mathbf{Z}_1^{k+1} \\ (\Theta_{\varepsilon,2}^{k+1}, \dots, \Theta_{\varepsilon,N+1}^{k+1}) - \mathbf{Z}_2^{k+1} \end{bmatrix}. \quad (\text{D.7})$$

**The  $\mathbf{Z}$  step:**

The updating rule in (D.6) is easily recognized to be the element-wise soft thresholding operator. However, we need to split it into two updates since  $(\mathbf{Z}_1, \mathbf{Z}_2)$  have to be updated jointly. Therefore, the update for  $\mathbf{Z}_{j,0}^{k+1}$  will be:

$$\mathbf{Z}_{j,0}^{k+1} \equiv S_{\alpha/\rho}(\Theta_{\varepsilon,j}^{k+1} + \mathbf{U}_{j,0}^k), \quad (\text{D.8})$$

where  $S_{\alpha/\rho}(\cdot)$  is the element-wise soft-thresholding operator.

We will solve a separate update for each  $(\mathbf{Z}_{j,2}, \mathbf{Z}_{j-1,1})$  pair for  $j = 2, \dots, N + 1$ :

$$\begin{aligned} (\mathbf{Z}_{j,2}^{k+1}, \mathbf{Z}_{j-1,1}^{k+1}) = \arg \min_{\mathbf{Z}_{j,2}, \mathbf{Z}_{j-1,1}} & \left( \frac{\rho}{2} \right) \left( \|\Theta_{\varepsilon,j} - \mathbf{Z}_{j,2} + \mathbf{U}_{j,2}\|_F^2 \right. \\ & \left. + \|\Theta_{\varepsilon,j-1} - \mathbf{Z}_{j-1,1} + \mathbf{U}_{j-1,1}\|_F^2 + \beta\psi(\mathbf{Z}_{j,2} - \mathbf{Z}_{j-1,1}) \right). \end{aligned} \quad (\text{D.9})$$

Note that (D.9) is guaranteed to converge to a fixed point since it can be written as a proximal operator:

$$(\mathbf{Z}_{j,2}^{k+1}, \mathbf{Z}_{j-1,1}^{k+1}) = \text{prox}_{\frac{\beta}{\rho}\psi(\cdot)} \left( \Theta_{\varepsilon,j} + \mathbf{U}_{j,2}, \Theta_{\varepsilon,j-1} + \mathbf{U}_{j-1,1} \right) \quad (\text{D.10})$$

**Remark 3.** A proximal operator of the scaled function  $\nu f$ , where  $\nu > 0$  can be expressed as:

$$\text{prox}_{\nu f}(v) = \underset{x}{\operatorname{argmin}} \left( f(x) + \frac{1}{2\nu} \|x - v\|_2^2 \right),$$

where  $f$  is a closed proper convex function. Note:

$$\text{prox}_{\nu f}(v) \approx v - \tau \nabla f(v).$$

*Parikh and Boyd (2014)* show that the fixed points of the proximal operator of  $f$  are precisely the minimizers of  $f$ , i.e.,  $\text{prox}_{\nu f}(x^*) = x^*$  if and only if  $x^*$  minimizes  $f$ .

**The  $\Theta$  step:**

The updating rule in (D.5) can be further simplified to obtain a closed-form solution. Rewrite (D.5):

$$\Theta_{\varepsilon,j}^{k+1} = \underset{\Theta_{\varepsilon,j} > 0}{\operatorname{argmin}} \operatorname{tr} \left( \widehat{\Sigma}_j \Theta_{\varepsilon,j} \right) - \log \det \Theta_{\varepsilon,j} + \frac{1}{2\eta} \left\| \Theta_{\varepsilon,j} - \mathbf{A}^k \right\|_F^2, \quad (\text{D.11})$$

where  $\mathbf{A}^k = \frac{\mathbf{Z}_{i,0}^k + \mathbf{Z}_{j-1,1}^k + \mathbf{Z}_{j,2}^k - \mathbf{U}_{j,0}^k - \mathbf{U}_{j-1,1}^k - \mathbf{U}_{j,2}^k}{3}$ , and  $\eta = \frac{n_j}{3\rho}$ .

Take the gradient of the updating rule in (D.11) in order to get an analytical solution:

$$\widehat{\Sigma}_{\varepsilon,j} - \Theta_{\varepsilon,j}^{-1,(k+1)} + \frac{1}{\eta} \left( \Theta_{\varepsilon,j}^{k+1} - \mathbf{A}^k \right) = 0, \quad (\text{D.12})$$

$$\frac{1}{\eta} \Theta_{\varepsilon,j}^{k+1} - \Theta_{\varepsilon,j}^{-1,(k+1)} = \frac{1}{\eta} \mathbf{A}^k - \widehat{\Sigma}_{\varepsilon,j}. \quad (\text{D.13})$$

Equation (D.13) implies that  $\Theta_{\varepsilon,j}^{k+1}$  and  $\frac{1}{\eta} \mathbf{A}^k - \widehat{\Sigma}_{\varepsilon,j}$  share the same eigenvectors.

Let  $\mathbf{Q}_j \mathbf{\Lambda}_j \mathbf{Q}_j'$  be the eigendecomposition of  $\frac{1}{\eta} \mathbf{A}^k - \widehat{\Sigma}_{\varepsilon,j}$ , where  $\mathbf{\Lambda}_j = \operatorname{diag}(\lambda_{1,j}, \dots, \lambda_{p,j})$ , and  $\mathbf{Q}_j' \mathbf{Q}_j = \mathbf{Q}_j \mathbf{Q}_j' = \mathbf{I}$ .<sup>8</sup> Pre-multiply (D.13) by  $\mathbf{Q}_j'$  and post-multiply it by  $\mathbf{Q}_j$ :

$$\frac{1}{\eta} \widetilde{\Theta}_{\varepsilon,j}^{k+1} - \widetilde{\Theta}_{\varepsilon,j}^{-1,(k+1)} = \mathbf{\Lambda}_j. \quad (\text{D.14})$$

---

<sup>8</sup>Note that in practice we need to check that  $\mathbf{A}^k$  is symmetric. If it is not, then we can define  $\widetilde{\mathbf{A}} \equiv \frac{\mathbf{A}^k + (\mathbf{A}^k)'}{2}$  and use it in the described algorithm instead of  $\mathbf{A}^k$ . Since  $\Theta_j$  is symmetric the results will not be affected.

Now construct a diagonal solution of (D.14):

$$\frac{1}{\eta} \tilde{v}_{i,j} - \frac{1}{\tilde{v}_{i,j}} = \lambda_{i,j}, \quad (\text{D.15})$$

where  $\tilde{v}_{i,j}$  denotes the  $i$ -th eigenvalue of  $\tilde{\Theta}_{\varepsilon,j}$ . Solving for  $\tilde{v}_{i,j}$  we get:

$$\tilde{v}_{i,j} = \frac{\lambda_{i,j} + \sqrt{\lambda_{i,j}^2 + \frac{4}{\eta}}}{2\eta^{-1}}. \quad (\text{D.16})$$

Now we can calculate  $\Theta_{\varepsilon,j}^{k+1}$  which satisfies the optimality condition in (D.14):

$$\Theta_{\varepsilon,j}^{k+1} = \frac{1}{2\eta^{-1}} \mathbf{Q}_j \left( \Lambda_j + \sqrt{\Lambda_j^2 + 4\eta^{-1} \mathbf{I}} \right) \mathbf{Q}_j'. \quad (\text{D.17})$$

Use the definition of  $\eta = \frac{n_j}{3\rho}$ :

$$\Theta_{\varepsilon,j}^{k+1} = \frac{n_j}{6\rho} \mathbf{Q}_j \left( \Lambda_j + \sqrt{\Lambda_j^2 + \frac{12\rho}{n_j} \mathbf{I}} \right) \mathbf{Q}_j'. \quad (\text{D.18})$$

Step (D.18) is the most computationally intensive task in the algorithm since the runtime of decomposing a  $p \times p$  matrix is  $\mathcal{O}(p^3)$ . Also, note that compared to standard ADMM without smoothing penalty  $\beta$ , (D.18) enforces stronger shrinkage. This is consistent with our motivation for the additional constraint - to smooth the estimator of precision matrix.

# Appendix E Additional Simulations

## E.1 No Break

In this section we present simulation results that augment the results in Section 5 by assuming there is no break in the DGP.

First, we explore behavior of precision matrix and weights estimates. The setup is the same as in Subsection 5.1, but there is no break in  $\Theta_\varepsilon$ : the value of the smallest coefficient is set to 0.1 and the value of the largest coefficient is set to 0.3.

Figure E.1 shows the averaged (over Monte Carlo simulations) errors of the estimators of the precision matrix  $\Theta$  and the optimal combination weight versus the sample size  $T$  in the logarithmic scale (base 2). For comparison, we include RD-FGL in all simulations. Since there is no break in the DGP, the tuning parameter for the factor loadings  $\gamma = 1$  and the value of  $\beta$  is estimated to be zero. Henceforth, all specifications of the smoothing function  $\psi(\cdot)$  yield similar results and we only include one of them RD-FGL ( $\ell_2$ ). As evidenced by Figure E.1, FGL and RD-FGL demonstrate superior performance over EW and non-factor based model (GL). FGL and RD-FGL have comparable performance, but since there is no break in DGP, FGL is more efficient. Furthermore, FGL and RD-FGL achieve lower estimation error in the combination weights, which leads to lower risk of the combined forecast. Also, note that the precision matrix estimated using the EW method also shows good convergence properties.

Second, we explore behavior of MSFE under no breaks. We set  $c_1 \in \{0, 0.75\}$  and  $c_2 = 0.9$ . We set  $M = 100$  and generate  $r = 5$  factors. To create  $\Lambda$  in (5.2) we take the first  $r$  rows of an upper triangular matrix from a Cholesky decomposition of the  $M \times M$  Toeplitz matrix parameterized by  $\rho = 0.9$ . The ranking of competing models was not very sensitive to varying values of  $\phi$ ,  $\rho$ ,  $c_2$ , and  $r$  – the results examining sensitivity to a grid of 10 different AR(1) coefficients  $\phi$  equidistant between 0 and 0.9, a grid of 10 different values of  $\rho$  equidistant between 0 and 0.9,  $c_2 \in \{0.6, 0.7, 0.8, 0.9\}$ , and  $r \in \{1, \dots, 7\}$  are available upon request.

One-step ahead forecasts are estimated from the factor-augmented autoregressive (FAR) models of orders  $k, l$ , denoted as FAR( $k, l$ ), defined in (5.3). We consider the FAR models of various orders, with  $k = 1, \dots, K$  and  $l = 1, \dots, L$ . We also consider the models without any lagged  $y$  or any factors. Therefore, the total number of forecasting models is  $p \equiv (1 + K) \times (1 + L)$ , which includes the forecasting models using naive average or no factors. We set  $K = 2$  and  $L = 7$ .



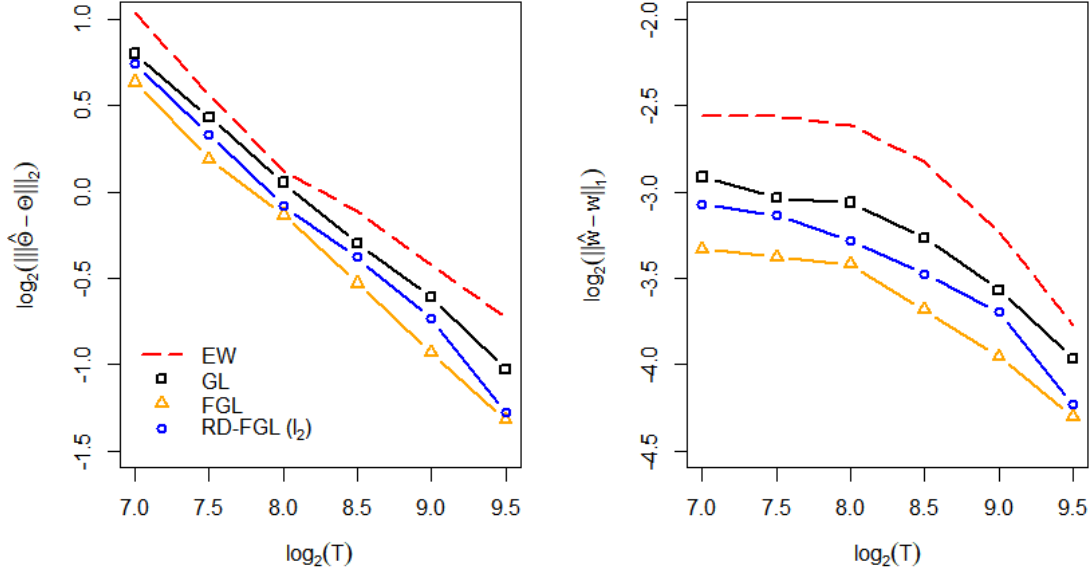


Figure E.1: **Averaged errors of the estimators of  $\Theta$  (left) and  $w$  on logarithmic scale (base 2).**  $p = T^{0.85}$ ,  $q = 2(\log(T))^{0.5}$ ,  $s_T = \mathcal{O}(T^{0.05})$ .

The total number of observations is  $m$ . The period for training the models is set to be  $m_1 = m/2$  – this is used to train competing FAR models in (5.3). The remaining part of the sample,  $m_2 = m - m_1$  is split as follows: the estimation window for training competing models (that is, EW, GL, FGL, and RD-FGL) is set to be  $\text{window} = m_2/2$ . We roll the estimation window over the the test sample of the size  $m_2/2$  to update all the estimates in each point of time. Recall that  $q$  denotes the number of factors in the forecast errors as in equation (2.1).

Similarly to the previous subsection, we include RD-FGL in all simulations. When there is no break in the DGP, the tuning parameter for the factor loadings,  $\gamma$ , is set to one, and the penalty that controls the change of idiosyncratic precision matrix over time,  $\beta$ , is zero. Figure E.2 shows the MSFE for different sample sizes and fixed parameters: we report the results for two values of  $c_1 \in \{0, 0.75\}$ . As evidenced from Figure E.2, the models that use the factor structure outperform EW combination and non-factor based counterparts for both values of  $c_1$ .

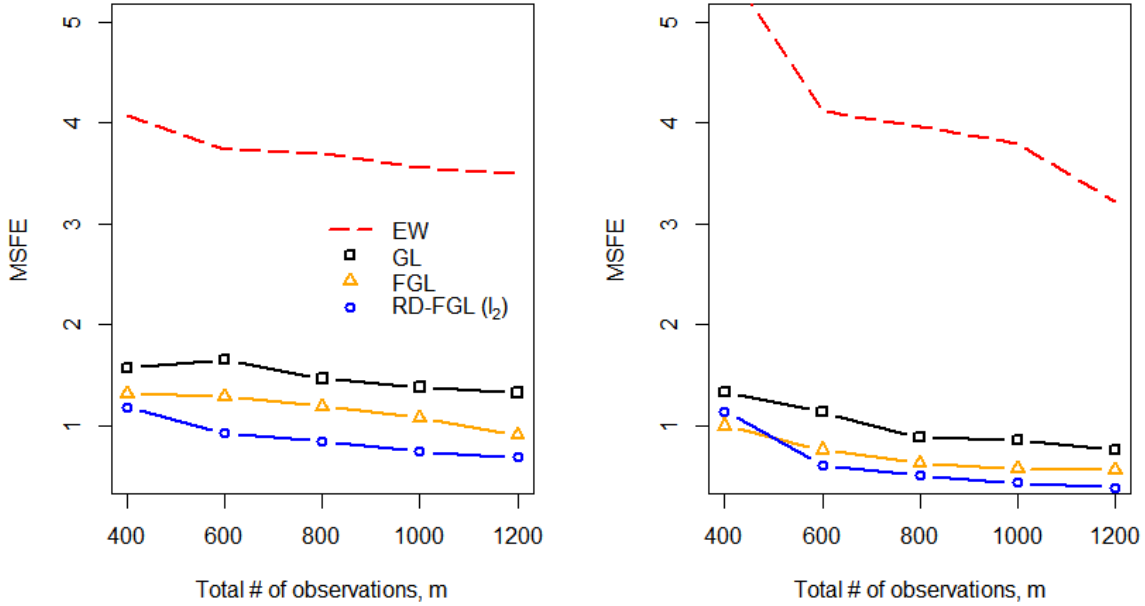


Figure E.2: **Plots of the MSFE over the total number of observations  $m$ .**  $c_1 = 0$  (left),  $c_1 = 0.75$  (right),  $c_2 = 0.9$ ,  $M = 100$ ,  $r = 5$ ,  $\sigma_\xi = 1$ ,  $L = 7$ ,  $K = 2$ ,  $p = 24$ ,  $q = 3$ ,  $\rho = 0.9$ ,  $\phi = 0.8$ .

## E.2 Break Only in Idiosyncratic Precision Matrix

This section presents the results for the case when there is a single break in  $\Theta_\varepsilon$ . The DGP is the same as described in Subsection 5.1: the break point is fixed in the middle of the sample  $T/2$ . Before the break, the value of the largest coefficient in  $\Theta_{\varepsilon,1}$  is set to 0.4, after the break it changes to 0.6.

Figure E.3 shows the averaged (over Monte Carlo simulations) errors of the estimators of the precision matrix  $\Theta$  and the optimal combination weight versus the sample size  $T$  in the logarithmic scale (base 2). The estimate of the precision matrix of the EW forecast combination is obtained using the fact that diagonal covariance and precision matrices imply equal weights. To determine the values of the diagonal elements we use the shrinkage intensity coefficient calculated as the average of the eigenvalues of the sample covariance matrix of the forecast errors (see Ledoit and Wolf (2004)).

Figure E.3 shows the performance of all models including RD-FGL when the break is only in  $\Theta_\varepsilon$  ( $\gamma = 1$ ): accounting for the break significantly reduces the estimation error of precision matrix and combination weights.

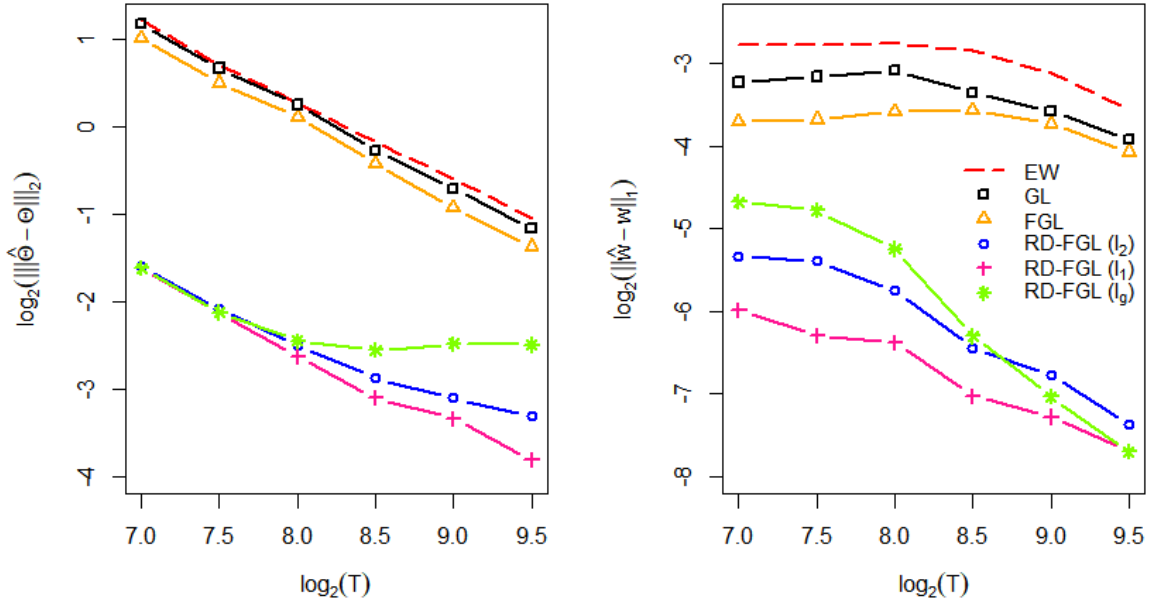


Figure E.3: **Averaged errors of the estimators of  $\Theta$  (left) and  $w$  on logarithmic scale (base 2): break in  $\Theta_\varepsilon$ .**  $p = T^{0.85}$ ,  $q = 2(\log(T))^{0.5}$ ,  $s_T = \mathcal{O}(T^{0.05})$ .

### E.3 Multiple Breaks

We examine the performance of RD-FGL and competing methods for the case of two known breaks.

First, we explore behavior of precision matrix and weights estimates. To incorporate two structural breaks in  $\Theta_\varepsilon$ , we add the following modification to the DGP setup in Subsection 5.1. We fix two break points: one at  $t_1 = T/4$  and the other at  $t_1 = 3T/4$ . Define the following idiosyncratic precision matrices:  $\Theta_{\varepsilon,1}$  before  $t_1$ ,  $\Theta_{\varepsilon,2}$  between  $t_1$  and  $t_2$ , and  $\Theta_{\varepsilon,3}$  after  $t_2$ . The value of the largest coefficient in the three aforementioned matrices is set to 0.2, 0.4, and 0.6, accordingly.

As demonstrated in Figure E.4, similarly to the findings in the main manuscript for the case with one break, accounting for the break significantly reduces the estimation error of precision matrix and combination weights.

Second, we explore behavior of MSFE under two breaks. To incorporate two structural breaks we add the following modification to the DGP in Subsection 5.2. The total number of observations is  $m$ . The period for training the models is set to be  $m_1 = T/3$  – this is used to train competing FAR models in (5.3). The remaining part of the sample,  $m_2 = m - m_1$

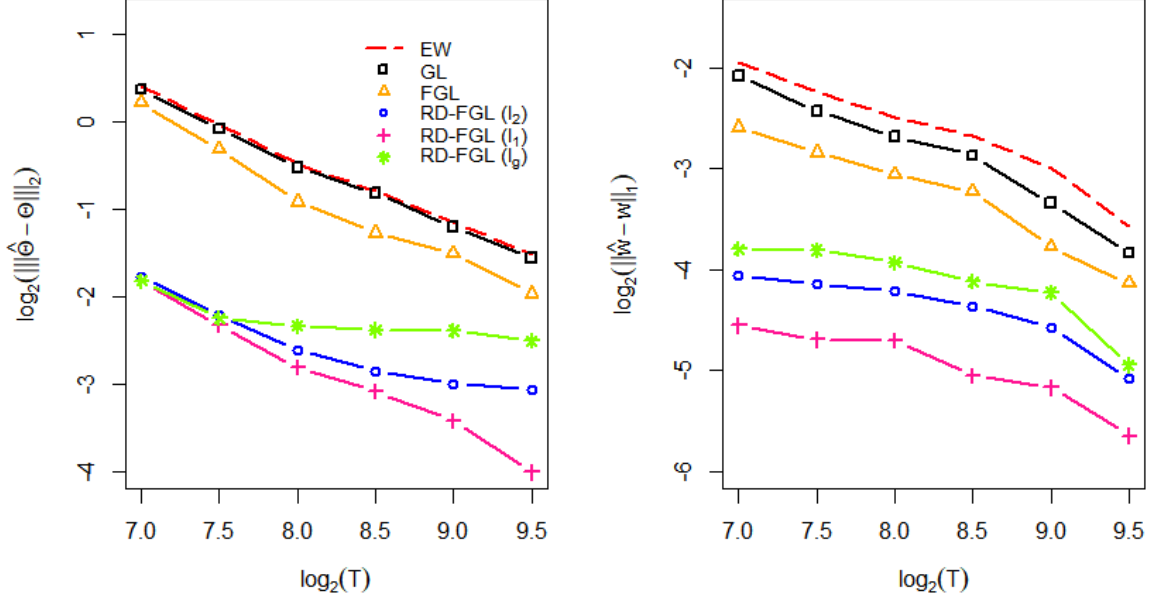


Figure E.4: **Averaged errors of the estimators of  $\Theta$  (left) and  $w$  on logarithmic scale (base 2): two breaks in  $\Theta_\varepsilon$ .  $p = T^{0.85}$ ,  $q = 2(\log(T))^{0.5}$ ,  $s_T = \mathcal{O}(T^{0.05})$ .**

is split similarly to Subsection 5.2: the estimation window for training competing models is set to be  $\text{window} = m_2/2$ . We roll the estimation window over the the test sample. The break points are fixed at  $1/3$  and  $3/4$  of the first estimation window, and will be referred to as  $t_1$  and  $t_2$ .

When generating  $\theta_s$  we set  $c_2$  as follows:  $c_2 = 0.3$  before  $t_1$ ,  $c_2 = 0.6$  between  $t_1$  and  $t_2$ ,  $c_3 = 0.9$  after  $t_2$ .

Before the break, when generating  $\theta_s$  we set  $c_2 = 0.3$ , and after the break  $c_2 = 0.9$ . All other parameters stay unchanged. Notice that the break in  $c_2$  can propagate into both a break in precision matrix and factor loadings.

Similarly to the main manuscript, we include different specifications of the smoothing function  $\psi(\cdot)$ . Figure E.5 shows the performance of all models including RD-FGL with  $\gamma$  estimated using cross-validation: similarly to the conclusions in Subsection 5.2, accounting for the break significantly reduces MSFE of the combined forecast.

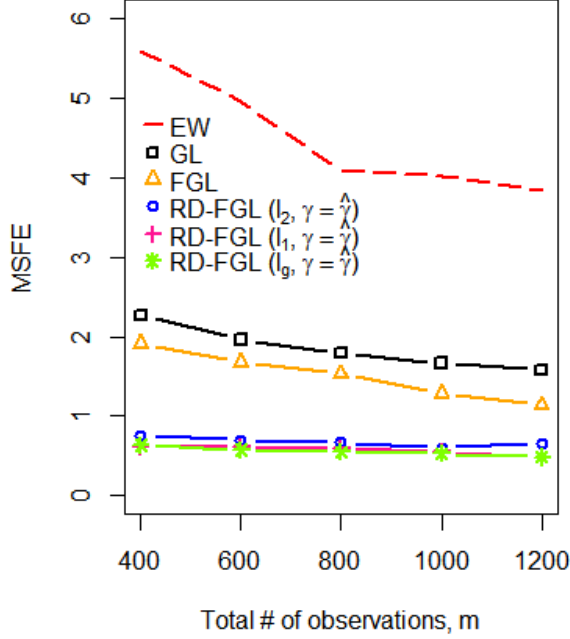


Figure E.5: **Plots of the MSFE over the total number of observations  $m$ .**  $c_1 = 0.75$ ,  $c_2 = 0.3$  (before  $t_1$ ),  $c_2 = 0.6$  (between  $t_1$  and  $t_2$ ),  $c_3 = 0.9$  (after  $t_2$ ),  $M = 100$ ,  $r = 5$ ,  $\sigma_\xi = 1$ ,  $L = 7$ ,  $K = 2$ ,  $p = 24$ ,  $q = 3$ ,  $\rho = 0.9$ ,  $\phi = 0.8$ .

## E.4 Varying Break Magnitude

We examine the performance of RD-FGL and competing methods for the case of one known break of smaller magnitude.

First, we explore behavior of precision matrix and weights estimates. The setup is the same as in Subsection 5.1: we fix a single break point in the middle of the sample size,  $T/2$ : in the precision matrix of the idiosyncratic errors before the break, referred to as  $\Theta_{\varepsilon,1}$ , the value of the largest coefficient is set to 0.4; whereas in the precision matrix of the idiosyncratic errors after the break,  $\Theta_{\varepsilon,2}$ , the value of the largest coefficient is set to 0.45. We use  $\Theta_{\varepsilon,1}$  and  $\Theta_{\varepsilon,2}$  to generate  $\varepsilon_t$  in (5.1).

As demonstrated in Figure E.6, similarly to the findings in the main manuscript, accounting for the break significantly reduces the estimation error of precision matrix and combination weights even if the break magnitude is small.

Second, we explore behavior of MSFE for smaller break magnitude. The setup is the same as in Subsection 5.1: the period for training the models is set to be  $m_1 = m/3$  – this is used to train competing FAR models in (5.3). The remaining part of the sample,

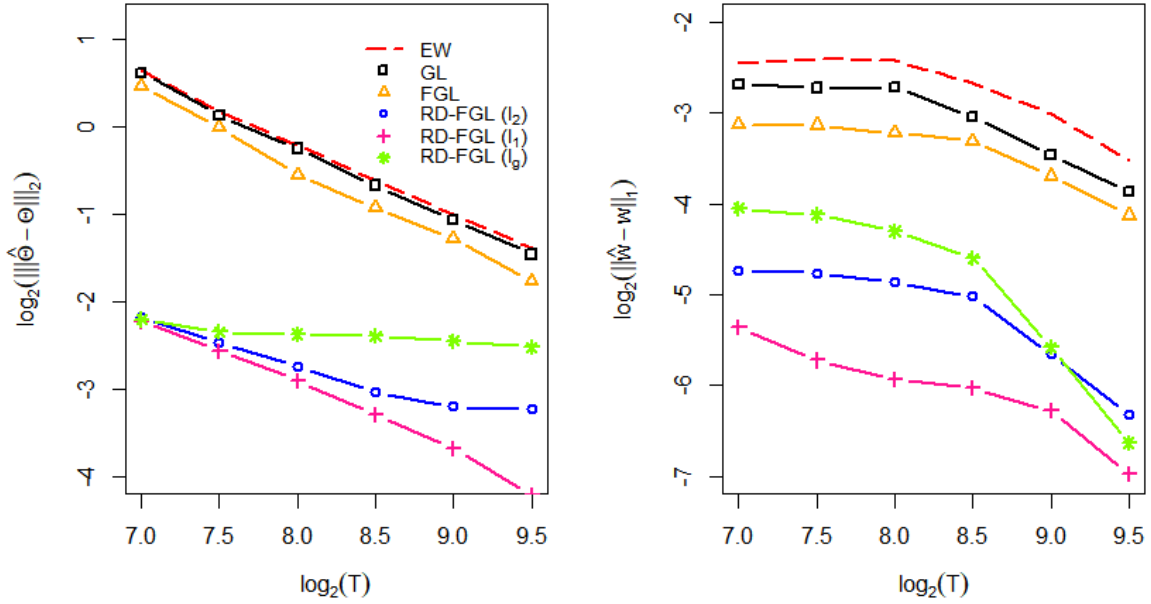


Figure E.6: **Averaged errors of the estimators of  $\Theta$  (left) and  $w$  on logarithmic scale (base 2): one break in  $\Theta_\varepsilon$ .  $p = T^{0.85}$ ,  $q = 2(\log(T))^{0.5}$ ,  $s_T = \mathcal{O}(T^{0.05})$ .**

$m_2 = m - m_1$  is split as follows: the estimation window for training competing models is set to be  $\text{window} = m_2/2$ . We roll the estimation window over the test sample of the size  $m_2/2$ . The break point is fixed at  $1/2$  of the first estimation window. Before the break, when generating  $\theta_s$  we set  $c_2 = 0.3$ , and after the break  $c_2 = 0.4$ . All other parameters stay unchanged.

Similarly to the main manuscript, we include different specifications of the smoothing function  $\psi(\cdot)$ . Figure E.7 shows the performance of all models including RD-FGL with  $\gamma$  estimated using cross-validation: similarly to the conclusions in Subsection 5.2, accounting for the break significantly reduces MSFE of the combined forecast even if the break magnitude is small.

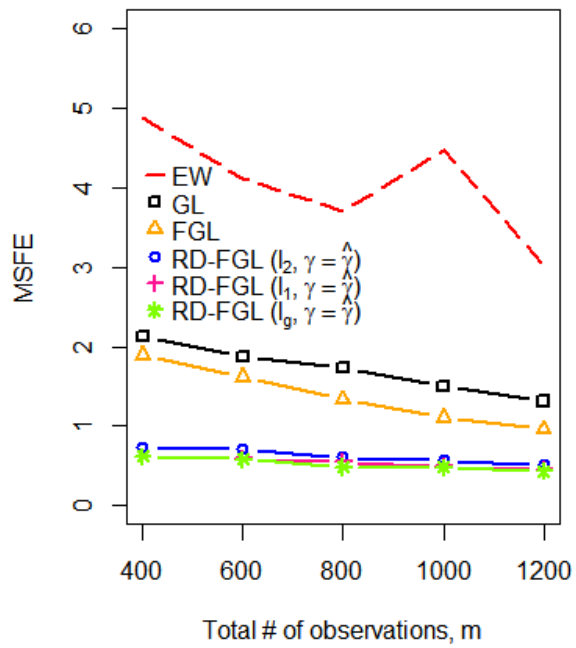


Figure E.7: **Plots of the MSFE over the total number of observations  $m$ .**  $c_1 = 0.75$ ,  $c_2 = 0.3$  (before the break),  $c_2 = 0.4$  (after the break),  $M = 100$ ,  $r = 5$ ,  $\sigma_\xi = 1$ ,  $L = 7$ ,  $K = 2$ ,  $p = 24$ ,  $q = 3$ ,  $\rho = 0.9$ ,  $\phi = 0.8$ .