Probabilistic Targeted Factor Analysis*

Miguel C. Herculano[†] Santiago Montoya-Blandón[‡]

December 10, 2024

Abstract

We develop a probabilistic variant of Partial Least Squares (PLS) we call Probabilistic Targeted Factor Analysis (PTFA), which can be used to extract common factors in predictors that are useful to predict a set of predetermined target variables. Along with the technique, we provide an efficient expectation-maximization (EM) algorithm to learn the parameters and forecast the targets of interest. We develop a number of extensions to missing-at-random data, stochastic volatility, and mixed-frequency data for real-time forecasting. In a simulation exercise, we show that PTFA outperforms PLS at recovering the common underlying factors affecting both features and target variables delivering better in-sample fit, and providing valid forecasts under contamination such as measurement error or outliers. Finally, we provide two applications in Economics and Finance where PTFA performs competitively compared with PLS and Principal Component Analysis (PCA) at out-of-sample forecasting.

Keywords: Probabilistic Targeted Factor Model, High-dimensional data, Expectation-Maximization algorithm, Missing data, Stochastic Volatility.

JEL Classification: C38, C53, C55, G12, G17

1 Introduction

In the age of big data, reducing dimensionality of the information in large datasets in order to uncover patterns and obtain interpretable predictions is crucial. Examples of well-established techniques for dimensionality reduction include Principal Component Analysis (PCA) and Partial Least Squares (PLS). The PLS was developed by the Econometrician Herman Wold in a series of papers including his seminal work (Wold, 1975) that introduces the main algorithm used for its computation, laying the foundation for

^{*}The implementation of PTFA is made openly available on PyPI.

[†]Adam Smith Business School, University of Glasgow.

E: miguel.herculano@glasgow.ac.uk — W: mcherculano.github.io

[‡]Adam Smith Business School, University of Glasgow.

E: santiago.montoya-blandon@glasgow.ac.uk — W: smontoyablandon.com

the method. It works by constructing a set of latent vectors maximizing the correlation between (potentially many) predictor variables and targets. The method has enjoyed increasing popularity in Economics and Finance due to its ability to handle high-dimensional data and multicollinearity effectively; for a host of applications and extensions see Welch and Goyal (2007); Kelly and Pruitt (2013, 2015); Giglio et al. (2016); Groen and Kapetanios (2016); Goyal et al. (2024).

In this paper we develop a technique we term Probabilistic Targeted Factor Analysis (PTFA), which works by finding common latent factors to predictors (X) and targets (Y) that are then used to forecast the targets Y. Both the latent factors and the parameters are jointly estimated via an EM algorithm that iteratively maximizes the model's observeddata likelihood. This estimation strategy is different from the two-step procedure used to estimate a PLS Regression. In contrast to PLS, PTFA recognizes the stochastic nature of the generative model of latent factors and parameters. This affords two key advantages to our methodology. First, PTFA offers a more flexible and robust approach by explicitly modeling the uncertainty in both the latent variables and the observed data. This probabilistic framework allows for the incorporation of noise into the model, leading to potentially more accurate and interpretable results, especially in noisy or incomplete-data environments. Second, the use of the Expectation-Maximization (EM) algorithm to maximize the likelihood function in PTFA ensures that the latent factors are estimated in a way that optimally reflects the underlying data structure. This contrasts with the deterministic nature of traditional dimensionality reduction techniques, which may be more prone to overfitting and less capable of handling complex, real-world data scenarios where uncertainty plays a significant role.

That is, unlike traditional techniques for dimentionality reduction, PTFA is based on a probability model, making it an attractive choice in complex modelling environments where missing data, stochastic volatility, noise and outliers abound (for a survey of the literature, see Wold et al., 2001; Groen and Kapetanios, 2016). Much like Tipping and Bishop (1999), who present a probabilistic version of PCA, the main contribution of this paper is to introduce a probabilistic foundation of PLS into the economics literature by developing a Probabilistic Targetted Factor Analysis framework. In addition to the theory, we provide an expectation-maximization (EM) algorithm to recover common predictability factors and an open-source implementation of the framework (currently available in Python: pypi.org/project/ptfa).

The PTFA framework allows us to provide two extensions that are valuable for applied work, all of which are implemented in our software. First, we derive the necessary extension to missing data, including mixed-frequency data, where high-frequency and low-frequency data are jointly modeled. This allows the applied researcher to recover common factors from unbalanced and incomplete datasets, even if the data is not missing at random. This extension is key to deal with the realities of data availability, data-release schedules, and non-overlapping or otherwise unbalanced datasets, such as occurs in try-

¹While the algorithm used to estimate PLS is an iterative process, the underlying structure of PLS is a two-step procedure: i) latent variable (component) extraction, followed by ii) regression of the response matrix **Y** on these latent variables (for additional details, see Butler and Denham, 2002).

ing to predict a low-frequency target using high-frequency (or real-time) information. Second, our framework allows us to incorporate stochastic volatility in both features and targets, which is a common feature in time-series forecasting with economic variables.

Alternative probabilistic versions of PLS have been developed to suit the needs of different fields and literatures, particularly in chemometrics (Gustafsson, 2001; Li et al., 2011; Zheng et al., 2016; el Bouhaddani et al., 2018). These papers generally focus on the interpretation of PLS as maximizing the covariance between targets and projected features, and tend to be more algorithmic in nature. Therefore, we see our paper as providing a unified approach to a probabilistic foundation for PLS, while importing this powerful technique to the forecasting of economic and financial data and time-series more generally. Our paper additionally contributes to the existing literature along several dimensions. We provide a full characterization of the potential solutions to the probabilistic PLS framework, which are parallels of the Tipping and Bishop (1999) solutions with an added special case that arises in this setting. This is contrast to existing work that only studies identification and construction of EM steps. Additionally, we explore the host of extensions made possible by the probabilistic setup, which are of key interest. Finally, by providing an open-source implementation of the package, we aim to make both the study and extension of the method accessible to researchers and practitioners.

To compare the performance of our method to popular alternatives in the literature, we use both Monte Carlo exercises and real-world data. We first set up a simple simulation exercise showcasing that PTFA provides better in-sample fit compared to standard PLS regardless of the generating distribution of the data, and show this performance gap can increase with the level of noise in the variables (particularly in targets). We then explore the technique's out-of-sample forecasting performance in two popular applications to Economics and Finance. The first application uses PTFA to forecast three key macroeconomic variables: industrial production, consumer price index (CPI) inflation and unemployment. We conduct a forecasting exercise that harvests the information contained in 126 Federal Reserve Economic Data monthly time series (FRED-MD) mimicking the setup in McCracken and Ng (2016). Second, we use our model to predict the equity premium using 26 signals made available by Goyal et al. (2024). In both applications PTFA performs well compared to both PLS and PCA, using similar computational complexity in fitting all methods.

The remaider of the paper is organized as follows. Section 2 provides a refresher on PLS and outlines our probabilistic foundation resulting in the PTFA method. Section 4 presents the setup for our simulation exercises and discusses the findings. In Section 5, we provide 2 applications of PTFA using popular datasets in Economics and Finance. The appendices provide additional theoretical and implementation details on our method: Appendix A provides a theoretical analysis of the properties of the maximum likelihood estimator resulting from the observed-data likelihood of our probabilistic model; Appendix B derives an EM algorithm based on iterative maximization of this likelihood; and Appendix C provides all algorithms.

2 Methodological Framework

Throughout this section, we let x be a p-dimensional vector of features and y be a q-dimensional vector of prediction targets. Our main assumption is that there are k common components or factors collected in vector f, where one typically expects $p \gg k$. This is generally the case when one aims to extract signals from high-dimensional data with a large feature space.

To understand the motivation behind the development of the PTFA, it is helpful to compare it with PCA. Both PTFA and PCA aim to reduce the dimensionality of a large set of variables, via a dense low-rank representation of the data-generating process (DGP). However, they do so with different objectives in mind. PCA transforms a set of possibly correlated variables into a set of linearly uncorrelated latent vectors called principal components. These components are constructed in an unsupervised way, such that the first principal component captures the maximum variance in the data, the second captures the maximum variance after projecting out the first component, and so on. PTFA, on the other hand, constructs latent vectors that maximize the covariance between the predictor (X) and response (Y) variables, targeting the former. Unlike PCA, which focuses solely on the predictors, PTFA considers both predictors and responses, aiming to find the directions in the features that best predict the responses. This makes PTFA particularly attractive in contexts where interpretability of the latent vectors is important.

Our setup is akin to the DGP assumed in the theoretical results supporting the popular forecasting technique Three-Pass Regression Filter (Kelly and Pruitt, 2015, 3PRF), of which PLS is a special case. We additionally assume throughout the paper that both x and y have been standardized and purged of the influence of any common observable effects. Before proceeding with PTFA, we also provide a short summary of the intuition underlying PLS, for which PTFA provides a probabilistic foundation.

2.1 Review: PLS regression

Partial Least Squares regression aims to find k factors from x that best predict y. While techniques like PCA also perform dimensionality reduction on the set of potential predictors x, they are silent about the relevance of the principal components to predict y. On the contrary, PLS directly recovers scores from x with predictability of y in mind. To be specific, PLS regression searches for two sets of scores f_x and f_y that perform a simultaneous decomposition of x and y such as to maximize the correlation between y and the recovered f_x . By focusing on recovering only k scores, we project the feature space to the directions that maximize predictability of y in the mean-squared error sense. Therefore, the independent and dependent variables are decomposed as linear transformations of the scores with loadings \mathbf{P} and \mathbf{Q} such as

$$x = \mathbf{P} f_{x}$$
 and $y = \mathbf{Q} f_{y}$. (1)

²Let all common observable effects between targets and features be captured by z, an r-dimensional vector of controls (including a constant for de-meaning). We can then assume that x and y are the residuals from a linear projection of the original features and targets onto z.

Loadings **P** are chosen so as to maximize $Cov(y, Pf_x)$. The Nonlinear Iterative Partial Least Squares (NIPALS) algorithm (Wold, 1975) — commonly used in the literature to estimate and motivate PLS — can then be used to efficiently recover the loadings, and these can be used to forecast y for any out-of-sample value of x. Note that, in contrast to PCA, the loadings recovered from PLS are not necessarily orthogonal.

Importantly, note that this representation does not have a probabilistic foundation in mind as there is no randomness embedded into the factor or loading recovery process. Additionally, the factor extraction process for these techniques is usually thought of in an algorithmic or geometric manner, rather than a statistical one. This also means a standard PLS approach does not acknowledge additional sources of variation in the data such as noise or incomplete data patterns.

2.2 Probabilistic Targeted Factor Analysis

We now provide a simple statistical formulation for performing factor extraction that embodies the objective of PLS. This framework, which we denote as Probabilistic Targeted Factor Analysis (PTFA), provides a simple unifying setting to probabilistic extraction of factors from features x that optimally predict a pre-specified target y. We assume the following model for x and y as generated from some *common* latent components f as

$$x = \mathbf{P}f + e_x \,, \tag{2}$$

$$y = \mathbf{Q}f + e_{y}, \tag{3}$$

where **P** and **Q** again represent loadings, and $e_x \sim \mathcal{N}_p(\mathbf{0}_p, \sigma_x^2 \mathbf{I}_p)$ and $e_y \sim \mathcal{N}_q(\mathbf{0}_q, \sigma_y^2 \mathbf{I}_q)$ are isotropic Gaussian noise terms.³ The structure provided by Eqs. (2)–(3) and the normality of the errors is not to be taken as a literal distributional assumption, but rather a probabilistic framework to embed PLS and similar targeted factor extration techniques.

Similar to the 3PRF, PTFA considers predictors x and targets y to be correlated solely through the latent common factors they share, such that $Cov(e_x, e_y) = \mathbf{0}_{p \times q}$. This choice of correlation structure results in a particularly simple yet consistent (quasi-)maximum likelihood estimator of the loadings \mathbf{P} and \mathbf{Q} (see Appendix \mathbf{A} for details). Other models, such as that considered by Groen and Kapetanios (2016), explicitly allow for additional predictors to be directly correlated with the target variable. While we can easily allow for additional correlation between features and targets conditional on the factors, for simplicity of exposition this paper focuses on the case where no additional correlation is assumed.

The latent scores are assumed to be normally distributed $f \sim \mathcal{N}_k(\mathbf{0}_k, \mathbf{V}_F)$ with positive-definite prior variance \mathbf{V}_F . We simply set $\mathbf{V}_F = \mathbf{I}_k$ in the absence of any prior information on the latent scores or if they are meant to represent structurally uncorrelated components. Letting d := p + q, and using the conditional independence between x and y, the conditional likelihood is given by

$$p(x, y|f) = p(x|f)p(y|f) = \mathcal{N}_d(\mu, \Sigma), \tag{4}$$

 $[\]overline{}^3$ Zero-mean errors are without loss of generality as both x and y are centered prior to any processing.

a multivariate Gaussian with d-dimensional mean vector $\boldsymbol{\mu} \coloneqq [\boldsymbol{f}^{\top} \mathbf{P}^{\top}, \boldsymbol{f}^{\top} \mathbf{Q}^{\top}]^{\top}$ and $d \times d$ variance-covariance matrix $\boldsymbol{\Sigma} := \operatorname{diag}(\sigma_x^2 \mathbf{I}_p, \sigma_y^2 \mathbf{I}_q)$. To derive the posterior distribution $p(\boldsymbol{f}|\boldsymbol{x}, \boldsymbol{y})$ we use Bayes' rule to find

$$p(\boldsymbol{f}|\boldsymbol{x},\boldsymbol{y}) \propto p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}) \propto \exp\left\{-\frac{1}{2}\left(\frac{\|\boldsymbol{x}-\mathbf{P}\boldsymbol{f}\|_{2}^{2}}{\sigma_{x}^{2}} + \frac{\|\boldsymbol{y}-\mathbf{Q}\boldsymbol{f}\|_{2}^{2}}{\sigma_{y}^{2}} + \boldsymbol{f}^{\top}\mathbf{V}_{F}^{-1}\boldsymbol{f}\right)\right\}.$$

Completing the squares, we can derive the factor posterior as $f \mid x, y \sim \mathcal{N}_k(m, \Omega)$ with posterior mean and covariance matrix given as⁴

$$\mathbf{\Omega} := \left(\mathbf{V}_F^{-1} + \frac{\mathbf{P}^\top \mathbf{P}}{\sigma_x^2} + \frac{\mathbf{Q}^\top \mathbf{Q}}{\sigma_y^2} \right)^{-1}, \tag{5}$$

$$m \coloneqq \Omega \left(\frac{\mathbf{P}^{\top} x}{\sigma_{x}^{2}} + \frac{\mathbf{Q}^{\top} y}{\sigma_{y}^{2}} \right).$$
 (6)

2.3 Implementation

We provide a fast and simple expectation-maximization (EM) solution to efficiently learn the parameters of our PTFA formulation, collected into $\theta = (\mathbf{P}, \mathbf{Q}, \sigma_x^2, \sigma_y^2)$. We assume we have access to a random sample $\{x_t, y_t\}_{t=1}^T$, collected into the matrices $\mathbf{X} \in \mathbb{R}^{T \times p}$ and $\mathbf{Y} \in \mathbb{R}^{T \times q}$. Similarly, we assume factors $\{f_t\}_{t=1}^T$ are collected into a matrix $\mathbf{F} \in \mathbb{R}^{T \times k}$.

The independence across rows of factor components in matrix **F** is translated into a prior $f_t \sim p(f)$ independently across observations t = 1, ..., T. Letting $\text{vec}(\mathbf{F})$ be the column-vectorized version of **F** and recalling $p(f) = \mathcal{N}_k(f \mid \mathbf{0}_k, \mathbf{V}_F)$, we can obtain the posterior of the stacked scores as

$$\operatorname{vec}(\mathbf{F}) \mid \mathbf{X}, \mathbf{Y}; \theta \sim \mathcal{N}_{Tk}(\operatorname{vec}(\mathbf{M}), \mathbf{\Omega} \otimes \mathbf{I}_T).$$
 (7)

The posterior mean matrix can be expressed succinctly as

$$\mathbf{M} = \mathbf{Z}\mathbf{\Sigma}^{-1}\mathbf{L}\mathbf{\Omega}\,,\tag{8}$$

where we stack the data into $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$ and all loadings into $\mathbf{L} = [\mathbf{P}^\top, \mathbf{Q}^\top]^\top$. Under standard statistical loss functions (such as quadratic, absolute, or zero-one losses), decision theory arguments guarantee that \mathbf{M} is the optimal prediction of the scores \mathbf{F} in this model (see, e.g., Greenberg, 2012, pp. 29–31).

The first step in deriving an EM algorithm to learn the parameters and latent vectors in the PTFA model is to derive the complete data log-likelihood log $p(X, Y, F \mid \theta)$. The expectation (E) step finds the observed-data likelihood by integrating out the factors under their posterior distribution:

$$Q(\theta) := \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\theta}[\log p(\mathbf{X},\mathbf{Y},\mathbf{F} \mid \theta)],$$

= $\mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\theta}\left[\log p(\mathbf{X} \mid \mathbf{F};\mathbf{P},\sigma_x^2) + \log p(\mathbf{Y} \mid \mathbf{F};\mathbf{Q},\sigma_y^2) + \log p(\mathbf{F})\right].$ (9)

⁴See Appendix B for a thorough derivation of these equations.

The maximization (M) step consists in optimizing $Q(\theta)$ (given an initial value) with a view of deriving updating rules for θ . We use $\mathbf{M} := \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\theta}[\mathbf{F}]$ and $\mathrm{Cov}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\theta}(\mathbf{F}|\mathbf{X},\mathbf{Y}) = \mathbf{\Omega} \otimes \mathbf{I}_T$ to obtain $\mathbf{V} := \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\theta}[\mathbf{F}^{\top}\mathbf{F}] = T \cdot \mathbf{\Omega} + \mathbf{M}^{\top}\mathbf{M}$. The result of maximizing (9) is a joint update rule for the loadings given by

$$\mathbf{L} = \begin{bmatrix} \mathbf{P} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^{\top} \\ \mathbf{Y}^{\top} \end{bmatrix} \mathbf{M} \mathbf{V}^{-1} = \mathbf{Z}^{\top} \mathbf{M} \mathbf{V}^{-1}.$$
 (10)

Similar simple update rules can be found for the variance parameters σ_x^2 and σ_y^2 as

$$\sigma_x^2 = \frac{1}{Tp} \left[\|\mathbf{X}\|_F^2 - \text{Tr}(\mathbf{P}^\top \mathbf{P} \mathbf{V}) \right] \quad \text{and} \quad \sigma_y^2 = \frac{1}{Tq} \left[\|\mathbf{Y}\|_F^2 - \text{Tr}(\mathbf{Q}^\top \mathbf{Q} \mathbf{V}) \right]. \tag{11}$$

By efficiently computing all updates using simple matrix operations, our algorithm achieves performance gains compared to PLS without incurring in large computational complexity. A full implementation of our EM algorithm that computes these steps along with their formal derivation can be found in Appendices B and C.

3 Extensions

In this section, we preview a host of possible extensions that become simple to implement once one has a probabilistic framework for targeted factor analysis. Specifically, we provide extensions to incomplete data (both under at-random and mixed-frequency designs) and stochastic volatility in the features and targets. All of these are of particular interest in areas such as economics and finance, where data is likely to exhibit such patterns. Additionally, while we focus on providing simple and computationally efficient extensions through the EM approach, by specifying priors over θ PTFA could be augmented to variational or Bayesian inference providing access to all the benefits of these frameworks. This means, for example, sparsity in the loadings via shrinkage priors, structural consideration of stochastic volatility, variable selection and model uncertainty, among many others. Some of these have been partially considered in the literature (Vidaurre et al., 2013; Li et al., 2018; Zheng and Song, 2018; Xie, 2019; Yang et al., 2021; el Bouhaddani et al., 2022), and we will continue exploring such extensions in future research.

3.1 Incomplete Data

3.1.1 Missing at Random

PTFA offers a natural approach to the estimation of the principal axes in cases where some of the data in **X** and **Y** are missing at random. We follow standard methodology for maximizing the likelihood of a Gaussian model in the presence of missing values (Little and Rubin, 2019). We explain now the changes that we make to the standard algorithm to account for *incomplete data*.

For any row $t \in \{1, ..., T\}$ and feature $j \in \{1, ..., p\}$, we let $\tau_{tj}^{(X)}$ be an indicator for whether that particular observation is missing in the feature matrix **X**. That is, $\tau_{tj}^{(X)} = 1$

only when observation tj is missing, and is 0 otherwise. We can analogously define a missing indicator for the target matrix **Y** and denote it as $\tau_{tj}^{(Y)}$, where $j \in \{1, \dots, q\}$.

Note that, after standardization, the unconditional mean for all columns of X and Y is zero. Therefore, a natural initial imputation strategy is to replace all missing observations (those with $\tau_{tj}^{(X)} = 1$ or $\tau_{tj}^{(Y)} = 1$) by 0. Let the matrices with imputed values be denoted as \widetilde{X} and \widetilde{Y} . One pass of our EM algorithm allows us to obtain estimated values for \widetilde{P} and \widetilde{Q} as well as the predicted scores by using Eq. (8) and (10) on the imputed matrices, with the output denoted as \widetilde{M} . Finally, given these update values, we can provide a more accurate imputation of \widetilde{X} and \widetilde{Y} by setting

$$\widetilde{X}_{ij} = \sum_{c=1}^{k} \widetilde{M}_{ic} \widetilde{P}_{jc} \quad \text{if} \quad \tau_{ij}^{(X)} = 1, \quad \text{and}$$
 (12)

$$\widetilde{Y}_{ij} = \sum_{c=1}^{k} \widetilde{M}_{ic} \widetilde{Q}_{jc} \quad \text{if} \quad \tau_{ij}^{(Y)} = 1.$$
 (13)

By iteratively applying equations (8), (10), (12) and (13), our algorithm can adapt to relatively large contamination rates for both features and targets. This feature of our method is also explored through a Monte Carlo simulation in the next section.

3.1.2 Mixed-Frequency Data

Many economic time series do not follow common release schedules and information availability itself can change across time. This creates a similar missing-data problem, but not one where we can claim the data is missing at random given there is a clear pattern to the unobserved data points. As a running example, consider \boldsymbol{x} to be a set of *monthly* indicators, where our targets \boldsymbol{y} are economic indicators (such as GDP or inflation) available at a *quarterly* frequency. As one cannot observe the quarterly variables in the intermediate months for which one has feature data available, the pattern of missingness is clearly not random.

In order to provide real-time indices of economic activity under the problem of mixed-frequency data, many proposal have been introduced in the literature (for a recent review, see Foroni and Marcellino, 2014). We now show how the PTFA framework can be extended to account for mixed-frequency observations, particularly when the missing data is on the target. We can modify our initial model equations as follows by introducing a latent set of targets y^* that is available at the higher frequency of x. By aggregating the latent variable into the lower-frequency analog, our model produces a likelihood that can directly be used to modify the EM algorithm introduced previously:

$$\boldsymbol{x}^{(l)} = \mathbf{P}\boldsymbol{f}^{(l)} + \boldsymbol{e}_{x}^{(l)}, \tag{14}$$

$$\mathbf{y}^{*(l)} = \mathbf{Q}\mathbf{f}^{(l)} + e_{\mathbf{y}}^{*(l)},$$
 (15)

$$y = \frac{1}{L} \sum_{l=1}^{L} y^{*(l)}, \qquad (16)$$

for l = 1, ..., L, where L represents the amount of period of high-frequency observations with respect to the lower-frequency ones (e.g., if x is monthly and y is quarterly, then L = 3). This aggregation scheme is drawn from the parsimonious approach to mixed-frequency modelling advocated by Giannone et al. (2008). Combining equations (15) and (16), we can directly derive the following equation for the observable target in terms of the latent factors:

$$y = \frac{1}{L} \mathbf{Q} \left(\sum_{l=1}^{L} f^{(l)} \right) + \frac{1}{L} \sum_{l=1}^{L} e_y^{*(l)}.$$
 (17)

Assuming as before that $f^{(l)} \sim \mathcal{N}_k(\mathbf{0}_k, \mathbf{V}_F)$ and $(\mathbf{e}_x, \mathbf{e}_y^{*(l)}) \sim \mathcal{N}_d(\mathbf{0}_d, \mathbf{\Sigma})$ independently across l = 1, ..., L, the conditional likelihood can be expressed as

$$p(\boldsymbol{x}^{1},\ldots,\boldsymbol{x}^{L},\boldsymbol{y}\mid\boldsymbol{f}^{1},\ldots,\boldsymbol{f}^{L}) = \prod_{l=1}^{L} \mathcal{N}_{p}(\boldsymbol{x}^{(l)}\mid\boldsymbol{P}\boldsymbol{f}^{(l)},\sigma_{x}^{2}\boldsymbol{I}_{p}) \times \mathcal{N}_{q}\left(\boldsymbol{y}\left|\frac{1}{L}\boldsymbol{Q}\sum_{l=1}^{L}\boldsymbol{f}^{(l)},\frac{1}{L}\sigma_{y}^{2}\boldsymbol{I}_{q}\right)\right). \tag{18}$$

Define the full set of features $\boldsymbol{x} = [\boldsymbol{x}^{(1)\top}, \dots, \boldsymbol{x}^{(L)\top}]^{\top}$ and factors $\boldsymbol{f} = [\boldsymbol{f}^{(1)\top}, \dots, \boldsymbol{f}^{(L)\top}]^{\top}$, of size $p \cdot L$ and $k \cdot L$, respectively. That is, these vectors collect the higher-frequency observations to match the lower frequency of the observed target \boldsymbol{y} . As before, we can use Bayes' rule to obtain a posterior for the factors as $\boldsymbol{f} \mid \boldsymbol{x}, \boldsymbol{y} \sim \mathcal{N}_{kL}(\boldsymbol{m}, \boldsymbol{\Omega})$, with posterior mean and covariance given by

$$\mathbf{\Omega}_{\mathrm{MF}} = \left[\mathbf{I}_{L} \otimes \left(\frac{1}{\sigma_{x}^{2}} \cdot \mathbf{P}^{\top} \mathbf{P} + \mathbf{V}_{F}^{-1} \right) + \frac{1}{L \cdot \sigma_{y}^{2}} \cdot \mathbf{1}_{L \times L} \otimes \left(\mathbf{Q}^{\top} \mathbf{Q} \right) \right]^{-1}, \tag{19}$$

$$\boldsymbol{m} = \boldsymbol{\Omega} \left[\frac{1}{\sigma_x^2} (\mathbf{I}_L \otimes \mathbf{P}^\top) \boldsymbol{x} + \frac{1}{\sigma_y^2} \cdot \mathbf{I}_L \otimes (\mathbf{Q}^\top \boldsymbol{y}) \right]. \tag{20}$$

We obtain a joint posterior for the high-frequency factors $f^{(1)}, \ldots, f^{(L)}$ where the aggregation equation (16) results in a natural correlation structure within each low-frequency period (e.g., monthly factors are naturally correlated to predict a quarterly target).

We now present how the EM algorithm for PTFA can be adjusted to handle data observed at a mixed-frequency. Let T denote the amount of lower-frequency observations available, such that there is a total of $\bar{T} = L \cdot T$ high-frequency observations (e.g., T quarters and 3T months of data). Collect all feature observations into an $T \times (pL)$ matrix X (e.g., each row has all monthly features associated to each quarter) and all targets into a $T \times q$ matrix Y. Given the new posterior of the high-frequency factors, the E-step requires the construction of the expected likelihood over θ . Collect the mean vectors m_1, \ldots, m_T associated to (20) into a $T \times (kL)$ matrix given by

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{Y} \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_x^2} \cdot \mathbf{I}_L \otimes \mathbf{P} \\ \frac{1}{\sigma_u^2} \cdot \mathbf{1}_L^\top \otimes \mathbf{Q} \end{bmatrix} \boldsymbol{\Omega}_{\mathrm{MF}}$$
(21)

Letting $V := \mathbb{E}_{F|Y,X;\theta}[F^{\top}F] = T \cdot \mathbf{\Omega}_{\mathrm{MF}} + M^{\top}M$, this means the posterior expectation of the log-likelihood now takes the form

$$Q(\theta) = -\frac{LTp}{2}\log(\sigma_x^2) - \frac{Tq}{2}\log(\sigma_y^2) - \frac{1}{2\sigma_x^2}\operatorname{Tr}\left[\boldsymbol{X}^{\top}\boldsymbol{X} - 2\boldsymbol{M}^{\top}\boldsymbol{X}(\mathbf{I}_L \otimes \mathbf{P}) + \boldsymbol{V}(\mathbf{I}_L \otimes \mathbf{P}^{\top}\mathbf{P})\right] - \frac{1}{2\sigma_y^2}\operatorname{Tr}\left[\boldsymbol{L} \cdot \boldsymbol{Y}^{\top}\boldsymbol{Y} - 2\boldsymbol{M}^{\top}\boldsymbol{Y}(\mathbf{1}_L^{\top} \otimes \mathbf{Q}) + \frac{1}{L}\boldsymbol{V}(\mathbf{1}_{L \times L} \otimes \mathbf{Q}^{\top}\mathbf{Q})\right]$$
(22)

The M-step then simplifies to obtaining update rules for all components of θ . As the features are now aggregated to a different scale compared to the target, the update steps for **P** and **Q** are no longer simplified if stacked using matrix operations. Therefore, we present updating steps whose computation will remain efficient even if computed separately.

To this end, write $X = [X^{(1)}, ..., X^{(L)}]$ and $M = [M^{(1)}, ..., M^{(L)}]$, where each $X^{(\ell)}$ block is a $T \times p$ matrix and the $M^{(\ell)}$ block is a $T \times k$ matrix, respectively for each $\ell \in \{1, ..., L\}$. Similarly, let each $k \times k$ block of V be denoted as $V_{\ell,r}$ for $\ell, r \in \{1, ..., L\}$. The update rules for the loadings for the features and targets under a mixed-frequency setting can then be expressed as

$$\mathbf{P} = \left(\sum_{\ell=1}^{L} \mathbf{X}^{(\ell)\top} \mathbf{M}^{(\ell)}\right) \left(\sum_{\ell=1}^{L} \mathbf{V}_{\ell,\ell}\right)^{-1}$$
(23)

$$\mathbf{Q} = L \cdot \left(\mathbf{Y}^{\top} \sum_{\ell=1}^{L} \mathbf{M}^{(\ell)} \right) \left(\sum_{r=1}^{L} \sum_{\ell=1}^{L} \mathbf{V}_{\ell,r} \right)^{-1}$$
(24)

As before, the first-order conditions for these updates allow us to obtain particularly simple and computationally efficient updates for σ_x^2 and σ_y^2 . Additional details are presented in Appendix B.

Finally, we note that all previous derivations can be adapted to the case when L itself changes with time, such that there are L_t high-frequency observations per low-frequency period, which is also a common occurrence in practice. For example, in macroeconomic now-casting of monthly targets such as inflation and industrial production, due to lags and complex interaction between release schedules of useful high-frequency predictors. In finance, differences in trading cycle definitions and firm-specific factors can also cause high-frequency information to be available at differing lengths. By defining a sequence (L_1, \ldots, L_T) of information availability at each time t, we can use summation notation instead of matrix operations to efficiently compute update rules without modifying the core derivations.

3.2 Stochastic Volatility

When working with economic or financial data, it is often the case that, the assumption that the parameters that govern the volatility processes in the model are constant is unrealistic and a source of misspecification. Next, we show how to allow for stochastic

volatility in the context of PTFA and the necessary changes to the EM algorithm to do so. Error covariance matrices in the context of multivariate time series models are usually modeled using multivariate stochastic volatility models, introducing significant computational costs (see, e.g., Primiceri, 2005). However, note that in our model the Gaussian noise terms are assumed isotropic, thus depending on a single constant parameter. The computational burden can be further simplified by considering recursive, simulation-free variance matrix discounting methods as in Quintana and West (1988). For σ_x and σ_y we use Exponential Weighted Moving Average (EWMA) estimators. These depend on decay factors λ_x and λ_y as follows

$$\sigma_x^2(t) = (1 - \lambda_x) \cdot \widehat{\sigma}_x^2(t) + \lambda_x \cdot \sigma_x^2(t - 1),$$

$$\sigma_y^2(t) = (1 - \lambda_y) \cdot \widehat{\sigma}_y^2(t) + \lambda_y \cdot \sigma_y^2(t - 1),$$

where $\hat{\sigma}_x^2(t)$ and $\hat{\sigma}_y^2(t)$ are the per-period estimates obtained from our model. In practice, the decay factors λ_x and λ_y are set to values close to 1, placing more weight on past volatility estimates, thereby making the process smoother and ensuring progressive learning from new data. Setting $\lambda_x = \lambda_y = 0$ mutes stochastic volatility completely and can be made equivalent to the static case by choosing the final estimate of the variances as the time-averages of $\sigma_x^2(t)$ and $\sigma_y^2(t)$.

These EWMA processes allow us to dynamically adjust the volatility estimates as the model iterates through time, capturing the time-varying nature of the volatility in both features and targets. The estimated volatilities, $\sigma_x(t)$ and $\sigma_y(t)$, are then used in the next iteration of the model, ensuring volatility is incorporated into parameter estimation. This iterative procedure ensures that the volatilities evolve over time, reflecting the dynamic nature of the system. We note that beyond being computationally trivial and very flexible, the EWMA provides an accurate approximation to an integrated GARCH process. The full EM implementation with this extension can be found in Algorithm 4 of Appendix C.

4 Simulation Exercises

In this section, we present several simulation exercises conducted to evaluate the performance of PTFA compared to traditional factor extraction techniques. The goal is to assess the models' accuracy in predicting response variables under data-generating processes for the noise in predictor and response variables.

The first step of the simulation revolves around the factor structure of predictors and targets before adding noise. Throughout all exercises, we set T = 200, p = 10, q = 3, and k = 2. We first draw all entries in the loadings **P** and **Q** from a uniform distribution between 0 and 1. Then, we generate $f_t \sim \mathcal{N}_k(\mathbf{0}_k, \mathbf{I}_k)$ independently for each time period $t \in \{1, ..., T\}$. Given the factors and loadings, we finally generate features and targets according to equations (2) and (3), respectively.

The main differences across each of the DGPs is the distribution of the errors e_x and e_y . For

the simplest (and correctly specified) DGP, we consider isotropic Gaussian errors

$$e_x \sim \mathcal{N}_p(\mathbf{0}_p, \sigma_x^2 \mathbf{I}_p)$$
 and $e_y \sim \mathcal{N}_q(\mathbf{0}_q, \sigma_y^2 \mathbf{I}_q)$. (DGP.1)

Figure 1 summarizes the key finding on a single realization of simulated data from DGP.1, where we fix $\sigma_x = \sigma_y = 1$. Note how the predicted targets from PTFA align more closely to the true targets when compared to standard PLS. The R^2 score value for each is also higher resulting in an average score of 68.1% for PTFA compared to 56.5% in standard PLS on this single realization. Figure 2 presents the path taken by the values of R^2 of the fit as the EM iterations of our algorithm progress. Notice how the algorithm quickly adapts to a large level of explained variance in the targets and levels off once the estimates reach numerical convergence as measured by the ℓ_2 distance between iterates.

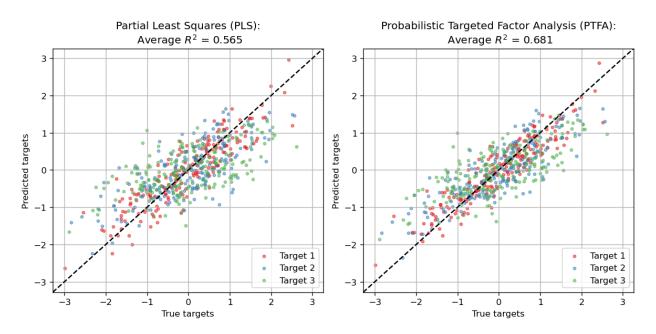


Figure 1: Comparison of PLS and PTFA on a single realization of simulated data with independent Gaussian errors (DGP.1)

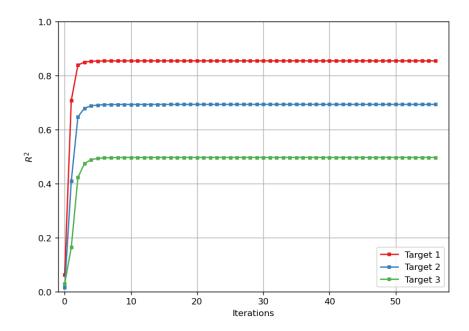


Figure 2: Path of R^2 of fit on a single realization of simulated data with independent Gaussian errors (DGP.1)

Crucially, Figure 3 showcases that these performance gains do not depend on any given realization of data. By comparing the average (across targets) R^2 statistics over 1000 replications of the previous setting, we find that PTFA first-order stochastic-dominates PLS in generating better in-sample fit.

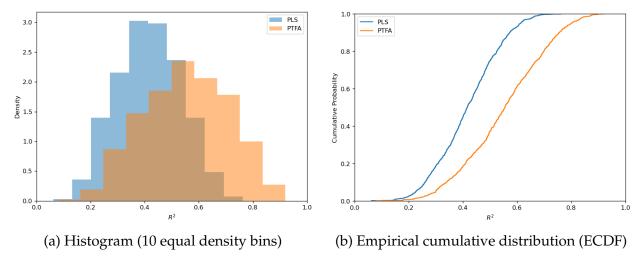


Figure 3: Comparison of the distributions of R^2 statistics between PLS and PTFA across 1000 replications of DGP.1

As discussed in the main text and formally shown in Appendix A, PTFA uses the assumption of isotropic Gaussian noise terms simply to provide a probabilistic framework to targeted factor extraction. Performance of PTFA should therefore not depend on whether

feature (**X**) and target (**Y**) variables are correlated or even normally distributed. Through this and the next simulation exercises, we show that the relative performance between PTFA and PLS does not depend on the assumed distribution of the variables being decomposed.

As a first extension, we dispose of the isotropic assumption and allow for the noises to be multivariate normal distributions with non-diagonal covariance matrices. For this example, we assume the following Toeplitz covariance structure for both features and targets:

$$e_{x} \sim \mathcal{N}_{p} \begin{pmatrix} \mathbf{0}_{p}, \begin{bmatrix} 1 & \rho_{x} & \cdots & \rho_{x}^{p-1} \\ \rho_{x} & 1 & \cdots & \rho_{x}^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{x}^{p-1} & \rho_{x}^{p-2} & \cdots & 1 \end{bmatrix} \end{pmatrix} \text{ and } e_{y} \sim \mathcal{N}_{q} \begin{pmatrix} \mathbf{0}_{q}, \begin{bmatrix} 1 & \rho_{y} & \cdots & \rho_{y}^{q-1} \\ \rho_{y} & 1 & \cdots & \rho_{y}^{q-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{y}^{q-1} & \rho_{y}^{q-2} & \cdots & 1 \end{bmatrix} \end{pmatrix},$$

$$(DGP.2)$$

where $\rho_x, \rho_y \in [-1, 1]$ are correlation parameters. Figures 4 and 5 present the same statistics as before for a realization of data from DGP.2 using $\rho_x = \rho_y = 0.5$ (keeping the remaining values the same as in the previous exercise). Similar results as in the isotropic Gaussian case are obtained, with PTFA dominating PLS in terms of in-sample fit within only a small number of iterations of the EM algorithm.

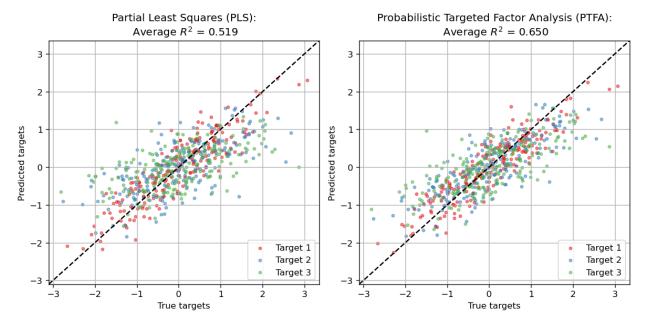


Figure 4: Comparison of PLS and PTFA on a single realization of simulated data with correlated Gaussian errors (DGP.2)

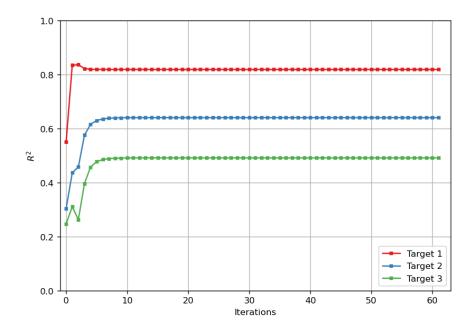


Figure 5: Path of R^2 of fit on a single realization of simulated data with correlated Gaussian errors (DGP.2)

Finally, we present evidence that similar results hold once we dispense the assumption of Gaussian noise altogether. Specifically, we consider the following setup design to produce heavy-tailed and asymmetric noise that results in clear deviations from Gaussian features and targets. Errors in features are drawn independently from a Student-t distributions with 3 degrees-of-freedom and scale σ_x , while target noise is drawn from a χ^2 distribution with 1 degree of freedom.

$$e_{x,j} \stackrel{iid}{\sim} \sigma_x \cdot t_3, j \in \{1, \dots, p\}$$

$$e_{y,j} \stackrel{iid}{\sim} \chi_1^2, j \in \{1, \dots, q\}.$$
(DGP.3)

Figures 6 and 7 showcase that similar results to before arise from specification DGP.3. As long as the data is standardized prior to processing, it can be observed that PTFA will deliver targeted factors that are generally more accurate to summarize the information in the targets regardless of the distributions of the variables involved.

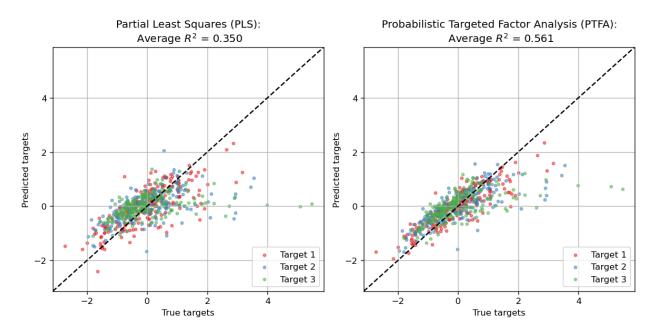


Figure 6: Comparison of PLS and PTFA on a single realization of simulated data with heavy-tailed non-Gaussian errors (DGP.3)

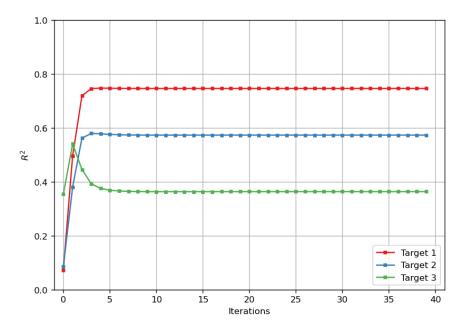


Figure 7: Path of R^2 of fit on a single realization of simulated data with heavy-tailed non-Gaussian errors (DGP.3)

Finally, given the critical role of noise in explaining the virtue of PLS, we additionally simulate data with differing levels of noise in both features **X** and targets **Y**. That is, once again we simulate noisy data from DGP.1, adjusting both error scales σ_x and σ_y over a grid between 0.1 and 5, covering a wide range of signal-to-noise ratios.

Figure 8 shows the median value of average R^2 statistics across targets over 1000 replications of this simulation setup. The superior performance of PTFA is evident from the heatmap. The gains in terms of goodness-of-fit of PTFA when compared to PLS are more salient when noise is increases, in particular when the noise is in the targets instead of the features. This is as seen in the PCA case, where perturbations to the data in the form of noise or outliers creates issues for consistently recovering the axes of maximal variance (Chen et al., 2021).

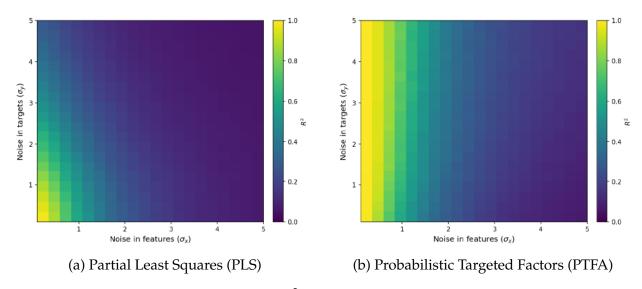


Figure 8: Comparison of the median R^2 statistics for PLS (a) and PTFA (b) across 1000 replications from DGP.1, varying noise in features (σ_x) and targets (σ_y)

5 Empirical Applications

5.1 Macroeconomic Forecasting

Partial Least Squares is particularly valuable when the key motivation for dimensionality reduction is prediction. To demonstrate the practical relevance of PTFA, we conduct a simple macroeconomic forecasting exercise using the Federal Reserve Economic Data – Monthly Database (FRED-MD; for details, see, McCracken and Ng, 2016). The data includes 126 variables that track macroeconomic developments in the United States at a monthly frequency. In this section, we discuss the main results and compare PTFA with popular alternatives in the literature.

Table 1: Out-of-Sample Performance of PTFA

Horizon	PCA	PLS	PTFA	PTFA-SV				
Industrial Production								
1	0.7990	2.6746	0.6996	0.7003				
4	0.8747	1.2164	0.8363	0.8367				
6	0.9192	1.3295	0.8716	0.8739				
12	0.9966	1.4466	0.9116	0.9138				
CPI Inflation								
1	1.0155	1.2575	0.9903	0.9912				
4	1.0460	1.6858	1.0033	1.0038				
6	1.0277	1.8232	1.0018	1.0011				
12	1.0368	1.7118	1.0119	1.0123				
Unemployment rate								
1	1.2254	1.0781	0.9048	0.9051				
4	1.1147	1.3735	1.0069	1.0094				
6	1.0569	2.2511	1.0117	1.0121				
12	1.0507	1.5848	1.0089	1.0094				

Notes: MSFE statistics calculated out-of-sample, with a rolling window of 200 monthly observations for forecast horizons of 1, 4, 6, and 12 months and a full-sample 1961M7-2023M3. PTFA, PTFA-SV and PLS are implemented according to algorithms 1 and 2 and PCA as used by McCracken and Ng (2016) to construct FRED-MD factors. 7 factors are used across models, consistent with FRED-MD. Both targets and predictors are standardized prior to estimation.

Table 1 shows out-of-sample Mean Squared Forecast Error (MSFE) of prediction for each target variable using 7 factors extracted with either PLS, PTFA, PTFA-SV or and PCA. The targets in our exercise are Industrial Production, CPI inflation and the Unemployment Rate. The number of factors is chosen to be consistent with the FRED-MD factors calculated according to McCracken and Ng (2016), with code made available by the authors. The key message from Table 1 is that the PTFA outperforms both PLS and PCA, adding value to forecasts across forecast horizons and macroeconomic variables considered. However, we also observe that the variant of PTFA with stochastic volatility (PTFA-SV) does not seem to add value to forecasts, above and beyond the baseline model.

5.2 Predicting the Equity Premium

Attempts to predict stock returns or the equity premium are in no short supply in the finance literature. Welch and Goyal (2007) and Goyal et al. (2024), provide a review and comprehensive assessment of the performance of 46 different variables that have been suggested by the academic literature to be good predictors of the equity premium. Following this large body of empirical work, our financial application studies the predictability of U.S. aggregate stock returns, using the Goyal et al. (2024) dataset.

The goal of this exercise is to predict the equity risk premia, using 26 signals which are available at a monthly frequency. Thus, in this case Y is the equity premium and X are the various predictors, lagged by one period, following standard practice. In this setting, it is less clear how many factors k should be considered. Therefore, we choose k by cross-validation and use the same parameter across all competing models but also report results for different choices of k. As for the forecasting horizon, we only consider 1 month ahead forecasts given the nature of the problem of forecasting stock returns, since information is priced-in quite fast.

Table 2: Out-of-Sample Performance of PTFA

Model	k = 1	k = 2	k = 3	k = 4	k = 5
PCA	1.0678	1.1182	1.1261	1.1570	1.1780
PLS	1.1043	1.2110	1.3280	1.5425	1.7775
PTFA	1.0139	1.0363	1.0557	1.0680	1.1027
PTFA-SV	1.0136	1.0403	1.0648	1.0772	1.1335

Notes: MSFE statistics calculated one-month-ahead out-of-sample on a rolling window of 60 monthly observations on a sample 1926M1-2023M12. The table shows results for different values of *k* (number of factors) using PLS, PTFA, PTFA-SV and PCA models. Both targets (equity premium) and predictors (26 monthly signals from Goyal et al., 2024) are standardized prior to estimation.

The main message from Table 2 is that PTFA adds value as compared to PLS and PCA in predicting the equity risk premia. Similar to our application with macroeconomic data, our model with stochastic volatility (PTFA-SV) does not seem to outperform PTFA, that is relatively more parsimonious. We observe that MSFE associated to PTFA forecasts are quite competitive, regardless of how many factors one chooses.

6 Conclusion

We introduce a probabilistic framework for targeted factor extraction called PTFA and derive a fast Expectation-Maximization (EM) algorithm to estimate the model. PTFA is flexible and naturally handles parameter uncertainty, noise, and missing data in estimation. Through simulation exercises and two real-world applications in macroeconomic forecasting and stock return prediction, we demonstrate the superior performance of PTFA, especially in noisy and incomplete data environments. Along the way, we provide additional contributions to mixed-frequency data, stochastic volatility in time series, and give further theoretical insight to the probabilistic PLS solutions. This probabilistic foundation also opens many avenues for future research, including interesting methodological extensions using probabilistic (fully Bayesian) or variational inference. By providing an open-source implementation of the method, our hope is that practitioners of time-series forecasting and researchers alike will continue to expand and improve upon the technique.

References

- Butler, N. A. and Denham, M. C. (2002). The Peculiar Shrinkage Properties of Partial Least Squares Regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 62(3):585–593.
- Chen, Y., Chi, Y., Fan, J., and Ma, C. (2021). Spectral Methods for Data Science: A Statistical Perspective. *Foundations and Trends® in Machine Learning*, 14(5).
- el Bouhaddani, S., Uh, H.-W., Hayward, C., Jongbloed, G., and Houwing-Duistermaat, J. (2018). Probabilistic partial least squares model: Identifiability, estimation and application. *Journal of Multivariate Analysis*, 167:331–346.
- el Bouhaddani, S., Uh, H.-W., Jongbloed, G., and Houwing-Duistermaat, J. (2022). Statistical Integration of Heterogeneous Omics Data: Probabilistic Two-Way Partial Least Squares (PO2PLS). *Journal of the Royal Statistical Society Series C: Applied Statistics*, 71(5):1451–1470.
- Foroni, C. and Marcellino, M. (2014). Mixed-frequency Structural Models: Identification, Estimation, and Policy Analysis. *Journal of Applied Econometrics*, 29(7):1118–1144. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jae.2396.
- Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135. Publisher: Taylor & Francis.
- Giannone, D., Reichlin, L., and Small, D. (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics*, 55(4):665–676.
- Giglio, S., Kelly, B., and Pruitt, S. (2016). Systemic risk and the macroeconomy: An empirical evaluation. *Journal of Financial Economics*, 119(3):457–471.
- Gong, G. and Samaniego, F. J. (1981). Pseudo Maximum Likelihood Estimation: Theory and Applications. *The Annals of Statistics*, 9(4):861–869. Publisher: Institute of Mathematical Statistics.
- Gourieroux, C., Monfort, A., and Trognon, A. (1984). Pseudo Maximum Likelihood Methods: Theory. *Econometrica*, 52(3):681–700. Publisher: [Wiley, Econometric Society].
- Goyal, A., Welch, I., and Zafirov, A. (2024). A Comprehensive 2022 Look at the Empirical Performance of Equity Premium Prediction. *Review of Financial Studies*, page hhae044.
- Greenberg, E. (2012). Introduction to Bayesian Econometrics. Cambridge University Press.
- Groen, J. J. J. and Kapetanios, G. (2016). Revisiting useful approaches to data-rich macroe-conomic forecasting. *Computational Statistics & Data Analysis*, 100:221–239.
- Gustafsson, M. G. (2001). A Probabilistic Derivation of the Partial Least-Squares Algorithm. *Journal of Chemical Information and Computer Sciences*, 41(2):288–294. Publisher: American Chemical Society.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.
- Kelly, B. and Pruitt, S. (2013). Market Expectations in the Cross-Section of Present Values. *The Journal of Finance*, 68(5):1721–1756.

- Kelly, B. and Pruitt, S. (2015). The three-pass regression filter: A new approach to forecasting using many predictors. *Journal of Econometrics*, 186(2):294–316. Publisher: Elsevier.
- Li, Q., Pan, F., Zhao, Z., and Yu, J. (2018). Process Modeling and Monitoring With Incomplete Data Based on Robust Probabilistic Partial Least Square Method. *IEEE Access*, 6:10160–10168. Conference Name: IEEE Access.
- Li, S., Gao, J., Nyagilo, J. O., and Dave, D. P. (2011). Probabilistic Partial Least Square Regression: A Robust Model for Quantitative Analysis of Raman Spectroscopy Data. In 2011 IEEE International Conference on Bioinformatics and Biomedicine, pages 526–531.
- Little, R. J. A. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.
- McCracken, M. W. and Ng, S. (2016). FRED-MD: A Monthly Database for Macroeconomic Research. *Journal of Business & Economic Statistics*, 34(4):574–589. Publisher: Taylor & Francis.
- Primiceri, G. E. (2005). Time varying structural vector autoregressions and monetary policy. *Review of Economic Studies*, 72(3):821–852. Publisher: Wiley-Blackwell.
- Quintana, J. M. and West, M. (1988). Time series analysis of compositional data. *Bayesian Statistics*, 3:747–756.
- Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622. Publisher: Oxford University Press.
- Vidaurre, D., van Gerven, M. A. J., Bielza, C., Larrañaga, P., and Heskes, T. (2013). Bayesian Sparse Partial Least Squares. *Neural Computation*, 25(12):3318–3339. Conference Name: Neural Computation.
- Welch, I. and Goyal, A. (2007). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *Review of Financial Studies*, 21(4):1455–1508.
- Wold, H. (1975). Soft Modelling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach. *Journal of Applied Probability*, 12(S1):117–142.
- Wold, S., Sjöström, M., and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58(2):109–130.
- Xie, Y. (2019). Fault monitoring based on locally weighted probabilistic kernel partial least square for nonlinear time-varying processes. *Journal of Chemometrics*, 33(12):e3196. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.3196.
- Yang, X., Liu, X., and Xu, C. (2021). Robust Mixture Probabilistic Partial Least Squares Model for Soft Sensing With Multivariate Laplace Distribution. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9. Conference Name: IEEE Transactions on Instrumentation and Measurement.
- Zheng, J. and Song, Z. (2018). Semisupervised learning for probabilistic partial least squares regression model and soft sensor application. *Journal of Process Control*, 64:123–131.

Zheng, J., Song, Z., and Ge, Z. (2016). Probabilistic learning of partial least squares regression model: Theory and industrial applications. *Chemometrics and Intelligent Laboratory Systems*, 158:80–90.

Appendices

A MLE Theory

Our PTFA is formulated to provide a probabilistic foundation to PLS. In the body of the paper we propose an iterative procedure to estimate the loadings and data variances associated to our model. In this section we show that a maximum likelihood estimator (MLE) of these quantities reproduces the standard PLS solution. This is akin to the results by Tipping and Bishop (1999) who propose their probabilistic version of PCA and show the same axes of maximal variance are obtained from a MLE of their model. Therefore, our setup also allows us to recover the frequentist properties of estimators for factor loadings and data variances.

Recall from (4) that the distribution of one realization of the data conditional on the factors is Gaussian with mean vector $\boldsymbol{\mu} := [\boldsymbol{f}^{\top} \mathbf{P}^{\top}, \boldsymbol{f}^{\top} \mathbf{Q}^{\top}]^{\top}$ and covariance matrix $\boldsymbol{\Sigma} := \operatorname{diag}(\sigma_x^2 \mathbf{I}_p, \sigma_y^2 \mathbf{I}_q)$. Recall we have also stacked the data and loadings into $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$ and $\mathbf{L} = [\mathbf{P}^{\top}, \mathbf{Q}^{\top}]^{\top}$, respectively. Collecting all disturbances $e_t := [e_{x,t}^{\top}, e_{y,t}^{\top}]$ and stacking into $\mathbf{E} := [e_1^{\top}, \dots, e_n^{\top}]$, we can express our model equations (2) and (3) succinctly as

$$\mathbf{Z} = \mathbf{F} \mathbf{L}^{\top} + \mathbf{E}. \tag{A.1}$$

Integrating out the factors according to their $\mathcal{N}(\mathbf{0}_k, \mathbf{V}_F)$ prior distribution results in the marginal log-likelihood of the data as a function of the loadings, denoted as $\ell(\theta)$. Write $\mathbf{S} := n^{-1}\mathbf{Z}^{\top}\mathbf{Z}$ for the sample data covariance (the columns of \mathbf{X} and \mathbf{Y} are standardized independently) and $\mathbf{C} := \mathbf{L}\mathbf{V}_F\mathbf{L}^{\top} + \mathbf{\Sigma}$ for the model variance. The log-likelihood can then be expressed as:

$$\ell(\theta) = -\frac{T}{2} \left[d \log(2\pi) + \log |\mathbf{C}| + \text{Tr}(\mathbf{C}^{-1}\mathbf{S}) \right]. \tag{A.2}$$

The MLE of θ , denoted as $\widehat{\theta}=(\widehat{\mathbf{P}},\widehat{\mathbf{Q}},\widehat{\sigma}_x^2,\widehat{\sigma}_y^2)$, is therefore given as the solution to

$$\widehat{\theta} := \arg\max_{\theta} \ell(\theta) \,. \tag{A.3}$$

Note that the assumption of uncorrelated Gaussian errors is made for convenience as it results in a particularly simple likelihood structure. If we assume the factor decomposition to be correctly specified —i.e., that there exists an $\mathbf{L} \in \mathbb{R}^{d \times k}$ such that $\mathbb{E}[\mathbf{Z} \mid \mathbf{F}] = \mathbf{F}\mathbf{L}^{\top})$ — then, under regularity conditions, our estimator will remain consistent even if the data is subject to other kind of more complex error processes (due to standard quasi-maximum likelihood results; see for example Gong and Samaniego, 1981; Gourieroux et al., 1984).

While we use the maximum likelihood moniker following earlier work by Tipping and Bishop (1999), $\hat{\theta}$ is technically a *maximum a-posteriori* (MAP) estimator as it can depend on the assumed factor prior variance V_F . For the remainder of the derivation and to

economize on notation we set $V_F = I_k$, as it also results in the canonical PLS formulation (for more details, see Frank and Friedman, 1993; Hastie et al., 2001).

The parameters θ enter the likelihood only through the value of **C**, with a gradient equal to

$$\frac{\partial \ell(\theta)}{\partial \mathbf{C}} = -\frac{n}{2} \left(\mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{S} \mathbf{C}^{-1} \right). \tag{A.4}$$

Writing $\hat{\mathbf{L}} = [\hat{\mathbf{P}}^{\top}, \hat{\mathbf{Q}}^{\top}]^{\top}$, $\hat{\mathbf{\Sigma}} := \operatorname{diag}(\hat{\sigma}_x^2 \mathbf{I}_p, \hat{\sigma}_y^2 \mathbf{I}_q)$, and $\hat{\mathbf{C}} := \hat{\mathbf{L}}\hat{\mathbf{L}}^{\top} + \hat{\mathbf{\Sigma}}$, this means that the maximum likelihood solutions for the loadings \mathbf{L} are characterized by

$$\mathbf{0}_{d\times k} = \left[\frac{\partial \ell(\widehat{\boldsymbol{\theta}})}{\partial \mathbf{C}}\right]^{\top} \frac{\partial \ell(\widehat{\boldsymbol{\theta}})}{\partial \mathbf{L}} \implies (\mathbf{S}\widehat{\mathbf{C}}^{-1} - \mathbf{I}_d) \begin{bmatrix} \widehat{\mathbf{P}} \\ \widehat{\mathbf{Q}} \end{bmatrix} = \mathbf{0}_{d\times k}. \tag{A.5}$$

The first-order condition (A.5) exhibits the same three classes of solutions as explored by Tipping and Bishop (1999), plus an additional interesting special case. First, the trivial solution sets $\hat{\mathbf{L}} = \mathbf{0}_{d \times k}$, which represents a minimum of the log-likelihood $\ell(\theta)$. Second, we obtain a solution if we assume our implied model variance to equal the data variance, such that $\hat{\mathbf{C}} = \mathbf{S}$. Letting \mathbf{S}_X , \mathbf{S}_Y and \mathbf{S}_{XY} denote the blocks of \mathbf{S} partitioned according to features and targets, we can then identify the components $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ from the set of equations given by

$$\widehat{\mathbf{P}}\widehat{\mathbf{P}}^{\top} = \mathbf{S}_{X} - \widehat{\sigma}_{x}^{2} \mathbf{I}_{p},$$

$$\widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^{\top} = \mathbf{S}_{Y} - \widehat{\sigma}_{y}^{2} \mathbf{I}_{q},$$

$$\widehat{\mathbf{P}}\widehat{\mathbf{Q}}^{\top} = \mathbf{S}_{XY}.$$
(A.6)

Note that equations (A.6) require the last p - k eigenvalues of S_X to be equal to each other, and similarly for the last q - k eigenvalues of S_Y . In this case, $\hat{\mathbf{P}}$ is constructed from the eigenvectors of S_X , $\hat{\mathbf{Q}}$ from the eigenvectors of S_Y , and the components are rotated to ensure $\hat{\mathbf{P}}\hat{\mathbf{Q}}^{\top} = \mathbf{S}_{XY}$.

An explicit construction for the solutions in this case can be provided as follows. Let $\mathbf{X}_k := \mathbf{U}_X \mathbf{D}_X \mathbf{V}_X^\top$ and $\mathbf{Y}_k := \mathbf{U}_Y \mathbf{D}_Y \mathbf{V}_Y^\top$ be the best rank k approximations to the data matrices \mathbf{X} and \mathbf{Y} , respectively. Therefore, \mathbf{U}_X and \mathbf{U}_Y are $T \times k$ matrices with orthogonal columns containing the left singular vectors, \mathbf{V}_X and \mathbf{V}_Y are $k \times k$ orthogonal matrices containing the right singular vectors, and \mathbf{D}_X and \mathbf{D}_Y are $k \times k$ diagonal matrices holding the largest k singular values for features \mathbf{X} and targets \mathbf{Y} , respectively. One can then check the following solutions satisfy (A.6):

$$\widehat{\mathbf{P}} = \mathbf{V}_{X} \left(\frac{1}{T} \mathbf{D}_{X}^{2} - \widehat{\sigma}_{x}^{2} \mathbf{I}_{k} \right)^{1/2} \mathbf{V}_{P}^{\top},$$

$$\widehat{\mathbf{Q}} = \mathbf{V}_{Y} \left(\frac{1}{T} \mathbf{D}_{Y}^{2} - \widehat{\sigma}_{y}^{2} \mathbf{I}_{k} \right)^{1/2} \mathbf{V}_{Q}^{\top},$$

$$\mathbf{V}_{P} = \mathbf{V}_{Q} \left(\mathbf{D}_{Y}^{2} - T \cdot \widehat{\sigma}_{y}^{2} \mathbf{I}_{k} \right)^{-1/2} \mathbf{D}_{Y} \mathbf{U}_{Y}^{\top} \mathbf{U}_{X} \mathbf{D}_{X} \left(\mathbf{D}_{X}^{2} - T \cdot \widehat{\sigma}_{x}^{2} \mathbf{I}_{k} \right)^{-1/2}$$
(A.7)

This solution leaves \mathbf{V}_Q as an arbitrary $k \times k$ orthogonal matrix, and sets $\widehat{\sigma}_x^2$ equal to the smallest eigenvalue of \mathbf{S}_X and $\widehat{\sigma}_y^2$ equal to the smallest eigenvalue of \mathbf{S}_Y (recall the trailing eigenvalues after the k-th one are assumed equal for both \mathbf{S}_X and \mathbf{S}_X). Note that this solution corresponds to extracting factors for \mathbf{X} and \mathbf{Y} independently using PCA, and then rotating the principal axes of the feature loadings according to the weighted covariance \mathbf{S}_{XY} between features and targets.

The third class of solutions represents the most interesting case likely to be encountered in practice, where (A.5) is satisfied but $\hat{C} \neq S$ so that our covariance model is misspecified. That is, we recognize that the isotropic Gaussian error terms in (2)–(3) are not to be taken as true modeling choices for the features nor targets, and are rather used as a working assumption to obtain a framework for probabilistic targeted factor recovery.

Define the marginal covariance matrices implied from the model as $\widehat{\mathbf{C}}_X := \widehat{\mathbf{P}}\widehat{\mathbf{P}}^\top + \widehat{\sigma}_x^2\mathbf{I}_p$ and $\widehat{\mathbf{C}}_Y := \widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^\top + \widehat{\sigma}_y^2\mathbf{I}_q$. Additionally, define the conditional covariance of \mathbf{Y} given \mathbf{X} implied from the model as $\widehat{\mathbf{C}}_{Y|X} := \widehat{\mathbf{C}}_Y - \widehat{\mathbf{Q}}\widehat{\mathbf{P}}^\top\widehat{\mathbf{C}}_X^{-1}\widehat{\mathbf{P}}\widehat{\mathbf{Q}}^\top$. Using these definitions, the system of equations (A.5) can be expressed as

$$(\mathbf{S}_{X}\widehat{\mathbf{C}}_{X}^{-1} - \mathbf{I}_{p})\widehat{\mathbf{P}} + \frac{1}{T}\mathbf{X}^{\top}(\mathbf{Y} - \mathbf{X}\widehat{\mathbf{C}}_{X}^{-1}\widehat{\mathbf{P}}\widehat{\mathbf{Q}}^{\top})\widehat{\mathbf{C}}_{Y|X}^{-1}\widehat{\mathbf{Q}}(\mathbf{I}_{k} - \widehat{\mathbf{P}}^{\top}\widehat{\mathbf{C}}_{X}^{-1}\widehat{\mathbf{P}}) = \mathbf{0}_{p \times k}$$

$$(\mathbf{S}_{XY}^{\top}\widehat{\mathbf{C}}_{X}^{-1}\widehat{\mathbf{P}} - \widehat{\mathbf{Q}}) + \frac{1}{T}\mathbf{Y}^{\top}(\mathbf{Y} - \mathbf{X}\widehat{\mathbf{C}}_{X}^{-1}\widehat{\mathbf{P}}\widehat{\mathbf{Q}}^{\top})\widehat{\mathbf{C}}_{Y|X}^{-1}\widehat{\mathbf{Q}}(\mathbf{I}_{k} - \widehat{\mathbf{P}}^{\top}\widehat{\mathbf{C}}_{X}^{-1}\widehat{\mathbf{P}}) = \mathbf{0}_{q \times k}$$
(A.8)

This representation of the first-order conditions showcases that the loadings can be obtained as the solution to the simpler system

$$\widehat{\mathbf{P}} = \mathbf{S}_X \widehat{\mathbf{C}}_X^{-1} \widehat{\mathbf{P}} \tag{A.9}$$

$$\widehat{\mathbf{Q}} = \mathbf{S}_{XY}^{\top} \widehat{\mathbf{C}}_{X}^{-1} \widehat{\mathbf{P}} \tag{A.10}$$

$$\mathbf{Y} = \mathbf{X}\widehat{\mathbf{C}}_X^{-1}\widehat{\mathbf{P}}\widehat{\mathbf{Q}}^{\top} \tag{A.11}$$

Equations (A.9) and (A.10) jointly imply that the columns of $\hat{\mathbf{P}}$ take into account information from the eigenvectors of the feature variance \mathbf{S}_X and covariance matrix \mathbf{S}_{XY} . Finally, equation (A.11) provides the explicit decomposition satisfied by the estimated loadings $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$. This is precisely the prediction produced by the canonical PLS setting with multiple targets. These results showcase that our PTFA framework indeed provides a probabilistic foundation for standard PLS as it reproduces the spirit of the non-random solution from the NIPALS algorithm.

Finally, we note one more special case of interest that does not have an analogue to the PPCA framework in Tipping and Bishop (1999). Specifically, this occurs when one is interested in targeted recovery of factors for a set of response variables with a *smaller* number of variables than the suspected number of components, such that q < k. In this case, as the column space of **Y** can be spanned by linear combinations of the k scores without loss of generality (as long as they are orthogonal), we can directly assume $\hat{\mathbf{C}}_Y = \mathbf{S}_Y$. In this case, loadings $\hat{\mathbf{Q}}$ can be assumed orthogonal and can be recovered using PCA. However, we should still allow for $\hat{\mathbf{C}}_X \neq \mathbf{S}_X$. In this case, the solution is a hybrid solution between (A.7) and (A.9)–(A.11).

B EM Derivation

We let $\widetilde{\theta}$ represent an initial or current fixed value of the parameters. The expectation (E) step requires us to obtain the observed-data likelihood by integrating out the factors according to their *posterior* distribution. This results in the objective function:

$$Q(\theta \mid \widetilde{\theta}) = \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y}:\widetilde{\theta}}[\log p(\mathbf{X},\mathbf{Y},\mathbf{F} \mid \theta)]$$

Our framework allows us to produce a closed-form solution for the E-step. The maximization (M) step, then consists in optimizing $Q(\cdot)$ with respect to the variables and parameters with a view of deriving updating rules. These updating rules are derived using the following steps:

1. To update **P**, we maximize the expected log-likelihood term involving **X**

$$Q_X(\mathbf{P}) := \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\widetilde{\theta}} \left[-\frac{1}{2\sigma_x^2} \|\mathbf{X} - \mathbf{F}\mathbf{P}^\top\|_F^2 \right]$$

expanding the norm and taking expectations we get

$$Q_X(\mathbf{P}) = -\frac{1}{2\sigma_v^2} \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\widetilde{\boldsymbol{\theta}}} \left[\text{Tr} \left((\mathbf{X} - \mathbf{F} \mathbf{P}^\top)^\top (\mathbf{X} - \mathbf{F} \mathbf{P}^\top) \right) \right]$$

This simplifies to:

$$Q_X(\mathbf{P}) = -\frac{1}{2\sigma_x^2} \left[\|\mathbf{X}\|_F^2 - 2\operatorname{Tr}(\mathbf{X}^\top \mathbf{M} \mathbf{P}^\top) + \operatorname{Tr}(\mathbf{P} \mathbf{V} \mathbf{P}^\top) \right]$$

Maximizing this quadratic form in **P** gives the first-order condition $\mathbf{X}^{\mathsf{T}}\mathbf{M} = \mathbf{P}\mathbf{V}$, which can be solved to yield:

$$\mathbf{P} = \mathbf{X}^{\top} \mathbf{M} \mathbf{V}^{-1} \tag{B.1}$$

2. To update **Q**, we maximize the expected log-likelihood term involving **Y**:

$$Q_{Y}(\mathbf{Q}) := \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\widetilde{\theta}} \left[-\frac{1}{2\sigma_{y}^{2}} \|\mathbf{Y} - \mathbf{F}\mathbf{Q}^{\top}\|_{F}^{2} \right]$$

Expanding and simplifying, we obtain an expression similar to before:

$$Q_{Y}(\mathbf{Q}) = -\frac{1}{2\sigma_{y}^{2}} \left[\|\mathbf{Y}\|_{F}^{2} - 2\operatorname{Tr}(\mathbf{Y}^{\top}\mathbf{M}\mathbf{Q}^{\top}) + \operatorname{Tr}(\mathbf{Q}\mathbf{V}\mathbf{Q}^{\top}) \right]$$

Maximizing this quadratic form in Q gives the first-order condition $Y^TM = QV$, which can be solved to yield:

$$\mathbf{Q} = \mathbf{Y}^{\mathsf{T}} \mathbf{M} \mathbf{V}^{-1} \tag{B.2}$$

3. To update σ_x^2 , We only need the term involving σ_x^2 :

$$\begin{split} Q(\sigma_x^2) &= -\frac{Tp}{2}\log(\sigma_x^2) - \frac{1}{2\sigma_x^2} \mathbb{E}_{\mathbf{F}|\mathbf{X},\mathbf{Y};\widetilde{\theta}} \left[\|\mathbf{X} - \mathbf{F}\mathbf{P}^\top\|_F^2 \right] \\ &= -\frac{Tp}{2}\log(\sigma_x^2) - \frac{1}{2\sigma_x^2} \left[\|\mathbf{X}\|_F^2 - 2\operatorname{Tr}(\mathbf{X}^\top \mathbf{M}\mathbf{P}^\top) + \operatorname{Tr}(\mathbf{P}\mathbf{V}\mathbf{P}^\top) \right] \end{split}$$

To maximize $Q(\sigma_x^2)$ with respect to σ_x^2 , take the derivative and set it to zero:

$$\frac{\partial Q(\sigma_x^2)}{\partial \sigma_x^2} = -\frac{Tp}{2\sigma_x^2} + \frac{1}{2\sigma_x^4} \left[\|\mathbf{X}\|_F^2 - 2\operatorname{Tr}(\mathbf{X}^\top \mathbf{M} \mathbf{P}^\top) + \operatorname{Tr}(\mathbf{P} \mathbf{V} \mathbf{P}^\top) \right] = 0$$

Solving for σ_x^2 :

$$\sigma_x^2 = \frac{1}{Tp} \left[\|\mathbf{X}\|_F^2 - 2 \operatorname{Tr}(\mathbf{X}^\top \mathbf{M} \mathbf{P}^\top) + \operatorname{Tr}(\mathbf{P} \mathbf{V} \mathbf{P}^\top) \right]$$

Using the first-order condition satisfied by **P** and combining the terms in the previous expression, our estimate can be succinctly computed as:

$$\sigma_x^2 = \frac{1}{Tp} \left[\|\mathbf{X}\|_F^2 - \text{Tr}(\mathbf{P}^\top \mathbf{P} \mathbf{V}) \right]$$

4. To update σ_y^2 , similar calculations can be performed as in the last step to find:

$$\sigma_y^2 = \frac{1}{Tq} \left[\|\mathbf{Y}\|_F^2 - \text{Tr}(\mathbf{Q}^\top \mathbf{Q} \mathbf{V}) \right]$$

C Algorithms

Algorithm 1 EM Algorithm for Probabilistic Targeted Factor Extraction

Require: Predictor matrix $\mathbf{X} \in \mathbb{R}^{T \times p}$, target matrix $\mathbf{Y} \in \mathbb{R}^{T \times q}$, number of components k, starting values $(\mathbf{P}_0, \mathbf{Q}_0, \sigma^2_{x,0}, \sigma^2_{y,0})$, prior variance \mathbf{V}_F , tolerance ϵ , maximum iterations

1: Center and scale the Data:

$$\mathbf{X} \leftarrow (\mathbf{X} - \mathbf{1}_T \mathbf{m}_{\mathbf{x}}^{\top}) \operatorname{diag}(\mathbf{s}_{\mathbf{x}})^{-1}, \quad \mathbf{Y} \leftarrow (\mathbf{Y} - \mathbf{1}_T \mathbf{m}_{\mathbf{y}}^{\top}) \operatorname{diag}(\mathbf{s}_{\mathbf{y}})^{-1}$$

where for the columns of **X** and **Y**, $m_x := (1/T) \sum_{t=1}^T x_t$ and $m_y := (1/T) \sum_{t=1}^T y_t$ are the vector of means, whereas s_x and s_y are the vectors of standard deviations, respectively.

2: repeat

3: **E-step: Expectation**

4: Collect all initial parameters into $\theta_0 \leftarrow (\mathbf{P}_0, \mathbf{Q}_0, \sigma_{x,0}^2, \sigma_{y,0}^2)$

5: Compute Posterior Covariance (Ω):

$$\mathbf{\Omega} \leftarrow \left(\mathbf{V}_F^{-1} + \frac{1}{\sigma_{x,0}^2} \mathbf{P}_0^\top \mathbf{P}_0 + \frac{1}{\sigma_{y,0}^2} \mathbf{Q}_0^\top \mathbf{Q}_0 \right)^{-1}$$

6: Compute Posterior Mean (**M**):

$$\mathbf{M} \leftarrow \left(\frac{1}{\sigma_{x,0}^2} \mathbf{X} \mathbf{P}_0 + \frac{1}{\sigma_{y,0}^2} \mathbf{Y} \mathbf{Q}_0\right) \mathbf{\Omega}$$

- M-step: Maximization 7:
- $\mathbf{V} \leftarrow \hat{T} \cdot \mathbf{\Omega} + \mathbf{M}^{\mathsf{T}} \mathbf{M}$ 8:
- Update **P** and **Q** jointly as: 9:

$$\begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{Q}_1 \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{X}^\top \\ \boldsymbol{Y}^\top \end{bmatrix} \boldsymbol{M} \boldsymbol{V}^{-1}$$

Update σ_x^2 : 10:

$$\sigma_{x,1}^2 \leftarrow \frac{1}{Tp} \left[\|\mathbf{X}\|_F^2 - \text{Tr}(\mathbf{P}_1^\top \mathbf{P}_1 \mathbf{V}) \right]$$

Update σ_y^2 : 11:

$$\sigma_{y,1}^2 \leftarrow \frac{1}{Tq} \left[\|\mathbf{Y}\|_F^2 - \text{Tr}(\mathbf{Q}_1^\top \mathbf{Q}_1 \mathbf{V}) \right]$$

- Collect updated parameters as $\theta_1 \leftarrow (\mathbf{P}_1, \mathbf{Q}_1, \sigma_{x,1}^2, \sigma_{y,1}^2)$ 12:
- 13: **until** convergence $\|\theta_1 \theta_0\| < \epsilon$ or S iterations are reached 14: **return** Loading matrices $\mathbf{P} \in \mathbb{R}^{p \times k}$ and $\mathbf{Q} \in \mathbb{R}^{q \times k}$, as well as noise variances σ_x^2 and σ_{ν}^2 from final estimate θ_1

Algorithm 2 EM Algorithm for Probabilistic Targeted Factor Extraction with Missing-at-Random Data

Require: Predictor matrix $\mathbf{X} \in \mathbb{R}^{T \times p}$, target matrix $\mathbf{Y} \in \mathbb{R}^{T \times q}$, number of components k, starting values $(\mathbf{P}_0, \mathbf{Q}_0, \sigma_{x,0}^2, \sigma_{y,0}^2)$, prior variance \mathbf{V}_F , tolerance ϵ , maximum iterations S

- 1: Missing value indices: let $\tau_{t,j}^{(X)} = 1$ if entry i,j of matrix \mathbf{X} is missing, 0 otherwise. Define $\tau_{t,i}^{(Y)}$ similarly for \mathbf{Y}
- 2: **Initial imputation step:** Replace $\mathbf{X}_{t,j} \leftarrow 0$ and $\mathbf{Y}_{t,j} \leftarrow 0$ if $\tau_{t,j}^{(X)} = 1$ and $\tau_{t,j}^{(Y)} = 1$, respectively
- 3: Center and scale the Data:

$$\mathbf{X} \leftarrow (\mathbf{X} - \mathbf{1}_T \mathbf{m}_{\mathbf{x}}^{\top}) \operatorname{diag}(\mathbf{s}_{\mathbf{x}})^{-1}, \quad \mathbf{Y} \leftarrow (\mathbf{Y} - \mathbf{1}_T \mathbf{m}_{\mathbf{y}}^{\top}) \operatorname{diag}(\mathbf{s}_{\mathbf{y}})^{-1}$$

where for the columns of **X** and **Y**, $m_x := (1/T) \sum_{t=1}^T x_t$ and $m_y := (1/T) \sum_{t=1}^T y_t$ are the vector of means, whereas s_x and s_y are the vectors of standard deviations, respectively.

- 4: repeat
- 5: **E-step: Expectation**
- 6: Collect all initial parameters into $\theta_0 \leftarrow (\mathbf{P}_0, \mathbf{Q}_0, \sigma_{x.0}^2, \sigma_{v.0}^2)$
- 7: Compute Posterior Covariance (Ω):

$$\mathbf{\Omega} \leftarrow \left(\mathbf{V}_F^{-1} + \frac{1}{\sigma_{x,0}^2} \mathbf{P}_0^\top \mathbf{P}_0 + \frac{1}{\sigma_{y,0}^2} \mathbf{Q}_0^\top \mathbf{Q}_0 \right)^{-1}$$

8: Compute Posterior Mean (M):

$$\mathbf{M} \leftarrow \left(\frac{1}{\sigma_{x,0}^2} \mathbf{X} \mathbf{P}_0 + \frac{1}{\sigma_{y,0}^2} \mathbf{Y} \mathbf{Q}_0 \right) \mathbf{\Omega}$$

- 9: **M-step: Maximization**
- 10: $\mathbf{V} \leftarrow T \cdot \mathbf{\Omega} + \mathbf{M}^{\top} \mathbf{M}$
- 11: Update the missing value entries with the latest EM fit:

$$\mathbf{X}_{t,j} \leftarrow \sum_{c=1}^{k} \mathbf{M}_{ic} \mathbf{P}_{jc}$$
 if $\tau_{t,j}^{(X)} = 1$

$$\mathbf{Y}_{t,j} \leftarrow \sum_{c=1}^{k} \mathbf{M}_{ic} \mathbf{Q}_{jc}$$
 if $\tau_{t,j}^{(Y)} = 1$

Update **P** and **Q** jointly as: 12:

$$\begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{Q}_1 \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{X}^\top \\ \boldsymbol{Y}^\top \end{bmatrix} \boldsymbol{M} \boldsymbol{V}^{-1}$$

Update σ_x^2 : 13:

$$\sigma_{x,1}^2 \leftarrow \frac{1}{Tp} \left[\|\mathbf{X}\|_F^2 - \text{Tr}(\mathbf{P}_1^\top \mathbf{P}_1 \mathbf{V}) \right]$$

Update σ_y^2 : 14:

$$\sigma_{y,1}^2 \leftarrow \frac{1}{Tq} \left[\|\mathbf{Y}\|_F^2 - \text{Tr}(\mathbf{Q}_1^\top \mathbf{Q}_1 \mathbf{V}) \right]$$

Collect updated parameters as $\theta_1 \leftarrow (\mathbf{P}_1, \mathbf{Q}_1, \sigma_{x,1}^2, \sigma_{y,1}^2)$

16: **until** convergence $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\| < \epsilon$ or S iterations are reached 17: **return** Loading matrices $\mathbf{P} \in \mathbb{R}^{p \times k}$ and $\mathbf{Q} \in \mathbb{R}^{q \times k}$, as well as noise variances σ_x^2 and σ_y^2 from final estimate θ_1

Algorithm 3 EM Algorithm for Probabilistic PLS with Mixed-Frequency Data

Require: High-frequency predictor matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{(TL)\times p}$, low-frequency target matrix $\mathbf{Y} \in \mathbb{R}^{T\times q}$, low-to-high-frequency period L, number of components k, starting values $(\mathbf{P}_0, \mathbf{Q}_0, \sigma_{x,0}^2, \sigma_{y,0}^2)$, prior variance \mathbf{V}_F , tolerance ϵ , maximum iterations S

1: Reshape $\widetilde{\mathbf{X}}$ into a $T \times (pL)$ matrix \mathbf{X} :

$$\widetilde{\mathbf{X}} = egin{bmatrix} oldsymbol{x}_1^{ op} \ dots \ oldsymbol{x}_L^{ op} \ dots \ oldsymbol{x}_L^{ op} \ dots \ oldsymbol{x}_{(T-1)L+1}^{ op} \ dots \ oldsymbol{x}_{(T-1)L+1}^{ op} \ dots \ oldsymbol{x}_{(T-1)L+1}^{ op} \ dots \ oldsymbol{x}_{TL}^{ op} \end{bmatrix} = \mathbf{X}$$

2: Center and scale the Data:

$$\mathbf{X} \leftarrow (\mathbf{X} - \mathbf{1}_T \mathbf{m}_x^\top) \operatorname{diag}(\mathbf{s}_x)^{-1}, \quad \mathbf{Y} \leftarrow (\mathbf{Y} - \mathbf{1}_T \mathbf{m}_y^\top) \operatorname{diag}(\mathbf{s}_y)^{-1}$$

where for the columns of **X** and **Y**, $m_x := (1/T) \sum_{t=1}^T x_t$ and $m_y := (1/T) \sum_{t=1}^T y_t$ are the vector of means, whereas s_x and s_y are the vectors of standard deviations, respectively.

3: repeat

4: E-step: Expectation

5: Collect all initial parameters into $\theta_0 \leftarrow (\mathbf{P}_0, \mathbf{Q}_0, \sigma_{x,0}^2, \sigma_{y,0}^2)$

6: Compute Posterior Covariance (Ω):

$$\mathbf{\Omega} \leftarrow \left[\mathbf{I}_L \otimes \left(\frac{1}{\sigma_{x,0}^2} \mathbf{P}_0^\top \mathbf{P}_0 + \mathbf{V}_F^{-1} \right) + \frac{1}{L \cdot \sigma_{y,0}^2} \mathbf{1}_{L \times L} \otimes \left(\mathbf{Q}_0^\top \mathbf{Q}_0 \right) \right]^{-1}$$

7: Compute Posterior Mean (**M**):

$$\boldsymbol{M} \leftarrow \begin{bmatrix} \frac{1}{\sigma_{x,0}^2} \boldsymbol{X}^{(1)} \boldsymbol{P}_0 + \frac{1}{\sigma_{y,0}^2} \boldsymbol{Y} \boldsymbol{Q}_0 & \cdots & \frac{1}{\sigma_{x,0}^2} \boldsymbol{X}^{(L)} \boldsymbol{P}_0 + \frac{1}{\sigma_{y,0}^2} \boldsymbol{Y} \boldsymbol{Q}_0 \end{bmatrix} \boldsymbol{\Omega}$$

8: **M-step: Maximization**

9:
$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_{1,1} & \cdots & \mathbf{V}_{1,L} \\ \vdots & \ddots & \vdots \\ \mathbf{V}_{L,1} & \cdots & \mathbf{V}_{L,L} \end{bmatrix} \leftarrow T \cdot \mathbf{\Omega} + \mathbf{M}^{\top} \mathbf{M}$$

10: Update **P** as:

$$\mathbf{P}_1 \leftarrow \left(\sum_{\ell=1}^L \mathbf{X}^{(\ell)\top} \mathbf{M}^{(\ell)}\right) \left(\sum_{\ell=1}^L \mathbf{V}_{\ell,\ell}\right)^{-1}$$

11: Update **Q** as:

$$\mathbf{Q}_1 \leftarrow L \cdot \left(\mathbf{Y}^\top \sum_{\ell=1}^L \mathbf{M}^{(\ell)} \right) \left(\sum_{r=1}^L \sum_{\ell=1}^L \mathbf{V}_{\ell,r} \right)^{-1}$$

12: Update σ_x^2 :

$$\sigma_{x,1}^2 \leftarrow \frac{1}{TLp} \left\{ \|\mathbf{X}\|_F^2 - \operatorname{Tr} \left[\mathbf{P}_1^\top \mathbf{P}_1 \left(\sum_{\ell=1}^L \mathbf{V}_{\ell,\ell} \right) \right] \right\}$$

13: Update σ_y^2 :

$$\sigma_{y,1}^2 \leftarrow \frac{L}{Tq} \operatorname{Tr} \left\{ \mathbf{Y}^\top \left[\mathbf{Y} - \frac{1}{L} \left(\sum_{\ell=1}^L \mathbf{M}^{(\ell)} \right) \mathbf{Q}_1^\top \right] \right\}$$

14: Collect updated parameters as $\theta_1 \leftarrow (\mathbf{P}_1, \mathbf{Q}_1, \sigma_{x,1}^2, \sigma_{y,1}^2)$

15: **until** convergence $\| \boldsymbol{\theta}_1 - \boldsymbol{\theta}_0 \| < \epsilon$ or S iterations are reached

16: **return** Loading matrices $\mathbf{P} \in \mathbb{R}^{p \times k}$ and $\mathbf{Q} \in \mathbb{R}^{q \times k}$, as well as noise variances σ_x^2 and σ_y^2 from final estimate θ_1

Algorithm 4 EM Algorithm for Probabilistic PLS with Exponential Weighted Moving Average (EWMA) Stochastic Volatility

Require: Predictor matrix $\mathbf{X} \in \mathbb{R}^{T \times p}$, target matrix $\mathbf{Y} \in \mathbb{R}^{T \times q}$, number of components k, starting values $(\mathbf{P}_0, \mathbf{Q}_0, \bar{\sigma}_x^2, \bar{\sigma}_y^2)$, prior variance \mathbf{V}_F , EWMA smoothing parameters (λ_x, λ_y) , tolerance ϵ , maximum iterations S

1: Center and scale the Data:

$$\mathbf{X} \leftarrow (\mathbf{X} - \mathbf{1}_T \mathbf{m}_{\mathbf{x}}^{\top}) \operatorname{diag}(\mathbf{s}_{\mathbf{x}})^{-1}, \quad \mathbf{Y} \leftarrow (\mathbf{Y} - \mathbf{1}_T \mathbf{m}_{\mathbf{y}}^{\top}) \operatorname{diag}(\mathbf{s}_{\mathbf{y}})^{-1}$$

where for the columns of **X** and **Y**, $m_x := (1/T) \sum_{t=1}^T x_t$ and $m_y := (1/T) \sum_{t=1}^T y_t$ are the vector of means, whereas s_x and s_y are the vectors of standard deviations, respectively.

2: Start the *T*-dimensional stochastic volatility vectors $\sigma_{x,0}^2$ and $\sigma_{y,0}^2$ as constant:

$$\sigma_{x,0}^2(t) \leftarrow \bar{\sigma}_x^2$$
 and $\sigma_{y,0}^2(t) \leftarrow \bar{\sigma}_y^2$ for all $t = 1, \dots, T$

3: repeat

E-step: Expectation

Collect initial parameters into $\theta_0 \leftarrow (\mathbf{P}_0, \mathbf{Q}_0, \sigma_{x,0}^2, \sigma_{y,0}^2)$ 5:

6: for $t = 1, \ldots, T$ do

Compute and store posterior covariance per period (Ω_t): 7:

$$\mathbf{\Omega}_t \leftarrow \left(\mathbf{V}_F^{-1} + \frac{1}{\sigma_{x,0}^2(t)}\mathbf{P}_0^{\top}\mathbf{P}_0 + \frac{1}{\sigma_{y,0}^2(t)}\mathbf{Q}_0^{\top}\mathbf{Q}_0\right)^{-1}$$

Compute posterior mean per period (m_t): 8:

$$m{m}_t \leftarrow m{\Omega}_t \left(rac{1}{\sigma_{x,0}^2(t)} \mathbf{P}_0^ op m{x}_t + rac{1}{\sigma_{y,0}^2(t)} \mathbf{Q}_0^ op m{y}_t
ight)$$

end for 9:

10: Stack posterior means into $T \times k$ matrix:

$$\mathbf{M} \leftarrow \begin{bmatrix} \boldsymbol{m}_1^\top \\ \vdots \\ \boldsymbol{m}_T^\top \end{bmatrix}$$

11:

M-step: Maximization $\mathbf{V} \leftarrow \sum_{t=1}^{T} \mathbf{\Omega}_t + \mathbf{M}^{\top} \mathbf{M}$

13: Update **P** and **Q** jointly as:

$$\begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{Q}_1 \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{X}^\top \\ \boldsymbol{Y}^\top \end{bmatrix} \boldsymbol{M} \boldsymbol{V}^{-1}$$

Compute Residuals: 14:

$$\widehat{\mathbf{E}}_{x} = \begin{bmatrix} \widehat{e}_{x,1}^{\top} \\ \vdots \\ \widehat{e}_{x,T}^{\top} \end{bmatrix} \leftarrow \mathbf{X} - \mathbf{M} \mathbf{P}_{1}^{\top} \quad \text{and} \quad \widehat{\mathbf{E}}_{y} = \begin{bmatrix} \widehat{e}_{y,1}^{\top} \\ \vdots \\ \widehat{e}_{y,T}^{\top} \end{bmatrix} \leftarrow \mathbf{Y} - \mathbf{M} \mathbf{Q}_{1}^{\top}$$

15: Update first-period stochastic volatility estimates:

$$\sigma_{x,1}^2(1) \leftarrow \frac{1}{p} \left[\|\widehat{\boldsymbol{e}}_{x,1}\|_2^2 + \operatorname{Tr}(\mathbf{P}_1^\top \mathbf{P}_1 \mathbf{\Omega}_1) \right]$$

$$\sigma_{y,1}^2(1) \leftarrow \frac{1}{q} \left[\|\widehat{\boldsymbol{e}}_{y,1}\|_2^2 + \operatorname{Tr}(\mathbf{Q}_1^\top \mathbf{Q}_1 \mathbf{\Omega}_1) \right]$$

Update remaining stochastic volatility estimates in $\sigma_{x,1}^2$ and $\sigma_{y,1}^2$ using EWMA: 16:

for t = 2, ..., T **do** 17:

$$\sigma_{x,1}^2(t) \leftarrow \lambda_x \cdot \sigma_{x,1}^2(t-1) + (1-\lambda_x) \cdot \frac{1}{p} \left[\|\widehat{\boldsymbol{e}}_{x,t}\|_2^2 + \operatorname{Tr}(\mathbf{P}_1^\top \mathbf{P}_1 \mathbf{\Omega}_t) \right]$$

$$\sigma_{y,1}^2(t) \leftarrow \lambda_y \cdot \sigma_{y,1}^2(t-1) + (1-\lambda_y) \cdot \frac{1}{q} \left[\|\widehat{\boldsymbol{e}}_{y,t}\|_2^2 + \operatorname{Tr}(\mathbf{Q}_1^\top \mathbf{Q}_1 \mathbf{\Omega}_t) \right]$$

18: end for

Collect updated parameters into $\theta_1 \leftarrow (\mathbf{P}_1, \mathbf{Q}_1, \sigma_{x,1}^2, \sigma_{y,1}^2)$ 19:

20: **until** convergence $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0\| < \epsilon$ or S iterations are reached 21: **return** Loading matrices $\mathbf{P} \in \mathbb{R}^{p \times k}$ and $\mathbf{Q} \in \mathbb{R}^{q \times k}$, as well as time-varying noise variances σ_x^2 and σ_y^2 from final estimates $\boldsymbol{\theta}_1$

D Additional Results

D.1 Missing data scenario

An additional advantage of the probabilistic formulation of PLS is that it also allows us to directly deal with missing observations. Missing observations are the staple in most real-world scenarios as these can arise from distinct data release schedules, different collection techniques, data corruption, etc. To compare our PTFA method, we introduce an additional modification allowing us to deal with missing data.

This is a standard approach when using EM-type algorithms, as the expectation step with missing data can be easily computed by a simple imputation step based on the current EM estimate. The full procedure is presented in Algorithm 2. In our provided package, we allow all of our methods and extensions to deal with missing data using this same idea.

To test the performance of our proposed method under missing data, we generate observations for **X** and **Y** as in the previous examples, additionally introducing a given percentage of missing-at-random observations for both. That is, we choose corruption levels ρ_x % and ρ_y % to set that fraction of elements randomly as missing. We compare two potential solutions to the missing data issue:

- 1. Create an imputed version of the data $(\widetilde{\mathbf{X}})$ and $\widetilde{\mathbf{Y}}$ that sets the missing values in \mathbf{X} and \mathbf{Y} to a fixed value, such as 0 or the sample average. We can then apply both PLS or our PTFA $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$.
- 2. On the other hand, we directly implement our PTFA with an imputation step on the inner loop of the EM algorithm based on the current predicted value for **X** and **Y** (as proposed in Eqs. 12 and 13).

Figure D.1 presents median R^2 statistics across 100 replications for this exercise, when we vary the level of missing-at-random observations in both features (ρ_x) and targets (ρ_y). The R^2 in this scenario is computed with respect to the true, infeasible values for the targets **Y** that have no missing observations. The first and second panels represent a direct comparison of applying either PLS or PTFA on the imputed data ($\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$), while the third represents the results from including the missing values into our EM algorithm.

Note that PLS never achieves a large value for the R^2 with imputation, which is not the case for our PTFA method. However, by implementing the imputation step in the inner loop, our method is able to deal with large amounts of missing observations in the features, only breaking down with extreme amounts of missing observations close to 50%.

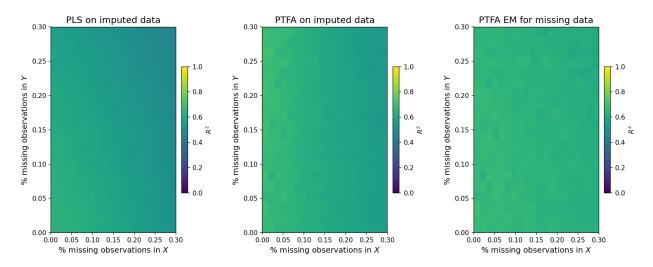


Figure D.1: Comparison of PLS vs PTFA based on % of missing-at-random observations in features (*X*) and targets (*Y*)