



Fast Variational Inference for Bayesian Factor Analysis in Single and Multi-Study Settings

Blake Hansen, Alejandra Avalos-Pacheco, Massimiliano Russo & Roberta De Vito

To cite this article: Blake Hansen, Alejandra Avalos-Pacheco, Massimiliano Russo & Roberta De Vito (15 May 2024): Fast Variational Inference for Bayesian Factor Analysis in Single and Multi-Study Settings, Journal of Computational and Graphical Statistics, DOI: [10.1080/10618600.2024.2356173](https://doi.org/10.1080/10618600.2024.2356173)

To link to this article: <https://doi.org/10.1080/10618600.2024.2356173>



View supplementary material [↗](#)



Accepted author version posted online: 15 May 2024.



Submit your article to this journal [↗](#)



Article views: 38



View related articles [↗](#)



View Crossmark data [↗](#)

Fast Variational Inference for Bayesian Factor Analysis in Single and Multi-Study Settings

Blake Hansen

Department of Biostatistics, Brown University

Alejandra Avalos-Pacheco,

Applied Statistics Research Unit, TU Wien

Harvard-MIT Center for Regulatory Science, Harvard Medical School

Massimiliano Russo*,

Department of Statistics, The Ohio State University

Roberta De Vito*

Department of Biostatistics and Data Science Institute, Brown University

*These authors contributed equally to this work.

Abstract—Factors models are commonly used to analyze high-dimensional data in both single-study and multi-study settings. Bayesian inference for such models relies on Markov Chain Monte Carlo (MCMC) methods, which scale poorly as the number of studies, observations, or measured variables increase. To address this issue, we propose new variational inference algorithms to approximate the posterior distribution of Bayesian latent factor models using the multiplicative gamma process shrinkage prior. The proposed algorithms provide fast approximate inference at a fraction of the time and memory of MCMC-based implementations while maintaining comparable accuracy in characterizing the data covariance matrix. We conduct extensive simulations to evaluate our proposed algorithms and show their utility in estimating the model for high-dimensional multi-study gene expression data in ovarian cancers. Overall, our proposed approaches enable more efficient and scalable inference for factor models, facilitating their use in high-dimensional settings. An R package VIMSFA implementing our methods is available on GitHub (github.com/blhansen/VI-MSFA).

Keywords: Factor Analysis, Shrinkage prior, Variational Bayes, Multi-Study

1 Introduction

Factor Analysis (FA) models are popular tools for providing low-dimensional data representations through latent factors. These factors are critical to visualize, denoise and explain patterns of interest of the data, making FA models useful in several fields of application such as biology (Pournara and Wernisch, 2007), finance (Ludvigson and Ng, 2007), public policy (Samartsidis et al., 2020), and nutrition (Edefonti et al., 2012; Joo et al., 2018)—among many others.

In high-dimensional settings, FA models shrinking many latent factors' components to zero are particularly helpful. These models allow focusing only on a small set of important variables, making the interpretation of the factors easier (Carvalho et al., 2008). Moreover, they can lead to more accurate parameter estimations compared to models that do not include shrinkage or sparsity (Avalos-Pacheco et al., 2022). In a Bayesian setting, many priors have been proposed to induce shrinkage or sparsity in the latent factors (e.g., Archambeau and Bach (2008); Carvalho et al. (2008); Legramanti et al. (2020); Cadonna et al. (2020); Frühwirth-Schnatter (2023)). A widely used one is the gamma process shrinkage prior (Bhattacharya and Dunson, 2011; Durante, 2017) that induces a shrinkage effect increasing with the number of factors. This prior adopts an adaptive approach for automatically choosing the latent dimension (i.e., the number of latent factors), facilitating posterior computation.

In a Bayesian context, inference on the posterior distribution of FA typically relies on Markov-Chain Monte Carlo (MCMC) algorithms (Lopes and West, 2004), which scale poorly to high-dimensional settings (Rajaratnam and Sparks, 2015). In addition, computing posterior expectations from MCMC can be computationally intensive. Specifically, many parameters are invariant to orthogonal transformations and must be post-processed before being averaged for inference, exacerbating time and memory needs. We refer to Papastamoulis and Ntzoufras (2022) for examples of post-processing algorithms. In high-dimensional settings, MCMC for FA can also show poor mixing (Frühwirth-Schnatter

et al., 2023), further increasing the difficulty of obtaining high-quality posterior inference without extensive computational resources.

While some alternatives to MCMC that seek the posterior mode via Expectation-Maximization (EM) algorithms have been developed for Bayesian FA (e.g., Ročková and George, 2016), fast approximation of the posterior distribution that has the potential to scale inference in high-dimensional settings are still under-explored. Variational Bayes (VB) is one popular class of approximate Bayesian inference methods which has been successfully applied to approximate the posterior distribution of several models, such as logistic regression (Jaakkola and Jordan, 1997; Durante and Rigon, 2019), probit regression (Fasano et al., 2022), latent Dirichlet allocation (Blei et al., 2017), and network factor models (Aliverti and Russo, 2022), among others. VB has also been shown to scale Bayesian inference to high dimensional datasets for models closely related to FA. For example, Ghahramani and Beal (1999) develop a VB approximation for Bayesian mixtures of factor analysers, Wang et al. (2020) consider item response theory models with latent factors, while Dang and Maestrini (2022) propose VB algorithms for structural equation models. However, VB for FA models with shrinkage priors is still under-explored, and to our knowledge, no VB implementations for FA with gamma process shrinkage priors exist. To address this issue and enable the practical use of FA in high-dimensional settings, we present two VB algorithms to approximate the posterior of this model and extend them to settings where multiple data from different sources are available.

There are many applications where data are collected from multiple sources or studies. These sources of information are then combined to produce more precise estimates and results that are robust to study-specific biases. For example, in biology, different microarray cancer studies can collect the same gene expression, measured in different platforms and/or with different late-stage patients. One standard approach to analyzing these multiple data is to stack them into a single dataset and perform FA (Wang et al., 2011). This approach can lead to misleading conclusions by ignoring potential study-to-study variability arising from both biological and technical sources (Garrett-Mayer et al., 2008). Therefore, statistical methods that are able to estimate concurrently

common and study-specific signals should be adopted. Recent methods have been proposed to enable FA to integrate multiple sources of information in a single statistical model. These include perturbed factor analysis (Roy et al., 2021), Multi-Study Factor Analysis (MSFA, De Vito et al., 2019, 2021), Multi-Study Factor Regression (De Vito and Avalos-Pacheco, 2023), and Bayesian combinatorial MSFA (Grabski et al., 2023).

In this paper, first, we present VB algorithms to conduct fast Bayesian estimation for FA with a continuous response, and second, we extend these algorithms to a multi-study setting. We adopt the MSFA model, which decomposes the covariance matrix of the data in terms of common and study-specific latent factors and extends the gamma process shrinkage prior in this context (De Vito et al., 2019, 2021). Compared to current MCMC implementations, we show that VB approximations for FA and MSFA greatly reduce computation time while still providing accurate posterior inference.

We emphasize that our proposed VB algorithms are tailored for efficient point estimation of the latent factor loadings. This is typically the main goal in FA, which is mainly used as an exploratory tool. In fact, the non-identifiability and the unknown dimensions of the factor loadings make uncertainty quantification of the latent factors themselves challenging.

The structure of this paper is as follows: §2 defines single-study FA and describes VB estimation of the model parameters through Coordinate Ascent Variational Inference (CAVI) and Stochastic Variational Inference (SVI) algorithms; §3 generalizes these algorithms in a multi-study setting; §4 provides simulation studies benchmarking the performance of the proposed VB algorithms in comparison to previous MCMC based algorithms; §5 applies the proposed algorithms to a dataset containing gene expression values from 1, 198 patients with ovarian cancer across 4 studies (Ganzfried et al., 2013); finally, §6 provides a brief discussion.

2 Factor Analysis

Let $\mathbf{x}_i = (x_{i1}, \dots, x_{iP}) \in \mathbb{R}^P$ be a vector of P centered observed variables for individual $i = 1, \dots, N$. FA assumes that \mathbf{x}_i can be modelled as a function of $J \ll P$ latent factors or scores $\mathbf{l}_i \in \mathbb{R}^J$, and a corresponding factor loading matrix $\mathbf{\Lambda} \in \mathbb{R}^{P \times J}$ via

$$\mathbf{x}_i = \mathbf{\Lambda} \mathbf{l}_i + \mathbf{e}_i, i = 1, \dots, N. \quad (1)$$

Both the factors $\{\mathbf{l}_i\}$ and the *idiosyncratic errors* $\{\mathbf{e}_i\}$ are assumed to be normally distributed, with $\mathbf{l}_i \sim \mathcal{N}_J(\mathbf{0}, \mathbf{I}_J)$, and $\mathbf{e}_i \sim \mathcal{N}_P(\mathbf{0}, \mathbf{\Psi})$ where $\mathbf{\Psi} = \text{diag}(\psi_1^2, \dots, \psi_P^2)$. Factors $\{\mathbf{l}_i\}$ are assumed to be independent from the errors $\{\mathbf{e}_i\}$, and the latent dimension $J \ll P$ is typically unknown. As a consequence, the observed covariance matrix, $\mathbf{\Sigma} = \text{Cov}(\mathbf{x}_i)$, can be expressed as $\mathbf{\Sigma} = \mathbf{\Lambda} \mathbf{\Lambda}^\top + \mathbf{\Psi}$. Given that $\mathbf{\Psi}$ is a diagonal matrix, the non-diagonal elements of $\mathbf{\Lambda} \mathbf{\Lambda}^\top$ represent

the pairwise covariances of the P variables, i.e.,
$$\text{Cov}(x_{ip}, x_{iq}) = \sum_{j=1}^J \lambda_{pj} \lambda_{qj} \quad \text{for } p \neq q.$$

In the following, we will use $\mathbf{X} = \{\mathbf{x}_i\}$ to indicate the $N \times P$ data matrix used for the analysis.

2.1 Prior Specification

Following a Bayesian approach to inference for Model (1), we use the gamma process shrinkage prior of (Bhattacharya and Dunson, 2011) for each element $\{\lambda_{pj}\}$ of the loading matrix $\mathbf{\Lambda}$. Indicating with $\Gamma(a, b)$ a gamma distribution with mean a/b and variance a/b^2 , this prior can be expressed via:

$$\begin{aligned} \lambda_{pj} | \omega_{pj}, \tau_j &\sim \mathcal{N}(0, \omega_{pj}^{-1} \tau_j^{-1}), \quad p = 1, \dots, P, \quad j = 1, \dots, \infty, \\ \omega_{pj} &\sim \Gamma\left(\frac{\nu}{2}, \frac{\nu}{2}\right), \quad \tau_j = \prod_{l=1}^j \delta_l, \quad \delta_1 \sim \Gamma(a_1, 1), \quad \delta_l \sim \Gamma(a_2, 1), l \geq 2, \end{aligned} \quad (2)$$

where δ_l ($l = 1, 2, \dots$) are independent, τ_j is the global shrinkage parameter for column j , and ω_{pj} is the local shrinkage parameter for element p in column j .

In practice, prior (2) is truncated to a conservative upper bound J^* , smaller than P . We refer to Bhattacharya and Dunson (2011) for more details on the choice of J^* .

Lastly, an inverse-gamma prior is placed on the diagonal entries of Ψ , which is a common choice in FA (Lopes and West, 2004; Ročková and George, 2016):

$$\psi_p^{-2} \sim \Gamma(a^\psi, b^\psi). \quad (3)$$

For simplicity, we denote all the model parameters with $\theta = \left(\left\{ \Lambda_p \right\}, \left\{ \mathbf{1}_i \right\}, \left\{ \psi_p^2 \right\}, \left\{ \omega_{pi} \right\}, \left\{ \delta_l \right\} \right)$ where $\Lambda_p = (\lambda_{1p}, \dots, \lambda_{J^*p})$ is row p of Λ , and with $\pi(\theta)$ the prior (2)–(3).

2.2 Variational Inference for Factor Analysis

The goal of VB algorithms is to approximate the posterior distribution $p(\theta | \mathbf{X})$, with a density $q^*(\theta)$ in a class \mathcal{Q} (see Blei et al. (2017) for further details). Specifically, the VB approximation of the posterior is defined as the distribution $q^*(\theta) \in \mathcal{Q}$ closest to $p(\theta | \mathbf{X})$ in Kullback-Leibler (KL) divergence, satisfying the minimization:

$$\arg \min_{q(\theta) \in \mathcal{Q}} \left\{ \text{KL} \left(q(\theta) \parallel p(\theta | \mathbf{X}) \right) \right\} = \arg \min_{q(\theta) \in \mathcal{Q}} \left\{ \int_{\Theta} q(\theta) \log \frac{q(\theta)}{p(\theta | \mathbf{X})} d\theta \right\},$$

or equivalently the maximization of the *evidence based lower bound* (ELBO):

$$\arg \max_{q(\theta) \in \mathcal{Q}} \left\{ \text{ELBO}(q(\theta)) \right\} = \arg \max_{q(\theta) \in \mathcal{Q}} \left\{ \mathbb{E}_{q(\theta)} [\log p(\theta, \mathbf{X})] - \mathbb{E}_{q(\theta)} [\log q(\theta)] \right\}, \quad (4)$$

where $p(\theta, \mathbf{X}) = p(\mathbf{X} | \theta) \pi(\theta)$ is the joint distribution of the data and θ , and the expectations are with respect to $q(\theta)$. A common approach to select \mathcal{Q} is to use a mean-field variational

family, $\mathcal{Q}^{\text{MF}} = \{q : q(\theta) = \prod_{m=1}^M q(\theta_m)\}$, that considers a partition of the parameters θ into M blocks, $(\theta_1, \dots, \theta_M)$ and approximate the posterior with a product of M independent density

functions $q(\theta_m)$, referred to as the variational factor for θ_m . When using \mathcal{Q}^{MF} , the variational factors $\{q^*(\theta_m)\}$ composing the VB approximation of the posterior can be obtained from

$$q_m^*(\theta_m) \propto \exp\left(\mathbb{E}_{q(\theta_{-m})}[\log(p(\theta_m | \theta_{-m}, \mathbf{X}))]\right), \quad (5)$$

where $\theta_{-m} = (\theta_1, \dots, \theta_{m-1}, \theta_{m+1}, \dots, \theta_M)$ is the vector of parameters excluding the m -th one (Chapter 10, Bishop (2006)).

When the conditional distribution $p(\theta_m | \theta_{-m}, \mathbf{X})$ belongs to the exponential family, the corresponding optimal variational factor $q^*(\theta_m)$ also belongs to the same exponential family. Therefore, the maximization problem (4) reduces to learning the value of the parameters characterizing the distribution $q_m^*(\theta_m)$. We will indicate these parameters as ϕ_m^* and the corresponding optimal variational factor as $q^*(\theta_m; \phi_m^*)$. Choosing \mathcal{Q}^{MF} in such a way that $\{\pi(\theta_m)\}$ are conditionally conjugate prior for $\{\theta_m\}$ enables the development of efficient algorithms such as the *Coordinate-Ascent Variational Inference* (CAVI) which learns ϕ^* by iteratively updating each parameter of $q^*(\theta_m; \phi_m^*)$ conditional to the others variational factors until convergence is reached (Chapter 10, Bishop (2006)).

To implement CAVI for FA with priors (2)–(3), we propose the following mean-field variational family:

$$q(\theta; \phi) = \left[\prod_{l=1}^{J^*} q(\delta_l; \alpha_l^\delta, \beta_l^\delta) \right] \left[\prod_{p=1}^P \prod_{j=1}^{J^*} q(\omega_{pj}; \alpha_{pj}^\omega, \beta_{pj}^\omega) \right] \left[\prod_{i=1}^N q(\mathbf{l}_i; \mu_i^l, \Sigma_i^l) \right] \times \left[\prod_{p=1}^P q(\Delta_p; \mu_p^\Delta, \Sigma_p^\Delta) \right] \left[\prod_{p=1}^P q(\psi_p^{-2}; \alpha_p^\psi, \beta_p^\psi) \right], \quad (6)$$

with variational parameters $\phi = (\{\alpha_l^\delta\}, \{\beta_l^\delta\}, \{\alpha_{pj}^\omega\}, \{\beta_{pj}^\omega\}, \{\mu_i^l\}, \{\Sigma_i^l\}, \{\mu_p^\Delta\}, \{\Sigma_p^\Delta\}, \{\alpha_p^\psi\}, \{\beta_p^\psi\})$.

The mean-field approximation in equation (6) factorizes the posterior distribution into conditionally conjugate elements, so maximizing the ELBO with respect to $q(\theta_m)$ leads to closed-form expressions for each variational parameter (Supplementary §F for

details). All the steps to obtain the VB posterior approximation are detailed in Algorithm S1 (Supplementary §A).

An important aspect of CAVI is that each iteration requires optimizing the *local parameters* Σ_i^l and μ_i^l , relative to the scores \mathbf{l}_i for $i = 1, \dots, N$, which is computationally expensive in settings with large sample size. Thus, we propose a *Stochastic Variational Inference* (SVI) algorithm, which uses stochastic optimization to reduce the computational cost of each iteration (Hoffman et al., 2013). The key idea of this algorithm is to compute at each iteration the local parameters only for a small subset of available data and use this subset to approximate the remaining parameters common to all individuals, namely *global parameters*. In the following we use $\mathcal{M}^G \subset \{1, \dots, M\}$ for the indices of the global parameters.

As a first step to illustrate the SVI algorithm, we reframe CAVI as a gradient-based optimization algorithm for the global parameters. Recall that the mean field approximation is constructed so that the optimal variational factors are distributions in the exponential family, i.e., a function of the natural parameters $\eta(\mathbf{X}, \theta)$ (see Table S1 of Supplementary §A for details). With this parameterization, the derivative of the ELBO in equation (4) with respect to the natural parameters is (Hoffman et al., 2013):

$$\nabla ELBO(q(\theta_m)) = \mathbb{E}_{q(\theta_m)} [\eta(\mathbf{X}, \theta)] - \phi_m \quad \text{for } m \in \mathcal{M}^G. \quad (7)$$

Equating the gradient (7) to 0, the solution of (4) can be written as $\phi_m = \mathbb{E}_{q(\theta_m)} [\eta(\mathbf{X}, \theta)]$.

The SVI replaces the gradient with an estimate much easier to compute. This is accomplished by selecting a batch-size parameter $b \in (0, 1)$ and drawing a random sample of the data of dimension $\tilde{N} = \lfloor b \times N \rfloor$, at each iteration of the algorithm. Here $\lfloor x \rfloor$, indicates the greatest integer less than or equal to x . At iteration t , let

$\mathcal{I}(t) \subseteq \{1, \dots, N\}$ be the subset of sampled observations, and $\mathbf{X}^t = \{\mathbf{x}_i \text{ for } i \in \mathcal{I}(t)\}$ the corresponding $\tilde{N} \times P$ data matrix. SVI proceed by first updating the local parameters

$\{\Sigma_i^l\}$ and $\{\mu_i^l\}$ for all $i \in \mathcal{I}(t)$, and then computing an approximated gradient for the global parameters via

$$\nabla ELBO(q(\theta_m)) = \mathbb{E}_{q(\theta_{-m})} [\eta(\mathbf{X}^t, \theta)] - \phi_m, \quad \text{for } m \in \mathcal{M}^G, \quad (8)$$

where the expectation $\mathbb{E}_{q(\theta_{-m})} [\eta(\mathbf{X}^t, \theta)]$ is computed giving weight N / \tilde{N} to each individual in $\mathcal{I}(t)$ (Hoffman et al., 2013). We call ϕ_m the solution of (8). At each iteration t of the SVI algorithm, the update for the $\{\phi_m(t)\}$ of the global parameters is obtained as a weighted average of ϕ_m and $\phi_m(t-1)$ at the previous iteration:

$$\phi_m(t) = (1 - \rho_t) \phi_m(t-1) + \rho_t \phi_m, \quad \text{for } m \in \mathcal{M}^G, \quad (9)$$

where ρ_t is a *step size* parameter such that $\sum_t \rho_t \rightarrow \infty$ and $\sum_t \rho_t^2 < \infty$ (Robbins and Monro, 1951). The choice of ρ_t can impact convergence. Step sizes that are too large may overweight estimated gradients and lead to unreliable estimates of variational parameter updates for some iterations, while step sizes that are too small may take many iterations to converge. A typical choice is to define $\rho_t = (t + \tau)^{-\kappa}$, where $\kappa \in (0.5, 1]$ is the forgetting rate, which controls how quickly the variational parameters change across several iterations, and $\tau > 0$ is the delay, which down-weights early iterations (Hoffman et al., 2013).

To implement SVI for FA, we reparameterize the global parameters in (6) in terms of the natural parameterization of the exponential family (Supplementary §A Table S1). The details are provided in Algorithm S2 (Supplementary §A).

3 Multi-Study Factor Analysis

We extend the two algorithms presented in §2 to the setting where there are $S > 1$ datasets measuring the same P variables, using the MSFA Model (De Vito et al., 2021).

We indicate with $\mathbf{x}_{si} = (x_{si1}, \dots, x_{siP}) \in \mathbb{R}^P$ the vector of P centered observed variables for individual $i = 1, \dots, N_s$ in study $s = 1, \dots, S$ and use the model

$$\mathbf{x}_{si} = \mathbf{\Phi} \mathbf{f}_{si} + \mathbf{\Lambda}_s \mathbf{l}_{si} + \mathbf{e}_{si} \quad i = 1, \dots, N_s, \quad s = 1, \dots, S, \quad (10)$$

where $\mathbf{f}_{si} \sim \mathcal{N}_K(\mathbf{0}, \mathbf{I}_K)$ are the K -dimensional scores which correspond to the shared loading matrix $\mathbf{\Phi} \in \mathbb{R}^{P \times K}$, and $\mathbf{l}_{si} \sim \mathcal{N}_{J_s}(\mathbf{0}, \mathbf{I}_{J_s})$ are the J_s -dimensional scores with study-specific loading matrices $\mathbf{\Lambda}_s \in \mathbb{R}^{P \times J_s}$. Finally, $\mathbf{e}_{is} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi}_s)$ are idiosyncratic errors with $\mathbf{\Psi}_s = \text{diag}(\psi_{s1}^2, \dots, \psi_{sp}^2)$.

Model (10) can be seen as a generalization of Model (1) that incorporates shared latent components, \mathbf{f}_{si} and $\mathbf{\Phi}$, and study-specific latent components, \mathbf{l}_{si} and $\mathbf{\Lambda}_s$. The covariance matrices of each study $\mathbf{\Sigma}_s = \text{Cov}(\mathbf{x}_{si})$ for $s = 1, \dots, S$ can be decomposed as $\mathbf{\Sigma}_s = \mathbf{\Phi} \mathbf{\Phi}^\top + \mathbf{\Lambda}_s \mathbf{\Lambda}_s^\top + \mathbf{\Psi}_s$. The terms $\mathbf{\Phi} \mathbf{\Phi}^\top$ and $\mathbf{\Lambda}_s \mathbf{\Lambda}_s^\top$ represent the portion of the covariance matrix that is attributable to the common factors and the study-specific factors, respectively. We refer to the $N_s \times P$ data matrix from study s as \mathbf{X}_s .

3.1 Prior Specification

Following De Vito et al. (2021), we use the gamma process shrinkage prior (Bhattacharya and Dunson, 2011) for both the study-specific and the shared loading matrix. The prior for each study-specific loading element λ_{spj} is

$$\begin{aligned} \lambda_{spj} \mid \omega_{spj}, \tau_{sj} &\sim \mathcal{N}\left(0, \omega_{spj}^{-1} \tau_{sj}^{-1}\right), \quad s = 1, \dots, S \quad p = 1, \dots, P, \quad j = 1, \dots, \infty, \\ \omega_{spj} &\sim \Gamma\left(\frac{\nu_s}{2}, \frac{\nu_s}{2}\right), \quad \tau_{sj} = \prod_{l=1}^j \delta_{sl}, \quad \delta_{s1} \sim \Gamma(a_{s1}, 1), \quad \delta_{sl} \sim \Gamma(a_{sl}, 1), \quad l \geq 2, \end{aligned} \quad (11)$$

where δ_{sl} ($l = 1, 2, \dots$) are independent, τ_{sj} is the global shrinkage parameter for column j , and ω_{spj} is the local shrinkage for the element p in column j . Similarly, the prior for each shared loading element ϕ_{pk} is

$$\begin{aligned} \phi_{pk} | \omega_{pk}, \tau_k &\sim \mathcal{N}(0, \omega_{pk}^{-1} \tau_k^{-1}), \quad p = 1, \dots, P, \quad k = 1, \dots, \infty, \\ \omega_{pk} &\sim \Gamma\left(\frac{\nu}{2}, \frac{\nu}{2}\right), \quad \tau_k = \prod_{l=1}^k \delta_l, \quad \delta_1 \sim \Gamma(a_1, 1), \quad \delta_l \sim \Gamma(a_2, 1), \quad l \geq 2, \end{aligned} \quad (12)$$

where δ_l ($l = 1, 2, \dots$) are independent, τ_k is the global shrinkage parameter for column k and ω_{pk} is the local shrinkage parameter for element p in column k .

These two priors introduce infinitely many factors for both the study-specific and the shared components (i.e., $J_s = \infty$, for $s = 1, \dots, S$, and $K = \infty$). As detailed in §(2), we truncate the latent dimensions using upper bounds K^* and J_s^* , to all the studies. Finally, an inverse-gamma prior is used for the diagonal entries of Ψ_s

$$\psi_{sp}^{-2} \sim \Gamma(a^\psi, b^\psi). \quad (13)$$

We denote $\theta = \left(\left\{ \Phi_p \right\}, \left\{ \Lambda_{sp} \right\}, \left\{ \psi_{ps}^2 \right\}, \left\{ \omega_{pk} \right\}, \left\{ \delta_l \right\}, \left\{ \omega_{spj} \right\}, \left\{ \delta_{sl} \right\} \right)$ as the vector of all model parameters, where Φ_p is row p of Φ and Λ_{sp} is row p of Λ_s .

3.2 Variational Inference for Multi-Study Factor Analysis

The computational cost of Model (10) is larger than S times the computational cost of Model (1) because we need to estimate both the shared and study-specific parameters. Therefore, the benefit of fast VB posterior approximation is even higher than in the single-study setting, especially if several high-dimensional studies are available for analysis. This is increasingly common in many applied settings, such as cancer genomics, where the expression levels of a large set of genes are measured across multiple cancer types.

To extend CAVI in a multi-study setting, we use the following mean field factorization

$$\begin{aligned}
q(\theta; \varphi) = & \left[\prod_{l=1}^{K^*} q(\delta_l; \alpha_l^\delta, \beta_l^\delta) \right] \left[\prod_{s=1}^S \prod_{l=1}^{J_s^*} q(\delta_{sl}; \alpha_{sl}^\delta, \beta_{sl}^\delta) \right] \left[\prod_{p=1}^P \prod_{k=1}^{K^*} q(\omega_{pk}; \alpha_{pk}^\omega, \beta_{pk}^\omega) \right] \\
& \times \left[\prod_{s=1}^S \prod_{p=1}^P \prod_{j=1}^{J_s^*} q(\omega_{spj}; \alpha_{spj}^\omega, \beta_{spj}^\omega) \right] \left[\prod_{s=1}^S \prod_{i=1}^{N_s} q(f_{si}; \mu_{si}^f, \Sigma_{si}^f) \right] \left[\prod_{s=1}^S \prod_{i=1}^{N_s} q(1_{si}; \mu_{si}^l, \Sigma_{si}^l) \right] \quad (14) \\
& \times \left[\prod_{p=1}^P q(\Phi_p; \mu_p^\Phi, \Sigma_p^\Phi) \right] \left[\prod_{s=1}^S \prod_{p=1}^P q(\Lambda_{sp}; \mu_{sp}^\Lambda, \Sigma_{sp}^\Lambda) \right] \left[\prod_{s=1}^S \prod_{p=1}^P q(\psi_{sp}^{-2}; \alpha_{sp}^\psi, \beta_{sp}^\psi) \right]
\end{aligned}$$

where Φ_p is the p^{th} row of Φ , Λ_{sp} is the p^{th} row of Λ_s , and

$$\begin{aligned}
\varphi = & \left(\{\alpha_l^\delta\}, \{\beta_l^\delta\}, \{\alpha_{sl}^\delta\}, \{\beta_{sl}^\delta\}, \{\alpha_{pk}^\omega\}, \{\beta_{pk}^\omega\}, \{\alpha_{spj}^\omega\}, \{\beta_{spj}^\omega\}, \right. \\
& \left. \{\mu_{si}^f\}, \{\Sigma_{si}^f\}, \{\mu_{si}^l\}, \{\Sigma_{si}^l\}, \{\mu_p^\Phi\}, \{\Sigma_p^\Phi\}, \{\mu_{sp}^\Lambda\}, \{\Sigma_{sp}^\Lambda\}, \{\alpha_{sp}^\psi\}, \{\beta_{sp}^\psi\} \right)
\end{aligned}$$

is the vector of variational parameters. Maximizing the ELBO with equation (5) leads to closed-form expressions for optimal variational parameters for each factor, conditional on the others (see Supplementary §F for supporting calculations). Our implementation of CAVI for Bayesian MSFA is detailed in Algorithm S3 (Supplementary §A).

CAVI for MSFA requires computing both shared and study-specific scores for each observation in each study at each iteration. When $\{N_s\}$ are large, this can become very computationally demanding. Thus, we generalize the SVI algorithm described in §2 to the multi-study setting using the natural parameterization in Table S2 (Supplementary §A). The sub-sampling step of SVI is generalized to a multi-study setting by taking a sub-sample of size $\tilde{N}_s = \lfloor b_s \times N_s \rfloor$ from each study. Our implementation of SVI for Bayesian MSFA is presented in Algorithm S4 (Supplementary §A).

4 Simulation Study

To assess the accuracy and computational performance of the CAVI and SVI algorithms for FA (§2) and MSFA (§3), we simulated different scenarios with varying numbers of subjects, variables, and studies. We compared the performance of the proposed algorithms (available at github.com/blhansen/VI-MSFA) against several competitors, including the Gibbs Sampler (GS) and Expectation Conditional Maximization (ECM)

algorithms (available at github.com/rdevito/MSFA), the principal components based POET algorithm (Fan et al., 2013), and the Automatic Differentiation Variational Inference (ADVI) algorithm (Kucukelbir et al., 2017), implemented in STAN (Carpenter et al., 2017). The algorithms were evaluated in terms of: 1) computation time measured in seconds, 2) maximum RAM usage in megabytes (Mb), and, 3) estimation accuracy of $\Sigma = \Lambda\Lambda^\top + \Psi$ and $\Sigma_s = \Phi\Phi^\top + \Lambda_s\Lambda_s^\top + \Psi_s$ in the single and multi-study simulations, respectively. Computation time and RAM usage are monitored using the R package `peakRAM` (Quinn, 2017), while estimation accuracy is evaluated using the RV coefficient between the true, Σ or Σ_s , and the estimated covariance matrices, $\hat{\Sigma}$ or $\hat{\Sigma}_s$. The RV coefficient is a measure of similarity between two matrices varying between 0 and 1 defined as

$$RV(\Sigma, \hat{\Sigma}) = \frac{\text{Tr}(\Sigma\hat{\Sigma}^\top \Sigma\hat{\Sigma}^\top)}{\sqrt{\text{Tr}(\Sigma\hat{\Sigma}^\top)^2 \text{Tr}(\Sigma\hat{\Sigma}^\top)^2}}. \quad (15)$$

An RV coefficient close to 1 (0) indicates strong similarity (dissimilarity) between the two matrices (Robert and Escoufier, 1976). Additional metrics to evaluate estimation accuracy, including the Frobenius norm and L1 norm of the difference between the estimated covariance matrices and the ground truth, are provided in Supplementary §C (Tables S3, S5, S7–S13).

For SVI Algorithms S2 and S4, we set the forgetting rate κ and delay parameter τ , which control the step-size schedule for ρ_t , to $\kappa = 0.75$ and $\tau = 1$. We must also specify the batch size parameters b , which control the number of points sampled at each iteration to estimate the gradient of the ELBO in Equation (8). The choice of batch size is non-trivial and consistently affects the accuracy and computational cost of the SVI algorithms. We refer to Tan (2017) as a recent work that considers the choice of batch size. To assess the effect of batch size on model performance in our settings, we consider batch sizes corresponding to 5%, 20%, and 50% of the sample size, $b \in \{0.05, 0.20, 0.50\}$. For convenience, we will refer to SVI algorithms with these batch sizes as SVI-005, SVI-02, and SVI-05, respectively. Parameter initialization for CAVI

and SVI is described in Supplementary §B. Convergence is monitored using the mean squared difference in parameters across iterations.

For GS, we draw 10, 000 posterior samples after a burn-in of 5, 000 and save in memory the values of the parameters every 5 iterations, i.e., a total of 2, 000 samples. We then use these samples to compute the averages of $\Phi\Phi^\top$ and $\Lambda_s\Lambda_s^\top$ as point estimates of their respective posterior means. Note that $\Phi\Phi^\top$ and $\Lambda_s\Lambda_s^\top$ are identifiable functionals of the posterior of Model (10), therefore posterior means can be estimated directly by averaging the GS draws.

4.1 Single-Study Simulations

We consider simulation scenarios for all combinations of $P = \{100, 500, 5000\}$ and $N = \{100, 500, 1000\}$, generating the data from Model (1) with $J = 4$ factors. For each scenario, we set each value of Λ to zero with probability 2/3 or randomly generated it from an $\text{Uniform}(0,1)$ with probability 1/3. Further simulation studies with sparsity levels of 50% and 80% are presented in Tables S4-S7. The values of Ψ are drawn from an $\text{Uniform}(0.1,1)$. Using the generated $\Sigma = \Lambda\Lambda^\top + \Psi$, we sample 50 datasets for each simulation scenario by taking N draws from $\mathcal{N}_p(\mathbf{0}, \Sigma)$. For each dataset, we estimate θ with $J^* = 5$ factors using CAVI, SVI, GS, ECM, ADVI, and POET as previously described. Bayesian methods (CAVI, SVI, GS, and ADVI) use priors (2)-(3) with hyperparameters $\nu = 3$, $a_1 = 2.1$, $a_2 = 3.1$, $a'' = 1$, and $b'' = 0.3$.

Table 1 reports, for each simulation scenario, the averages and standard deviations for the computation time, memory usage, and estimation accuracy. Comprehensive results are provided in Fig.s S1-S4 (Supplementary §C). Given the computational resources available, we found performing GS, ECM, and ADVI infeasible for some scenarios and therefore, results are not reported (see caption of Table 1 for details).

As expected, CAVI required much less computational time than GS for every scenario. For example, in the scenario with $P = 500$ and $N = 1, 000$, CAVI required, on average, 68 seconds compared to 873 for the GS. The average computation time of SVI was

lower than CAVI in every simulation scenario with $N > 100$, while in scenarios with $N = 100$ and $P = 100, 500$, SVI-005 required, on average, more computational time than CAVI. Note that in these scenarios, SVI-005 only uses 5 observations to approximate the gradient in (8), requiring many iterations to converge. When the sample size increases, SVI is faster than CAVI. This is particularly evident for scenarios with larger P . For example, when $P = 5,000$ and $N = 1,000$ CAVI required on average approximately 25 minutes compared to 12, 5 and 2 minutes for SVI-05, SVI-02, and SVI-005 respectively. Additionally, SVI with small batch sizes scales better with increasing N . In the scenario with $P = 5,000$, $N = 100$, SVI-005, required an average of 23.65 seconds. The computational time increases six times when $N = 1,000$, while the computational time for SVI-02 and SVI-05 increases by 7 and 8 times, respectively. We note that the frequentist algorithms ECM and POET require very little computational time in simulation scenarios with $P = 100$, requiring approximately 3 seconds of runtime with $N = 1000$, while CAVI required 13 seconds in the same scenario. However, ECM cannot be performed in scenarios with $P \geq N$, and POET does not scale with increasing P , requiring approximately 1017 seconds on average when $P = 5000$ and $N = 100$, which is approximately 6 times longer than CAVI, which required only 164 seconds on average. ADVI consistently required the most computational time out of any of the algorithms considered, taking approximately 150 times longer than CAVI even in the smallest simulation scenario considered (202 seconds vs 1.37 seconds).

In terms of memory usage, the computational burden of GS drastically increases with the number of variables. For example, GS required 396 Mb of RAM memory on average when $N = 1,000$ and $P = 100$, and 4,275 Mb when $P = 500$. In other words, when $P = 500$ the memory requirements are approximately 11 times the ones needed for $P = 100$. In the same scenarios, the memory requirements of CAVI increase by approximately 1.2 times, from 37 Mb ($P = 100$) to 45 Mb ($P = 500$). The ECM and SVI algorithms have memory requirements that are very similar to CAVI. POET and ADVI consistently required more memory than CAVI across all simulation scenarios, with the disparity in their memory requirements increasing in scenarios with P .

In terms of accuracy, CAVI and SVI estimate Σ with comparable accuracy to the estimates from GS, ECM, and POET in most scenarios. As expected, SVI performance deteriorates in scenarios with a small sample size and batch size. For example, in the scenario with $P = 5,000$ and $N = 100$, SVI-005 had an average RV of 0.62 compared to 0.74 of CAVI. However, the difference in performance between the two algorithms diminishes as the sample size increases. For example, when $P = 5,000$ and $N = 1,000$, SVI-005 has an average RV of 0.94 compared to 0.97 of CAVI. On average, ADVI had comparable estimation accuracy to CAVI, in terms of RV, but had worse estimation accuracy in terms of the Frobenius and L1 norms (Table S3, Supplementary §C).

4.2 Multi-Study Simulations

Analogously to §4.1, we assess the performance of the proposed VI algorithms in the multi-study setting. We consider simulation scenarios with $S = \{5, 10\}$, $P \in \{100, 500, 5000\}$ and $N_s = \{100, 500, 1000\}$, generating the data from Model (10) with $K = 4$ common factors and $J_s = 4, s = 1, \dots, S$ study-specific factors. For each scenario, factor matrices Φ and Λ_s are randomly generated setting each values to 0 with probability 2/3 or drawing from an $\text{Uniform}(0,1)$ with probability 1/3, while the values of $\Psi_s, s = 1, \dots, S$ are generated from an $\text{Uniform}(0,1,1)$. Using the generated Σ_s , we sample $50 \times S$ by taking N_s draws from $\mathcal{N}(\mathbf{0}, \Phi\Phi^\top + \Lambda_s\Lambda_s^\top + \Psi_s)$ for each study. We estimate θ with $K^* = 5, J_s^* = 5$ using the multi-study versions of CAVI, SVI, GS, ECM, and ADVI as described in §4. We excluded comparisons with POET in this section since it is not designed to analyze multi-study data. Bayesian methods (CAVI, SVI, GS, and ADVI) use priors (11)-(13) with hyperparameters $\nu = 3 = \nu_s = 3, a_1 = a_{s1} = 2.1, a_2 = a_{s2} = 3.1, a'' = 1$, and $b'' = 0.3$. As we found in the single-study simulations, we were unable to run some scenarios with GS, ECM, and ADVI within the limited computational resources available.

Table 2 reports the average computational times (refer to Fig. S5, Supplementary §C for comprehensive results). CAVI and SVI required much less time on average than GS in every simulation scenario. For example, in the scenario with $S = 10, P = 500$, and $N_s = 1,000$, CAVI required an average of approximately 22 minutes to converge while

GS took about 6 hours, making CAVI about 16 times faster than GS. In the same scenario, SVI algorithms were even faster, taking approximately 16, 7, and 6 minutes on average for SVI-05, SVI-02 and SVI-005 respectively, making SVI algorithms between 22 to 65 times faster than GS. ECM and ADVI require much more computational time than CAVI in all simulation scenarios.

Table 3 reports the average memory usage (refer to Fig. S6, Supplementary §C, for the comprehensive results). The proposed algorithms require much less memory than GS in all the scenarios. For example, GS required 3.5 times as much memory as CAVI in the setting with $S = 5$, $N_s = 100$, $P = 100$ (126 Mb compared to 36 Mb), while in the setting with $S = 10$, $N_s = 1000$, $P = 500$ GS required about 5.5 times more memory than CAVI (1557 Mb compared to 285 Mb). In all scenarios, CAVI and SVI have comparable memory requirements. ECM had memory requirements to CAVI for scenarios with $P = 100$, but appears to scale poorly with increasing P , having memory requirements comparable to GS in simulation scenarios with $P = 500$. ADVI consistently had the largest memory requirements in every scenario we could complete.

We report the averages and standard deviations of the RV coefficients between estimated Σ_s and the simulation truth (Table 4). We refer to Fig. S7 for comprehensive results for Σ_s and Fig.s S8–S9 for the RV coefficients of Φ and Λ_s , respectively. As before, CAVI estimates Σ_s with comparable accuracy to GS, ADVI, and ECM in most scenarios. In scenarios where the number of observations is small, CAVI is less accurate than GS. For example, when $P = 100$ and $N_s = 100$, CAVI has an average RV of 0.85 compared to 0.92 of the GS. When the sample size increases, the average RV of CAVI is closer to GS. For example, when $P = 500$ and $N_s = 1,000$ CAVI as an average RV of 0.96 compared to 0.99 of the GS. When both the sample and batch size are small, the SVI algorithms can present poor performance. When N_s is low, the performance of SVI with a small batch size can be inferior to CAVI, especially in high dimensional settings. For example, with $P = 5,000$ and $N_s = 100$, the average RV of SVI-005 was 0.68 while CAVI had an average RV of 0.86. The difference between SVI and CAVI is much smaller with larger batch sizes—the RVs were comparable to CAVI

for SVI-02 and SVI-05 across all 18 simulation scenarios. When $N_s > 100$, SVI-005 presented an accuracy comparable with CAVI for all the considered values of P and S .

5 Case Study: Ovarian Cancer Gene Expression Data

We apply the proposed algorithms to four high-dimensional datasets containing microarray gene expression data available in the R package `curatedOvarianData` 1.36 (Ganzfried et al., 2013). The curated ovarian cancer studies contain $N_s = \{578, 285, 195, 140\}$ patient observations. We selected genes with observed variances in the highest 15% in at least one study, resulting in a final set of $P = 3643$ genes. Each dataset was centered and scaled before analysis.

We start by evaluating the ability of MSFA to predict out-of-sample \mathbf{x}_{si} compared to alternative FA models in a 10-fold cross-validation. In particular, we consider three models: 1) A MSFA as described in §3; 2) A FA fitted on a dataset stacking all the $S = 4$ studies (Stacked FA); 3) Independent FA models for each study (Independent FA). For the MSFA model 1) we set $K^* = 10$ and $J_s^* = 10$, and use CAVI and SVI (with $b_s = 0.5$ for $s = 1, \dots, 4$). For FA in 2) and 3), we set $J^* = 10$ using CAVI and SVI ($b = 0.5$). It is important to emphasize that fitting these models with GS, ECM, or ADVI would have required more than 128Gb of RAM to be allocated (the limit set in §4); therefore, this was considered not feasible.

Predictions are computed as follows:

$$MSFA: \mathbf{x}_{si} = \Phi \hat{\mathbf{f}}_{si} + \Lambda_s \hat{\mathbf{1}}_{si} \quad \text{Stacked FA: } \mathbf{x}_{si} = \Phi \hat{\mathbf{f}}_{si} \quad \text{Independent FA: } \mathbf{x}_{si} = \Lambda_s \hat{\mathbf{1}}_{si}.$$

Factor scores for out-of-sample observations are derived by adapting Bartlett's method (Bartlett, 1937) for MSFA:

$$\hat{\mathbf{f}}_{si} = \left((\Phi, \Lambda_s)^\top \Psi_s^{-1} (\Phi, \Lambda_s) \right)^{-1} \Phi^\top \Psi_s^{-1} \mathbf{x}_{si}, \quad (16)$$

$$\hat{\mathbf{I}}_{si} = \left((\mathbf{\Phi}, \mathbf{\Lambda}_s)^\top \mathbf{\Psi}_s^{-1} (\mathbf{\Phi}, \mathbf{\Lambda}_s) \right)^{-1} \mathbf{\Lambda}_s^\top \mathbf{\Psi}_s^{-1} \mathbf{x}_{si}, \quad (17)$$

where $(\mathbf{\Phi}, \mathbf{\Lambda}_s)$ is obtained by stacking the common and study-specific factor loading matrices, and $\mathbf{\Phi}$ and $\mathbf{\Lambda}_s, \hat{\mathbf{\Psi}}_s$ denote the posterior mode of the respective VI approximation. The Mean Squared Error (MSE) is computed as the average of the prediction errors across all

$$\text{MSE} = \left\{ \sum_{s=1}^S N_s \right\}^{-1} \sum_{s=1}^S \sum_{i=1}^{N_s} \sum_{p=1}^P (x_{sip} - \hat{x}_{sip})^2.$$

observations in every study:

Results of the 10-fold cross-validation are reported in Table 5. Despite the high dimensionality of the dataset, the proposed VB algorithms were able to fit the MSFA model with an average computation time of approximately 11 minutes for CAVI and 9 minutes for SVI. Also, the two considered FA approaches were computed in a small amount of time, approximately 14 minutes for CAVI and 7 minutes for SVI. In terms of MSE, CAVI and SVI share a similar performance, decreasing the MSE of 10% compared to Stacked FA and 6% compared to Independent FA.

We then estimated the common covariance $\hat{\Sigma}_\Phi = \mathbf{\Phi}\mathbf{\Phi}^\top$ with MSFA via CAVI and represented it via a gene co-expression network (Figure 1). A gene co-expression network is an undirected graph where the nodes correspond to genes, and the edges correspond to the degree of co-expression between genes. The number of connections between genes is visualized in the plot by the size of each node, i.e., the bigger, the more connected. We identified two different and important clusters. The first cluster contains genes associated with the immune system and cell signaling, such as *CD53* (Dunlock, 2020), *LAPTM5* (Glowacka et al., 2012), *PTPRC* (Hermiston et al., 2003), *TYROBP* (Lanier et al., 1998), *C1QA/C1QB* (Liang et al., 2022). Also, in the first cluster, some genes play a crucial role in cancer, such as *SAMSN1* (Yan et al., 2013) and *FCER1G* (Yang et al., 2023). The second network includes genes such as *FBN1* known to promote metastasis in ovarian cancer (Wang et al., 2015), and *SERPINF1* crucial to the prognosis of cancer (Zhang et al., 2022).

The gene network in Fig. 1 reveals the ability of the MSFA to discover common signal pathways across different studies, critically helpful in ovarian cancer and generalizable to new context areas. These results further show the ability of the proposed VB algorithms to provide reliable results in a short computation time.

6 Discussion

We have proposed VB algorithms that provide fast Bayesian estimation for FA in both single and multi-study settings. These algorithms provided significant advantages over MCMC-based implementations, requiring substantially less memory and time while achieving comparable performance in characterizing data covariance matrices. Moreover, in the ovarian cancer application case, we showed how these algorithms could help reveal biological pathways in the high-dimensional setting, using computational resources typically available on a laptop rather than a high-performance computing server.

In Supplementary §D, we present additional simulations to evaluate the predictive performance of our VB algorithms compared to GS. We consider scenarios with $S = 1$, sample size $N = 100$ and $P = 100, 500$. These results suggest that CAVI and GS have similar performance in terms of out-of-sample prediction errors and coverage and length of prediction intervals. SVI performed worse than CAVI and GS, achieving inappropriate 95% prediction interval coverage (ranging from 0.92 to 0.99) and higher MSE. For example, with $P = 500$, SVI algorithms have an MSE ranging from 0.6 to 0.9, compared to 0.3 of GS and CAVI.

While VB algorithms offer numerous advantages, they have some limitations. First, convergence can be sensitive to parameter initialization. This is common to many iterative algorithms, including the EM algorithm used in frequentist MSFA (De Vito et al., 2019). Thus, we have provided effective informative initializations (Supplementary §B), helpful especially in scenarios with $P > N$. Second, the performance of SVI algorithms can vary substantially depending on the chosen batch size parameter. Tuning the batch size parameter remains an active area of research (Tan, 2017). Third,

compared to MCMC and GS, VB algorithms tend to underestimate uncertainty. However, when the main focus is on point estimates and exploratory analysis—typically the main goals of FA—the underestimation of uncertainty is less of a concern than in other settings.

A known issue with FA is the non-identifiability of the factor loading matrix, i.e., the factor loadings can be rotated and produce the same covariance structure. This orthogonal indeterminacy is not a concern if inference only targets an identifiable function of the loading matrix, such as the covariance matrix. Alternatively, the loading matrices in Models 1 and 10 can be further constrained to be identifiable (e.g, Lopes and West, 2004). These or alternative constraints can be incorporated in the proposed algorithms by adapting parameter-expansions techniques (see for example Ročková and George, 2016). We also refer to De Vito et al. (2019) for two methods to recover factor loading matrices from MSFA, which can be applied to estimates provided by our algorithms.

We have even observed that the impact of the shrinkage priors (2), and (11)–(12) on the GS posterior distribution differs from the effect on the approximate VB posterior (Fig.s S10-S11, Supplementary §C). When employing the same hyperparameter values, the VB algorithm induces a stronger shrinkage effect compared to GS. Similar findings were reported by Zhao and Yu (2009) while studying FA under a different shrinkage prior. Further research is required to understand if this behavior is shared when using VB with other popular shrinkage priors, such as the Besov prior (Lassas et al., 2009), the cumulative shrinkage process prior (Legramanti et al., 2020), the triple-gamma prior (Cadonna et al., 2020), and non-local shrinkage priors (Avalos-Pacheco et al., 2022).

In conclusion, the proposed VB algorithms enable scaling inference for FA and MSFA to high-dimensional data, enabling new research opportunities in previously inaccessible settings without extensive computational resources.

7 Acknowledgements

This work was supported by the US National Institutes of Health, under grants NIGMS/NIH COBRE CBHD P20GM109035, and 5R01CA262710-03.

8 Supplementary Materials

Supplementary A - F: §A: Descriptions for CAVI and SVI algorithms for Bayesian FA and Bayesian MSFA as described in §2-3; §B: Description of initialization strategies for VB Algorithms; §C: Additional figures and tables for simulation study; §D: Prediction accuracy and prediction uncertainty quantification simulations; §E: Observed factor loading shrinkage per column index; §F: Derivations supporting coordinate updates in Algorithms 1-4. (pdf)

Supporting code: VIMSFA: the current version of our R package available at github.com/blhansen/VI-MSFA

References

- Aliverti, E. and Russo, M. (2022). Stratified stochastic variational inference for high-dimensional network factor model. *Journal of Computational and Graphical Statistics*, 31(2):502–511.
- Archambeau, C. and Bach, F. (2008). Sparse probabilistic projections. *Advances in Neural Information Processing Systems*, 21.
- Avalos-Pacheco, A., Rossell, D., and Savage, R. S. (2022). Heterogeneous large datasets integration using Bayesian factor regression. *Bayesian Analysis*, 17(1):33–66.
- Bartlett, M. S. (1937). The statistical conception of mental factors. *British Journal of Psychology. General Section*, 28(1):97–104.
- Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. *International AAAI Conference on Weblogs and Social Media*.
- Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information science and statistics. Springer, New York.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.
- Cadonna, A., Frühwirth-Schnatter, S., and Knaus, P. (2020). Triple the gamma—a unifying shrinkage prior for variance and variable selection in sparse state space and tvp models. *Econometrics*, 8(2):20.
- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., et al. (2017). Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76:1–32.

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., et al. (2008). High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456.

Dang, K.-D. and Maestrini, L. (2022). Fitting structural equation models via variational approximations. *Structural Equation Modeling: A Multidisciplinary Journal*, 29(6):839–853.

De Vito, R. and Avalos-Pacheco, A. (2023). Multi-study factor regression model: an application in nutritional epidemiology. arXiv:2304.13077.

De Vito, R., Bellio, R., Trippa, L., and Parmigiani, G. (2019). Multi-study factor analysis. *Biometrics*, 75(1):337–346.

De Vito, R., Bellio, R., Trippa, L., and Parmigiani, G. (2021). Bayesian multistudy factor analysis for high-throughput biological data. *The Annals of Applied Statistics*, 15(4):1723–1741.

Dunlock, V. E. (2020). Tetraspanin CD53: an overlooked regulator of immune cell function. *Medical Microbiology and Immunology*, 209(4):545–552.

Durante, D. (2017). A note on the multiplicative gamma process. *Statistics & Probability Letters*, 122:198–204.

Durante, D. and Rigon, T. (2019). Conditionally conjugate mean-field variational Bayes for logistic models. *Statistical Science*, 34(3):472–485.

Edefonti, V., Hashibe, M., Ambrogi, F., Parpinel, M., et al. (2012). Nutrient-based dietary patterns and the risk of head and neck cancer: a pooled analysis in the International Head and Neck Cancer Epidemiology consortium. *Annals of Oncology*, 23(7):1869–1880.

Fan, J., Liao, Y., and Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 75(4):603–680.

Fasano, A., Durante, D., and Zanella, G. (2022). Scalable and accurate variational Bayes for high-dimensional binary regression models. *Biometrika*, 109(4):901–919.

Frühwirth-Schnatter, S. (2023). Generalized cumulative shrinkage process priors with applications to sparse Bayesian factor analysis. *Philosophical Transactions of the Royal Society A*, 381:20220148–20220148.

Frühwirth-Schnatter, S., Hosszejni, D., and Lopes, H. F. (2023). Sparse Bayesian factor analysis when the number of factors is unknown. arXiv:2301.06459.

Ganzfried, B. F., Riester, M., Haibe-Kains, B., Risch, T., et al. (2013). curatedOvarianData: clinically annotated data for the ovarian cancer transcriptome. *Database*, 2013.

Garrett-Mayer, E., Parmigiani, G., Zhong, X., Cope, L., et al. (2008). Cross-study validation and combined analysis of gene expression microarray data. *Biostatistics*, 9(2):333–354.

Ghahramani, Z. and Beal, M. (1999). Variational inference for Bayesian mixtures of factor analysers. *Advances in Neural Information Processing Systems*, 12.

Glowacka, W. K., Alberts, P., Ouchida, R., Wang, J. Y., et al. (2012). LAPTM5 protein is a positive regulator of proinflammatory signaling pathways in macrophages. *The Journal of Biological Chemistry*, 287(33):27691–27702.

Grabski, I. N., Vito, R. D., Trippa, L., and Parmigiani, G. (2023). Bayesian combinatorial MultiStudy factor analysis. *The Annals of Applied Statistics*, 17(3):2212 – 2235.

Hermiston, M. L., Xu, Z., and Weiss, A. (2003). CD45: a critical regulator of signaling thresholds in immune cells. *Annual Review of Immunology*, 21:107–137.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Jaakkola, T. S. and Jordan, M. I. (1997). A variational approach to Bayesian logistic regression models and their extensions. In Madigan, D. and Smyth, P., editors, *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, volume R1 of *Proceedings of Machine Learning Research*, pages 283–294.

Joo, J., Williamson, S. A., Vazquez, A. I., Fernandez, J. R., et al. (2018). Advanced dietary patterns analysis using sparse latent factor models in young adults. *The Journal of Nutrition*, 148:1984–1992.

Kucukelbir, A., Blei, D. M., Gelman, A., Ranganath, R., et al. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(1):1–45.

Lanier, L. L., Corliss, B. C., Wu, J., Leong, C., et al. (1998). Immunoreceptor DAP12 bearing a tyrosine-based activation motif is involved in activating NK cells. *Nature*, 391(6668):703–707.

Lassas, M., Saksman, E., and Siltanen, S. (2009). Discretization-invariant Bayesian inversion and Besov space priors. *Inverse Problems and Imaging*, 3(1):87–122.

Legramanti, S., Durante, D., and Dunson, D. B. (2020). Bayesian cumulative shrinkage for infinite factorizations. *Biometrika*, 107(3):745–752.

Liang, Z., Pan, L., Shi, J., and Zhang, L. (2022). C1QA, C1QB, and GZMB are novel prognostic biomarkers of skin cutaneous melanoma relating tumor microenvironment. *Scientific Reports*, 12:20460.

- Lopes, H. F. and West, M. (2004). Bayesian model assessment in factor analysis. *Statistica Sinica*, 14(1):41–67.
- Ludvigson, S. C. and Ng, S. (2007). The empirical risk–return relation: A factor analysis approach. *Journal of Financial Economics*, 83(1):171–222.
- Papastamoulis, P. and Ntzoufras, I. (2022). On the identifiability of bayesian factor analytic models. *Statistics and Computing*, 32(2):23.
- Pournara, I. and Wernisch, L. (2007). Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinformatics*, 8:1–20.
- Quinn, T. (2017). peakRAM: Monitor the Total and Peak RAM Used by an Expression or Function. R package version 1.0.2.
- Rajaratnam, B. and Sparks, D. (2015). MCMC-based inference in the era of big data: A fundamental analysis of the convergence complexity of high-dimensional chains. arXiv:1508.00947.
- Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407.
- Robert, P. and Escoufier, Y. (1976). A unifying tool for linear multivariate statistical methods: The RV-coefficient. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 25(3):257–265.
- Roy, A., Lavine, I., Herring, A. H., and Dunson, D. B. (2021). Perturbed factor analysis: Accounting for group differences in exposure profiles. *The Annals of Applied Statistics*, 15(3):1386 – 1404.
- Ročková, V. and George, E. I. (2016). Fast Bayesian Factor Analysis via Automatic Rotations to Sparsity. *Journal of the American Statistical Association*, 111(516):1608–1622.

Samartsidis, P., Seaman, S. R., Montagna, S., Charlett, A., et al. (2020). A Bayesian multivariate factor analysis model for evaluating an intervention by using observational time series data on multiple outcomes. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 183(4):1437–1459.

Tan, L. S. L. (2017). Stochastic variational inference for large-scale discrete choice models using adaptive batch sizes. *Statistics and Computing*, 27(1):237–257.

Wang, X. V., Verhaak, R. G. W., Purdom, E., Spellman, P. T., et al. (2011). Unifying gene expression measures from multiple platforms using factor analysis. *PLOS ONE*, 6(3):e17691.

Wang, Z., Gu, Y., Lan, A., and Baraniuk, R. (2020). VarFA: A variational factor analysis framework for efficient Bayesian learning analytics. *Proceedings of The 13th International Conference on Educational Data Mining (EDM)*.

Wang, Z., Liu, Y., Lu, L., Yang, L., et al. (2015). Fibrillin-1, induced by Aurora-A but inhibited by BRCA2, promotes ovarian cancer metastasis. *Oncotarget*, 6(9):6670–6683.

Yan, Y., Zhang, L., Xu, T., Zhou, J., et al. (2013). SAMS1 is highly expressed and associated with a poor survival in Glioblastoma Multiforme. *PLOS ONE*, 8(11):e81905.

Yang, R., Chen, Z., Liang, L., Ao, S., et al. (2023). Fc fragment of IgE receptor Ig (FCER1G) acts as a key gene involved in cancer immune infiltration and tumour microenvironment. *Immunology*, 168(2):302–319.

Zhang, C., Yang, W., Zhang, S., Zhang, Y., et al. (2022). Pan-cancer analysis of osteogenesis imperfecta causing gene SERPINF1. *Intractable & Rare Diseases Research*, 11(1):15–24.

Zhao, J.-h. and Yu, P. L. (2009). A note on variational Bayesian factor analysis. *Neural Networks*, 22(7):988–997.

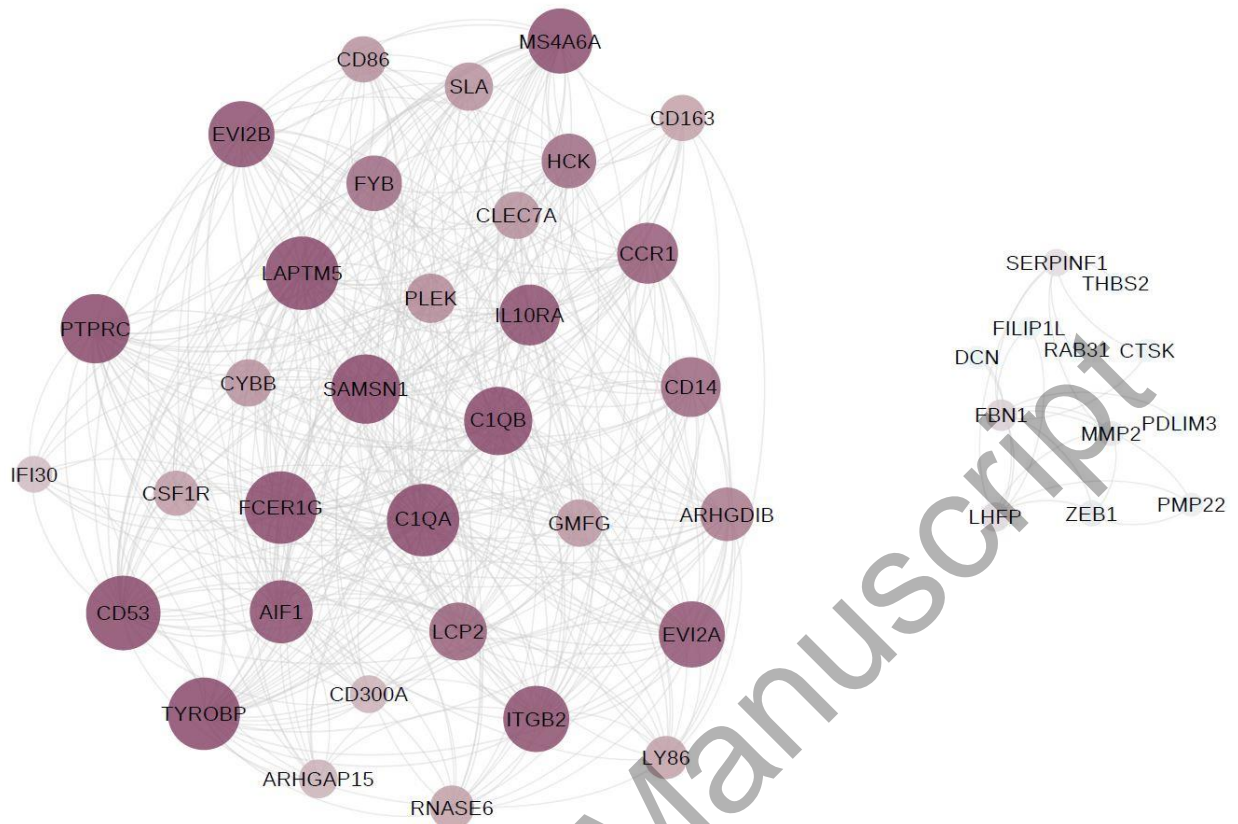


Fig. 1 Common gene co-expression network estimated via $\Phi\Phi^T$ across the four ovarian cancer studies. The network is obtained using Gephi via the Fruchterman-Reingold algorithm (Bastian et al., 2009). We include edges if $|\Phi\Phi^T| \geq 0.5$, and exclude from the figure nodes with no edges. The figure includes 92 nodes and 851 edges.

Table 1 Average computational cost (in seconds), used RAM (in Mb), and estimation accuracy (RV coefficients) across 50 simulation replicates, mean(sd). Note: we were unable to run GS, ADVI, and POET for some scenarios due to computation time and memory limitations, as the maximum allocation of 24 hours of run time and 128Gb of RAM for each iteration was insufficient. Additionally, we are unable to run ECM in scenarios with $P \geq N$. This is indicated with — in the table.

										Estimation Accuracy					
Time (Seconds)										Memory (Mb)			$(RV(\Sigma, \Sigma))$		
0															
ADV	10	202.48(12				180.56(2				0.76(0					
I	0	0.32)	—	—	1.37)	—	—	.17)	—	—					
	50	157.97(2.4	745.66(1				293.63(0	4133.59(0.99(0	0.98(0		
GS	0	8)	4.21)	—	.16)	1.02)	—	.01)	.01)	—					
CAV	50	34.94(0.		789.25(65	37.11(0.	46.65(0.	696.27(85.1	0.98(0	0.94(0	0.93(0					
I	0	6.29(0.07)	5)	.87)	06)	33)	9)	.01)	.01)	.01)					
SVI-	50	19.78(0.		369.37(24				748.79(61.4	0.97(0	0.92(0	0.91(0				
05	0	3.63(0.05)	3)	.09)	37.1(0)	46.5(0)	5)	.01)	.01)	.01)					
SVI-	50	9.62(0.1		169.66(10				745.16(66.4	0.97(0	0.92(0	0.91(0				
02	0	1.78(0.04)	4)	.74)	32.4(0)	46.5(0)	7)	.01)	.01)	.01)					
SVI-	50	4.49(0.0		69.01(4.0	24.82(0.	46.23(1.	745.4(72.09	0.94(0	0.9(0.	0.89(0					
005	0	0.86(0.04)	8)	8)	13)	57))	.01)	.01)	.01)					
POE	50	22.86(1.		2596.15(1	103.6(0.	1195.96(100903.5(1	0.98(0	0.97(0	0.98(0					
T	0	1.17(0.09)	9)	37.93)	03)	0.07)	101.22)	.01)	.01)	.01)					
	50				37.3(0.0				0.98(0						
ECM	0	3.19(0.07)	—	—	3)	—	—	.01)	—	—					
ADV	50	988.78(10				233.59(5				0.97(0					
I	0	29.77)	—	—	9.72)	—	—	.02)	—	—					
GS	10	182.65(2.2	873.28(3	—	396.72(1	4275.09(—	0.99(0	0.99(0	—					

		Time (Seconds)			Memory (Mb)			Estimation Accuracy (RV(Σ , Σ))		
	00	5)	2.11)		2.59)	1.39)))	
CAV	10	13.42(2.03	68.07(1.14	97.61(5	36.71(0.44	65(0.78	4.08(106	0.98(0	0.97(0	0.97(0
I	00)	38)	0.72)	06)	34)	86)	.01)	.01)	.01)
SVI-	10		38.23(0.70	2.83(54			747.39(109	0.98(0	0.96(0	0.96(0
05	00	6.9(0.08)	66)	.63)	36.7(0)	60.7(0)	88)	.01)	.01)	.01)
SVI-	10		18.01(0.31	8.39(24		60.38(2.76	1.84(104	0.97(0	0.96(0	0.96(0
02	00	3.28(0.05)	3)	.57)	36.7(0)	28)	72)	.01)	.01)	.01)
SVI-	10		7.89(0.1	127.56(8	33.57(0.60	42(1.74	8.56(130	0.96(0	0.94(0	0.94(0
005	00	1.49(0.04)	2)	06)	17)	94)	9)	.01)	.01)	.01)
POE	10		38.87(3.			2385.51(0.99(0	0.99(0	
T	00	3.18(0.16)	02)	—	203(0)	0.07)	—))	—
	10		76.9(0.8			61.04(0.		0.99(0	0.97(0	
ECM	00	3.08(0.07))	—	37(0.03)	3)	—)	.01)	—
ADV	10	2011.14(2		183.79(1				0.95(0		
I	00	119.07)	—	—	5.67)	—	—	.09)	—	—

Table 2 Average computation time in seconds, mean(sd), for 50 multi-study simulation replicates. Note: we were unable to run GS and ADVI for some scenarios due to computation time and memory limitations, as the maximum allocation of 24 hours

of run time and 128Gb of RAM for each iteration was insufficient. Additionally, we are unable to run ECM in scenarios with $P \geq N_s$. This is indicated with — in the table.

		S = 5			S = 10		
Meth							
od	N _s	P= 100	P=500	P=5000	P=100	P=500	P=5000
GS	10	1742.43(39.3	7907.73(254.		3288.87(73.3	14001.49(85	
	0	9)	45)	—	7)	7.06)	—
CAVI	10			704.38(10.			1434.77(28.4
	0	14.03(0.3)	68.38(0.77)	27)	26.49(0.32)	138.1(1.93)	6)
SVI-05	10						1216.65(15.5
	0	11.15(0.14)	54.53(0.56)	608(5.73)	22.12(0.3)	109.7(0.79)	4)
SVI-02	10			343.61(4.6			
	0	6.06(0.11)	30.58(0.33)	2)	11.95(0.19)	59.3(0.67)	686.83(7.86)
SVI-005	10			207.91(3.0			
	0	5.15(0.42)	17.86(0.14)	1)	7.44(0.46)	34.2(0.24)	409.53(8.76)
ECM	10						
	0	—	—	—	—	—	—
ADVI	10	1596(1200.04			2851.41(1667		
	0)	—	—	.1)	—	—
GS	50	2100.87(42.6	9534.65(446.		3962.02(84.6	16861.88(18	
	0	8)	68)	—	6)	35.69)	—
CAVI	50	62.96(0.68)	316.34(5.56)		126.77(2.1)		
	50			3303.78(66		649.44(27.64	7001.73(419.

S = 5				S = 10			
	0			.45))		42)
SVI-	50			2548.9(63.			5055.43(460.
05	0	49.62(3.61)	232.96(2.9)	29)	94.74(1.3)	467.26(5.47)	32)
SVI-	50			1231.84(13			
02	0	42.69(2.42)	111.5(1.18)	.37)	64.11(2.54)	224.64(2.61)	2475(49.49)
SVI-	50						1140.81(33.7
005	0	39.65(2.1)	82.77(2.44)	581.16(9)	58.95(2.08)	129.35(1.88)	6)
	50	1219.83(572.			3063.55(742.		
ECM	0	05)	—	—	01)	—	—
	50	8642.38(7060			12078.54(234		
ADVI	0	.36)	—	—	0.81)	—	—
	10	2657.03(56.3	11638.49(624			21271.9(125	
GS	00	2)	.67)	—	5048.3(99.63)	6.6)	—
	10			6669.16(71		1332.73(19.1	11674.25(13
CAVI	00	128.28(1.87)	665.95(11.99)	2.57)	253.37(5.38)	5)	14.02)
SVI-	10			5072.69(34		965.94(20.17	8785.58(808.
05	00	144.7(7.69)	473.39(7.28)	4.52)	218.7(12.65))	99)
SVI-	10			2349.63(45			4417.48(496.
02	00	123.46(5.25)	259.83(15.78)	.61)	181.71(7.47)	434(7.62)	26)
SVI-	10			1044.76(21			2115.91(92.7
005	00	106.77(4.21)	213.44(5.5)	.22)	143.88(4.02)	329.09(9.48)	4)

S = 5				S = 10			
	10	1023.11(443.	61570.47(197		4088.28(903.		
ECM	00	41)	74.57)	—	69)	—	—
	10	14602.13(133			25977.91(174		
ADVI	00	89.31)	—	—	21.21)	—	—

Accepted Manuscript

Table 3 Average peak RAM usage in Mb, mean(sd), for 50 multi-study simulation replicates. Note: we were unable to run GS and ADVI for some scenarios due to computation time and memory limitations, as the maximum allocation of 24 hours of run time and 128Gb of RAM for each iteration was insufficient. Additionally, we are unable to run ECM in scenarios with $P \geq N_s$. This is indicated with — in the table.

		S = 5			S = 10			
Metho	d	N _s	P= 100	P=500	P=5000	P=100	P=500	P=5000
			126.24(2.33	528.48(22.94		184.89(1.07		
GS	100))	—)	856.76(17.8)	—
					3130.33(315.			5708.93(710.
CAVI	100	35.81(0.08)	57.93(1.88)		86)	40.49(6.68)	69.49(2.16)	89)
					3123.61(313.			5720.95(701.
SVI-05	100	35.51(0.65)	77.93(7.92)		18)	47.14(3.05)	82.62(1.27)	25)
					2556.88(231.			5707.01(703.
SVI-02	100	35.51(0.65)	58.15(0.35)		39)	47.41(2.44)	82.62(1.27)	05)
					2556.78(231.			5710.53(697.
SVI-005	100	35.51(0.65)	57.93(1.23)		38)	47.14(3.05)	82.6(1.39)	44)
ECM	100		—	—	—	—	—	—
			359.08(23.3					
ADVI	100		9)	—	—	610(26.92)	—	—
GS	500		735.71(51.2)		—			—
		211.76(15.6				308.18(40.6	1170.03(53.3	

S = 5				S = 10			
2)				2)		3)	
				3235.65(323.		4736.93(650.	
CAVI	500	47.66(1.73)	116.52(4.45)	23)	60.92(1.99)	155.69(0.1)	13)
				2532.19(191.		4652.65(633.	
SVI-05	500	47.83(0.2)	117.13(1.17)	86)	61.01(2.01)	155.59(0.08)	34)
				2532.29(191.		4621.51(501.	
SVI-02	500	47.6(1.43)	115.31(7.34)	88)	61.01(2.01)	155.59(0.08)	91)
SVI-	109.23(13.42			2533.01(191.		4613.68(499.	
005	500	47.56(1.71))	98)	61.01(2.01)	155.59(0.08)	66)
ECM	500	64.31(0.27)	—	—	80.61(0.42)	—	—
362.45(14.2				518.52(28.3			
ADVI	500	5)	—	—	9)	—	—
100 213.15(13.7 1001.85(91.5				294.72(14.7 1557.5(108.8			
GS	0	7)	5)	—	2)	3)	—
100 168.55(10.13 2663.43(258. 87.94(12.32				5080.33(560.			
CAVI	0	46.03(0.18))	27))	285.26(1.73)	21)
100 2674.08(225.				5276.57(641.			
SVI-05	0	60.42(4.8)	171.02(2.67)	38)	76.06(2.4)	285.35(1.74)	45)
100 2697.99(168.				274.18(14.38 6068.98(888.			
SVI-02	0	61.06(3.75)	171.02(2.67)	79)	75.67(3.6))	85)

S = 5

S = 10

SVI-	100			2697.99(168.			5901.31(645.
005	0	61.71(2.08)	169.57(8.52)	79)	75.29(4.46)	285.35(1.74)	49)
	100		779.87(59.36				
ECM	0	62.42(0.27))	—	99.76(0.43)	—	—
	100	375.78(28.6			577.74(82.7		
ADVI	0	3)	—	—	1)	—	—

Accepted Manuscript

Table 4 Average estimation accuracy, reported as RV between estimated $\Sigma_s = \Phi\Phi^\top + \Lambda_s\Lambda_s^\top + \Psi_s$ and the simulation truth, mean(sd), for 50 multi-study simulation replicates. Note: we were unable to run GS and ADVI for some scenarios due to computation time and memory limitations, as the maximum allocation of 24 hours of run time and 128Gb of RAM for each iteration was insufficient. Additionally, we are unable to run ECM in scenarios with $P \geq N_s$. This is indicated with — in the table.

		S = 5			S = 10		
Method	N_s	P= 100	P=500	P=5000	P=100	P=500	P=5000
GS	100	0.923(0.025)	0.894(0.047)	—	0.927(0.025)	0.904(0.044)	—
CAVI	100	0.851(0.032)	0.842(0.026)	0.85(0.021)	0.862(0.031)	0.855(0.027)	0.864(0.02)
				0.857(0.022)			0.871(0.021)
SVI-05	100	0.835(0.032)	0.842(0.026))	0.843(0.037)	0.856(0.028))
				0.842(0.025)			0.849(0.024)
SVI-02	100	0.827(0.033)	0.827(0.027))	0.83(0.038)	0.837(0.03))
SVI-005	100	0.713(0.049)	0.676(0.042))	0.704(0.047)	0.679(0.043)	0.679(0.04)
ECM	100	—	—	—	—	—	—
ADVI	100	0.798(0.071)	—	—	0.761(0.061)	—	—
		0.986(0.0036)	0.984(0.0058)		0.982(0.0059)	0.987(0.0046)	
GS	500))	—))	—
				0.956(0.018)			0.966(0.013)
CAVI	500	0.932(0.022)	0.933(0.03))	0.921(0.02)	0.958(0.013))

S = 5

S = 10

				0.954(0.018		0.965(0.013	
SVI-05	500	0.92(0.025)	0.927(0.031))	0.892(0.032)	0.952(0.016))
				0.952(0.018		0.963(0.013	
SVI-02	500	0.925(0.024)	0.926(0.03))	0.896(0.03)	0.951(0.015))
SVI-				0.935(0.018		0.943(0.013	
005	500	0.932(0.019)	0.92(0.028))	0.907(0.023)	0.938(0.015))
ECM	500	0.907(0.012)	—	—	0.907(0.015)	—	—
ADVI	500	0.91(0.045)	—	—	0.906(0.033)	—	—
	100	0.992(0.0021	0.993(0.0021		0.993(0.0022	0.993(0.0019	
GS	0))	—))	—
	100			0.966(0.018	0.948(0.0071	0.977(0.011	
CAVI	0	0.94(0.029)	0.949(0.011)))	0.961(0.015))
	100			0.964(0.019		0.974(0.012	
SVI-05	0	0.934(0.028)	0.944(0.012))	0.936(0.012)	0.955(0.018))
	100			0.963(0.019		0.974(0.013	
SVI-02	0	0.938(0.027)	0.945(0.012))	0.941(0.011)	0.954(0.018))
SVI-	100			0.957(0.018	0.948(0.0085	0.966(0.012	
005	0	0.946(0.022)	0.946(0.011)))	0.954(0.016))
	100						
ECM	0	0.929(0.011)	0.915(0.039)	—	0.904(0.028)	—	—

		S = 5			S = 10		
	100						
ADVI	0	0.92(0.039)	—	—	0.909(0.031)	—	—

Table 5 Computation time (in minutes) and MSE, reported as mean(sd), across the 10 cross-validation folds. Relative MSE is the average MSE divided by the average MSE of MSFA fitted using CAVI (the best-performing algorithm). For independent FA the time include fitting $S = 4$ serially.

Method	Time (minutes)	MSE	Relative MSE
MSFA (CAVI)	11.06(0.24)	2296(53)	1
MSFA (SVI)	8.83(0.24)	2315(52)	1.01
Independent FA (CAVI)	14.41(0.16)	2435(55)	1.06
Independent FA (SVI)	6.77(0.036)	2447(50)	1.06
Stacked FA (CAVI)	13.58(0.19)	2531(52)	1.10
Stacked FA (SVI)	6.07(0.093)	2527(53)	1.10