

FITTING MULTILEVEL FACTOR MODELS*

TETIANA PARSHAKOVA[†], TREVOR HASTIE[‡], AND STEPHEN BOYD[§]

Abstract. We examine a special case of the multilevel factor model, with covariance given by multilevel low rank (MLR) matrix [T. Parshakova et al., *Factor fitting, rank allocation, and partitioning in multilevel low rank matrices*, in Optimization, Discrete Mathematics, and Applications to Data Sciences, SOIA 220, Springer, 2024, pp. 135–173]. We develop a novel, fast implementation of the expectation–maximization algorithm, tailored for multilevel factor models, to maximize the likelihood of the observed data. This method accommodates any hierarchical structure and maintains linear time and storage complexities per iteration. This is achieved through a new efficient technique for computing the inverse of the positive definite MLR matrix. We show that the inverse of positive definite MLR matrix is also an MLR matrix with the same sparsity in factors, and we use the recursive Sherman–Morrison–Woodbury matrix identity to obtain the factors of the inverse. Additionally, we present an algorithm that computes the Cholesky factorization of an expanded matrix with linear time and space complexities, yielding the covariance matrix as its Schur complement. This paper is accompanied by an open-source package that implements the proposed methods.

Key words. multivariate analysis, sparse matrix factorization, mathematical software, scalable algorithm, multilevel factor model

MSC codes. 62H12, 15A23, 65F05

DOI. 10.1137/24M1697992



See reproducibility of computational results at end of the article.

1. Introduction. Factor models are used to explain the variation in the observed variables through a smaller number of factors. In fields like biology, economics, and social sciences, the data often has hierarchical structures. To capture this structure specialized multilevel factor models were developed. Existing methods for fitting these models do not scale well with large datasets.

In this work, we introduce an efficient algorithm for fitting multilevel factor models. Our method is compatible with any hierarchical structure and achieves linear time and storage complexity per iteration.

1.1. Prior work.

Factor models. Factor analysis was initially developed to address problems in psychometrics about 120 years ago [55], and it later found applications in psychology, finance, economics, and statistics. The idea behind factor analysis is to describe variability among the observed variables using a small number of unobserved variables called factors. Factor models decompose a covariance matrix into a sum of a low rank matrix, associated with underlying factors, and a diagonal matrix, representing idiosyncratic variances. Since the early 20th century, factor analysis has seen significant methodological advancements [21, 11, 30, 20, 19], with several books dedicated to its theory and application [27, 12].

*Received by the editors September 30, 2024; accepted for publication (in revised form) by N. Gillis May 20, 2025; published electronically August 25, 2025. Associated software is available at https://github.com/cvxgrp/multilevel_factor_model.

[†]<https://doi.org/10.1137/24M1697992>

[‡]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 USA (tetianap@stanford.edu).

[§]Department of Statistics, Stanford University, Stanford, CA 94305 USA (hastie@stanford.edu).

[§]Electrical Engineering Department, Stanford University, Stanford, CA 94305 USA (boyd@stanford.edu).

Hierarchically structured data. Data from fields such as biology, economics, social sciences, and medical sciences often exhibit a hierarchical, nested, or clustered structure. This has led to the development of specialized techniques in factor analysis aimed specifically at handling hierarchically structured data such as hierarchical factor models [54, 57] and multilevel factor models [1, 38].

Hierarchical factor models. In hierarchical factor models, factors are organized into a hierarchy, where general factors at the top influence more specific factors positioned beneath them [54, 9, 59, 50]. This model type does not necessarily reflect a hierarchy in the data (e.g., individuals within groups) but rather in the latent variables themselves. Widely used in psychometrics, these models are crucial for distinguishing between higher-order and lower-order factors [10, 39]. For instance, [17] identified a hierarchical structure of personality with two general factors, stability and plasticity, at the top, and the so-called Big Five personality factors below them: neuroticism, agreeableness, and conscientiousness are under stability, while extraversion and openness are under plasticity.

Multilevel factor models. Multilevel factor models are statistical frameworks developed in the 1980s to handle hierarchical data structures; see [1, 23, 38, 52, 46], and the books [14, 24]. These models partition factors into global and local components, allowing the decomposition of the variances of observed variables into components attributable to each level of the hierarchy. There is a wide variety of multilevel factor models discussed in the literature, with the general form for a 2-level factor model presented in [24, section 8.2].

Multilevel (dynamic) factor models have also been applied to time series data [26, 3, 56, 4]. They have been particularly effective in modeling the co-movement of economic quantities across different levels [26, 4]. For example, [33, 13, 32] used these models to characterize the co-movement of international business cycles on global, regional, and country levels.

In this paper we focus on a special case of the multilevel factor model, that has no intercept and no linear covariates. The framework can be easily extended to more general case as needed, see Appendix E. We assume the observations follow a normal distribution, so the model is defined by a covariance matrix that is a multilevel low rank (MLR) matrix [43]. In [43] the authors consider two problems beyond fitting, namely, rank allocation and capturing partition. Here, we assume that both rank allocation and hierarchical partition are fixed, and focus solely on fitting factors.

Fitting methods. Several methods have been employed to fit multilevel models, each with its advantages and challenges. Among the most prominent are maximum likelihood and Bayesian estimation techniques [15], and Frobenius norm-based fitting methods [43]. Commonly utilized algorithms for these methods include the expectation-maximization (EM) algorithm [53, 49], the Newton–Raphson algorithm [36], iterative generalized least squares [23], the Fisher scoring algorithm, and Markov Chain Monte Carlo [25]. Despite the efficacy of these approaches, no single method proves entirely satisfactory under all possible data conditions encountered in research. As a result, statisticians are continually developing alternative techniques to enhance model fitting and accuracy [15, 35].

Software packages. Several commercial packages offer capabilities for handling multilevel modeling, including LISREL [31], Mplus [2, 41, 42], and MLwiN [48]. The open-source packages include lavaan [51, 29], gllamm [47]. Additional resources and software recommendations can be found in [14, section 1.7] and [24, section 18]. These tools are primarily designed for multilevel linear models [22], and most of them do not support the specific requirements of factor analysis within multilevel frameworks that

involve an arbitrary number of levels in hierarchical structures. Although OpenMx [7, 44], an open-source package that implements MLE-based fitting methods, does support multiple levels of hierarchy, it was unable to handle our large-scale examples. Additionally, we found no high-quality, open-source implementations of MCMC-based fitting methods; thus these were not included in our comparison.

In this paper, leveraging the MLR structure of the covariance matrix, we derive a novel fast implementation of the EM algorithm for multilevel factor modeling that works with any hierarchical structure and requires linear time and storage complexities per iteration.

1.2. Our contribution. The main contributions of this paper are the following:

1. We present a novel computationally efficient algorithm for fitting multilevel factor models, which operates with linear time and storage complexities per iteration.
2. We show that the inverse of an invertible PSD MLR matrix is also an MLR matrix with the same sparsity in factors, and we use the recursive Sherman–Morrison–Woodbury (SMW) matrix identity to obtain the factors of the inverse.
3. We present an algorithm that computes the Cholesky factorization of an expanded matrix with linear time and space complexities, yielding the covariance matrix as its Schur complement. We also show that Cholesky factor has the same sparsity pattern as its inverse.
4. We provide an open-source package that implements the fitting method, available at https://github.com/cvxgrp/multilevel_factor_model. We also provide several examples that illustrate our method.

2. Multilevel factor model. In this section we review the multilevel low rank (MLR) matrix along with notations necessary for our method. We then present a variant of the multilevel factor model that will be the focus of this paper.

2.1. Multilevel low rank matrices. An MLR matrix [43] is a row and column permutation of a sum of matrices, each one a block diagonal refinement of the previous one, with all blocks low rank, given in the factored form. We focus on the special case of symmetric positive semidefinite (PSD) MLR matrices.

An $n \times n$ contiguous PSD MLR matrix Σ with L levels has the form

$$(2.1) \quad \Sigma = \Sigma^1 + \cdots + \Sigma^L,$$

where Σ_l is a PSD block diagonal matrix,

$$\Sigma_l = \text{blkdiag}(\Sigma_{l,1}, \dots, \Sigma_{l,p_l}), \quad l = 1, \dots, L,$$

where **blkdiag** is the direct sum of blocks $\Sigma_{l,k} \in \mathbf{R}^{n_{l,k} \times n_{l,k}}$ for $k = 1, \dots, p_l$. Here p_l is the size of the partition at level l , and

$$\sum_{k=1}^{p_l} n_{l,k} = n, \quad l = 1, \dots, L.$$

Throughout this paper we consider $L \geq 2$ and $p_L = n$; therefore, Σ_L is a diagonal matrix. Also for all $l = 1, \dots, L$ define matrices

$$\Sigma_{l+} = \Sigma_l + \cdots + \Sigma_L, \quad \Sigma_{l-} = \Sigma_1 + \cdots + \Sigma_l.$$

By definition, we have $\Sigma = \Sigma_{1+} = \Sigma_{L-}$.

The block dimensions on level l partition the n indices into p_l groups, which are contiguous. Let J_1, \dots, J_L be partitions of the set $\{1, \dots, n\}$. (By symmetry of Σ_l , these partitions are the same for rows and columns.)

For each $l = 1, \dots, L$, the level l partition of the indices is the set of p_l index sets

$$J_l = \{\{1, \dots, n_{l,1}\}, \{n_{l,1} + 1, \dots, n_{l,1} + n_{l,2}\}, \dots, \{n - n_{l,p_l} + 1, \dots, n\}\}.$$

We require that these partitions be hierarchical, meaning that for all $l = 2, \dots, L$, the partition J_l is a *refinement* of J_{l-1} . We write

$$J_l \preceq J_{l-1}$$

to indicate that for every index set $X \in J_l$, there exists index set $Y \in J_{l-1}$ such that $X \subseteq Y$.

We require that blocks on level l have rank not exceeding r_l , given in the factored form as

$$\Sigma_{l,k} = F_{l,k} F_{l,k}^T, \quad F_{l,k} \in \mathbf{R}^{n_{l,k} \times r_l}, \quad l = 1, \dots, L-1, \quad k = 1, \dots, p_l,$$

and refer to $F_{l,k}$ as the factor (of block k on level l).

Define a diagonal matrix $D = \Sigma_L$, which forces $r_L = 1$. See Figure 1. We refer to $r = r_1 + \dots + r_{L-1} + 1$ as the MLR-rank of Σ . The MLR-rank of A is, in general, not the same as the rank of Σ . We refer to $(r_1, \dots, r_{L-1}, 1)$ as the rank allocation.

Factor form. For each level $l = 1, \dots, L-1$ define

$$F_l = \text{blkdiag}(F_{l,1}, \dots, F_{l,p_l}) \in \mathbf{R}^{n \times p_l r_l}.$$

Then we have

$$\Sigma_l = F_l F_l^T, \quad l = 1, \dots, L-1.$$

Define

$$F = [F_1 \quad \dots \quad F_{L-1}] \in \mathbf{R}^{n \times s},$$

with $s = \sum_{l=1}^{L-1} p_l r_l$. Then we can write Σ as

$$\Sigma = [F \quad D^{1/2}] [F \quad D^{1/2}]^T = FF^T + D,$$

where F has s columns, and a very specific sparsity structure, with column blocks that are block diagonal, and D is diagonal; see Figure 2.

Define F_{l+} as the concatenation of left factors from levels $l, \dots, L-1$, and similarly F_{l-} , i.e.,

$$F_{l+} = [F_l \quad \dots \quad F_{L-1}], \quad F_{l-} = [F_1 \quad \dots \quad F_l].$$

Thus the number of nonzero coefficients in F_{l+} is $n \sum_{l'=l}^{L-1} r_{l'}$ and in F_{l-} is $n \sum_{l'=1}^l r_{l'}$. By definition, we also have $F = F_{1+} = F_{(L-1)-}$.

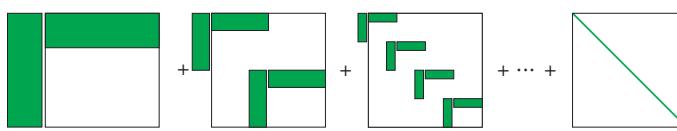


FIG. 1. (Contiguous) PSD MLR matrix given as a sum of block diagonal matrices with each block being low rank. The coefficients of the factors are depicted in green. (Figure in color online.)

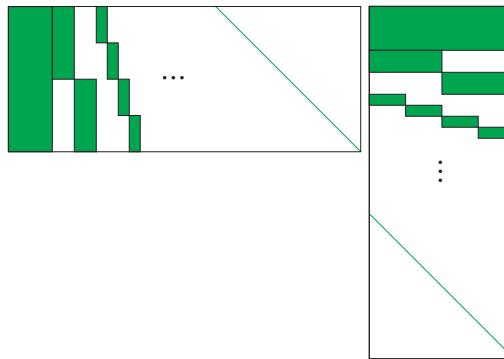


FIG. 2. (Contiguous) PSD MLR matrix given as a product of two sparse structured matrices. The coefficients of the factors are depicted in green. (Figure in color online.)

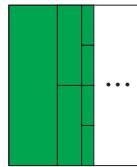


FIG. 3. (Contiguous) PSD MLR matrix given in compressed form.

Compressed factor form. We can also arrange the factors into one dense matrix with dimensions $n \times r$. We vertically stack the factors at each level to form matrices

$$\bar{F}^l = \begin{bmatrix} F_{l,1} \\ \vdots \\ F_{l,p_l} \end{bmatrix} \in \mathbf{R}^{n \times r_l}, \quad l = 1, \dots, L-1,$$

and lastly a diagonal of matrix D , $\text{diag}(D) \in \mathbf{R}^n$. We horizontally stack these matrices to obtain one matrix

$$\bar{F} = [\bar{F}^1 \quad \dots \quad \bar{F}^{L-1}] \in \mathbf{R}^{n \times (r-1)}.$$

All of the coefficients in the factors of a contiguous MLR matrix are contained in this matrix and vector $\text{diag}(D)$; see Figure 3. To fully specify a contiguous MLR matrix, we need to give the block dimension $n_{l,k}$ for $l = 1, \dots, L$, $k = 1, \dots, p_l$, and the ranks r_1, \dots, r_L .

PSD MLR matrix. We reviewed the contiguous PSD MLR matrix. The PSD MLR matrix is given by the symmetric permutation of rows and columns of a contiguous PSD MLR matrix. Therefore, the PSD MLR matrix uses a general hierarchical partition of the index set.

Example. To illustrate our notation we give an example with $L = 4$ levels, $p_1 = 1$, with the second level partitioned into $p_2 = 2$ groups, and the third level partitioned into $p_3 = 4$ groups. We take $n = 5$, with block row (and column) dimensions

$$\begin{aligned} n_{1,1} &= 5, \\ n_{2,1} &= 3, \quad n_{2,2} = 2, \\ n_{3,1} &= 1, \quad n_{3,2} = 2, \quad n_{3,3} = 1, \quad n_{3,4} = 1 \\ n_{4,1} &= 1, \quad n_{4,2} = 1, \quad n_{4,3} = 1, \quad n_{4,4} = 1, \quad n_{4,5} = 1. \end{aligned}$$

The sparsity patterns of Σ_1 , Σ_2 , and Σ_3 are shown below, with $*$ denoting a possibly nonzero entry, and all other entries zero (the sparsity pattern of Σ_4 matches that of a diagonal matrix),

$$\Sigma_1 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ & * & * \\ & * & * \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} * & & & \\ & * & * & \\ & * & * & \\ & & * & \\ & & & * \end{bmatrix}.$$

If we have ranks $r_1 = 2$, $r_2 = 1$, $r_3 = 1$, and $r_4 = 1$, the MLR-rank is $r = 5$, with factor sparsity pattern as below,

$$F_1 = \begin{bmatrix} * & * \\ * & * \\ * & * \\ * & * \\ * & * \end{bmatrix}, \quad F_2 = \begin{bmatrix} * \\ * \\ * \\ * \\ * \end{bmatrix}, \quad F_3 = \begin{bmatrix} * & & & \\ & * & & \\ & & * & \\ & & & * \end{bmatrix}.$$

This means that Σ_1 has rank 2, the $p_2 = 2$ blocks in Σ_2 each have rank 1, and the $p_3 = 4$ blocks in Σ_3 also have rank 1.

2.2. Partition notation. In this paper we consider matrices that are block diagonal, e.g., F_l , and matrices formed by concatenation of block diagonal matrices, e.g., F_{l+} . To formally describe the row and column sparsity patterns of these matrices, we define the following operators.

Define an operator \mathcal{J} , that for any block diagonal matrix $B \in \mathbf{R}^{m \times n}$ returns its column index partition. Similarly, define operator $\tilde{\mathcal{I}}$ to return the row index partition of B . Note by definition $\tilde{\mathcal{I}}(B) = \mathcal{J}(B^T)$.

Define operators \mathcal{I} and $\tilde{\mathcal{J}}$ that for any (horizontal or vertical) concatenation of block diagonal matrices $B = [B_1 \ \dots \ B_c] \in \mathbf{R}^{m \times n}$ return lists of partitions for each block diagonal matrix

$$\mathcal{J}(B) = (\tilde{\mathcal{J}}(B_1), \dots, \tilde{\mathcal{J}}(B_c)), \quad \mathcal{I}(B) = (\tilde{\mathcal{I}}(B_1), \dots, \tilde{\mathcal{I}}(B_c)).$$

We say a partition *refines* a list of partitions if it refines each partition in that list. Conversely, we say a list of partitions refines a partition if every partition in the list refines that partition. We denote this relation by \preceq .

Finally, define the sparsity pattern of any $B \in \mathbf{R}^{m \times n}$ as

$$\mathbf{supp}(B) = \{(i, j) \mid B_{ij} \neq 0, i = 1, \dots, m, j = 1, \dots, n\}.$$

Remark 2.1. If $B, C \in \mathbf{R}^{m \times n}$ are concatenations of block diagonal matrices with $\mathbf{supp}(B) = \mathbf{supp}(C)$, then $\mathcal{I}(B) = \mathcal{I}(C)$ and $\mathcal{J}(B) = \mathcal{J}(C)$.

Example. Applying these operators to the matrices from the previous section, we get

$$\mathcal{I}(\Sigma_l) = \mathcal{J}(\Sigma_l) = \mathcal{I}(F_l) = J_l,$$

and

$$\begin{aligned} \mathcal{I}(F_{l-}) &= (J_1, \dots, J_l), \\ \mathcal{J}(F_l) &= \{ \{1, \dots, r_l\}, \{r_l + 1, \dots, 2r_l\}, \dots, \{(p_l - 1)r_l + 1, \dots, p_l r_l\} \}, \\ \mathcal{I}(F_{l+}) &= (J_l, \dots, J_{L-1}). \end{aligned}$$

We also have

$$\mathcal{I}(F_{l+}) \preceq \mathcal{I}(F_l) \preceq \mathcal{I}(F_{l-}),$$

and

$$\mathbf{supp}(\Sigma_l) = \mathbf{supp}(\Sigma_{l+}).$$

2.3. Problem setting. We consider a multilevel factor model,

$$(2.2) \quad y = Fz + e,$$

where $F \in \mathbf{R}^{n \times s}$ is structured factor loading matrix, $z \in \mathbf{R}^s$ are factor scores, with $z \sim \mathcal{N}(0, I_s)$, and $e \in \mathbf{R}^n$ are the idiosyncratic terms, with $e \sim \mathcal{N}(0, D)$.

We assume that the n features can be hierarchically partitioned, with specific factors explaining the correlations within each group of this hierarchical partition. This can be modeled by taking F to be the factor matrix of PSD MLR. Then $y \in \mathbf{R}^n$ is a Gaussian random vector with zero mean and covariance matrix Σ that is PSD MLR,

$$\Sigma = FF^T + D.$$

We assume we have access to hierarchical partition and rank allocation. Therefore, we reorder n features so that the groups in hierarchical partition correspond to contiguous index ranges. We seek to fit the coefficients of $F \in \mathbf{R}^{n \times s}$ and diagonal $D \in \mathbf{R}^{n \times n}$ (with $\text{diag}(D) > 0$) from the observed samples.

We assume $s \ll n$, i.e., number of factors is smaller than the number of features.

3. Fitting methods. In this paper, we estimate parameters F and D using the maximum likelihood estimation (MLE). This approach is different from that in [43], which focuses on fitting the PSD MLR matrix to the empirical covariance matrix using a Frobenius norm-based loss. Notably, the Frobenius norm is not an appropriate loss for fitting covariance models. First, the Frobenius norm is coordinate-independent, it treats all coordinates equally, whereas MLE accounts for coordinate-specific differences, where changes across different coordinates have varying implications. This can lead to covariance models with small eigenvalues when using the Frobenius norm, a situation that MLE inherently guards against. Second, the Frobenius norm-based loss is distribution-agnostic. In contrast, MLE takes advantage of the known distribution of the data. Nevertheless, there is an intrinsic connection between the MLE and Frobenius norm, which we detail in section A of the appendix.

3.1. Frobenius norm-based estimation. One way to estimate coefficients of matrices F and D is by minimizing Frobenius norm-based distance with sample covariance. This means solving the following optimization problem:

$$(3.1) \quad \begin{aligned} & \text{minimize} && \|FF^T + D - \hat{\Sigma}\|_F^2 \\ & \text{subject to} && FF^T + D \text{ is PSD MLR,} \end{aligned}$$

with the hierarchical partition and sparsity structure of F (number of levels, block dimensions, and ranks) predefined and fixed, as previously proposed in [43].

Since the problem (3.1) is nonconvex, [43, section 4] introduce two complementary block coordinate descent methods to find an approximate solution. For example, alternating least squares minimizes the fitting error over the left factors, then over the right factors, and so on. The second method updates factors at one level in each iteration by minimizing the fitting error while cycling over the levels.

3.2. Maximum likelihood estimation. Alternatively, we can estimate matrices F and D using MLE. Suppose we observe samples $y_1, \dots, y_N \in \mathbf{R}^n$, organized in the matrix form as

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} \in \mathbf{R}^{N \times n}.$$

The log-likelihood of N samples is

$$(3.2) \quad \ell(F, D; Y) = -\frac{nN}{2} \log(2\pi) - \frac{N}{2} \log \det(FF^T + D) - \frac{1}{2} \mathbf{Tr}((FF^T + D)^{-1}Y^T Y).$$

For structured F , directly maximizing the log-likelihood $\ell(F, D; Y)$ is difficult. Instead, the expectation-maximization (EM) algorithm [16] is the preferred approach for MLE.

Simplification via data augmentation. Difficult maximum likelihood problems can be simplified by data augmentation. Suppose, along with Y , we also observed latent data $z_1, \dots, z_N \in \mathbf{R}^s$, organized in matrix $Z \in \mathbf{R}^{N \times s}$. Then the log-likelihood of complete data (Y, Z) for model (2.2) is

$$(3.3) \quad \ell(F, D; Y, Z) = -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log \det D - \frac{1}{2} \|(Y - ZF^T)D^{-1/2}\|_F^2 - \frac{1}{2} \|Z\|_F^2.$$

Maximizing the $\ell(F, D; Y, Z)$ with respect to F and D is now tractable. First, since D is diagonal, when F is known solving for D is trivial. Second, note that $\ell(F, D; Y, Z)$ is separable across the rows of F . The nonzero coefficients in each row of F can be found by solving the least squares problem.

For example, consider a simple factor model, where F is just a dense low rank matrix. Then from the optimality conditions, the solution to (3.3) is given by

$$(3.4) \quad F = Y^T Z (Z^T Z)^{-1}, \quad D = \frac{1}{N} \text{diag}(\text{diag}((Y - ZF^T)^T (Y - ZF^T))).$$

Since we only observe Y and Z are missing, we use the EM algorithm to simplify the problem through data augmentation.

4. EM algorithm. The EM algorithm iterates expectation and maximization steps until convergence. After each pair of E and M steps it can be shown that the log-likelihood of the observed data is nondecreasing, with equality at a local optimum.

4.1. Expectation step. In the expectation step we compute the conditional expectation of complete data log-likelihood with respect to the conditional distribution $(Y, Z | Y)$ governed by the the current estimate of parameters F^0 and D^0 :

$$(4.1) \quad Q(F, D; F^0, D^0) = \mathbf{E}(\ell(F, D; Y, Z) | Y, F^0, D^0).$$

To evaluate the $Q(F, D; F^0, D^0)$, we need to compute several expectations. First, using (2.2) we have

$$\begin{aligned} \mathbf{cov}(y, z) &= \mathbf{E} F z z^T = F, \\ \mathbf{cov}(y, y) &= FF^T + D = \Sigma. \end{aligned}$$

Thus (z, y) is a Gaussian random vector with zero mean and covariance

$$\mathbf{cov}((z, y), (z, y)) = \begin{bmatrix} I_s & F^T \\ F & \Sigma \end{bmatrix}.$$

Second, the conditional distribution $(z_i | y_i, F^0, D^0)$ is Gaussian,

$$\mathcal{N}\left(F^{0T}(\Sigma^0)^{-1}y_i, I_s - F^{0T}(\Sigma^0)^{-1}F^0\right).$$

Using the omitted derivations in section C.1, we can show that (4.1) equals

$$\begin{aligned} Q(F, D; F^0, D^0) = & -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log \det D - \frac{1}{2} \mathbf{Tr}(W) \\ (4.2) \quad & - \frac{1}{2} \mathbf{Tr}(D^{-1}(Y^T Y - 2FV + FWF^T)), \end{aligned}$$

where we defined matrices $V \in \mathbf{R}^{s \times n}$ and $W \in \mathbf{R}^{s \times s}$ as

$$(4.3) \quad V = \sum_{i=1}^N \mathbf{E}(z_i | y_i, F^0, D^0) y_i^T = F^{0T}(\Sigma^0)^{-1}Y^T Y,$$

$$\begin{aligned} (4.4) \quad W = & \sum_{i=1}^N \mathbf{E}(z_i z_i^T | y_i, F^0, D^0) \\ & = N(I_s - F^{0T}(\Sigma^0)^{-1}F^0) + F^{0T}(\Sigma^0)^{-1}Y^T Y(\Sigma^0)^{-1}F^0. \end{aligned}$$

Remark 4.1. Note that $(I_s - F^{0T}(\Sigma^0)^{-1}F^0) \succ 0$, as it is a Schur complement of matrix

$$\begin{bmatrix} I_s & F^T \\ F & \Sigma \end{bmatrix} \succ 0.$$

Consequently, it follows that $W \succ 0$.

4.2. Maximization step. In the maximization step we find updated parameters F^1 and D^1 by solving the following problem:

$$(4.5) \quad \begin{aligned} \text{maximize} \quad & Q(F, D; F^0, D^0) \\ \text{subject to} \quad & \begin{bmatrix} F & D^{1/2} \end{bmatrix} \text{ is the factor of PSD MLR.} \end{aligned}$$

Similar to (3.3), the maximization problem (4.5) is tractable. Observe, $Q(F, D; F^0, D^0)$ is separable across the rows of F (and respective diagonal elements of D). Moreover, using optimality conditions, the nonzero coefficients in each row of F can be determined by solving the least squares problem. For efficiency, we can group the rows by their sparsity pattern and instead solve the least squares problems for each row sparsity pattern of F at once, forming resulting matrix F^1 ; see section 5.2.2. Having F^1 , the diagonal matrix is then equal to

$$D^1 = \frac{1}{N} \text{diag}(\text{diag}(Y^T Y - 2F^1 V + F^1 W(F^1)^T)).$$

Thus F^1 and D^1 are the optimal solutions to problem (4.5), which we can also compute efficiently as discussed in section 5.

4.3. Initialization. The EM algorithm is a maximization-maximization procedure [28, section 8.5]; therefore, it converges to at least a local maximum. The trajectory of the EM algorithm depends on the initial values of F^0 and D^0 . We have observed that, depending on the initialization, it can converge to different local maxima. Additionally, when a good initial guess is not available, we have also observed that initializing matrices using a single sweep of the block coordinate descent method [43, section 4.2] from the top to bottom level works well.

5. Efficient computation.

5.1. Inverse of PSD MLR. In the maximization step, evaluating matrices V (4.3) and W (4.4) requires solving linear systems with the PSD MLR matrix. We will first address the efficient computation of Σ^{-1} , i.e.,

$$(F_1 F_1^T + \cdots + F_{L-1} F_{L-1}^T + D)^{-1}.$$

We will show that the inverse of the PSD MLR matrix is the MLR matrix with the same hierarchical partition and rank allocation, and

$$\Sigma^{-1} = -H_1 H_1^T - \cdots - H_{L-1} H_{L-1}^T + D^{-1},$$

where $H_l \in \mathbf{R}^{n \times p_l r_l}$ is a factor at level l with the same sparsity structure as F_l .

We compute the coefficients of the inverse by recursively applying the Sherman–Morrison–Woodbury (SMW) matrix identity.

5.1.1. Properties of structured matrices. We begin by giving useful properties of our structured matrices. Consider a factor matrix on level l , $F_l \in \mathbf{R}^{n \times p_l r_l}$, with p_l diagonal blocks of size $n_{l,k} \times r_l$, and row index partition set J_l for all $k = 1, \dots, p_l$.

Remark 5.1. Lemma C.1 states that if block diagonal matrices B and C are such that $\mathcal{J}(B) \preceq \mathcal{J}(C)$, then BC is block diagonal with $\mathcal{J}(BC) = \mathcal{J}(C)$, e.g.; see below. Moreover, if $\mathcal{J}(B) = \mathcal{J}(B)$, then $\text{supp}(BC) = \text{supp}(C)$.

Remark 5.2. The following properties are based on Lemma C.1, and they will be useful in the next section.

1. Matrix $F_l F_l^T \in \mathbf{R}^{n \times n}$ is a block diagonal matrix with blocks of size $n_{l,k} \times n_{l,k}$, with $\mathcal{J}(F_l F_l^T) = \mathcal{J}(F_l F_l^T) = J_l$, e.g., see illustration below.

For all $l' \geq l$, $J_{l'} \preceq J_l$ implies $\text{supp}(F_{l'} F_{l'}^T) \subseteq \text{supp}(F_l F_l^T)$. Then for matrix

$$F_{(l+1)+} F_{(l+1)+}^T = \sum_{l'=l+1}^{L-1} F_{l'} F_{l'}^T,$$

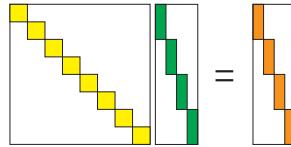
we obtain $\text{supp}(F_{l+1} F_{l+1}^T) = \text{supp}(F_{(l+1)+} F_{(l+1)+}^T)$.

2. For matrix $\Sigma_{(l+1)+}$ it holds that $\text{supp}(\Sigma_{(l+1)+}) = \text{supp}(F_{(l+1)}F_{(l+1)}^T)$.
3. The inverse of a block diagonal matrix is a block diagonal matrix consisting of the inverses of each block. Thus for

$$\Sigma_{(l+1)+}^{-1} = (F_{(l+1)+}F_{(l+1)+}^T + D)^{-1}$$

we get $\text{supp}(\Sigma_{(l+1)+}^{-1}) = \text{supp}(\Sigma_{(l+1)+})$.

4. Since $\mathcal{I}(\Sigma_{(l+1)+}^{-1}) = \mathcal{J}(\Sigma_{(l+1)+}^{-1}) \preceq \mathcal{I}(F_l)$, for $M_0 = \Sigma_{(l+1)+}^{-1}F_l$, $\text{supp}(M_0) = \text{supp}(F_l)$.

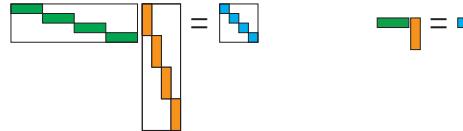


Thus matrix-vector product with M_0 can be computed in the order of $\sum_{k=1}^{p_l} n_{l,k} r_l = nr_l$ operations.

5. Since $\mathcal{J}(F_l^T) \preceq \mathcal{I}(F_{(l-1)-})$, we have $\mathcal{J}(M_0^T F_{(l-1)-}) = \mathcal{J}(F_{(l-1)-})$. Further, since $\mathcal{J}(\Sigma_{(l+1)+}) \preceq \mathcal{I}(F_{(l-1)-})$, it follows that

$$\text{supp}(\Sigma_{(l+1)+}^{-1} F_{(l-1)-}) = \text{supp}(F_{(l-1)-}).$$

6. For $F_l^T M_0 \in \mathbf{R}^{p_l r_l \times p_l r_l}$ it holds that $\mathcal{I}(F_l^T M_0) = \mathcal{J}(F_l^T M_0) = \mathcal{J}(F_l)$; see the figure below.



It is straightforward to check that each of the blocks is PSD.

5.1.2. Computing the inverse. We show that Σ^{-1} is an MLR matrix with factors having the same sparsity pattern as Σ . To establish this, we employ SMW matrix identity

$$(FF^T + D)^{-1} = D^{-1} - D^{-1}F(I_s + F^T D^{-1}F)^{-1}F^T D^{-1}.$$

We derive

$$(5.1) \quad \Sigma_{l+}^{-1} = \Sigma_{(l+1)+}^{-1} - H_l H_l^T,$$

where we defined matrix

$$H_l = \Sigma_{(l+1)+}^{-1} F_l (I_{p_l r_l} + F_l^T \Sigma_{(l+1)+}^{-1} F_l)^{-1/2};$$

see section C.2 for details. Remark 5.2 implies that $\text{supp}(H_l) = \text{supp}(\Sigma_{(l+1)+}^{-1} F_l) = \text{supp}(F_l)$. Applying recursion (5.1) from the bottom to the top level we get

$$\Sigma^{-1} = -H_1 H_1^T - \cdots - H_{L-1} H_{L-1}^T + D^{-1}.$$

Combining, we establish that Σ^{-1} is an MLR matrix with the same hierarchical partition as Σ .

Recursive SMW algorithm. We now show that the complexity of computing the coefficients of the MLR matrix Σ^{-1} is $O(nr^2 + p_{L-1}r_{\max}r^2)$ and extra memory used is less than $3nr + 2p_{L-1}r_{\max}r$, where $r_{\max} = \max\{r_1, \dots, r_L\}$. To do so, we recursively compute the coefficients of the matrices

$$(5.2) \quad \Sigma_{l+}^{-1} F_{(l-1)-}, \quad H_l,$$

from the bottom to the top level.

Suppose we have $n \sum_{l'=1}^l r_{l'}$ coefficients of $\Sigma_{(l+1)+}^{-1} F_{l-}$. This implies that we have the coefficients of $M_0 = \Sigma_{(l+1)+}^{-1} F_l$. We now show how to compute (5.2) using SMW matrix identity (5.1).

1. Compute $M_1 = M_0^T F_{(l-1)-}$ in $O(nr_l \sum_{l'=1}^{l-1} r_{l'})$ and store its $p_l r_l \sum_{l'=1}^{l-1} r_{l'}$ coefficients, since for $l' \leq l-1$ computing $M_0^T F_{l'}$ takes $nr_l r_{l'}$ operations, and compact form of $F_{(l-1)-}$ has $\sum_{l'=1}^{l-1} r_{l'}$ columns.
2. Compute $M_2 = (I_{p_l r_l} + F_l^T M_0)^{-1}$ in $O(nr_l^2 + p_l r_l^3)$ and store its $p_l r_l^2$ coefficients. Compute $H_l = M_0(I_{p_l r_l} + F_l^T M_0)^{-1/2}$ in $O(nr_l^2 + p_l r_l^3)$ and store its nr_l coefficients. Note that computing $I_{p_l r_l} + F_l^T M_0$ requires $O(nr_l^2)$ operations, and its eigendecomposition, $I_{p_l r_l} + F_l^T M_0 = Q_l \Lambda_l Q_l^T$, to compute H_l takes $O(p_l r_l^3)$ operations.
3. Compute $M_3 = M_2 M_1$ in $O(p_l r_l^2 \sum_{l'=1}^{l-1} r_{l'})$ and store its $p_l r_l \sum_{l'=1}^{l-1} r_{l'}$ coefficients, since $\mathcal{I}(M_2) \preceq \mathcal{I}(M_1)$ and compact form of M_1 has $\sum_{l'=1}^{l-1} r_{l'}$ columns. Note that $\text{supp}(M_3) = \text{supp}(M_1)$.
4. Compute $M_4 = M_0 M_3$ in $O(nr_l \sum_{l'=1}^{l-1} r_{l'})$ and store its $n \sum_{l'=1}^{l-1} r_{l'}$ coefficients, since $\mathcal{J}(M_0) \preceq \mathcal{J}(M_3)$, and compact form of M_3 has $\sum_{l'=1}^{l-1} r_{l'}$ columns. Note that $\text{supp}(M_4) = \text{supp}(F_{(l-1)-})$.
5. Compute $M_5 = \Sigma_{(l+1)+}^{-1} F_{(l-1)-} - M_4$ in $n \sum_{l'=1}^{l-1} r_{l'}$ and store its $n \sum_{l'=1}^{l-1} r_{l'}$ coefficients.

Therefore, the complexity at the level l is

$$O \left((nr_l + p_l r_l^2) \sum_{l'=1}^l r_{l'} \right).$$

Finally, we conclude that the total complexity is

$$T(n) = \sum_{l=1}^{L-1} O \left((nr_l + p_l r_l^2) \sum_{l'=1}^l r_{l'} \right) = O(nr^2 + p_{L-1}r_{\max}r^2),$$

and extra storage used is less than $3nr + 2p_{L-1}r_{\max}r$.

Recall that $s = \sum_{l=1}^{L-1} p_l r_l \ll n$; therefore, we have $p_{L-1} \ll n$. This implies that the time complexity is linear in n .

If we assume that the rank allocation is uniform $r_1 = \dots = r_{L-1} = \tilde{r}$ and that each block on one level is split into two nearly equal-sized blocks on the next level, $p_l = 2^{l-1}$, then the total complexity and storage are, respectively,

$$T(n) = O(n\tilde{r}^2 L^2 + 2^L \tilde{r}^3 L), \quad 3n\tilde{r}L + 2^L \tilde{r}^2 L.$$

Using the assumption that $s \ll n$ and $s = (2^{L-1} - 1)\tilde{r}$, we have

$$L \ll \log_2(n/\tilde{r} + 1) + 1.$$

Determinant. In Appendix D we show the covariance matrix Σ is the Schur complement of the expanded matrix. For this expanded matrix, we also provide an explicit Cholesky factorization method with linear time and space complexities. We leverage this connection to argue that the determinant of Σ equals to

$$\det(\Sigma) = \det(D) \prod_{l=1}^{L-1} \det(\Lambda_l).$$

Therefore, $\det(\Sigma)$ can be computed at no additional cost while recursively computing Σ^{-1} . Moreover, Cholesky factors enable feature-dependent linear transform that whitens the data and offer multiple useful interpretations; see [5, section 2]. See section D.4 for details.

5.2. EM iteration.

5.2.1. Selection matrices. Let s_i be the i th row sparsity pattern of F . We denote by $|s_i|$ the number of rows that share this sparsity. Then the number of unique sparsity patterns of rows of F equals the number of groups at level $L-1$, i.e., p_{L-1} . Note that we must have $\sum_{i=1}^{p_{L-1}} |s_i| = n$. Let $S_{r_i} \in \{0, 1\}^{|s_i| \times n}$ be a matrix that selects rows with i th sparsity pattern. Since any row sparsity pattern of F has $\sum_{l=1}^{L-1} r_l = r-1$ nonzero columns, we define $S_{c_i}^T \in \{0, 1\}^{s_i \times (r-1)}$ as a matrix that selects those columns of F . Thus, number of nonzero columns for row sparsity pattern s_i is $r-1$, and the matrices

$$S_{r_i} F S_{c_i}^T \in \mathbf{R}^{|s_i| \times (r-1)}, \quad i = 1, \dots, p_{L-1},$$

are dense in the coefficients of F ; see Figure 4.

Remark 5.3. For any matrix M with s rows we have

$$S_{r_i} F M = S_{r_i} F S_{c_i}^T S_{c_i} M, \quad i = 1, \dots, p_{L-1}.$$

5.2.2. EM iteration computation. Recall that $Q(F, D; F^0, D^0)$ (4.2) is separable across the rows of F . Therefore, to find F^1 we solve the reduced least squares problem for each sparsity pattern of F .

Recall matrices V (4.3) and W (4.4), where $W \succ 0$. To find the coefficients of F in problem (4.5), using section 5.2.1, it suffices to minimize the following:

$$\begin{aligned} & \mathbf{Tr}(FWF^T - 2FV) \\ &= \sum_{i=1}^{p_{L-1}} \mathbf{Tr}(S_{r_i} F W F^T S_{r_i}^T - 2S_{r_i} F V S_{r_i}^T) \\ &= \sum_{i=1}^{p_{L-1}} \mathbf{Tr}((S_{r_i} F S_{c_i}^T)(S_{c_i} W S_{c_i}^T)(S_{r_i} F S_{c_i}^T)^T - 2(S_{r_i} F S_{c_i}^T)(S_{c_i} V S_{r_i}^T)). \end{aligned}$$

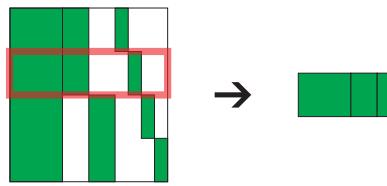


FIG. 4. Structured matrix F with $p_3 = 4$ row sparsity patterns is shown on the left. The second row sparsity pattern is highlighted in red. The dense matrix $S_{r_2} F S_{c_2}^T$ is shown on the right. (Figure in color online.)

To recover the coefficients of F , we solve the least squares problem,

$$(5.3) \quad S_{r_i} F S_{c_i}^T = (S_{c_i} V S_{r_i}^T)^T (S_{c_i} W S_{c_i}^T)^{-1}$$

for each $i = 1, \dots, p_L$. The inverse operation above is well-defined, since $W \succ 0$ implies $S_{c_i} W S_{c_i}^T \succ 0$.

We now derive the computational complexity for calculating F^1 . We first compute coefficients of MLR $(\Sigma^0)^{-1}$ in $T(n)$.

Next, we describe how to efficiently compute $S_{c_i} V S_{r_i}^T$ and $S_{c_i} W S_{c_i}^T$. Since $F^0 S_{c_i}^T \in \mathbf{R}^{n \times (r-1)}$, we compute $(\Sigma^0)^{-1} (F^0 S_{c_i}^T) \in \mathbf{R}^{n \times (r-1)}$ in $O(nr^2)$ using section 5.1. Next, we compute

$$\left((S_{c_i} F^0)^T (\Sigma^0)^{-1} \right) (F^0 S_{c_i}^T) \in \mathbf{R}^{(r-1) \times (r-1)}$$

in $O(nr^2)$. To evaluate the product $((S_{c_i} F^0)^T (\Sigma^0)^{-1}) Y^T \in \mathbf{R}^{(r-1) \times N}$ we need $O(nrN)$. Combining the above, we obtain

$$S_{c_i} V S_{r_i}^T = \left(S_{c_i} F^0 (\Sigma^0)^{-1} Y^T \right) (Y S_{r_i}^T) \in \mathbf{R}^{(r-1) \times |s_i|}$$

in $O(|s_i|rN)$. Also by computing

$$\left((S_{c_i} F^0)^T (\Sigma^0)^{-1} Y^T \right) (Y (\Sigma^0)^{-1} F^0 S_{c_i}^T) \in \mathbf{R}^{(r-1) \times (r-1)}$$

in $O(r^2N)$, we then get $S_{c_i} W S_{c_i}^T \in \mathbf{R}^{(r-1) \times (r-1)}$ in $O(r^2)$. Given $S_{c_i} V S_{r_i}^T$ and $S_{c_i} W S_{c_i}^T$, solving the linear system (5.3) takes $O(|s_i|r^3)$.

When solving for each sparsity pattern s_i , the total complexity of the maximization step is

$$T(n) + \sum_{i=1}^{p_{L-1}} O(nr^2 + nrN + |s_i|rN + r^2N + |s_i|r^3),$$

which simplifies to

$$T(n) + O(p_{L-1}nr^2 + p_{L-1}nrN + p_{L-1}r^2N + nr^3).$$

Plugging in the complexity of the inverse computation we arrive at

$$O(p_{L-1}nr^2 + nr^3 + p_{L-1}nrN + p_{L-1}r_{\max}r^2 + p_{L-1}r^2N).$$

Since $p_{L-1} \ll n$, the time complexity is linear in n .

As a stopping criteria we use the relative difference between consecutive log-likelihoods of observations (3.2). This requires computing the determinant of the covariance matrix, which we obtain at no cost during the inverse computation. See Appendices D and D.4 for details.

6. Numerical examples. We compare two factor fitting approaches based on Frobenius norm [43] and MLE. In the first example, we compare a traditional factor model (FM) with a multilevel factor model (MFM) using real data. We demonstrate that the multilevel factor model significantly improves the likelihood of the observations. In the second example, we consider a synthetic MFM to generate the observations. Our results show that the expected log-likelihood distribution of the MLE-based method significantly outperforms the Frobenius norm-based method. Finally, we apply our method to the real-world large-scale example.

6.1. Asset covariance matrix. We focus on the asset covariance matrix from [43, section 8.1]. In this example the daily returns of $n = 5000$ assets are found or derived from data from CRSP Daily Stock and CRSP/Compustat Merged Database 2023 Center for Research in Security Prices (CRSP), The University of Chicago Booth School of Business. We consider a $N = 300$ (trading) day period ending 2022/12/30, and for hierarchical partition use Global Industry Classification Standard (GICS) [6] codes from CRSP/Compustat Merged Database—Security Monthly during 2022/06/30 to 2023/01/31 which has $L = 6$ levels.

We use the GICS hierarchy and two different rank allocations; see Figure 5 and Table 1. For a rank allocation of $r_1 = 29, r_2 = \dots = r_5 = 0, r_6 = 1$ (i.e., a traditional factor model), our method’s average log-likelihood of realized returns improves by 98 compared to the Frobenius norm-based method. Alternatively, using ranks $r_1 = 14, r_2 = 6, r_3 = 4, r_4 = 3, r_5 = 2, r_6 = 1$, as determined by the rank allocation algorithm in [43] (i.e., multilevel factor model), the average log-likelihood increases by 158. Thus the best log-likelihood is achieved using the MFM fitted with MLE-based objective. Also note that a low Frobenius error does not necessarily indicate a better log-likelihood; see Table 1.

To assess whether the log-likelihoods of the two methods are significantly different, we can compare it to the standard deviation of the expectation of these log-likelihoods with respect to the true model. Since we do not have the density of the true model, we assume that the samples are drawn from (2.2). Under this assumption the standard deviation of the average log-likelihood is 2.887; see Appendix B. Therefore, we conclude that the log-likelihood for our method MLE is significantly better.

6.2. Synthetic MFM. We generate samples from a synthetic MFM with $n = 10,000$ features. We create a random hierarchical partition with $L = 6$. Starting with

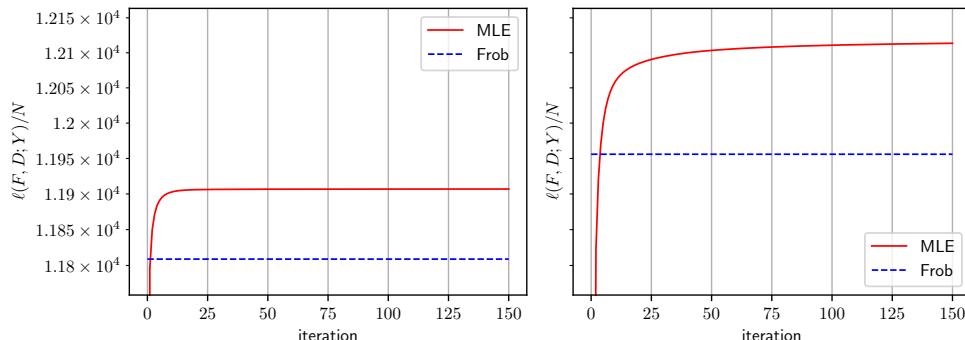


FIG. 5. Log-likelihood during the EM algorithm (red curve) and after Frobenius norm fitting (blue curve), for FM (left) and MFM (right) of the asset covariance matrix. (Figure in color online.)

TABLE 1

Frobenius errors and average log-likelihoods for factors fitted using either the Frobenius norm or MLE-based methods for the asset covariance matrix.

Fit	Model	$\ \hat{\Sigma} - \Sigma\ _F / \ \Sigma\ _F$	$\ell(F, D; Y)/N$
Frob	FM	0.1538	11809
MLE	FM	0.1617	11907
Frob	MFM	0.1648	11956
MLE	MFM	0.8497	12114

a single group, we evenly divide it across levels, resulting in 4, 8, 16, 32, and finally 10,000 groups at the bottom level. Each level is assigned ranks: $r_1 = 10, r_2 = 5, r_3 = 4, r_4 = 3, r_5 = 2, r_6 = 1$, respectively, yielding $s = 174$ unique factors in total. The resulting compression ratio is 200 : 1.

Following this, the coefficients of the structured factor matrix F are sampled from $\mathcal{N}(0, 1)$. Then we sample the noise variance in proportion to the average signal variance maintaining a signal-to-noise ratio (SNR) of 4. This is achieved by sampling D_{ii} uniformly from the interval

$$[0, 2(\mathbf{1}^T \text{diag}(FF^T)/n)/\text{SNR}], \quad i = 1, \dots, n.$$

To evaluate how effectively we can fit the factors using MLE, we use the rank allocation and hierarchical partition from the true model. The model is fitted with $N = 80$ samples and evaluated using expected log-likelihood (based on the density of the true model).

Since in this example we have access to the true model $\Sigma^{\text{true}} = F^{\text{true}} F^{\text{true}^T} + D^{\text{true}}$, we can compute the expected log-likelihood

$$\mathbf{E}(\ell(F, D; y)) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log \det(FF^T + D) - \frac{1}{2} \mathbf{Tr}((FF^T + D)^{-1} \Sigma^{\text{true}}).$$

We compare the average log-likelihood of two fitting approaches based on Frobenius norm and MLE; see Figure 6 and Table 2. Our method outperforms the Frobenius norm-based approach, showing a 284 higher average log-likelihood on the sampled data Y and a 372 greater expected log-likelihood.

We generate 200 samples Y , and for each Y , fit the model with two competing methods. The resulting histograms of expected log-likelihoods $\mathbf{E}(\ell(F, D; y))$ are shown in Figure 7. The histogram of differences $\mathbf{E}(\ell(F^{\text{MLE}}, D^{\text{MLE}}; y)) - \mathbf{E}(\ell(F^{\text{Frob}}, D^{\text{Frob}}; y))$ is displayed in Figure 8. The mean of the differences is 371, with a standard deviation

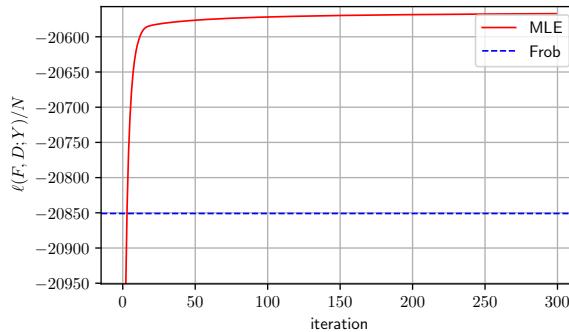


FIG. 6. Log-likelihood during the EM algorithm (red curve) and after Frobenius norm fitting (blue curve) for a single Y in the synthetic example. (Figure in color online.)

TABLE 2

Log-likelihood for models fitted using the Frobenius norm, MLE-based methods, and the true model for a single Y in the synthetic example.

Fit	$\ell(F, D; Y)/N$	$\mathbf{E}(\ell(F, D; y))$
Frob	-20851	-24843
MLE	-20567	-24471
True	-22031	-22068

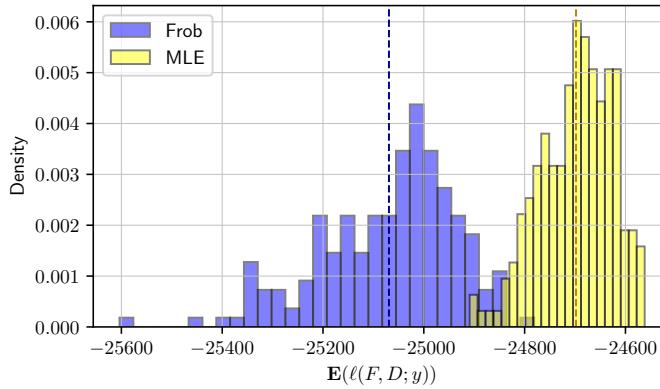


FIG. 7. Histograms of expected log-likelihoods for MLE and Frobenius norm-based fitting methods.

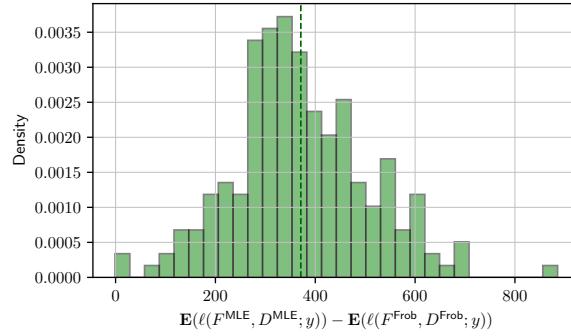


FIG. 8. Histogram of differences in expected log-likelihoods between MLE and Frobenius norm-based fitting methods.

of 136, and for 99.5% of the samples, the difference is positive. Based on these histograms, we conclude that the distribution of the MLE-based method is significantly better than that of Frobenius norm-based method.

6.3. Large-scale single-cell RNA sequencing dataset. Single-cell RNA sequencing generates transcript count matrices that contain gene expression profiles of individual cells. In this section we use the dataset from [18, 40, 45] that contains immune cells from human tissues.

The original dataset contains 329,762 cells with 36,398 genes, collected from 12 donors. Then we follow standard preprocessing steps for single-cell RNA sequencing data [37, 34]. We use Scanpy package [58] for quality control metrics [37] to filter out low-quality cells and uninformative genes. In particular, we filter cells with fewer than 200 genes and filter genes expressed in fewer than 200 cells. We also filter out cells with more than 20% of transcript counts from mitochondrial genes, or which contain more than 2,500 detected gene types. Next, we normalize gene counts per cell, and subsequently apply log-plus-one transformation. To reduce the dimensionality, we selected the top 500 most variable genes. The final feature matrix is standardized across cells and has $n = 280,535$ cells and $N = 500$ genes.

We use a hierarchy with $L = 3$ levels, grouping level $l = 2$ by donor IDs (i.e., making 12 groups in $l = 2$). We set the rank allocation to $r_1 = 12, r_2 = 8, r_3 = 1$.

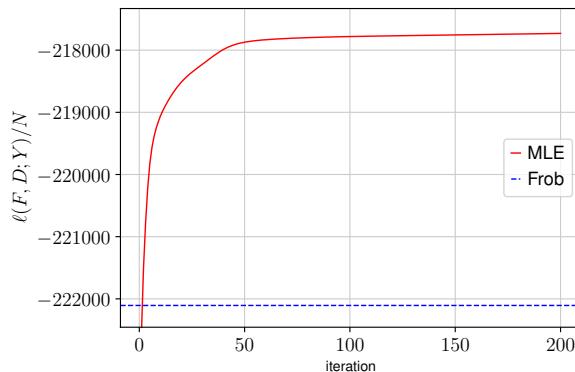


FIG. 9. Log-likelihood during the EM algorithm (red curve) and after Frobenius norm fitting (blue curve) in the single-cell RNA example. (Figure in color online.)

Our method achieves an average log-likelihood of $-217,730$, which is by 4,376 larger than the Frobenius norm-based method with $-222,106$, see Figure 9.

In this experiment we expect the $r_2 = 8$ factors on level $l = 2$ to capture donor-specific correlations, while the factors on level $l = 1$ are to be shared across all the donors and to describe the correlations across the cells. In Figure 10 we plot the factor loadings F_1 , reordered to display the cell types as contiguous groups, using CellTypist labels [18]. The horizontal yellow lines indicate the ranges of the cell types. We can see that some factors are strong predictors for specific cell types. For instance, the second factor (the second column) predicts B cells with large positive loadings, and both CD16+ and CD16- NK cells with large negative loadings. Similarly, the fifth factor is associated with classical monocytes, and macrophages through large positive loadings. Furthermore, $r_1 = 12$ factors on the first level explain on average 68% of their individual variances. In contrast, applying our fitting method to the factor model, i.e., flat hierarchy with $L = 2$ and 12 factors, results in factors that explain on average 62% of their individual variances.

7. Conclusion. In this work, we present a novel and computationally efficient algorithm for fitting multilevel factor model. We introduce a fast implementation of the EM algorithm that uses linear time and space complexities per iteration, making it scalable. This method relies on a novel fast algorithm for computing the inverse and a determinant of the PSD MLR matrix.

We also provide an open-source implementation of our methods that demonstrate their effectiveness on several examples, including the large-scale real-world example. Our MLE-based method consistently outperforms the Frobenius norm-based method. In this paper we assume that the hierarchy, as well as rank allocation, are known. Future research will focus on developing scalable heuristics for finding hierarchy and rank allocation while leveraging our fast factor-fitting method. The challenge is to keep storage and time complexities nearly linear. As demonstrated in Appendix A, minimizing the Frobenius norm-based error approximately maximizes the log-likelihood. Therefore, one promising approach is to adapt the techniques from [43] to avoid forming dense matrices and store all matrices in the factored form. For example, applying partial singular value decomposition to the matrices in factored form enables the rank exchange algorithm to be applied straightforwardly to our setting. However, the incremental hierarchy construction is less straightforward to apply as it requires forming

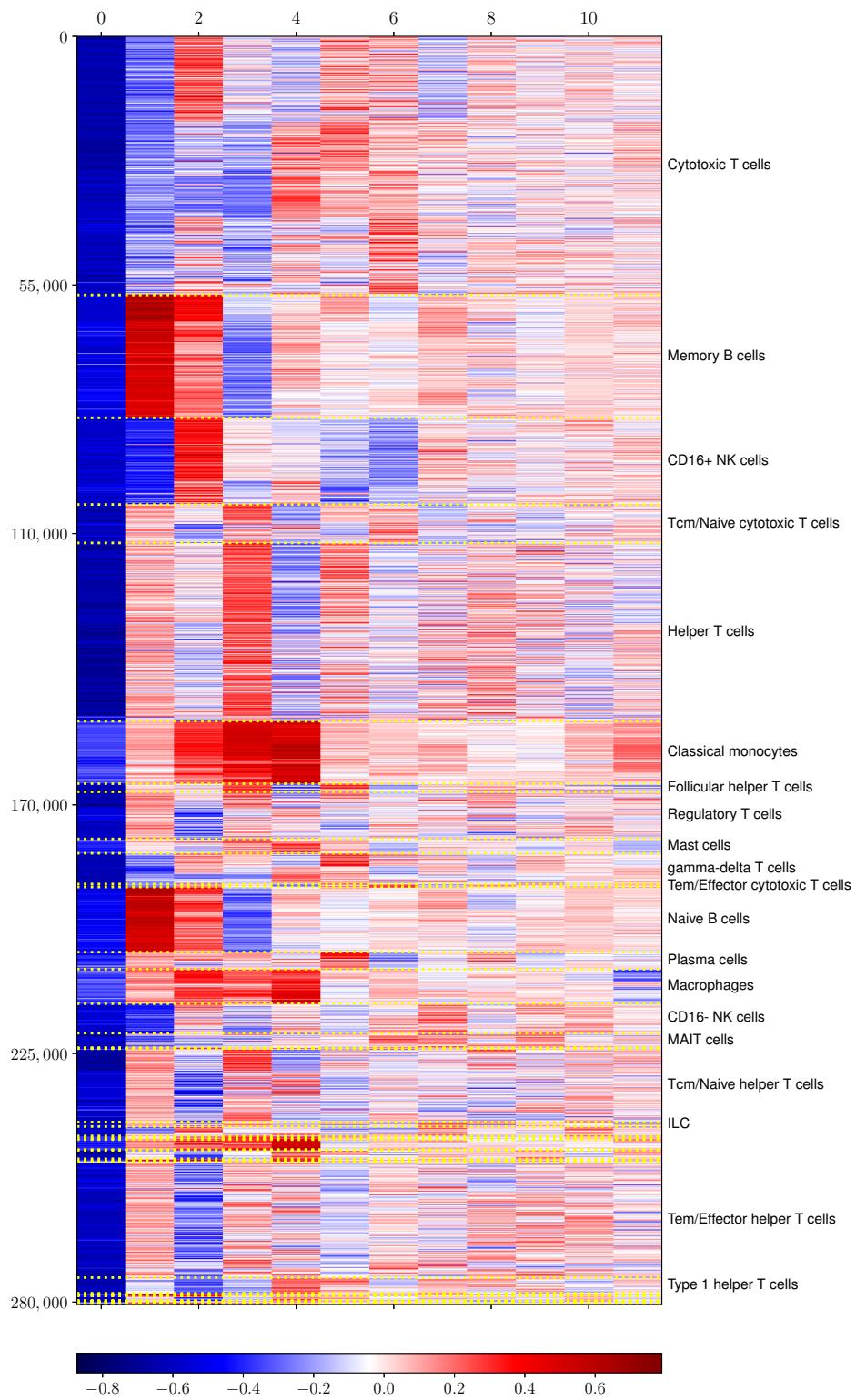


FIG. 10. Factor loading matrix $F_1 \in \mathbf{R}^{n \times r_1}$ with reordered rows to display the cell types as contiguous groups in the single-cell RNA example.

dense residual matrices. Specifically, it is based on the nested spectral dissection [43], which involves computation of the second smallest eigenvalue of Laplacian matrix for graph with adjacency matrix given by the squared valued in the residual matrix. And even though the residual matrix is factored, when we square it elementwise, this structure changes. Future work will focus on the development of spectral clustering methods for the factored residual matrix, while respecting the storage requirements.

Appendix A. Second order approximation of log-likelihood. In this section we explain the intricate relationship between Frobenius norm and MLE-based losses. Let $S = Y^T Y/N$ be a sample covariance matrix. Then the average log-likelihood of N data points for a Gaussian model $y \sim N(0, \Sigma)$ is

$$\frac{1}{N} \ell(\Sigma; Y) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log \det \Sigma - \frac{1}{2} \mathbf{Tr}(\Sigma^{-1} S).$$

We now derive the second-order approximation of the average log-likelihood. We start with finding the second-order approximation of the function $f: \mathbf{S}^n \rightarrow \mathbf{R}$,

$$f(\Sigma) = \log \det \Sigma, \quad \mathbf{dom} f = \mathbf{S}_{++}^n.$$

Following the derivation of [8, section A.4], let $\Delta \in \mathbf{S}^n$ be such that $(\Sigma + \Delta) \in \mathbf{S}_{++}^n$ is close to Σ . We have

$$\begin{aligned} \log \det(\Sigma + \Delta) &= \log \det \left(\Sigma^{1/2} (I + \Sigma^{-1/2} \Delta \Sigma^{-1/2}) \Sigma^{1/2} \right) \\ &= \log \det \Sigma + \log \det(I + \Sigma^{-1/2} \Delta \Sigma^{-1/2}) \\ &= \log \det \Sigma + \sum_{i=1}^1 \log(1 + \lambda_i), \end{aligned}$$

where λ_i is the i th eigenvalue of $\Sigma^{-1/2} \Delta \Sigma^{-1/2}$. Since Δ is small, then λ_i are small. Thus to second order, we have

$$\log(1 + \lambda_i) \approx \lambda_i - \frac{\lambda_i^2}{2}.$$

Combining the above we get

$$\log \det(\Sigma + \Delta) - \log \det \Sigma \approx \sum_{i=1}^n \left(\lambda_i - \frac{\lambda_i^2}{2} \right) = \mathbf{Tr}(\Sigma^{-1} \Delta) - \frac{1}{2} \mathbf{Tr}(\Sigma^{-1} \Delta \Sigma^{-1} \Delta).$$

We used the fact the sum of eigenvalues is the trace, and the eigenvalues of the product of a symmetric matrix with itself are the squares of the eigenvalues of the original matrix, and the cyclic property of trace.

Next, we find the second-order approximation of the function $g: \mathbf{S}^n \rightarrow \mathbf{R}$,

$$g(\Sigma) = \mathbf{Tr}(\Sigma^{-1} S), \quad \mathbf{dom} g = \mathbf{S}_{++}^n.$$

Since $\Sigma \succ 0$, we have

$$\mathbf{Tr}((\Sigma + \Delta)^{-1} S) = \mathbf{Tr} \left(\Sigma^{-1/2} (I + \Sigma^{-1/2} \Delta \Sigma^{-1/2})^{-1} \Sigma^{-1/2} S \right).$$

Recall $\Delta \in \mathbf{S}^n$ is small. Therefore, the spectral radius of $\Sigma^{-1/2} \Delta \Sigma^{-1/2}$ is smaller than 1. Thus using the Neuman series to second order, we have

$$(I + \Sigma^{-1/2} \Delta \Sigma^{-1/2})^{-1} \approx I - \Sigma^{-1/2} \Delta \Sigma^{-1/2} + \Sigma^{-1/2} \Delta \Sigma^{-1} \Delta \Sigma^{-1/2}.$$

Combining the above we get

$$\mathbf{Tr}((\Sigma + \Delta)^{-1}S) \approx \mathbf{Tr}(\Sigma^{-1}S) - \mathbf{Tr}(\Sigma^{-1}\Delta\Sigma^{-1}S) + \mathbf{Tr}(\Sigma^{-1}\Delta\Sigma^{-1}\Delta\Sigma^{-1}S).$$

Using the above derivations, the second-order approximation of the average log-likelihood at S is the quadratic function of Σ given by

$$(A.1) \quad \frac{1}{N}\ell(\Sigma; Y) \approx \frac{1}{N}\ell(S; Y) - \frac{1}{4}\|S^{-1}(S - \Sigma)\|_F^2 = \frac{1}{N}\ell(S; Y) - \frac{1}{4}\|I - S^{-1}\Sigma\|_F^2.$$

Finally, (A.1) gives the relationship between the log-likelihood and Frobenius norm.

Appendix B. Heuristic method for variance estimation. In section 6, we compare the log-likelihoods of models fitted using Frobenius-based loss or MLE. To assess if the difference in the log-likelihoods is significant, we present a heuristic method for estimating the variance of the average log-likelihood. We assume that the empirical data is coming from model (2.2) with parameters F and D . Then the average log-likelihood of N data points is

$$\begin{aligned} \frac{1}{N}\ell(F, D; Y) &= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det(FF^T + D) - \frac{1}{2N}\mathbf{Tr}((FF^T + D)^{-1}Y^TY) \\ &= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det(\Sigma) - \frac{1}{2N}\sum_{i=1}^N y_i^T \Sigma^{-1} y_i. \end{aligned}$$

Since $y_i \sim \mathcal{N}(0, \Sigma)$, then $\Sigma^{-1/2}y_i \sim \mathcal{N}(0, I)$. This implies

$$y_i^T \Sigma^{-1} y_i = (\Sigma^{-1/2}y_i)^T (\Sigma^{-1/2}y_i) \sim \chi^2(n).$$

Let $z_i = \Sigma^{-1/2}y_i$, thus

$$\mathbf{var}\left(\frac{1}{N}\ell(F, D; Y)\right) = \mathbf{var}\left(\frac{1}{2N}\sum_{i=1}^N z_i^T z_i\right) = \frac{1}{4N^2}\sum_{i=1}^N \mathbf{var}(z_i^T z_i) = \frac{n}{2N}.$$

Also the expectation is

$$\begin{aligned} \mathbf{E}\left(\frac{1}{N}\ell(F, D; Y)\right) &= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det(\Sigma) - \frac{1}{2N}\sum_{i=1}^N \mathbf{E}(z_i^T z_i) \\ &= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\det(\Sigma) - \frac{n}{2}. \end{aligned}$$

In the asset covariance example, we have $n = 5000$ and $N = 300$. Therefore, the approximation to the standard deviation is

$$\sqrt{\frac{n}{2N}} \approx 2.887,$$

and the expectation is

$$-\frac{n}{2}(1 + \log(2\pi)) - \frac{1}{2}\log\det(\Sigma) - \frac{n}{2} \approx -7095 - \frac{1}{2}\log\det(\Sigma).$$

Appendix C. Auxiliary derivations.

LEMMA C.1. *Let $B \in \mathbf{R}^{m \times n}$ be a block diagonal matrix with block sizes determined by row and column index partitions I and J , respectively. Similarly, let $C \in \mathbf{R}^{n \times \tilde{n}}$ be a block diagonal matrix with row and column index partitions \tilde{I} and \tilde{J} , respectively. If $J \preceq \tilde{I}$, then product BC is also a block diagonal matrix with column sparsity given by the partition \tilde{J} . Moreover, if $I = J$, then $\text{supp}(BC) = \text{supp}(C)$.*

Proof. Define matrices in terms of blocks explicitly as

$$B = \text{blkdiag}(B_1, \dots, B_p) \in \mathbf{R}^{m \times n}, \quad C = \text{blkdiag}(C_1, \dots, C_{\tilde{p}}) \in \mathbf{R}^{n \times \tilde{n}}.$$

Similarly, define index partitions as

$$\{b_1^J, \dots, b_p^J\} = I, \quad \{b_1^J, \dots, b_{\tilde{p}}^J\} = J, \quad \{\tilde{b}_1^{\tilde{I}}, \dots, \tilde{b}_{\tilde{p}}^{\tilde{I}}\} = \tilde{I}, \quad \{\tilde{b}_1^{\tilde{J}}, \dots, \tilde{b}_{\tilde{p}}^{\tilde{J}}\} = \tilde{J}.$$

For each $\tilde{k} = 1, \dots, \tilde{p}$, index set $\tilde{b}_{\tilde{k}}^{\tilde{I}} \in \tilde{I}$ is refined in J , because $J \preceq \tilde{I}$. Formally, there exist some indices $0 \leq k_1 < k_2 \leq p$ such that

$$\bigcup_{k'=k_1}^{k_2} b_{k'}^J = \tilde{b}_{\tilde{k}}^{\tilde{I}}, \quad b_0^J = \emptyset.$$

Hence, the product BC restricted to the rows indexed by $\bigcup_{k'=k_1}^{k_2} b_{k'}^J$ is nonzero only in the columns indexed by $\tilde{b}_{\tilde{k}}^{\tilde{I}}$.

Therefore, BC is block diagonal with \tilde{p} blocks, where \tilde{k} th block has size $|\bigcup_{k'=k_1}^{k_2} b_{k'}^J| \times |\tilde{b}_{\tilde{k}}^{\tilde{J}}|$ and is given by

$$\text{blkdiag}(B_{k_1}, \dots, B_{k_2})C_{\tilde{k}},$$

and \tilde{J} defines its column partition.

If $I = J$, then

$$\bigcup_{k'=k_1}^{k_2} b_{k'}^I = \bigcup_{k'=k_1}^{k_2} b_{k'}^J = \tilde{b}_{\tilde{k}}^{\tilde{I}}.$$

Therefore, for all $\tilde{k} = 1, \dots, \tilde{p}$ we have

$$\text{supp}(\text{blkdiag}(B_{k_1}, \dots, B_{k_2})C_{\tilde{k}}) = \text{supp}(C_{\tilde{k}}),$$

which implies $\text{supp}(BC) = \text{supp}(C)$. □

LEMMA C.2. *Let $F \in \mathbf{R}^{n \times pr}$ be a block diagonal matrix with p blocks of size $n_k \times r$ for all $k = 1, \dots, p$. Similarly, let $\tilde{F} \in \mathbf{R}^{n \times \tilde{p}r}$ be a block diagonal matrix with \tilde{p} blocks of size $\tilde{n}_k \times \tilde{r}$ for all $k = 1, \dots, \tilde{p}$. If $\mathcal{I}(F) \preceq \mathcal{I}(\tilde{F})$, then product $\tilde{F}^T F$ is also a block diagonal matrix with \tilde{p} blocks and $p\tilde{r}$ nonzero elements. Moreover, computing $\tilde{F}^T F$ takes $O(nr\tilde{r})$.*

Proof. Applying Lemma C.1, $\mathcal{I}(\tilde{F}^T F) = \mathcal{J}(\tilde{F})$. In other words $\tilde{F}^T F$ is a block diagonal matrix with \tilde{p} blocks.

Consider any group k in the partition $\mathcal{I}(\tilde{F})$, it is refined into $c_k \geq 1$ groups in $\mathcal{I}(F)$. Then the diagonal block corresponding to group k in $\mathcal{I}(\tilde{F})$ of size $\tilde{n}_k \times \tilde{r}$ interacts

with c_k respective block diagonal elements in $\mathcal{I}(F)$, forming a block diagonal matrix of size $\tilde{n}_k \times c_k r$. This block diagonal matrix has c_k blocks with column index partition

$$\{\{1, \dots, r\}, \{r+1, \dots, 2r\}, \dots, \{(c_k-1)r, \dots, c_k r\}\}.$$

Thus the matrix-vector multiplication with this matrix requires $O(\tilde{n}_k r)$ operations. Therefore, computing $\tilde{F}^T F$ requires the order of $\sum_{k=1}^{\tilde{p}} \tilde{n}_k r \tilde{r} = nr \tilde{r}$ operations. Finally, the number of nonzero elements in $\tilde{F}^T F$ is $\sum_{k=1}^{\tilde{p}} c_k r \tilde{r} = pr \tilde{r}$. \square

C.1. EM method. This section complements section 4.1. Using the joint distribution (y, z) and conditional distribution $z_i | y_i, F^0, D^0$ defined in section 4.1, we get

$$\begin{aligned} \sum_{i=1}^N \mathbf{E}(z_i z_i^T | y_i, F^0, D^0) &= \sum_{i=1}^N \mathbf{cov}((z_i, z_i) | y_i, F^0, D^0) \\ &\quad + \mathbf{E}(z_i | y_i, F^0, D^0) \mathbf{E}(z_i | y_i, F^0, D^0)^T \\ &= N(I_s - F^{0T}(\Sigma^0)^{-1}F^0) + F^{0T}(\Sigma^0)^{-1}Y^T Y(\Sigma^0)^{-1}F^0. \end{aligned}$$

We can now derive the expression for (4.1),

$$\begin{aligned} Q(F, D; F^0, D^0) &= \mathbf{E}(\ell(F, D; Y, Z) | Y, F^0, D^0) \\ &= -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log \det D - \frac{1}{2} \sum_{i=1}^N \mathbf{Tr}(\mathbf{E}(z_i z_i^T | y_i, F^0, D^0)) \\ &\quad - \frac{1}{2} \sum_{i=1}^N \mathbf{Tr}(D^{-1}\{(y_i y_i^T - 2F\mathbf{E}(z_i | y_i, F^0, D^0) y_i^T) \\ &\quad + F\mathbf{E}(z_i z_i^T | y_i, F^0, D^0) F^T\}) \\ &= -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log \det D \\ &\quad - \frac{1}{2} \mathbf{Tr}\left(\underbrace{N(I_s - F^{0T}(\Sigma^0)^{-1}F^0) + F^{0T}(\Sigma^0)^{-1}Y^T Y(\Sigma^0)^{-1}F^0}_W\right) \\ &\quad - \frac{1}{2} \mathbf{Tr}\left(D^{-1}\left\{Y^T Y - 2F\underbrace{F^{0T}(\Sigma^0)^{-1}Y^T Y}_V\right\}\right. \\ &\quad \left. + F\left(\underbrace{N(I_s - F^{0T}(\Sigma^0)^{-1}F^0) + F^{0T}(\Sigma^0)^{-1}Y^T Y(\Sigma^0)^{-1}F^0}_W\right) F^T\right). \end{aligned}$$

C.2. Inverse computation. The SMW matrix identity implies

$$\begin{aligned} ((F_{l+} F_{l+}^T + D)^{-1} &= F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} - \underbrace{(F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} F_l}_{=M_0} \\ &\quad (I_{p_l r_l} + F_l^T \underbrace{(F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} F_l}_{=M_0})^{-1} \underbrace{F_l^T (F_{(l+1)+} F_{(l+1)+}^T + D)^{-1}}_{=M_0^T} \\ &= (F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} - M_0 (I_{p_l r_l} + F_l^T M_0)^{-1} M_0^T \\ &= (F_{(l+1)+} F_{(l+1)+}^T + D)^{-1} - H_l H_l^T. \end{aligned}$$

Therefore, we have

$$\Sigma_{l+}^{-1} = \Sigma_{(l+1)+}^{-1} - H_l H_l^T.$$

Appendix D. Cholesky factorization. In this section we present a Cholesky factorization for the expanded matrix and show that the Cholesky factor has the same sparsity as its inverse.

D.1. Schur complement. Finding the inverse of Σ amounts to solving the linear system

$$(FF^T + D)X = DX + F_{L-1}F_{L-1}^T X + \cdots + F_1F_1^T X = I_n,$$

which is equivalent to solving the expanded system of equations

$$(D.1) \quad \begin{bmatrix} D & F_{L-1} & \cdots & F_1 \\ F_{L-1}^T & -I_{p_{L-1}r_{L-1}} & & \\ \vdots & & \ddots & \\ F_1^T & & & -I_{p_1r_1} \end{bmatrix} \begin{bmatrix} X \\ Y_{L-1} \\ \vdots \\ Y_1 \end{bmatrix} = \begin{bmatrix} I_n \\ \mathbf{0} \end{bmatrix}.$$

Denote the expanded matrix (D.1) by $E \in \mathbf{S}^{n+s}$. Note that E has the block sparsity pattern of the upward-left arrow.

Block Gaussian elimination on the matrix (D.1) leads to an LDL decomposition

$$(D.2) \quad E = \begin{bmatrix} I_n & -F_{L-1} & \cdots & -F_1 \\ & I_{p_{L-1}r_{L-1}} & & \\ & & \ddots & \\ & & & I_{p_1r_1} \end{bmatrix} \begin{bmatrix} FF^T + D & & \\ & -I_s & \\ & & \ddots \end{bmatrix} \begin{bmatrix} I_n & & & \\ -F_{L-1}^T & I_{p_{L-1}r_{L-1}} & & \\ \vdots & & \ddots & \\ -F_1^T & & & I_{p_1r_1} \end{bmatrix}.$$

And $FF^T + D$ is Schur complement of the block $-I_s$ of the matrix E .

D.2. Recursive Cholesky factorization. Let $s_{l+} = \sum_{l'=l}^{L-1} p_{l'}r_{l'}$ for all $l = 1, \dots, L-1$. Define E_l as the top left $(n + s_{l+}) \times (n + s_{l+})$ submatrix of E , i.e.,

$$E_l = \begin{bmatrix} D & F_{L-1} & \cdots & F_l \\ F_{L-1}^T & -I_{p_{L-1}r_{L-1}} & & \\ \vdots & & \ddots & \\ F_l^T & & & -I_{p_l r_l} \end{bmatrix} \in \mathbf{S}^{n+s_{l+}}.$$

We find the factors of E by recursively factorizing E_{L-1}, \dots, E_1 using the relation

$$E_l = \begin{bmatrix} E_{l+1} & \begin{bmatrix} F_l \\ \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} F_l^T & \mathbf{0} \end{bmatrix} & -I_{p_l r_l} \end{bmatrix}.$$

D.2.1. Sparsity patterns. The block Gaussian elimination on E_l gives the following factorization:

$$(D.3) \quad \begin{bmatrix} I_{n+s_{(l+1)+}} & E_{l+1} \\ \begin{bmatrix} F_l^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} & I_{p_l r_l} \end{bmatrix} \begin{bmatrix} E_{l+1} & \\ -\left(I_{p_l r_l} + \begin{bmatrix} F_l^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} F_l \\ \mathbf{0} \end{bmatrix} \right) & \end{bmatrix} \begin{bmatrix} I_{n+s_{(l+1)+}} & \\ \begin{bmatrix} F_l^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} & I_{p_l r_l} \end{bmatrix}^T.$$

Submatrices of E . In Lemma D.1 we show the sparsity pattern of matrices necessary for Cholesky factorization.

LEMMA D.1. *Let F and D be factors of PSD MLR Σ , and let E be its expanded matrix. Then for all $l = 1, \dots, L-1$, we have*

$$\begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1} \begin{bmatrix} I_n \\ \mathbf{0} \end{bmatrix} = F_{(l-1)-}^T \Sigma_{l+}^{-1},$$

and

$$\text{supp}\left(\begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1}\right) = \text{supp}(F_{(l-1)-}^T [D \ F_{L-1} \ \cdots \ F_l]).$$

Proof. It is easy to check that these properties hold for the base case, i.e., $E_L = D$. Now we demonstrate the properties of E_l for all $l = L-1, \dots, 1$.

Assume that

$$(D.4) \quad \begin{bmatrix} F_{l-}^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} I_n \\ \mathbf{0} \end{bmatrix} = F_{l-}^T \Sigma_{(l+1)+}^{-1},$$

and

$$(D.5) \quad \text{supp}(\begin{bmatrix} F_{l-}^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1}) = \text{supp}(F_{l-}^T [D \ F_{L-1} \ \cdots \ F_{l+1}]).$$

Note that the (negative) bottom block in the block diagonal matrix in (D.3) is equal to

$$(D.6) \quad I_{p_l r_l} + F_l^T \Sigma_{(l+1)+}^{-1} F_l \succ 0.$$

Recall from section 5.1.1, that this matrix is block diagonal, consisting of p_l blocks, each of which is of size $r_l \times r_l$. Let $R_l V_l R_l^T$ be the Cholesky factorization of (D.6).

Using the relation from (D.3), we can express the inverse as

$$E_l^{-1} = \begin{bmatrix} I_{n+s_{(l+1)+}} & \\ [-F_l^T \mathbf{0}] E_{l+1}^{-1} & I_{p_l r_l} \end{bmatrix}^T \begin{bmatrix} E_{l+1}^{-1} & \\ & -(R_l V_l R_l^T)^{-1} \end{bmatrix} \begin{bmatrix} I_{n+s_{(l+1)+}} & \\ [-F_l^T \mathbf{0}] E_{l+1}^{-1} & I_{p_l r_l} \end{bmatrix}.$$

Then the matrix $\begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1}$ is equal to

$$\begin{bmatrix} F_{(l-1)-}^T & \begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} -F_l \\ \mathbf{0} \end{bmatrix} \end{bmatrix} \begin{bmatrix} E_{l+1}^{-1} & \\ [(R_l V_l R_l^T)^{-1} F_l^T \mathbf{0}] E_{l+1}^{-1} & -(R_l V_l R_l^T)^{-1} \end{bmatrix},$$

which simplifies to

$$(D.7) \quad \begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \left[\left(I_{n+s_{(l+1)+}} - \begin{bmatrix} F_l \\ \mathbf{0} \end{bmatrix} (R_l V_l R_l^T)^{-1} [F_l^T \mathbf{0}] E_{l+1}^{-1} \right) \begin{bmatrix} F_l \\ \mathbf{0} \end{bmatrix} (R_l V_l R_l^T)^{-1} \right].$$

Combining (D.7), (D.4), and SMW (5.1) we get

$$\begin{aligned} \begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1} \begin{bmatrix} I_n \\ \mathbf{0} \end{bmatrix} &= F_{(l-1)-}^T \left(\Sigma_{(l+1)+}^{-1} - \Sigma_{(l+1)+}^{-1} F_l (R_l V_l R_l^T)^{-1} F_l^T \Sigma_{(l+1)+}^{-1} \right) \\ &= F_{(l-1)-}^T \Sigma_{l+}^{-1}. \end{aligned}$$

The coefficients of matrix

$$(D.8) \quad \begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1} \begin{bmatrix} \mathbf{0} \\ I_{p_l r_l} \end{bmatrix} = F_{(l-1)-}^T \Sigma_{(l+1)+}^{-1} F_l (R_l V_l R_l^T)^{-1} = M_3^T$$

are obtained during the inverse computation; see section 5.1.2. Furthermore, we have $\text{supp}(M_3^T) = \text{supp}(F_{(l-1)-}^T F_l)$.

Using assumption (D.5), for any $\tilde{l} \geq l + 1$ we have

$$\text{supp} \left(\begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} \mathbf{0} \\ I_{p_l r_l} \\ \mathbf{0} \end{bmatrix} \right) = \text{supp}(F_{(l-1)-}^T F_{\tilde{l}}).$$

Similarly, it holds that

$$\text{supp} \left(M_3^T \begin{bmatrix} F_l^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} \mathbf{0} \\ I_{p_l r_l} \\ \mathbf{0} \end{bmatrix} \right) = \text{supp}(F_{(l-1)-}^T F_l F_l^T F_{\tilde{l}}).$$

By Lemma C.2, for any $l_1 \leq l_2$ product $F_{l_1}^T F_{l_2}$ has $p_{l_2} r_{l_1} r_{l_2}$ nonzero entries, $\mathcal{I}(F_{l_1}^T F_{l_2}) = \mathcal{J}(F_{l_1})$, and can be computed $O(n r_{l_1} r_{l_2})$. By Lemma C.1, for any $l' \leq l - 1$ and $\tilde{l} \geq l + 1$, we have $\mathcal{I}(F_l F_{l'}^T) = \mathcal{J}(F_l F_{l'}^T) \preceq \mathcal{I}(F_{l'})$. This implies $\text{supp}(F_{l'}^T F_l F_l^T) = \text{supp}(F_{l'}^T)$, and consequently, $\text{supp}(F_{l'}^T F_l F_l^T F_{\tilde{l}}) = \text{supp}(F_{l'}^T F_{\tilde{l}})$. Combining this result with (D.7), for any $\tilde{l} \geq l + 1$ matrix

(D.9)

$$\begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1} \begin{bmatrix} \mathbf{0} \\ I_{p_l r_l} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} \mathbf{0} \\ I_{p_l r_l} \\ \mathbf{0} \end{bmatrix} - M_3^T \begin{bmatrix} F_l^T & \mathbf{0} \end{bmatrix} E_{l+1}^{-1} \begin{bmatrix} \mathbf{0} \\ I_{p_l r_l} \\ \mathbf{0} \end{bmatrix},$$

has the sparsity of $\text{supp}(F_{(l-1)-}^T F_l F_l^T F_{\tilde{l}}) = \text{supp}(F_{(l-1)-}^T F_{\tilde{l}})$. This implies

$$\text{supp} \left(\begin{bmatrix} F_{(l-1)-}^T & \mathbf{0} \end{bmatrix} E_l^{-1} \right) = \text{supp} \left(F_{(l-1)-}^T [D \ F_{L-1} \ \cdots \ F_l] \right).$$

The final result follows by induction. \square

Cholesky factors. Let the Cholesky factorization of a symmetric matrix E_{l+1} be given by

$$E_{l+1} = L^{(l+1)} D^{(l+1)} L^{(l+1)T}.$$

Using (D.3) we have

$$\begin{aligned} E_l &= \begin{bmatrix} I_{n+s_{(l+1)+}} & \\ [F_l^T \ \mathbf{0}] E_{l+1}^{-1} & I_{p_l r_l} \end{bmatrix} \begin{bmatrix} E_{l+1} & \\ -R_l V_l R_l^T & \end{bmatrix} \begin{bmatrix} I_{n+s_{(l+1)+}} & \\ [F_l^T \ \mathbf{0}] E_{l+1}^{-1} & I_{p_l r_l} \end{bmatrix}^T \\ (D.10) \quad &= \begin{bmatrix} L^{(l+1)} & \\ [F_l^T \ \mathbf{0}] E_{l+1}^{-1} L^{(l+1)} & R_l \end{bmatrix} \begin{bmatrix} D^{(l+1)} & \\ -V_l & \end{bmatrix} \begin{bmatrix} L^{(l+1)} & \\ [F_l^T \ \mathbf{0}] E_{l+1}^{-1} L^{(l+1)} & R_l \end{bmatrix}^T. \end{aligned}$$

Note that the matrix R_l is a block diagonal matrix consisting of p_l blocks, each of which is a lower triangular matrix of size $r_l \times r_l$ (i.e., $\mathcal{I}(R_l) = \mathcal{J}(R_l) \preceq \mathcal{I}(F_l)$); see section 5.1.1. Thus from (D.10), Cholesky factors of E_l are

$$(D.11) \quad L^{(l)} = \begin{bmatrix} L^{(l+1)} & \\ [F_l^T \ \mathbf{0}] (D^{(l+1)} L^{(l+1)T})^{-1} & R_l \end{bmatrix}, \quad D^{(l)} = \begin{bmatrix} D^{(l+1)} & \\ -V_l & \end{bmatrix}.$$

Then we also have

$$(L^{(l)})^{-1} = \begin{bmatrix} (L^{(l+1)})^{-1} & \\ -R_l^{-1} [F_l^T \ \mathbf{0}] E_{l+1}^{-1} & R_l^{-1} \end{bmatrix}.$$

Lemma D.2 establishes the sparsity pattern of Cholesky factors, and, in particular, $\text{supp}(L^{(l)}) = \text{supp}((L^{(l)})^{-1})$.

LEMMA D.2. *Let F and D be factors of PSD MLR Σ , and let E be its expanded matrix. Then for all $l = L, \dots, 1$ and $\tilde{l} = L, \dots, l$, we have*

$$\begin{aligned}\text{supp}(F_{\tilde{l}}^T [D \ F_{L-1} \ \cdots \ F_{l+1}]) &= \text{supp}([\mathbf{0} \ I_{p_{\tilde{l}}r_{\tilde{l}}} \ \mathbf{0}] (L^{(l+1)})^{-1}) \\ &= \text{supp}([\mathbf{0} \ I_{p_{\tilde{l}}r_{\tilde{l}}} \ \mathbf{0}] L^{(l+1)}).\end{aligned}$$

Proof. Assume for all $\tilde{l} = L, \dots, l+1$ we have

$$\begin{aligned}\text{supp}(F_{\tilde{l}}^T [D \ F_{L-1} \ \cdots \ F_{l+1}]) &= \text{supp}([\mathbf{0} \ I_{p_{\tilde{l}}r_{\tilde{l}}} \ \mathbf{0}] (L^{(l+1)})^{-1}) \\ &= \text{supp}([\mathbf{0} \ I_{p_{\tilde{l}}r_{\tilde{l}}} \ \mathbf{0}] L^{(l+1)}).\end{aligned}$$

Using (D.11) it suffices to show the sparsity of the bottom block of $L^{(l)}$ and $(L^{(l)})^{-1}$ of size $p_{\tilde{l}}r_{\tilde{l}} \times (n + s_{l+1})$. The assumptions above imply

$$\text{supp}([\mathbf{I}_n \ \mathbf{0}] (L^{(l+1)})^{-1}) = \text{supp}(D [D \ F_{L-1} \ \cdots \ F_{l+1}]),$$

thus since $D^{(l+1)}$ is diagonal we get

$$\text{supp}([F_{\tilde{l}}^T \ \mathbf{0}] (D^{(l+1)} L^{(l+1)T})^{-1}) = \text{supp}(F_{\tilde{l}}^T [D \ F_{L-1} \ \cdots \ F_{l+1}]).$$

Combining Lemma D.1 with $\text{supp}(R_{\tilde{l}}^{-1} F_{\tilde{l}}^T) = \text{supp}(F_{\tilde{l}}^T)$, it follows that

$$\text{supp}(R_{\tilde{l}}^{-1} [F_{\tilde{l}}^T \ \mathbf{0}] E_{l+1}^{-1}) = \text{supp}(F_{\tilde{l}}^T [D \ F_{L-1} \ \cdots \ F_{l+1}]).$$

Since $\text{supp}(R_{\tilde{l}}) = \text{supp}(R_{\tilde{l}}^{-1}) \subseteq \text{supp}(F_{\tilde{l}}^T F_{\tilde{l}})$, the following holds

$$\begin{aligned}\text{supp}(F_{\tilde{l}}^T [D \ F_{L-1} \ \cdots \ F_{l+1}]) &= \text{supp}([\mathbf{0} \ I_{p_{\tilde{l}}r_{\tilde{l}}}] L^{(l)}) \\ &= \text{supp}([\mathbf{0} \ I_{p_{\tilde{l}}r_{\tilde{l}}}] (L^{(l)})^{-1}).\end{aligned}$$

Combining these results with (D.11) we conclude that $\text{supp}(L^{(l)}) = \text{supp}((L^{(l)})^{-1})$.

Evidently, for the base case, $L^{(L)} = \mathbf{I}_n$ and $D^{(L)} = D$, these properties hold. By induction we showed $\text{supp}(L^{(1)}) = \text{supp}((L^{(1)})^{-1})$. \square

D.3. Efficient computation.

Recurrent term. Using (D.11) we recursively compute

$$[F_{(l-1)-}^T \ \mathbf{0}] (D^{(l)} L^{(l)T})^{-1} = [F_{(l-1)-}^T \ \mathbf{0}] \left[(D^{(l+1)} L^{(l+1)T})^{-1} \ E_{l+1}^{-1} \begin{bmatrix} F_l \\ \mathbf{0} \end{bmatrix} (V_l R_l^T)^{-1} \right].$$

Lemma D.1 implies

$$[F_{(l-1)-}^T \ \mathbf{0}] E_{l+1}^{-1} \begin{bmatrix} F_l \\ \mathbf{0} \end{bmatrix} (V_l R_l^T)^{-1} = M_3^T R_l.$$

The product $M_3^T R_l$ requires $\sum_{l'=1}^{l-1} O(p_{l'}r_l^2 r_{l'}) = O(p_{l'}r_l^2)$ operations. Thus we get identity

$$(D.12) \quad [F_{(l-1)-}^T \ \mathbf{0}] (D^{(l)} L^{(l)T})^{-1} = \left[[F_{(l-1)-}^T \ \mathbf{0}] (D^{(l+1)} L^{(l+1)T})^{-1} \ M_3^T R_l \right].$$

This indicates that constructing a recurrent term at the next level only requires computing $M_3^T R_l$.

Moreover, by Lemma D.2 the sparsity is

$$\text{supp} \left([F_{(l-1)-}^T \ \mathbf{0}] (D^{(l)} L^{(l)T})^{-1} \right) = \text{supp} \left(F_{(l-1)-}^T [D \ F_{L-1} \ \cdots \ F_l] \right).$$

Method. We now describe the algorithm for computing Cholesky factorization that recursively computes Cholesky factors of E_L, E_{L-1}, \dots, E_1 . This process is accompanied by the recursive computation of coefficients in Σ^{-1} ; see section 5.1. We include additional time and space complexities beyond those discussed in section 5.1.

We start with $L^{(L)} = I_n$ and $D^{(L)} = D$. Then for each level $l = L-1, \dots, 1$ repeat the following steps.

1. Compute Cholesky decomposition of (D.6), $R_l V_l R_l^T$, in $O(p_l r_l^3)$, and store its $O(p_l r_l^2)$ coefficients. The coefficients of (D.6) and its inverse are obtained in section 5.1.
2. Form $L^{(l)}$ and $D^{(l)}$ using stored coefficients of $[F_{l-}^T \ 0] (D^{(l+1)} L^{(l+1)})^{-1}$ according to (D.11).
3. Form $[F_{(l-1)-}^T \ 0] (D^{(l)} L^{(l)})^{-1}$ using (D.12). This requires computing $M_3^T R_l$ with $O(p_l r_l r_l^2)$ operations and $\sum_{l'=1}^{l-1} r_{l'}(n + \sum_{l=l}^{l-1} p_l r_l)$ coefficients. We use $\sum_{l'=1}^{l-1} p_l r_{l'} r_l$ coefficients of M_3 from section 5.1.

Cholesky factor of E , i.e., lower triangular matrix $L^{(1)}$, has less than

$$n + \sum_{l=L-1}^1 r_l(n + p_{L-1} r_{L-1} + \dots + p_l r_l) \leq nr + p_{L-1} r^2$$

nonzero entries. The total cost for computing the factors is

$$O(nr) + \sum_{l=L-1}^1 O(p_l r_l^3 + p_l r_l r_l^2) = O(nr + p_{L-1} r^3).$$

D.4. Determinant. Using the Cholesky decomposition of E we can easily compute the determinant of MLR covariance matrix Σ . Specifically, using (D.2) we have

$$\det(E) = \det(FF^T + D)(-1)^s,$$

since the eigenvalues of a triangular matrix are exactly its diagonal entries and because the determinant is a multiplicative map. Alternatively, using Cholesky decomposition, $E = L^{(1)} D^{(1)} L^{(1)}^T$, we have

$$\det(E) = \det(L^{(1)})^2 \det(D^{(1)}) = \det(D^{(1)}).$$

Since

$$\det(D^{(1)}) = (-1)^s \det(D) \prod_{l=1}^{L-1} \det(V_l),$$

we obtain

$$\det(FF^T + D) = \prod_{i=1}^{n+s} |D_{ii}^{(1)}|, \quad \log \det(FF^T + D) = \sum_{i=1}^{n+s} \log |D_{ii}^{(1)}|.$$

Remark D.3. The $\det(\Sigma)$ can be computed at no additional cost while recursively computing the coefficients in Σ^{-1} ; see section 5.1. For every $l = L-1, \dots, 1$ we compute the eigendecomposition of the matrix

$$R_l V_l R_l^T = I_{p_l r_l} + F_l^T \Sigma_{(l+1)+}^{-1} F_l = Q_l \Lambda_l Q_l^T,$$

which implies

$$\det(V_l) = \det(R_l V_l R_l^T) = \det(Q_l \Lambda_l Q_l^T) = \det(\Lambda_l).$$

Therefore,

$$(D.13) \quad \det(FF^T + D) = \det(D) \prod_{l=1}^{L-1} \det(\Lambda_l).$$

Note that, alternatively, determinant (D.13) can be interpreted as relying on the recursive application of the matrix determinant lemma, which states that if $A \in \mathbf{R}^{n \times n}$ is invertible, then for any $U, V \in \mathbf{R}^{n \times p}$, it holds that

$$\det(A + UV^T) = \det(A) \det(I_p + V^T A^{-1} U).$$

Appendix E. Factor model with linear covariates. In this section we show how to apply our fitting method to the factor model with linear covariates. Suppose we have samples $y_1, \dots, y_N \in \mathbf{R}^n$ along with covariates $x_1, \dots, x_N \in \mathbf{R}^p$. Then the factor model with the covariates is given by

$$y_i = Bx_i + Fz_i + e_i,$$

where $B \in \mathbf{R}^{n \times p}$ is a matrix with regression coefficients. Define

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \in \mathbf{R}^{N \times p}, \quad \tilde{Z} = [X \ Z] \in \mathbf{R}^{N \times (p+s)}, \quad \tilde{F} = [B \ F] \in \mathbf{R}^{n \times (p+s)}.$$

Similarly to steps in section 4.1, we have $y_i \sim \mathcal{N}(Bx_i, \Sigma)$, $z_i \sim \mathcal{N}(0, I_s)$, and the conditional distribution $(z_i | y_i, x_i, \tilde{F}^0, D^0)$ is Gaussian,

$$\mathcal{N}\left(F^{0T}(\Sigma^0)^{-1}(y_i - B^0 x_i), I_s - F^{0T}(\Sigma^0)^{-1}F^0\right).$$

Since the log-likelihood of complete data (Y, X, Z) is

$$\ell(\tilde{F}, D; Y, \tilde{Z}) = -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log \det D - \frac{1}{2} \|(Y - \tilde{Z}\tilde{F}^T)D^{-1/2}\|_F^2 - \frac{1}{2} \|Z\|_F^2,$$

we have

$$\begin{aligned} Q(\tilde{F}, D; \tilde{F}^0, D^0) &= \mathbf{E}\left(\ell(\tilde{F}, D; Y, \tilde{Z}) | Y, X, \tilde{F}^0, D^0\right) \\ &= -\frac{(n+s)N}{2} \log(2\pi) - \frac{N}{2} \log \det D - \frac{1}{2} \mathbf{Tr}(\tilde{W}) \\ &\quad - \frac{1}{2} \mathbf{Tr}\left(D^{-1} \left\{ Y^T Y - 2\tilde{F} \begin{bmatrix} X^T Y \\ \tilde{V} Y \end{bmatrix} + \tilde{F} \begin{bmatrix} X^T X & X^T \tilde{V}^T \\ \tilde{V} X & \tilde{W} \end{bmatrix} \tilde{F}^T \right\}\right), \end{aligned}$$

where the matrices \tilde{V} and \tilde{W} are defined as

$$\begin{aligned} \tilde{V} &= F^{0T}(\Sigma^0)^{-1}(Y - XB^{0T})^T, \\ \tilde{W} &= \sum_{i=1}^N \mathbf{E}\left(z_i z_i^T | y_i, x_i, \tilde{F}^0, D^0\right) = N(I_s - F^{0T}(\Sigma^0)^{-1}F^0) + \tilde{V}\tilde{V}^T. \end{aligned}$$

Similarly to (4.2), $Q(\tilde{F}, D; \tilde{F}^0, D^0)$ is separable across the rows of \tilde{F} ; therefore, our fast EM method can be applied directly.

Appendix F. Product of MLR matrices. In this section we show that the product of two MLR matrices, A with MLR-rank r and A' with MLR-rank r' , sharing the same symmetric hierarchical partition, is also an MLR matrix with the same hierarchical partition and an MLR-rank of $(r + r')$. We also show that it can be computed using $O(n \max\{r, r'\}^2)$ operations.

Since the hierarchical partition is symmetric, without loss of generality assume A and A' are contiguous MLR. Define

$$A_{l+} = A_l + \cdots + A_L,$$

then it is easy to check that

$$\begin{aligned} AA' &= \left(\sum_{l=1}^L A_l \right) \left(\sum_{l=1}^L A'_l \right) \\ (F.1) \quad &= \sum_{l=1}^{L-1} (A_l A'_{l+} + A_{(l+1)+} A'_l) + A_L A'_L. \end{aligned}$$

We now show that each term in the sum above can be decomposed into a product of block diagonal matrices, which are the factors of matrix AA' on level l .

Recall the notation from [43],

$$A_l = \text{blkdiag}(B_{l,1} C_{l,1}^T, \dots, B_{l,p_l} C_{l,p_l}^T), \quad A'_l = \text{blkdiag}(B'_{l,1} C'_{l,1}^T, \dots, B'_{l,p_l} C'_{l,p_l}^T),$$

where $B_{l,k}, B'_{l,k}, C_{l,k}, C'_{l,k} \in \mathbf{R}^{n_{l,k} \times r_l}$, for all $k = 1, \dots, p_l$, and $l = 1, \dots, L$.

Since for all levels $l \leq \tilde{l}$, $\text{supp}(A'_{\tilde{l}}) \subseteq \text{supp}(A_l)$, it follows that $\text{supp}(A_l A'_{\tilde{l}}) = \text{supp}(A_l)$; see section 5.1.1. Thus we also have $\text{supp}(A_l A'_{l+}) = \text{supp}(A_l)$. Similarly, $\text{supp}(A_{(l+1)+} A'_l) = \text{supp}(A'_l)$.

Consider levels $l \leq \tilde{l}$. Let the k th group on level l (for $k = 1, \dots, p_l$) be partitioned into $p_{l,k,\tilde{l}}$ groups on level \tilde{l} , indexed by $\tilde{k}, \dots, \tilde{k} + p_{l,k,\tilde{l}} - 1$. Let the partition of $C_{l,k}$ into $p_{l,k,\tilde{l}}$ blocks for each group be defined as follows:

$$C_{l,k} = \begin{bmatrix} C_{l,k,1} \\ \vdots \\ C_{l,k,p_{l,k,\tilde{l}}} \end{bmatrix}.$$

Then the k th diagonal block of the $A_l A'_{\tilde{l}}$ is given by

$$\begin{aligned} (A_l A'_{\tilde{l}})_k &= B_{l,k} C_{l,k}^T \text{blkdiag} \left(B'_{\tilde{l},\tilde{k}} C'_{\tilde{l},\tilde{k}}^T, \dots, B'_{\tilde{l},p_{l,k,\tilde{l}}} C'_{\tilde{l},p_{l,k,\tilde{l}}}^T \right) \\ &= B_{l,k} \begin{bmatrix} C_{l,k,1}^T B'_{\tilde{l},\tilde{k}} C'_{\tilde{l},\tilde{k}}^T & \cdots & C_{l,k,p_{l,k,\tilde{l}}}^T B'_{\tilde{l},\tilde{k}+p_{l,k,\tilde{l}}-1} C'_{\tilde{l},\tilde{k}+p_{l,k,\tilde{l}}-1}^T \end{bmatrix} \\ &= B_{l,k} \bar{C}_{l,k,\tilde{l}}^T, \end{aligned}$$

where $B_{l,k}, \bar{C}_{l,k,\tilde{l}} \in \mathbf{R}^{n_{l,k} \times r_l}$ are left and right factors of $(A_l A'_{\tilde{l}})_k$. Computing

$$(C_{l,k,j}^T B'_{\tilde{l},j}) C'_{\tilde{l},\tilde{k}+j-1}^T \in \mathbf{R}^{r_l \times n_{\tilde{l},\tilde{k}+j-1}}, \quad j = 1, \dots, p_{l,k,\tilde{l}},$$

where $C_{l,k,j} \in \mathbf{R}^{n_{l,k+j-1} \times r_l}$ and $C'_{\tilde{l},\tilde{k}+j-1}, B'_{\tilde{l},\tilde{k}+j-1} \in \mathbf{R}^{n_{\tilde{l},\tilde{k}+j-1} \times r_{\tilde{l}}}$, takes $O(n_{\tilde{l},\tilde{k}+j-1} r_l r_{\tilde{l}})$ operations. Computing all coefficients of the right factor of $A_l A'_{\tilde{l}}$ requires

$$\sum_{\tilde{k}=1}^{p_{\tilde{l}}} \sum_{j=1}^{p_{l,k,\tilde{l}}} O(n_{\tilde{l},\tilde{k}+j-1} r_l r_{\tilde{l}}) = O(nr_l r_{\tilde{l}}).$$

Therefore, we have the following factorization:

$$A_l A'_{\tilde{l}} = \text{blkdiag}(B_{l,1} \bar{C}_{l,1,\tilde{l}}^T, \dots, B_{l,p_l} \bar{C}_{l,p_l,\tilde{l}}^T) = B_l \bar{C}_{l,\tilde{l}}^T,$$

where $\text{supp}(\bar{C}_{l,\tilde{l}}) = \text{supp}(B_l)$.

Similarly, for levels $l \geq \tilde{l}$, we have

$$A_l A'_{\tilde{l}} = \text{blkdiag}(\bar{B}_{\tilde{l},1,l} C_{\tilde{l},1,l}^T, \dots, \bar{B}_{\tilde{l},p_{\tilde{l}},l} C_{\tilde{l},p_{\tilde{l}},l}^T) = \bar{B}_{\tilde{l},l} C_{\tilde{l}}^T,$$

where $\text{supp}(\bar{B}_{\tilde{l},l}) = \text{supp}(C_{\tilde{l}})$, and it can be computed in $O(nr_l r_{\tilde{l}})$. Thus $A_l A'_{\tilde{l}}$ has the same sparsity as $A'_{\tilde{l}}$.

Combining the above we have the following factorization:

$$\begin{aligned} A_l A'_{l+} + A_{(l+1)+} A'_l &= \sum_{\tilde{l}=l}^L B_l \bar{C}_{l,\tilde{l}}^T + \sum_{\tilde{l}=l+1}^L \bar{B}_{\tilde{l},l} C_l^T \\ (F.2) \quad &= \begin{bmatrix} B_l & \sum_{\tilde{l}=l+1}^L \bar{B}_{\tilde{l},l} \end{bmatrix} \begin{bmatrix} \sum_{\tilde{l}=l}^L \bar{C}_{l,\tilde{l}} & C_l' \end{bmatrix}^T, \end{aligned}$$

which we can compute in

$$O\left(nr_l \sum_{\tilde{l}=l+1}^L r_{\tilde{l}}' + nr_l' \sum_{\tilde{l}=l}^L r_{\tilde{l}}\right).$$

Note that $\mathcal{I}(\sum_{\tilde{l}=l+1}^L \bar{B}_{\tilde{l},l}) = \mathcal{I}(B_l)$, and similarly, $\mathcal{I}(\sum_{\tilde{l}=l}^L \bar{C}_{l,\tilde{l}}) = \mathcal{I}(C_l')$. Therefore, we can equivalently represent (F.2) as a product of two block diagonal matrices by permuting the columns in the left and right factors accordingly. The resulting two block diagonal matrices are the factors of AA' on level l , and in the compressed form have size $n \times (r_l + r_l')$ each.

Finally, from (F.1) we see that matrix AA' is an MLR matrix with MLR-rank $(r + r')$. Moreover, computing factors requires $O(n \max\{r, r'\}^2)$ operations.

Reproducibility of computational results. This paper has been awarded the “SIAM Reproducibility Badge: Code and data available” as a recognition that the authors have followed reproducibility principles valued by SIMAX and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://github.com/cvxgrp/multilevel_factor_model (see notebooks in /examples folder).

REFERENCES

- [1] M. AITKIN, D. ANDERSON, AND J. HINDE, *Statistical modelling of data on teaching styles*, J. R. Stat. Soc. Ser. A: Statist. Soc., 144 (1981), pp. 419–448, <https://doi.org/10.2307/2981826>.
- [2] T. ASPAROUHOV AND B. MUTHÉN, *Multilevel modeling of complex survey data*, in Proceedings of the Joint Statistical Meeting in Seattle, Citeseer, 2006, pp. 2718–2726.
- [3] J. BAI AND S. NG, *Determining the number of factors in approximate factor models*, Econometrica, 70 (2002), pp. 191–221, <https://doi.org/10.1111/1468-0262.00273>.
- [4] J. BAI AND P. WANG, *Identification and Bayesian estimation of dynamic factor models*, J. Bus. Econ. Stat., 33 (2015), pp. 221–240, <https://doi.org/10.1080/07350015.2014.941467>.

- [5] S. BARRATT AND S. BOYD, *Covariance prediction via convex optimization*, Optim. Eng., 24 (2023), pp. 2045–2078, <https://doi.org/10.1007/s11081-022-09765-w>.
- [6] S. BHOJRAJ, C. M. C. LEE, AND D. K. OLER, *What's my line? A comparison of industry classification schemes for capital market research*, J. Account. Res., 41 (2003), pp. 745–774, <https://doi.org/10.1046/j.1475-679X.2003.00122.x>.
- [7] S. BOKER, M. NEALE, H. MAES, M. WILDE, M. SPIEGEL, T. BRICK, J. SPIES, R. ESTABROOK, S. KENNY, T. BATES, P. MEHTA, AND J. FOX, *Openmx: An open source extended structural equation modeling framework*, Psychometrika, 76 (2011), pp. 306–317, <https://doi.org/10.1007/s11336-010-9200-6>.
- [8] S. P. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2004.
- [9] M. BRUNNER, G. NAGY, AND O. WILHELM, *A tutorial on hierarchically structured constructs*, J. Pers., 80 (2012), pp. 796–846, <https://doi.org/10.1111/j.1467-6494.2011.00749.x>.
- [10] J. CARROLL, *Human Cognitive Abilities: A Survey of Factor-Analytic Studies*, Vol. 1, Cambridge University Press, 1993.
- [11] R. CATTELL, *A biometrics invited paper. Factor analysis: An introduction to essentials I. The purpose and underlying models*, Biometrics, 21 (1965), pp. 190–215, <https://doi.org/10.2307/2528364>.
- [12] D. CHILD, *The Essentials of Factor Analysis*, A&C Black, 2006.
- [13] M. CRUCINI, A. KOSE, AND C. OTROK, *What are the driving forces of international business cycles?*, Rev. Econ. Dynam., 14 (2011), pp. 156–175, <https://doi.org/10.1016/j.red.2010.09.001>.
- [14] J. DE LEEUW, E. MEIJER, AND H. GOLDSTEIN, *Handbook of Multilevel Analysis*, Vol. 401, Springer, 2008.
- [15] R. DEDRICK, J. FERRON, M. HESS, K. HOGARTY, J. KROMREY, T. LANG, J. NILES, AND R. LEE, *Multilevel modeling: A review of methodological issues and applications*, Rev. Educ. Res., 79 (2009), pp. 69–102, <https://doi.org/10.3102/0034654308325581>.
- [16] A. DEMPSTER, N. LAIRD, AND D. RUBIN, *Maximum likelihood from incomplete data via the EM algorithm*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 39 (1977), pp. 1–22, <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>.
- [17] C. DEYOUNG, *Higher-order factors of the big five in a multi-informant sample*, J. Pers. Soc. Psychol., 91 (2006), pp. 1138–1151, <https://doi.org/10.1037/0022-3514.91.6.1138>.
- [18] C. DOMÍNGUEZ CONDE, C. XU, L. B. JARVIS, D. B. RAINBOW, S. B. WELLS, T. GOMES, S. K. HOWLETT, O. SUCHANEK, K. POLANSKI, H. W. KING, L. MAMANOVA, N. HUANG, P. A. SZABO, L. RICHARDSON, L. BOLT, E. S. FASOULI, K. T. MAHBUBANI, M. PRETE, L. TUCK, N. RICHOZ, Z. K. TUONG, L. CAMPOS, H. S. MOUSA, E. J. NEEDHAM, S. PRITCHARD, T. LI, R. ELMONTAITE, J. PARK, E. RAHMANI, D. CHEN, D. K. MENON, O. A. BAYRAKTAR, L. K. JAMES, K. B. MEYER, N. YOSEF, M. R. CLATWORTHY, P. A. SIMS, D. L. FARBER, K. SAEB-PARSY, J. L. JONES, AND S. A. TEICHMANN, *Cross-tissue immune cell analysis reveals tissue-specific features in humans*, Science, 376 (2022), eab15197, <https://doi.org/10.1126/science.abl5197>.
- [19] L. FABRIGAR, D. WEGENER, R. MACCALLUM, AND E. STRAHAN, *Evaluating the use of exploratory factor analysis in psychological research*, Psychol. Methods, 4 (1999), pp. 272–299, <https://doi.org/10.1037/1082-989X.4.3.272>.
- [20] E. FAMA AND K. FRENCH, *Common risk factors in the returns on stocks and bonds*, J. Financ. Econ., 33 (1993), pp. 3–56, [https://doi.org/10.1016/0304-405X\(93\)90023-5](https://doi.org/10.1016/0304-405X(93)90023-5).
- [21] B. FRUCHTER, *Introduction to Factor Analysis*, Van Nostrand, 1954.
- [22] A. GELMAN AND J. HILL, *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge University Press, 2007.
- [23] H. GOLDSTEIN, *Multilevel mixed linear model analysis using iterative generalized least squares*, Biometrika, 73 (1986), pp. 43–56, <https://doi.org/10.1093/biomet/73.1.43>.
- [24] H. GOLDSTEIN, *Multilevel Statistical Models*, John Wiley & Sons, 2011.
- [25] H. GOLDSTEIN AND W. BROWNE, *Multilevel factor analysis modelling using Markov chain Monte Carlo estimation*, in *Latent Variable and Latent Structure Models*, Psychology Press, 2014, pp. 237–256.
- [26] A. GREGORY AND A. HEAD, *Common and country-specific fluctuations in productivity, investment, and the current account*, J. Monetary Econ., 44 (1999), pp. 423–451, [https://doi.org/10.1016/S0304-3932\(99\)00035-5](https://doi.org/10.1016/S0304-3932(99)00035-5).
- [27] H. HARMAN, *Modern Factor Analysis*, University of Chicago Press, 1976.
- [28] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009.
- [29] F. HUANG, *Conducting multilevel confirmatory factor analysis using R*, Working paper, 2017, <https://doi.org/10.13140/RG.2.2.12391.34724>.

- [30] K. JÖRESKOG, *A general approach to confirmatory maximum likelihood factor analysis*, *Psychometrika*, 34 (1969), pp. 183–202, <https://doi.org/10.1007/BF02289343>.
- [31] K. JÖRESKOG AND D. SÖRBOM, *LISREL 8: User's Reference Guide*, Scientific Software International, 1996.
- [32] B. JÖRG AND E. SANDRA, *Analyzing international business and financial cycles using multi-level factor models: A comparison of alternative approaches*, in *Dynamic Factor Models*, Vol. 35, Emerald Group Publishing Limited, 2016, pp. 177–214.
- [33] A. KOSE, C. OTROK, AND C. WHITEMAN, *International business cycles: World, region, and country-specific factors*, *Am. Econ. Rev.*, 93 (2003), pp. 1216–1239, <https://doi.org/10.1257/000282803769206278>.
- [34] D. LEVINE, S. A. RIZVI, S. LÉVY, N. PALLIKKAVALIYAVEETIL, D. ZHANG, X. CHEN, S. GHADERMARZI, R. WU, Z. ZHENG, I. VRKIC, A. ZHONG, D. RASKIN, I. HAN, A. H. DE OLIVEIRA FONSECA, J. O. CARO, A. KARBASI, R. M. DHODAPKAR, AND D. VAN DIJK, *Cell2sentence: Teaching large language models the language of biology*, *BioRxiv*, (2023), pp. 2023–2009.
- [35] T. LIN, *A comparison of multiple imputation with EM algorithm and MCMC method for quality of life missing data*, *Qual. Quant.*, 44 (2010), pp. 277–287, <https://doi.org/10.1007/s11135-008-9196-5>.
- [36] M. LINDSTROM AND D. BATES, *Newton-Raphson and EM algorithms for linear mixed-effects models for repeated-measures data*, *J. Amer. Statist. Assoc.*, 83 (1988), pp. 1014–1022, <https://doi.org/10.1080/01621459.1988.10478693>.
- [37] D. McCARTHY, K. CAMPBELL, A. LUN, AND Q. WILLS, *Scater: Pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R*, *Bioinformatics*, 33 (2017), pp. 1179–1186, <https://doi.org/10.1093/bioinformatics/btw777>.
- [38] R. McDONALD AND H. GOLDSTEIN, *Balanced versus unbalanced designs for linear structural relations in two-level data*, *British J. Math. Statist. Psychol.*, 42 (1989), pp. 215–232, <https://doi.org/10.1111/j.2044-8317.1989.tb00911.x>.
- [39] K. MCGREW, *CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research*, *Intelligence*, 37 (2009), pp. 1–10.
- [40] C. MEGILL, B. MARTIN, C. WEAVER, S. BELL, L. PRINS, S. BADAJOZ, B. MCCANDLESS, A. O. PISCO, M. KINSELLA, F. GRIFFIN, J. KIGGINS, G. HALIBURTON, A. MANI, M. WEIDEN, M. DUNITZ, M. LOMBARDO, T. HUANG, T. SMITH, S. CHAMBERS, J. FREEMAN, J. COOL, AND A. CARR, *Cellxgene: A performant, scalable exploration platform for high dimensional sparse matrices*, *BioRxiv*, (2021), <https://doi.org/10.1101/2021.04.05.438318>.
- [41] B. MUTHÉN AND L. MUTHÉN, *Mplus*, in *Handbook of Item Response Theory*, Chapman and Hall/CRC, 2017, pp. 507–518.
- [42] B. MUTHÉN, *Mplus: A brief overview of its unique analysis capabilities*, in *Cambridge Handbook of Research Methods and Statistics for the Social and Behavioral Sciences: Volume Three*, 2024, in press.
- [43] T. PARSHAKOVA, T. HASTIE, E. DARVE, AND S. BOYD, *Factor fitting, rank allocation, and partitioning in multilevel low rank matrices*, in *Optimization, Discrete Mathematics, and Applications to Data Sciences*, SOIA 220, Springer, 2024, pp. 135–173.
- [44] J. PRITIKIN, M. HUNTER, T. VON OERTZEN, T. BRICK, AND S. BOKER, *Many-level multilevel structural equation modeling: An efficient evaluation strategy*, *Structural Equation Model.: A Multidisciplinary J.*, 24 (2017), pp. 684–698.
- [45] C. C. S. PROGRAM, S. ABDULLA, B. AEVERMANN, P. ASSIS, S. BADAJOZ, S. BELL, E. BEZZI, B. CAKIR, J. CHAFFER, S. CHAMBERS, J. M. CHERRY, T. CHI, J. CHIEN, L. DORMAN, P. GARCIA-NIETO, N. GLORIA, M. HASTIE, D. HEGEMAN, J. HILTON, T. HUANG, A. INFELD, A.-M. ISTRATE, I. JELIC, K. KATSUYA, Y. J. KIM, K. LIANG, M. LIN, M. LOMBARDO, B. MARSHALL, B. MARTIN, F. McDADE, C. MEGILL, N. PATEL, A. PREDEUS, B. RAYMOR, B. ROBATMILI, D. ROGERS, E. RUTHERFORD, D. SADGAT, A. SHIN, C. SMALL, T. SMITH, P. SRIDHARAN, A. TARASHANSKY, N. TAVARES, H. THOMAS, A. TOLOPKO, M. URISKO, J. YAN, G. YERETSSIAN, J. ZAMANIAN, A. MANI, J. COOL, AND A. CARR, *Cz cellxgene discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data*, *Nucleic Acids Res.*, 53 (2025), pp. D886–D900, <https://doi.org/10.1093/nar/gkae1142>.
- [46] S. RABE-HESKETH, A. SKRONDAL, AND A. PICKLES, *Generalized multilevel structural equation modeling*, *Psychometrika*, 69 (2004), pp. 167–190, <https://doi.org/10.1007/BF02295939>.
- [47] S. RABE-HESKETH, A. SKRONDAL, AND A. PICKLES, *GLLAMM Manual*, Technical report 1160, U.C. Berkeley Division of Biostatistics Working Paper Series, 2004.
- [48] J. RASBASH, W. BROWNE, H. GOLDSTEIN, M. YANG, I. PLEWIS, M. HEALY, G. WOODHOUSE, D. DRAPER, I. LANGFORD, AND T. LEWIS, *A user's guide to MLwiN*, Institute of Education, London, 286, 2000.

- [49] S. RAUDENBUSH, *Maximum likelihood estimation for unbalanced multilevel covariance structure models via the EM algorithm*, British J. Math. Stat. Psychol., 48 (1995), pp. 359–370, <https://doi.org/10.1111/j.2044-8317.1995.tb01068.x>.
- [50] S. RAUDENBUSH AND A. BRYK, *Hierarchical Linear Models: Applications and Data Analysis Methods*, Vol. 1, SAGE, 2002.
- [51] Y. ROSSEEL, *An R package for structural equation modeling*, J. Stat. Softw., 48 (2012), pp. 1–36, <https://doi.org/10.18637/jss.v048.i02>.
- [52] K. ROWE AND P. HILL, *Modeling educational effectiveness in classrooms: The use of multi-level structural equations to model students' progress*, Educational Research and Evaluation, 4 (1998), pp. 307–347, <https://doi.org/10.1076/edre.4.4.307.6953>.
- [53] D. RUBIN AND D. THAYER, *EM algorithms for ML factor analysis*, Psychometrika, 47 (1982), pp. 69–76, <https://doi.org/10.1007/BF02293851>.
- [54] J. SCHMID AND J. LEIMAN, *The development of hierarchical factor solutions*, Psychometrika, 22 (1957), pp. 53–61, <https://doi.org/10.1007/BF02289209>.
- [55] C. SPEARMAN, “General intelligence,” objectively determined and measured, Amer. J. Psychol., 15 (1904), pp. 201–292, <https://doi.org/10.2307/1412107> (accessed 2023-01-16).
- [56] P. WANG, *Large dimensional factor models with a multi-level factor structure: Identification, estimation, and inference*, 2012.
- [57] R. WHERRY, *Hierarchical factor solutions without rotation*, Psychometrika, 24 (1959), pp. 45–51, <https://doi.org/10.1007/BF02289762>.
- [58] A. WOLF, P. ANGERER, AND F. THEIS, *Scanpy: Large-scale single-cell gene expression data analysis*, Genome Biol., 19 (2018), pp. 1–5, <https://doi.org/10.1186/s13059-017-1382-0>.
- [59] Y.-F. YUNG, D. THISSEN, AND L. D. MCLEOD, *On the relationship between the higher-order factor model and the hierarchical factor model*, Psychometrika, 64 (1999), pp. 113–128, <https://doi.org/10.1007/BF02294531>.