Statistica Sinica Preprint No: SS-2025-0196	
Title	Exploratory Hierarchical Factor Analysis with an Application to Psychological Measurement
Manuscript ID	SS-2025-0196
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0196
Complete List of Authors	Jiawei Qiao, Yunxiao Chen and Zhiliang Ying
Corresponding Authors	Yunxiao Chen
E-mails	y.chen186@lse.ac.uk
Notice: Accepted author version.	

EXPLORATORY HIERARCHICAL FACTOR ANALYSIS WITH AN APPLICATION TO PSYCHOLOGICAL MEASUREMENT

Jiawei Qiao[1], Yunxiao Chen[2] and Zhiliang Ying[3]

Abstract: Hierarchical factor models, which include the bifactor model as a special case, are useful in social and behavioural sciences for measuring hierarchically structured constructs. Specifying a hierarchical factor model involves imposing hierarchically structured zero constraints on a factor loading matrix, which is often challenging. Therefore, an exploratory analysis is needed to learn the hierarchical factor structure from data. Unfortunately, there does not exist an identifiability theory for the learnability of this hierarchical structure, nor a computationally efficient method with provable performance. The method of Schmid-Leiman transformation, which is often regarded as the default method for exploratory hierarchical factor analysis, is flawed and likely to fail. The contribution of this paper is three-fold. First, an identifiability result is established for general hierarchical factor models, which shows that the hierarchical factor structure is learnable under mild regularity conditions. Second, a computationally efficient divide-and-conquer approach is proposed for learning the hierarchical factor structure. Finally, asymptotic theory is established for the proposed method, showing that it can consistently recover the true hierarchical factor structure as the sample size grows to infinity. The power of the proposed method is shown via simulation studies and a real data application to a personality test. The computation code for the proposed method is publicly available at https://github.com/EmetSelch97/EHFA/.

Key words and phrases: Hierarchical factor model, augmented Lagrangian method, exploratory hierarchical factor analysis

1 Introduction

Many constructs in social and behavioural sciences are conceptualized to be hierarchically structured, such as psychological traits (e.g., Carroll, 1993; DeYoung, 2006), economic factors (e.g., Kose et al., 2008; Moench et al., 2013), health outcomes measures (e.g., Chen et al., 2006; Reise et al., 2007), and constructs in marketing research (e.g., Sharma et al., 2022).

Hierarchical factor models (Brunner et al., 2012; Schmid and Leiman, 1957; Thomson, 1939; Yung et al., 1999), which include the bi-factor model (Holzinger and Swineford, 1937) as a special case with two factor layers, are commonly used to measure hierarchically structured constructs. In these models, hierarchically structured zero constraints are imposed on factor loadings to define the hierarchical factors. When the hierarchical factor structure is known or hypothesized a priori, the statistical inference of a hierarchical factor model only requires standard confirmatory factor analysis techniques (Brunner et al., 2012). However, for many real-world scenarios, little prior information about the hierarchical factor structure is available, so we need to learn this structure from data. This analysis is referred to as exploratory hierarchical factor analysis.

Exploratory hierarchical factor analysis is a structured extension of classical exploratory factor analysis (e.g., Anderson, 2003; Chen et al., 2019). In conventional exploratory factor analysis, rotation methods (e.g., Browne, 2001) are typically employed to achieve a sparse loading structure (Thurstone, 1947) for interpreting the factors. Exploratory hierarchical factor analysis builds on this principle but imposes a hierarchical sparsity pattern on the loading matrix, requiring that zero loadings be placed nonarbitrarily and follow a hierarchical structure. Compared with classical exploratory factor analysis, exploratory hierarchical factor analysis faces theoretical and computational challenges. First, we lack a theoretical understanding of its identifiability, i.e., the conditions under which the hierarchical factor structure is uniquely determined by the distribution of manifest variables. This is an important question, as learning a hierarchical factor structure is only sensible when it is identifiable. Although identifiability theory has been established for exploratory bi-factor analysis in Qiao et al. (2025), to our knowledge, no results are available under the general hierarchical factor model. Second, learning the hierarchical factor structure is a model selection problem, which is computationally challenging due to its combinatorial nature. For a moderately large number of manifest variables, it is computationally infeasible to compare all the possible hi-

erarchical factor structures using relative fit measures. However, it is worth noting that a computationally efficient method is available and commonly used for this problem, known as the Schmid–Leiman transformation (Schmid and Leiman, 1957). This method involves constructing a constrained higher-order factor model by iteratively applying an exploratory factor analysis method with oblique rotation and, further, performing orthogonal transformations to turn the higher-order factor model solution into a hierarchical factor model solution. However, as shown in Yung et al. (1999), the Schmid-Leiman transformation imposes unnecessary proportionality constraints on the factor loadings. As a result, it may not work well for more general hierarchical factor models. Jennrich and Bentler (2011) gave an example in which the Schmid–Leiman transformation fails to recover a bi-factor loading structure. Not only theoretically flawed, the implementation of the Schmid–Leiman transformation can also be a challenge for practitioners due to several decisions one needs to make, including the choice of oblique rotation method for the exploratory factor analysis and how the number of factors is determined in each iteration.

This paper fills these gaps. Specifically, we establish an identifiability result for exploratory hierarchical factor analysis, showing that the hierarchical factor structure is learnable under mild regularity conditions. We also propose a computationally efficient divide-and-conquer approach for learning the hierarchical factor structure. This approach divides the learning problem into many subtasks of learning the factors nested within a factor, also known as the child factors of this factor. It conquers these subtasks layer by layer, starting from the one consisting only of the general factor. Our method for solving each subtask has two building blocks – (1) a constraint-based continuous optimization algorithm and (2) a search algorithm based on an information criterion. The former is used to explore the number and loading structure of the child factors, and the latter serves as a refinement step that ensures the true structure of the child factors is selected with high probability. Finally, asymptotic theory is established for the proposed method, showing that it can consistently

recover the true hierarchical factor structure as the sample size grows to infinity.

The proposed method is closely related to the method proposed in Qiao et al. (2025) for exploratory bi-factor analysis, which can be seen as a special case of the current method when the hierarchical factor structure is known to have only two layers. However, we note that the current problem is substantially more challenging as the complexity of a hierarchical factor structure grows quickly as the number of factor layers increases. Nevertheless, the constraint-based continuous optimization algorithm that serves as a building block of the proposed method is similar to the algorithm used for exploratory bi-factor analysis in Qiao et al. (2025). This algorithm turns a computationally challenging combinatorial model selection problem into a relatively easier-to-solve continuous optimization problem, enabling a more efficient global search of the factor structure.

The rest of the paper is organized as follows. In Section 2, we establish the identifiability of the general hierarchical factor model and, further, propose a divide-and-conquer approach for exploratory hierarchical factor analysis and establish its consistency. In Section 3, the computation of the divide-and-conquer approach is discussed. Simulation studies and a real data example are presented in Sections 4 and 5, respectively, to evaluate the performance of the proposed method. We conclude with discussions in Section 6.

2 Exploratory Hierarchical Factor Analysis

2.1 Constraints of hierarchical factor model

Consider a factor model for J observed variables, with K orthogonal factors. The population covariance matrix can be decomposed as $\Sigma = \Lambda\Lambda^\top + \Psi$, where $\Lambda = (\lambda_{jk})_{J\times K}$ is the loading matrix and Ψ is a $J \times J$ diagonal matrix, which is typically referred as the unique variance matrix (see, e.g., Fabrigar and Wegener, 2012), with diagonal entries $\psi_1, \ldots, \psi_J > 0$ that

record the unique variances. We say this factor model is a hierarchical factor model if the loading matrix Λ satisfies certain zero constraints that encode a factor hierarchy.

Specifically, let $v_k = \{j : \lambda_{jk} \neq 0\}$ be the variables loading on the kth factor. The factor model becomes a hierarchical factor model if $v_1, \ldots, v_K$ satisfy the following constraints:

C1. $v_1 = \{1, \ldots, J\}$ corresponds to a general factor that is loaded on by all the items.

C2. For any $k < l$, it holds that either $v_l \subsetneq v_k$ or $v_l \subset \{1, \ldots, J\} \setminus v_k$. That is, the variables that load on factor l are either a subset of those that load on factor k or do not overlap with them. When $v_l \subsetneq v_k$, we say factor l is a descendant factor of factor k. If further that there does not exist k' such that $k < k' < l$ and $v_l \subsetneq v_{k'} \subsetneq v_k$, we say factor l is a child factor of factor k, and factor k is a parent factor of factor l.

C3. For a given factor k, we denote all its child factors as Ch_k. Then its cardinality $|\mathrm{Ch}_k|$ satisfies that $|\mathrm{Ch}_k| = 0$ or $|\mathrm{Ch}_k| \geq 2$. That is, a factor either does not have any child factor or at least two child factors. Moreover, when a factor k has two or more child factors, these child factors satisfy that $v_l \cap v_{l'} = \emptyset$, for any $l, l' \in \mathrm{Ch}_k$, and $\cup_{l \in \mathrm{Ch}_k} v_l = v_k$. That is, the sets of variables that load on the child factors of a factor are a partition of the variables that load on this factor. We note that one child node is not allowed due to identification issues. To avoid ambiguity in the labelling of the factors, we further require that

   (a) $k < l$ if factors k and l are the child factors of the same factor and $\min\{v_k\} < \min\{v_l\}$. That is, we label the child factors of the same factor based on the labels of the variables that load on each factor.

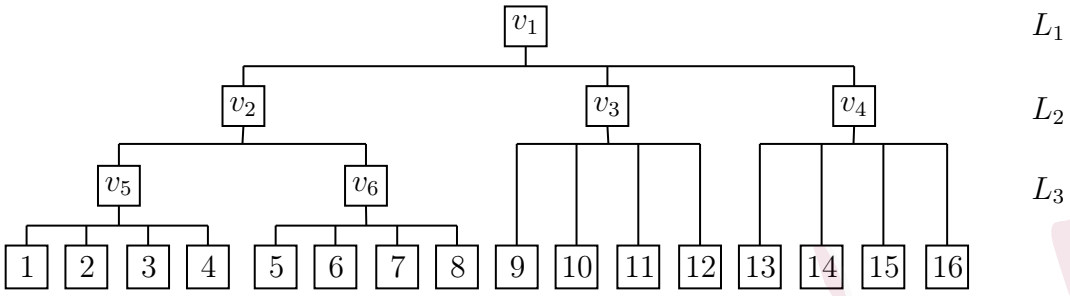
   (b) $k < l$ if factors k and l do not have the same parent factor, and the parent factor of k has a smaller label than the parent factor of l.

The requirement $|\mathrm{Ch}_k| = 0$ or $|\mathrm{Ch}_k| \geq 2$ in constraint C3 is necessary for the hierarchical factor model to be identifiable. When a factor k has a unique child factor (i.e. $|\mathrm{Ch}_k| = 1$), it is easy to show that the two columns of the loading matrix that correspond to factor k and its single child factor are not determined up to an orthogonal rotation.

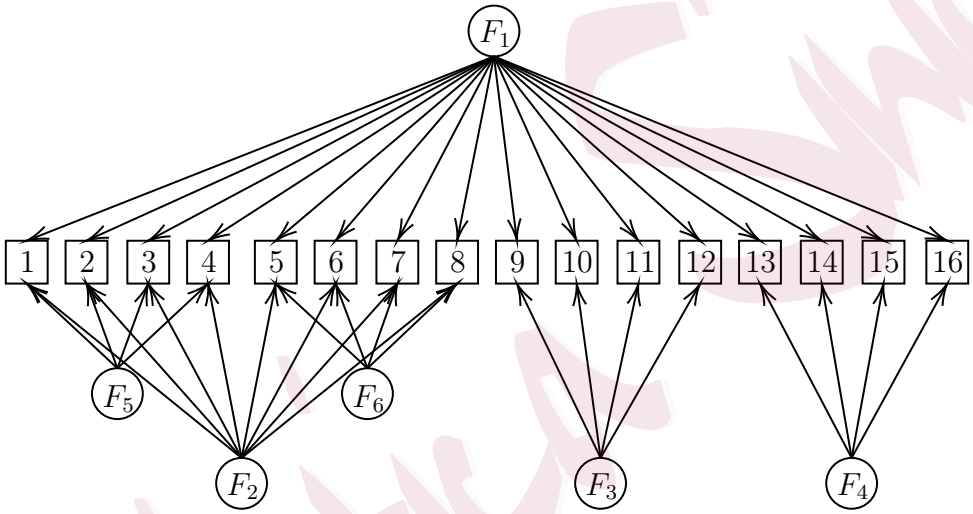
We note that when the above constraints hold, the hierarchical factor structure can be visualized as a tree, where each internal node represents a factor, and each leaf node represents an observed variable. In this tree, factor l being a child factor of factor k, is represented by node l being a child node of node k. The variables that load on each factor are indicated by its descendant leaf nodes.

When the factors follow a hierarchical structure, we can classify the factors into layers. The first factor layer only includes the general factor, denoted by $L_1 = \{1\}$. The rest of the layers can be defined recursively. That is, if a factor k is in the tth layer, then its child factors are in the $(t+1)$th layer. Let T be the total number of layers and $L_1, \ldots, L_T$ be the sets of factors for the T layers. It is worth noting that the way the layers are labelled here is opposite to how they are labelled in the literature. That is, we label the layers from the top to the bottom of the hierarchy of the factors. In contrast, they are labelled from the bottom to the top in the literature (see, e.g., Yung et al., 1999). We adopt the current labelling system because it is more convenient for the proposed method in Section 2.2 that learns the factor hierarchy from top to bottom.

An illustrative example of a three-layer hierarchical factor model is given in Figure 1, where Panel (a) shows the variables that load on each factor from the top layer to the bottom layer, and Panel (b) shows the corresponding path diagram. In this example, $J = 16$, $K = 6$, $v_1 = \{1, 2, \ldots, 16\}$, $v_2 = \{1, \ldots, 8\}$, $v_3 = \{9, \ldots, 12\}$, $v_4 = \{13, \ldots, 16\}$, $v_5 = \{1, \ldots, 4\}$ and $v_6 = \{5, \ldots, 8\}$. The factors are labeled following the constraints C3(a) and C3(b). Based on this hierarchical structure, we have $T = 3$, $L_1 = \{1\}$, $L_2 = \{2, 3, 4\}$ and $L_3 = \{5, 6\}$. The



(a) The hierarchical factor structure of a three-layer hierarchical factor model.



(b) The path diagram corresponding to the hierarchical factor model in Panel (a).

Figure 1: The illustrative example of a three-layer hierarchical factor model.

loading matrix Λ under the hierarchical structure takes the form

$$\Lambda = \begin{pmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} & \lambda_{41} & \lambda_{51} & \lambda_{61} & \lambda_{71} & \lambda_{81} & \lambda_{91} & \lambda_{10,1} & \lambda_{11,1} & \lambda_{12,1} & \lambda_{13,1} & \lambda_{14,1} & \lambda_{15,1} & \lambda_{16,1} \\ \lambda_{12} & \lambda_{22} & \lambda_{32} & \lambda_{42} & \lambda_{52} & \lambda_{62} & \lambda_{72} & \lambda_{82} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{93} & \lambda_{10,3} & \lambda_{11,3} & \lambda_{12,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{13,4} & \lambda_{14,4} & \lambda_{15,4} & \lambda_{16,4} \\ \lambda_{15} & \lambda_{25} & \lambda_{35} & \lambda_{45} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{56} & \lambda_{66} & \lambda_{76} & \lambda_{86} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^{\top}. \tag{1}$$

Under a confirmatory setting, the number of factors K and the variables associated with each factors, $v_1, v_2, \ldots, v_K$, are known. In that case, estimating the hierarchical factor model is a relatively simple problem, which involves solving an optimization problem with suitable

zero constraints on the loading parameters. However, in many real-world applications, we do not have prior knowledge about the hierarchical structure of the loading matrix. In these cases, we are interested in exploratory hierarchical factor analysis, i.e., simultaneously learning the hierarchical structure from data and estimating the corresponding parameters.

Before presenting a method for exploratory hierarchical factor analysis, we first show that the true factor hierarchy is unique under mild conditions, which is essential for the true structure to be learnable. Suppose that we are given a true covariance matrix $\Sigma^* = \Lambda^*(\Lambda^*)^\top + \Psi^*$, where the true loading matrix Λ^* satisfies the constraints of a hierarchical factor model. Theorem 1 below shows that the true loading matrix Λ^* is unique up to column sign-flips and thus yields the same hierarchical structure.

The following notation is needed in the rest of the paper. Given a hierarchical factor structure with loading sets v_i, let $D_i = \{j : v_j \subsetneq v_i\}$ be the set of all descendent factors of factor i. For example, in the hierarchical structure shown in Figure 1, $D_2 = \{5, 6\}$. For any matrix $A = (a_{i,j})_{m\times n}$ and sets $\mathcal{S}_1 \subset \{1, \ldots, m\}$ and $\mathcal{S}_2 \subset \{1, \ldots, n\}$, let $A_{[\mathcal{S}_1,\mathcal{S}_2]} = (a_{i,j})_{i\in\mathcal{S}_1, j\in\mathcal{S}_2}$ be the submatrix of A consisting of elements that lie in rows belonging to set $\mathcal{S}_1$ and columns belonging to set $\mathcal{S}_2$, where the rows and columns are arranged in ascending order based on their labels in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. For example, consider the loading matrix in (1), where $v_2 = \{1, 2, \ldots, 8\}$. Then, $\Lambda_{[v_2,\{1,2\}]}$ takes the form

$$\Lambda_{[v_2,\{1,2\}]} = \begin{pmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} & \lambda_{41} & \lambda_{51} & \lambda_{61} & \lambda_{71} & \lambda_{81} \\ \lambda_{12} & \lambda_{22} & \lambda_{32} & \lambda_{42} & \lambda_{52} & \lambda_{62} & \lambda_{72} & \lambda_{82} \end{pmatrix}^\top.$$

For any vector $\boldsymbol{a} = (a_1, \ldots, a_n)^\top$ and set $\mathcal{S} \subset \{1, \ldots, n\}$, we similarly define $\boldsymbol{a}_{[\mathcal{S}]} = (a_i)^\top_{i\in\mathcal{S}}$ be the subvector of $\boldsymbol{a}$ consisting of the elements belonging to $\mathcal{S}$, where the elements in $\boldsymbol{a}_{[\mathcal{S}]}$ are arranged in ascending order based on their labels in $\mathcal{S}$. For any set $\mathcal{S}_1 \subset \{1, 2, \ldots, n\}$, let $\text{vec}(\mathcal{S}_1)$ be a mapping that maps the set $\mathcal{S}_1$ to a vector whose elements are the same as $\mathcal{S}_1$

and arranged in ascending order. For two sets $\mathcal{S}_1 \subset \{1, 2, \ldots, n\}$ and $\mathcal{S}_2 \subset \{1, 2, \ldots, |\mathcal{S}_1|\}$, we denote $\mathcal{S}_1[\mathcal{S}_2]$ as the subset of $\mathcal{S}_1$, consisting of elements in $\text{vec}(\mathcal{S}_1)[\mathcal{S}_2]$.

Condition 1 *The population covariance matrix can be expressed as the form* $\Sigma^* = \Lambda^*(\Lambda^*)^\top + \Psi^*$, *where the true loading matrix* Λ^* *is of rank* K *and the loading sets* v_k^* *and child factors* Ch_k^* *defined by* Λ^* *satisfy the constraints C1–C3 of a hierarchical factor model.*

Condition 2 *Given another* $J \times K$ *matrix* Λ *and* $J \times J$ *diagonal matrix* Ψ *such that* $\Sigma^* = \Lambda^*(\Lambda^*)^\top + \Psi^* = \Lambda\Lambda^\top + \Psi$, *we have* $\Lambda\Lambda^\top = \Lambda^*(\Lambda^*)^\top$ *and* $\Psi = \Psi^*$.

Condition 3 *Let* D_k^* *be the corresponding true set of descendant factors of factor* k. *For any factor* i *with* $Ch_i^* \neq \emptyset$ *and any* $j \in Ch_i^*$, *it satisfies that (1) any two rows of* $\Lambda^*_{[v_j^*,\{i,j\}]}$ *are linearly independent, (2) for any* $k \in v_j^*$, $\Lambda^*_{[v_j^* \setminus \{k\},\{i,j\} \cup D_j^*]}$ *has full column rank, and (3) if* $|Ch_j^*| \geq 2$, *then, for any* $s_1, s_2 \in Ch_j^*$, $k_1, k_2 \in v_{s_1}^*$, *and* $k_3, k_4 \in v_{s_2}^*$, $\Lambda^*_{[\{k_1,\ldots,k_4\},\{i,j,s_1,s_2\}]}$ *is of full rank.*

Theorem 1 *Suppose that Conditions 1–3 hold. If there exists some hierarchical factor structure with* K *factors such that its loading matrix* Λ *and unique variance matrix* Ψ *satisfy* $\Sigma^* = \Lambda\Lambda^\top + \Psi$, *there exists some sign flip matrix* $Q \in \mathcal{Q}$ *such that* $\Lambda = \Lambda^* Q$, *where* $\mathcal{Q}$ *consists of all* $K \times K$ *diagonal matrices* Q *whose diagonal entries take values* 1 *or* -1.

Remark 1 *As far as we know, Theorem 1 is the first identifiability result for exploratory hierarchical factor analysis. This theorem establishes mild regularity conditions under which the true hierarchical factor model is identifiable when we do not know the true hierarchical factor structure. Condition 1 assumes that the true model is a hierarchical factor model. Under this model assumption, the identifiability result of Theorem 1 has two parts. The first part involves identifying the column space of the loading matrix based on the population covariance matrix, and the second part entails identifying the factors based on this column space. The identifiability result in the first part, which is assumed in Condition 2, has already*

been well studied in the literature. For example, Condition 4 below is a result in Theorem 5.1, Anderson and Rubin (1956), which gives a sufficient condition for Condition 2 to hold. On the other hand, the second part is more challenging and relies more on the hierarchical factor structure. Theorem 1 focuses on proving the second part.

Condition 4 *For each $j \in \{1, \ldots, J\}$, there exist two disjoint set $E_1, E_2 \subset \{1, \ldots, J\} \setminus \{j\}$ with $|E_1| = |E_2| = K$ such that $\Lambda^*_{[E_1,:]}$ and $\Lambda^*_{[E_2,:]}$ are of full rank, where $\Lambda^*_{[E_1,:]}$ and $\Lambda^*_{[E_2,:]}$ are the submatrices of Λ^* consisting of the rows belonging to E_1 and E_2.*

Remark 2 *Condition 2 implicitly imposes some minimum requirements on the parameter space for identifiable hierarchical factor models. In fact, Proposition 1 below implies a necessary condition for Condition 2. This necessary condition leads to the following constraint:*

C4. For all $k = 1, \ldots, K$, $|v_k| \geq 3$, and $|v_k| \geq 7$ if factor k has two or more child factors.

Proposition 1 *There exists another $J \times K$ matrix Λ following the same hierarchical factor structure as the true model and a $J \times J$ diagonal matrix Ψ such that $\Sigma^* = \Lambda^*(\Lambda^*)^\top + \Psi^* = \Lambda\Lambda^\top + \Psi$, if there exists a factor k such that (1) $|v^*_k| \leq 2$ or (2) $|Ch^*_k| \geq 2$ and $|v^*_k| \leq 6$.*

Proposition 1 follows directly from Theorem 1 in Fang et al. (2021).

Remark 3 *Condition 3 imposes three requirements. First, it requires that there do not exist two variables loading on factor j such that their loadings on any factor i and its child node j are linearly dependent. This is a mild assumption satisfied by almost all the models in the full parameter space of hierarchical factor models. Second, it requires that the submatrix $\Lambda^*_{[v^*_j,\{i,j\}\cup D^*_j]}$, which corresponds to variables in v^*_j and factors i, j, j's descendants, are still of full column rank after deleting any row. This condition mainly imposes a restriction on the number of descendant factors each factor can have. That is, the full-column-rank requirement implies that $|v^*_j| \geq 3+|D^*_j|$. As shown via Proposition 2 below, this requirement automatically*

holds for all the identifiable hierarchical factor models that satisfy constraints C1–C4. Other than that, the full-column-rank requirement is easily satisfied by most hierarchical factor models. These two requirements can be seen as an extension of Condition 2 of Qiao et al. (2025) to hierarchical factor models, where Qiao et al. (2025) consider a bi-factor model with possibly correlated bi-factors. Third, we require that when factor j has child factors s_1 and s_2, for any two variables k_1, k_2 loading on factor s_1 and any variables k_3, k_4 loading on factor s_2, the sub-loading matrix corresponding to variables $k_1, \ldots, k_4$ and factors i, j, s_1, s_2 is of full rank. Although the requirements in Condition 3 are quite mild, we acknowledge that they may be further weakened. For example, instead of requiring any two roles of $\Lambda^*_{[v_j^*,\{i,j\}]}$ to be linearly independent, we may only need to require a sufficient number pair of the rows of $\Lambda^*_{[v_j^*,\{i,j\}]}$ to be linearly independent; see Appendix S3 for further discussions. We leave the refinement of the condition for future investigation.

Proposition 2 *Suppose that the hierarchical factor structure satisfies constraints C1–C4. Then $|v_j^*| \geq 3 + |D_j^*|$ holds for each factor j.*

2.2 An Overview of Proposed Method

As the proposed method is quite sophisticated, we start with an overview of the proposed method to help readers understand it. Consider a dataset with N observation units from a certain population and J observed variables. Let S be the sample covariance matrix of observed data. The proposed method takes S as the input and outputs estimators:

1. $\widehat{T}$ and $\widehat{K}$ for the number of layers T and the number of factors K.
2. $\widehat{L}_1, \ldots, \widehat{L}_{\widehat{T}}$ for the factor layers $L_1, \ldots, L_T$ and $\widehat{v}_1, \widehat{v}_2, \ldots, \widehat{v}_{\widehat{K}}$ for the sets of variables loading on the K factors, $v_1, \ldots, v_K$.
3. $\widehat{\Lambda}$ and $\widehat{\Psi}$ for the loading matrix Λ and unique variance matrix Ψ.

As shown in Theorem 2 below, with the sample size N going to infinity, these estimates will converge to their true values.

The proposed method learns the hierarchical factor structure from the top to the bottom of the factor hierarchy. It divides the learning problem into many subproblems and conquers them layer by layer, starting from the first layer $\widehat{L}_1 = \{1\}$ with $\widehat{v}_1 = \{1, \ldots, J\}$. For each step t, $t = 2, 3, \ldots$, suppose the first to the $(t-1)$th layers have been learned. These layers are denoted by $\widehat{L}_i = \{k_{i-1} + 1, \ldots, k_i\}$, $i = 1, \ldots, t-1$, where $k_0 = 0$ and $k_1 = 1$, and the associated sets of variables are denoted by $\widehat{v}_1, \ldots, \widehat{v}_{k_{t-1}}$. We make the following decisions in the tth step:

1. For each factor $k \in \widehat{L}_{t-1}$, learn its child factors under the constraints C3 and C4. This is achieved by an Information-Criterion-Based (ICB) method described in Section 2.3 below. The labels of the child factors are denoted by $\widehat{\mathrm{Ch}}_k$. When $\widehat{\mathrm{Ch}}_k \neq \emptyset$, we denote the associated sets of variables as $\widehat{v}_l, l \in \widehat{\mathrm{Ch}}_k$.

2. If $\widehat{\mathrm{Ch}}_k = \emptyset$ for all $k \in \widehat{L}_{t-1}$, stop the learning algorithm and conclude that the factor hierarchy has $\widehat{T} = t - 1$ layers.

3. Otherwise, let $\widehat{L}_t = \{k_{t-1} + 1, \ldots, k_t\} = \cup_{k \in \widehat{L}_{t-1}} \widehat{\mathrm{Ch}}_k$ and proceed to the $(t+1)$th step.

We iteratively learn the structure of each layer until the preceding stopping criterion is met. Then we obtain the estimates $\widehat{\Lambda}$ and $\widehat{\Psi}$ by maximum likelihood estimation given $\widehat{K} = k_{\widehat{T}}, \widehat{v}_1, \ldots, \widehat{v}_{\widehat{K}}$:

$$\begin{aligned} (\widehat{\Lambda}, \widehat{\Psi}) = & \arg\min_{\Lambda, \Psi} l(\Lambda\Lambda^\top + \Psi; S), \\ & \text{s.t. } \lambda_{ij} = 0, i \notin \widehat{v}_j, i = 1, \ldots, J, j = 1, \ldots, \widehat{K}, \\ & \Psi_{[\{i\},\{i\}]} \geq 0, \Psi_{[\{i\},\{j\}]} = 0, i = 1, \ldots, J, j \neq i, \end{aligned} \tag{2}$$

where $l(\Lambda\Lambda^\top+\Psi;S)=N\big(\log(\det(\Lambda\Lambda^\top+\Psi))+\mathrm{tr}(S(\Lambda\Lambda^\top+\Psi)^{-1})-\log(\det(S))-J\big)$ equals twice the negative log-likelihood of the observed data up to a constant. We output $\widehat{T}$, $\widehat{K}$, $\widehat{L}_1,\ldots,\widehat{L}_{\widehat{T}}$, $\widehat{v}_1,\ldots,\widehat{v}_{\widehat{K}}$, $\widehat{\Lambda}$ and $\widehat{\Psi}$ as our final estimate of the hierarchical factor model.

To illustrate, consider the example in Figure 1. In the first step, we start with $\widehat{L}_1=\{1\}$ and $\widehat{v}_1=\{1,\ldots,16\}$. In the second step, we learn the child factors of Factor 1. If they are correctly learned, then we obtain $\widehat{\mathrm{Ch}}_1=\{2,3,4\}$ with $\widehat{v}_2=\{1,\ldots,8\}$, $\widehat{v}_3=\{9,\ldots,12\}$ and $\widehat{v}_4=\{13,\ldots,16\}$. This leads to $\widehat{L}_2=\{2,3,4\}$. In the third step, we learn the child factors of Factors 2, 3 and 4, one by one. If correctly learned, we have $\widehat{\mathrm{Ch}}_2=\{5,6\}$, $\widehat{\mathrm{Ch}}_3=\emptyset$, $\widehat{\mathrm{Ch}}_4=\emptyset$, $\widehat{L}_3=\{5,6\}$, $\widehat{v}_5=\{1,\ldots,4\}$ and $\widehat{v}_6=\{5,\ldots,8\}$. In the fourth step, if correctly learned, we have $\widehat{\mathrm{Ch}}_5=\widehat{\mathrm{Ch}}_6=\emptyset$, and the learning algorithm stops. We then have $\widehat{T}=3$, $\widehat{K}=6$, $\widehat{L}_1,\ldots,\widehat{L}_3$, $\widehat{v}_1,\ldots,\widehat{v}_6$ and further obtain $\widehat{\Lambda}$ and $\widehat{\Psi}$ using (2) given $\widehat{K}$ and $\widehat{v}_1,\ldots,\widehat{v}_6$.

We summarise the steps of the proposed method in Algorithm 1 below.

Algorithm 1 A Divide-and-Conquer method for learning the hierarchical factor structure

Input: Sample covariance matrix $S\in\mathbb{R}^{J\times J}$.

1: Set $\widehat{L}_1=\{1\}$ with $\widehat{v}_1=\{1,\ldots,J\}$.
2: Determine $\widehat{\mathrm{Ch}}_1$, the child factors of Factor 1, and $\widehat{v}_i$ for all $i\in\widehat{\mathrm{Ch}}_1$, the sets of variables loading on these child factors, by the ICB method in Algorithm 2.
3: Set $\widehat{L}_2=\widehat{\mathrm{Ch}}_1$ and $t=2$,
4: **while** $\widehat{L}_t\neq\emptyset$ **do**
5: **for** $k\in\widehat{L}_t$ **do**
6: Determine $\widehat{\mathrm{Ch}}_k$ and $\widehat{v}_i$ for all $i\in\widehat{\mathrm{Ch}}_k$ by the ICB method in Algorithm 2.
7: **end for**
8: Set $\widehat{L}_{t+1}=\cup_{k\in\widehat{L}_t}\widehat{\mathrm{Ch}}_k$.
9: $t=t+1$.
10: **end while**
11: Set $\widehat{T}=t-1$, $\widehat{K}=\Sigma_{l=1}^{\widehat{T}}|\widehat{L}_l|$.
12: Obatin $\widehat{\Lambda}$ and $\widehat{\Psi}$ using (2) given $\widehat{K}$ and $\widehat{v}_1,\ldots,\widehat{v}_{\widehat{K}}$.

Output: $\widehat{T}$, $\widehat{K}$, $\widehat{L}_1,\ldots,\widehat{L}_{\widehat{T}}$, $\widehat{v}_1,\ldots,\widehat{v}_{\widehat{K}}$, $\widehat{\Lambda}$ and $\widehat{\Psi}$.

2.3 ICB Method for Learning Child Factors

From the overview of the proposed method described above, we see that the proposed method solves the learning problem by iteratively applying an ICB method to learn the child factors of each given factor. We now give the details of this method. We start with the ICB method for learning the child factors of Factor 1, i.e., the general factor. In this case, the main questions the ICB method answers are: (1) how many child factors does Factor 1 have? and (2) what variables load on each child factor? It is worth noting that when learning these from data, we need to account for the fact that each child factor can have an unknown number of descendant factors. However, with a divide-and-conquer spirit, we do not learn the structure of the descendant factors (i.e., the hierarchical structure of these descendant factors and the variables loading on them) of each child factor in this step because this structure is too complex to learn at once.

The ICB method answers the two questions above by learning a loading matrix $\widetilde{\Lambda}_1$ with zero patterns that encode the number and loading structure of the child factors of Factor 1. More specifically, $\widetilde{\Lambda}_1$ is searched among the space of loading matrices that satisfy certain zero constraints that encode a hierarchical factor model. This space is defined as

$$\mathcal{A}_1 = \cup_{c\in\{0,2,\ldots,c_{\max}\},d_1,\ldots,d_c\in\{1,\ldots,d_{\max}\}}\mathcal{A}^1(c,d_1,\ldots,d_c),$$

where, if $c \geq 2$, for a pre-specified constant $\tau > 0$,

$$\begin{aligned}\mathcal{A}^1(c,d_1,\ldots,d_c) =&\{A=(a_{ij})_{J\times(1+d_1+\cdots+d_c)}:\ \text{there exists a partition of } \{1,\ldots,J\},\ \text{denoted}\\ &\text{by } v_1^1,\ldots,v_c^1, \text{satisfying } \min\{v_1^1\}<\min\{v_2^1\}<\cdots<\min\{v_c^1\},\ \text{such that}\\ &A_{[v_s^1,\{j\}]}=\mathbf{0},\ \text{for all } s=1,\ldots,c,\ \text{and } j\notin\{1,2+\sum_{s'<s}d_{s'},3+\sum_{s'<s}d_{s'},\ldots,\\ &1+\sum_{s'\leq s}d_{s'}\}\ \text{and } |a_{ij}|\leq\tau,\ \ \text{for all } i=1,\ldots,J \text{ and } j=1,\ldots,1+\sum_{s=1}^{c}d_c.\},\end{aligned} \tag{3}$$

and, if $c = 0$, $\mathcal{A}^1(0) = \{A = (a_{ij})_{J\times 1} : |a_{ij}| \le \tau\}$. Here, $c_{\max}$ and $d_{\max}$ are pre-specified constants typically decided by domain knowledge. τ is a universal upper bound for the loading parameters, which is needed for technical reasons for our theory. The space $\mathcal{A}^1(c, d_1, \ldots, d_c)$ includes all possible loading matrices for a hierarchical factor structure, where Factor 1 has c child factors, and each child factor has $d_s - 1$ descendant factors. The space $\mathcal{A}_1$ is the union of all the possible $\mathcal{A}^1(c, d_1, \ldots, d_c)$ for different combinations of the numbers of child factors and their descendant factors.

For example, consider the hierarchical factor model example in Figure 1, for which $\widehat{v}_1 = \{1, \ldots, 16\}$. Then, the matrix

$$\Lambda_1 = \begin{pmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} & \lambda_{41} & \lambda_{51} & \lambda_{61} & \lambda_{71} & \lambda_{81} & \lambda_{91} & \lambda_{10,1} & \lambda_{11,1} & \lambda_{12,1} & \lambda_{13,1} & \lambda_{14,1} & \lambda_{15,1} & \lambda_{16,1} \\ \lambda_{12} & \lambda_{22} & \lambda_{32} & \lambda_{42} & \lambda_{52} & \lambda_{62} & \lambda_{72} & \lambda_{82} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_{13} & \lambda_{23} & \lambda_{33} & \lambda_{43} & \lambda_{53} & \lambda_{63} & \lambda_{73} & \lambda_{83} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_{14} & \lambda_{24} & \lambda_{34} & \lambda_{44} & \lambda_{54} & \lambda_{64} & \lambda_{74} & \lambda_{84} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{95} & \lambda_{10,5} & \lambda_{11,5} & \lambda_{12,5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{13,6} & \lambda_{14,6} & \lambda_{15,6} & \lambda_{16,6} \end{pmatrix}^\top \tag{4}$$

lies in space $\mathcal{A}^1(3, 3, 1, 1)$. This loading matrix is what the ICB method aims to find, as it has the same blockwise zero pattern (ignoring the zero constraints implied by the lower-layer factors) as the true loading pattern in (1) after reordering the columns of Λ in (1).

We search for the best possible loading matrix in $\mathcal{A}_1$ using the information criterion defined as:

$$\begin{aligned} \mathrm{IC}_1(c, d_1, \ldots, d_c) = \min_{\Lambda_1, \Psi_1} & l\left(\Lambda_1\Lambda_1^\top + \Psi_1, S\right) + p_1(\Lambda_1)\log N, \\ \text{s.t. } & \Lambda_1 \in \mathcal{A}^1(c, d_1, \ldots, d_c), \kappa_1 \le (\Psi_1)_{[\{i\},\{i\}]} \le \kappa_2, \\ & (\Psi_1)_{[\{i\},\{j\}]} = 0, i = 1, \ldots, |\widehat{v}_1|, j \ne i, \end{aligned} \tag{5}$$

where κ_1 and κ_2 are pre-specified lower and upper bounds for the unique variance, and

$$p_1(\Lambda_1) = \begin{cases} \Sigma_{s=1}^{c}(|v_s^1|d_s - d_s(d_s-1)/2) \text{ if } d_s \leq |v_s^1| \text{ for all } s = 1, \dots c, \\ \infty, \text{ otherwise,} \end{cases} \tag{6}$$

is a penalty on the number of free parametersfor a matrix Λ_1 in $\mathcal{A}^1(c, d_1, \dots, d_c)$. The penalty ensures that in the selected factor loadings, one plus the number of descendant factors of each child factor of Factor 1 will not exceed the number of items loading on the corresponding child factor.

Ideally, we hope to find the loading matrix in $\mathcal{A}_1$ that minimises $\mathrm{IC}_1(c, d_1, \dots, d_c)$ among all $c \in \{0, 2, \dots, c_{\max}\}$ and $d_1, \dots, d_c \in \{1, \dots, d_{\max}\}$. More specifically, we define

$$(\bar{c}_1, \bar{d}_1^1, \dots, \bar{d}_{\bar{c}_1}^1) = \underset{c\in\{0,2,\dots,c_{\max}\}, 1\leq d_s\leq d_{\max}, s=1,\dots,c}{\arg\min} \mathrm{IC}_1(c, d_1, \dots, d_c) \tag{7}$$

and further

$$\begin{aligned} (\bar{\Lambda}_1, \bar{\Psi}_1) = \underset{\Lambda_1, \Psi_1}{\arg\min}\ & l\left(\Lambda_1\Lambda_1^\top + \Psi_1, S\right) \\ \text{s.t. } & \Lambda_1 \in \mathcal{A}^1(\bar{c}_1, \bar{d}_1^1, \dots, \bar{d}_{\bar{c}_1}^1), \kappa_1 \leq (\Psi_1)_{[\{i\},\{i\}]} \leq \kappa_2 \\ & (\Psi_1)_{[\{i\},\{j\}]} = 0, i = 1, \dots, |\widehat{v}_1|, j \neq i. \end{aligned} \tag{8}$$

We determine the variables loading on each child factor of Factor 1 based on the zero pattern of $\bar{\Lambda}_1$.

However, we note that $\mathcal{A}_1$ is highly complex, and thus, enumerating all the possible loading matrices in $\mathcal{A}_1$ is computationally infeasible. In other words, while the quantities in (7) and (8) are well-defined mathematically, they cannot be computed within a reasonable time. In this regard, we develop a greedy search method, presented in Algorithm 2, for searching over the space $\mathcal{A}_1$. This greedy search method will output $\widehat{c}_1$ and $\widehat{v}_1^1, \dots, \widehat{v}_{\widehat{c}_1}^1$.

As shown in Theorem 2, with probability tending to 1, they are consistent estimates of the corresponding true quantities for the factors in this layer. In other words, this greedy search is theoretically guaranteed to learn the correct hierarchical factor structure. Moreover, Algorithm 2 also solves a similar optimization as (8) for loading matrices in $\mathcal{A}^1(\widehat{c}_1, \widehat{d}_1, \ldots, \widehat{d}_{\widehat{c}_1})$, from which we obtain a consistent estimate of the first column of the loading matrix, denoted by $\widetilde{\boldsymbol{\lambda}}_1$. So far, we have learned the factors in the second layer of the factor hierarchy.

For $t \geq 3$, suppose that the first to the $(t-1)$th layers have been successfully learned, and we now need to learn the factors in the tth layer. This problem can be decomposed into learning the child factors of each factor k in $\widehat{L}_{t-1} = \{k_{t-2}+1, \ldots, k_{t-1}\}$. At this moment, we have the estimated variables loading on Factor k, denoted by $\widehat{v}_k$, and a consistent estimate of the loading parameters for the factors in the first to the $(t-2)$th layer, denoted by $\widetilde{\boldsymbol{\lambda}}_i$, $i = 1, \ldots, k_{t-2}$, which are obtained as a by-product of the ICB method in the previous steps. We define $\widetilde{\Sigma}_{k,0} := \sum_{i=1}^{k_{t-2}} (\widetilde{\boldsymbol{\lambda}}_i)_{[\widehat{v}_k]} (\widetilde{\boldsymbol{\lambda}}_i)_{[\widehat{v}_k]}^\top$ and $S_k := S_{[\widehat{v}_k, \widehat{v}_k]}$. Similar to the learning of child factors of Factor 1, we define the possible space for the loading submatrix associated with the descendant factors of Factor k as

$$\mathcal{A}_k = \cup_{c \in \{0,2,\ldots,c_{\max}\}, d_1, \ldots, d_c \in \{1,\ldots,d_{\max}\}} \mathcal{A}^k(c, d_1, \ldots, d_c),$$

where, if $c \geq 2$, for the same constant $\tau > 0$ as in $\mathcal{A}_1$

$$\begin{aligned}
&\mathcal{A}^k(c, d_1, \ldots, d_c) \\
=&\{A = (a_{ij})_{|\widehat{v}_k| \times (1+d_1+\cdots+d_c)} : \text{ there exists a partition of } \{1, \ldots, |\widehat{v}_k|\}, \text{denoted} \\
&\text{by } v_1^k, \ldots, v_c^k, \text{ satisfying } \min\{v_1^k\} < \min\{v_2^k\} < \cdots < \min\{v_c^k\}, \text{ such that} \\
&A_{[v_s^k, \{j\}]} = \mathbf{0}, \text{ for all } s = 1, \ldots, c, \text{ and } j \notin \{1, 2 + \sum_{s'<s} d_{s'}, 3 + \sum_{s'<s} d_{s'}, \ldots, \\
&1 + \sum_{s' \leq s} d_{s'}\} \text{ and } |a_{ij}| \leq \tau \text{ for all } i = 1, \ldots, |\widehat{v}_k| \text{ and } j = 1, \ldots, 1 + \sum_{s=1}^{c} d_s\},
\end{aligned} \tag{9}$$

and, if $c = 0$, $\mathcal{A}^k(0) = \{A = (a_{ij})_{|\widehat{v}_k|\times 1} : |a_{ij}| \leq \tau\}$. Here, c and $d_1, \ldots, d_c$ have similar meanings as in $\mathcal{A}^1(c, d_1, \ldots, d_c)$. That is, $\mathcal{A}^k(c, d_1, \ldots, d_c)$ includes the corresponding loading submatrices when Factor k has c child factors, and each child factor has $d_s - 1$ descendant factors. It should be noted that, however, each matrix in $\mathcal{A}^k(c, d_1, \ldots, d_c)$ has only $|\widehat{v}_k|$ rows, while those in $\mathcal{A}^1(c, d_1, \ldots, d_c)$ have J rows. This is because, given the results from the previous steps, we have already estimated that factor k and its descendant factors are only loaded by the variables in $\widehat{v}_k$. Therefore, we only focus on learning the rows of the loading matrix that correspond to the variables in $\widehat{v}_k$ in the current task. Similar to $\mathrm{IC}_1(c, d_1, \ldots, d_c)$, we define

$$\begin{aligned} \mathrm{IC}_k(c, d_1, \ldots, d_c) = \min_{\Lambda_k, \Psi_k} & l\left(\widetilde{\Sigma}_{k,0} + \Lambda_k \Lambda_k^\top + \Psi_k, S_k\right) + p_k(\Lambda_k) \log N, \\ & \text{s.t. } \Lambda_k \in \mathcal{A}^k(c, d_1, \ldots, d_c), \kappa_1 \leq (\Psi_k)_{[\{i\},\{i\}]} \leq \kappa_2, \\ & (\Psi_k)_{[\{i\},\{j\}]} = 0, i = 1, \ldots, |\widehat{v}_k|, j \neq i, \end{aligned} \tag{10}$$

where

$$p_k(\Lambda_k) = \begin{cases} \Sigma_{s=1}^c (|v_s^k| d_s - d_s(d_s - 1)/2) \text{ if } d_s \leq |v_s^k| \text{ for all } s = 1, \ldots c, \\ \infty, \text{ otherwise} \end{cases} \tag{11}$$

is a penalty term.

Again, we use the greedy search algorithm, Algorithm 2, to search for the best possible Λ_k in $\mathcal{A}_k$. It outputs $\widehat{c}_k$ and $\widehat{v}_1^k, \ldots, \widehat{v}_{\widehat{c}_k}^k$, and an estimate of the kth column of the loading matrix, $\widetilde{\boldsymbol{\lambda}}_k$. Under some regularity conditions, Theorem 2 shows that $\widehat{c}_k$, $\widehat{v}_1^k, \ldots, \widehat{v}_{\widehat{c}_k}^k$, and $\widetilde{\boldsymbol{\lambda}}_k$ are consistent estimates of the corresponding true quantities.

Remark 4 *The penalty term in the proposed information criterion is essential for learning the correct hierarchical factor structure that satisfies the constraints in C1-C4. It avoids asymptotically rank-degenerated solutions for the loading matrix and, thus, avoids selecting*

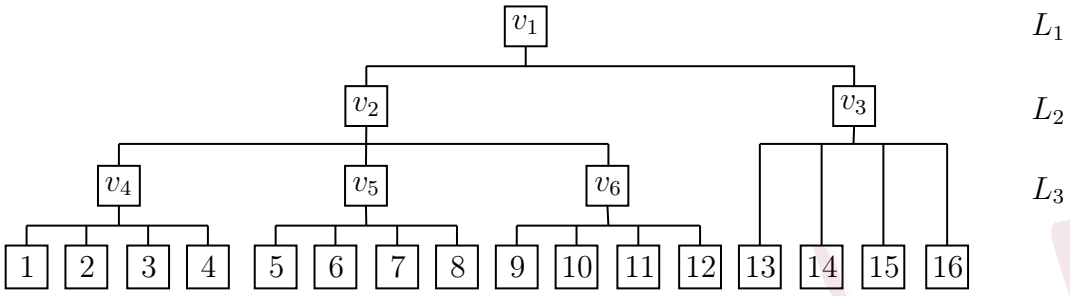


Figure 2: A correctly specified model with a redundant factor corresponding to v_2.

an over-specified hierarchical factor model with redundant parameters in the loading matrix and redundant factors, which affects the interpretation of the estimated factors. Consider the example in Figure 1. Without the penalty in the proposed information criterion, we may select the structure in Figure 2, which is still a correctly specified model but has a redundant factor (corresponding to v_2) that is not very interpretable.

We present the proposed greedy search algorithm for efficiently searching over the space $\mathcal{A}_k$ for each k. Recall that $\widetilde{\Sigma}_{k,0} := \sum_{i=1}^{k_{t-2}} (\widetilde{\boldsymbol{\lambda}}_i)_{[\widehat{v}_k]} (\widetilde{\boldsymbol{\lambda}}_i)_{[\widehat{v}_k]}^\top$ when $k \in \widehat{L}_{t-1}$ for $t \geq 3$. We further define $\widetilde{\Sigma}_{k,0}$ as a $J \times J$ zero matrix to cover the case when $t = 2$ and $k = 1$. We divide the search into two cases.

1. For $c = 0$, we simply compute

$$\begin{aligned} \widetilde{\mathrm{IC}}_{k,0} = \min_{\Lambda_k, \Psi_k} & l\left(\widetilde{\Sigma}_{k,0} + \Lambda_k \Lambda_k^\top + \Psi_k, S_k\right), \\ \text{s.t. } & \Lambda_k \in \mathcal{A}^k(0), \kappa_1 \leq (\Psi_k)_{[\{i\},\{i\}]} \leq \kappa_2, \\ & (\Psi_k)_{[\{i\},\{j\}]} = 0, i = 1, \ldots, |\widehat{v}_k|, j \neq i \end{aligned} \tag{12}$$

and use $(\widetilde{\Lambda}_{k,0}, \widetilde{\Psi}_{k,0})$ to denote the solution to (12). This is a relatively simple continuous optimization problem that a standard numerical solver can solve.

2. Set $d = d_{\max} + 2 - t$. For each $c \in \{2, \ldots, c_{\max}\}$, we perform the following steps:

(a) Solve the optimization in $\mathrm{IC}_k(c, d, \ldots, d)$. It is easy to check that the penalty term in

$\mathrm{IC}_k(c, d, \ldots, d)$ equals to $|\widehat{v}_k|d - cd(d-1)/2$, which does not depend on the loading matrix Λ_k as long as the number of item within each of the corresponding partition is no less than d. Therefore, the optimization problem becomes

$$\begin{aligned} &\min_{\Lambda_k, \Psi_k} l\left(\widetilde{\Sigma}_{k,0} + \Lambda_k \Lambda_k^\top + \Psi_k, S_k\right), \\ &\text{s.t. } \Lambda_k \in \mathcal{A}^k(c, d, \ldots, d), \kappa_1 \leq (\Psi_k)_{[\{i\},\{i\}]} \leq \kappa_2, \\ &\quad (\Psi_k)_{[\{i\},\{j\}]} = 0, i = 1, \ldots, |\widehat{v}_k|, j \neq i. \end{aligned} \tag{13}$$

Let $v_1^{k,c}, \ldots, v_c^{k,c}$ be the partition of $1, \ldots, |\widehat{v}_k|$ given by the solution to (13). We note that (13) is a discrete optimization problem, due to the combinatorial nature of the space $\mathcal{A}^k(c, d, \ldots, d)$. The theoretical properties in Theorem 2 are established under the ideal scenario that this optimization is solved exactly for all k. In reality, however, exactly solving (13) is computationally infeasible when J and c are large. To search for the solution to (13), we cast it into a continuous optimization problem with nonlinear zero constraints and solved by an augmented Lagrangian method; see Section 3 for the relevant details.

(b) Given the partition $v_1^{k,c}, \ldots, v_c^{k,c}$ from the previous step, we define the space for all $d_1, \ldots, d_c \in \{1, \ldots, d_{\max}\}$

$$\begin{aligned} &\widetilde{\mathcal{A}}^k(c, d_1, \ldots, d_c) \\ =&\{A = (a_{ij})_{|\widehat{v}_k| \times (1 + d_1 + \cdots + d_c)} : A_{[v_s^{k,c}, \{j\}]} = \mathbf{0}, \text{ for all } s = 1, \ldots, c, \text{ and} \\ &j \notin \{1, 2 + \sum_{s' < s} d_{s'}, 3 + \sum_{s' < s} d_{s'}, \ldots, 1 + \sum_{s' \leq s} d_{s'}\}, \text{ and } |a_{ij}| \leq \tau \text{ for} \\ &\text{all } i = 1, \ldots, |\widehat{v}_k|, j = 1, \ldots, 1 + \sum_{s=1}^{c} d_s\} \end{aligned} \tag{14}$$

for the same constant τ as in $\mathcal{A}^k(c, d_1, \ldots, d_c)$. We note that the space of $\widetilde{\mathcal{A}}^k(c, d_1, \ldots, d_c)$

is substantially smaller than $\mathcal{A}^k(c, d_1, \ldots, d_c)$ as the partition of the variables is fixed. Based on $\widetilde{\mathcal{A}}^k(c, d_1, \ldots, d_c)$, we define information criterion

$$\begin{aligned}\widetilde{\mathrm{IC}}_k(c, d_1, \ldots, d_c) = \min_{\Lambda_k, \Psi_k} l\left(\widetilde{\Sigma}_{k,0} + \Lambda_k\Lambda_k^\top + \Psi_k, S_k\right) + p_k(\Lambda_k)\log N, \\ \text{s.t. } \Lambda_k \in \widetilde{\mathcal{A}}^k(c, d_1, \ldots, d_c), \kappa_1 \le (\Psi_k)_{[\{i\},\{i\}]} \le \kappa_2, \\ (\Psi_k)_{[\{i\},\{j\}]} = 0, i = 1, \ldots, |\widehat{v}_k|, j \ne i.\end{aligned} \tag{15}$$

As the space $\widetilde{\mathcal{A}}^k(c, d_1, \ldots, d_c)$ is relatively simple, the optimization in (15) is a relatively simple continuous optimization problem that a standard numerical solver can solve.

(c) We then search for the best values for $d_1, \ldots, d_c$ for the given c. They are determined sequentially, one after another. More specifically, we first determine d_1 by

$$\widetilde{d}_1^c = \underset{1 \le d_1 \le \min(|v_1^{k,c}|, d)}{\arg\min} \widetilde{\mathrm{IC}}_k(c, d_1, \min(|v_2^{k,c}|, d), \ldots, \min(|v_c^{k,c}|, d)), \tag{16}$$

where we fix the value of $d_2, \ldots, d_c$ at $\min(|v_2^{k,c}|, d), \ldots, \min(|v_c^{k,c}|, d)$ and only vary the value of d_1. Solving (16) involves solving $\min(|v_1^{k,c}|, d)$ relatively simple continuous optimization problems. Then we proceed to d_2 and so on. For $s \ge 2$, suppose that we have learned $\widetilde{d}_1^c, \ldots, \widetilde{d}_{s-1}^c$, then d_s is determined by

$$\widetilde{d}_s^c = \underset{1 \le d_s \le \min(|v_s^{k,c}|, d)}{\arg\min} \widetilde{\mathrm{IC}}_k(c, \widetilde{d}_1^c, \ldots, \widetilde{d}_{s-1}^c, d_s, \min(|v_{s+1}^{k,c}|, d), \ldots, \min(|v_c^{k,c}|, d)),$$

where we fix $d_1, \ldots, d_{s-1}$ at their learned values and further fix $d_{s+1}, \ldots, d_c$ at the value of $\min(|v_{s+1}^{k,c}|, d), \ldots, \min(|v_c^{k,c}|, d)$.

(d) Given $\widetilde{d}_1^c, \ldots, \widetilde{d}_c^c$, we define

$$\widetilde{\mathrm{IC}}_{k,c} = \widetilde{\mathrm{IC}}_k(c, \widetilde{d}_1^c, \ldots, \widetilde{d}_c^c) \tag{17}$$

and $\widetilde{\Lambda}_{k,c}$, $\widetilde{\Psi}_{k,c}$ as the solution to (17).

The above steps yield $\widetilde{\mathrm{IC}}_{k,c}$, $c \in \{0, 2, \ldots, c_{\max}\}$. Then, we estimate the number of child factors of Factor k by the value of c that minimises the modified information criterion $\widetilde{\mathrm{IC}}_{k,c}$. That is, we let

$$\widehat{c}_k = \underset{c \in \{0,2,\ldots,c_{\max}\}}{\arg\min} \widetilde{\mathrm{IC}}_{k,c}.$$

Moreover, we define

$$\widehat{v}_s^k = \widehat{v}_k[v_s^{k,\widehat{c}_k}], s = 1, \ldots, \widehat{c}_k,$$

where $v_s^{k,\widehat{c}_k}$, $s = 1, \ldots, \widehat{c}_k$, is the partition of $\{1, \ldots, |\widehat{v}_k|\}$ learned above for $c = \widehat{c}_k$. Then $\widehat{v}_s^k$, $s = 1, \ldots, \widehat{c}_k$, give a partition of $\widehat{v}_k$, and we estimate that the sth child factor of Factor k is loaded by the variables in $\widehat{v}_s^k$. As a by-product, we obtain an estimate of the kth column of the loading matrix, denoted by $\widetilde{\boldsymbol{\lambda}}_k$, satisfying that $(\widetilde{\boldsymbol{\lambda}}_k)_{[\widehat{v}_k]}$ equals to the first column of $\widetilde{\Lambda}_{k,\widehat{c}_k}$ and $(\widetilde{\boldsymbol{\lambda}}_k)_{[\{1,\ldots,J\}\setminus\widehat{v}_k]}$ is a zero vector.

We summarise the steps described previously in Algorithm 2.

Remark 5 *$c \in \{0, 2, \ldots, c_{\max}\}$ represents the number of child factors of Factor k. In other words, $c_{\max}$ is an upper bound on the possible number of child factors of Factor k. On the one hand, we need to ensure that $c_{\max}$ is not too small so that Condition 9 is satisfied. On the other hand, we want to avoid $c_{\max}$ being too large to reduce the computational cost. Since the true value of c should satisfy constraints C3 and C4 in Section 2.1, $c_{\max}$ should be no more than $\lfloor |\widehat{v}_k|/3 \rfloor$ when $|\widehat{v}_k| \geq 7$ and $c_{\max} = 0$, when $|\widehat{v}_k| \leq 6$, where $\lfloor \cdot \rfloor$ is the floor function that returns the greatest integer less than or equal to the input. In the simulation study in Section 4, we set $c_{\max} = \min(4, \lfloor |\widehat{v}_k|/3 \rfloor)$ when $|\widehat{v}_k| \geq 7$ and $c_{\max} = 0$ when $|\widehat{v}_k| \leq 6$, which, according to the data generation model, is an upper bound for the value of c. For the real data analysis in Section 5, since the true structure is unknown, we set $c_{\max} = \min(6, \lfloor |\widehat{v}_k|/3 \rfloor)$ when $|\widehat{v}_k| \geq 7$ and $c_{\max} = 0$ when $|\widehat{v}_k| \leq 6$ as a more conservative choice than that of $c_{\max}$*

Algorithm 2 Information-Criterion-based method

Input: $\widehat{v}_k$, $c_{\max}, d_{\max} \in \mathbb{N}^+$, $\widetilde{\Sigma}_{k,0}$, S_k and layer t.

1: Set $d = \min(|\widehat{v}_k|, d_{\max} + 2 - t)$.

2: Solve $\widetilde{\mathrm{IC}}_{k,0}$ defined in (12). Let $(\widetilde{\Lambda}_{k,0}, \widetilde{\Psi}_{k,0})$ as the solution to $\widetilde{\mathrm{IC}}_{k,0}$.

3: **for** $c = 2, 3, \ldots, c_{\max}$ **do**

4: Solve the optimization problem (13). Set $v_1^{k,c}, \ldots, v_c^{k,c}$ as the partition of $\{1, \ldots, |\widehat{v}_k|\}$ by the solution to (13).

5: **for** $s = 1, \ldots, c$ **do**

6: Compute

$$\widetilde{d}_s^c = \underset{1 \le d_s \le \min(|v_s^{k,c}|, d)}{\arg\min} \widetilde{\mathrm{IC}}_k(c, \widetilde{d}_1^c, \ldots, \widetilde{d}_{s-1}^c, d_s, \min(|v_{s+1}^{k,c}|, d), \ldots, \min(|v_c^{k,c}|, d)),$$

where $\widetilde{\mathrm{IC}}_k$ is defined in (15).

7: **end for**

8: Define $\widetilde{\mathrm{IC}}_{k,c} = \widetilde{\mathrm{IC}}_k(c, \widetilde{d}_1^c, \ldots, \widetilde{d}_c^c)$ and $(\widetilde{\Lambda}_{k,c}, \widetilde{\Psi}_{k,c})$ as the solution to $\widetilde{\mathrm{IC}}_{k,c}$.

9: **end for**

10: Define $\widehat{c}_k = \arg\min_{c \in \{0,2,3,\ldots,c_{\max}\}} \widetilde{\mathrm{IC}}_{k,c}$.

11: Set $\widetilde{v}_1^k, \ldots, \widetilde{v}_{\widehat{c}_k}^k$ be the partition of $\{1, \ldots, |\widehat{v}_k|\}$ associated with $\widetilde{\Lambda}_{k,\widehat{c}_k}$. Define the partition of $\widehat{v}_k$ by $\widehat{v}_1^k = \widehat{v}_k[\widetilde{v}_1^k], \ldots, \widehat{v}_{\widehat{c}_k}^k = \widehat{v}_k[\widetilde{v}_{\widehat{c}_k}^k]$.

12: Define $\widetilde{\boldsymbol{\lambda}}_k$ following that $(\widetilde{\boldsymbol{\lambda}}_k)_{[\widehat{v}_k]}$ equals to the first column of $\widetilde{\Lambda}_{k,\widehat{c}_k}$ and $(\widetilde{\boldsymbol{\lambda}}_k)_{[\{1,\ldots,J\} \setminus \widehat{v}_k]}$ is a zero vector.

Output: $\widehat{c}_k$, $\widehat{v}_1^k, \ldots, \widehat{v}_{\widehat{c}_k}^k$ and $\widetilde{\boldsymbol{\lambda}}_k$.

for the simulation study. In practice, we may adjust our choice based on prior knowledge about the hierarchical factor structure.

Remark 6 *The input hyperparameter $d_{\max}$ is an upper bound of one plus the number of descendant factors of the factors in the second layer. When learning the factors on the tth layer for $t \geq 3$, we use $d_{\max} + 2 - t$ as an upper bound of one plus the number of descendant factors of the factors in the $(t+1)$th layer, as the number of descendant factors each factor has tends to decrease as t increases. Similar to the choice of $c_{\max}$, we want to choose a $d_{\max}$ that is neither too large nor too small. In the simulation study in Section 4, we start with $d_{\max} = 6$ when learning the factors in the second layer. In the real data analysis in Section 5, we start with $d_{\max} = 10$. In practice, we may adjust this choice based on the problem size (e.g., the number of variables) and prior knowledge of the hierarchical factor structure.*

Remark 7 *Efficiently learning the hierarchical structure from data is challenging due to the super-exponential growth of the search space with the number of items J, which creates a significant computational bottleneck. To overcome the computational issue, we convert the combinatorial optimization problems in* (5) *and* (10) *into the continuous optimization problems in* (13)*. A similar constraint-based continuous optimization method is proposed for learning directed acyclic graphs (DAGs) in Zheng et al. (2018) and the bi-factor model in Qiao et al. (2025). By integrating continuous optimization techniques with a breadth-first search strategy, our approach (presented in Algorithms 1 and 2) requires solving only $\mathcal{O}(Jc_{\max}^2 d_{\max})$ continuous optimization problems, thus significantly improving the computational efficiency.*

2.4 Theoretical Results

We now provide theoretical guarantees for the proposed method based on Algorithms 1 and 2. We start with introducing some notation. We use $\|\cdot\|_F$ to denote the Frobenius norm of any matrix and $\|\cdot\|$ as the Euclidean norm of any vector. We also use the notation

$a_N = O_{\mathbb{P}}(b_N)$ to denote that a_N/b_N is bounded in probability. In addition to the conditions required for the identifiability of the true hierarchical factor model, we additionally require Conditions 5–9 to ensure the proposed method is consistent.

Condition 5 *For any factor i with $Ch_i^* \neq \emptyset$ and any $j \in v_i^*$, there exist $E_1, E_2 \subset v_i^* \setminus \{j\}$ with $|E_1| = |E_2| = 1 + |D_i^*|$ and $E_1 \cap E_2 = \emptyset$, such that $\Lambda^*_{[E_1,\{i\}\cup D_i^*]}$ and $\Lambda^*_{[E_2,\{i\}\cup D_i^*]}$ are of full rank.*

Condition 6 *For any factor i with $Ch_i^* \neq \emptyset$ and any $k \in Ch_i^*$, there exist $E_1, E_2 \subset v_k^*$ with $|E_1| = 2 + |D_k^*|$, $|E_2| = 1 + |D_k^*|$ and $E_1 \cap E_2 = \emptyset$ such that $\Lambda^*_{[E_1,\{i,k\}\cup D_k^*]}$ and $\Lambda^*_{[E_2,\{k\}\cup D_k^*]}$ are of full rank.*

Condition 7 $\|S - \Sigma^*\|_F = O_{\mathbb{P}}(1/\sqrt{N})$.

Condition 8 *The true loading parameters and unique variance parameters satisfy $|\lambda_{ij}^*| \leq \tau$ and $\kappa_1 \leq \psi_i^* \leq \kappa_2$ for all i, j, where τ, κ_1 and κ_2 are constraints used in the ICB method.*

Condition 9 *When learning the child factors of each true factor k, the constants $c_{\max}$ and $d_{\max}$ are chosen such that $c_{\max} \geq |Ch_k^*|$ and $d_{\max} \geq \max_{s \in Ch_k^*} |D_s^*| + 1$.*

Theorem 2 *Suppose that Conditions 1,3, and 5–9 hold. Then, the outputs from Algorithm 1 are consistent. That is, as N goes to infinity, the probability of $\widehat{T} = T$, $\widehat{K} = K$, $\widehat{L}_t = L_t$, $t = 1, \ldots, T$, and $\widehat{v}_i = v_i^*$, $i = 1, \ldots, K$ goes to 1, and $\|\widehat{\Lambda} - \Lambda^* \widehat{Q}\|_F = O_{\mathbb{P}}(1/\sqrt{N})$ and $\|\widehat{\Psi} - \Psi^*\|_F = O_{\mathbb{P}}(1/\sqrt{N})$, where $\widehat{Q} \in \mathcal{Q}$ is the diagonal matrix with diagonal entries consisting of the signs of the corresponding entries of $\widehat{\Lambda}^\top \Lambda^*$.*

Theorem 2 guarantees that the true hierarchical factor structure can be consistently learned from data and its parameters can be consistently estimated after adjusting the sign for each column of the loading matrix by $\widehat{Q}$.

Remark 8 *It should be noted that in Theorem 2, Algorithm 1 applies Algorithm 2, which involves some nontrivial optimization problems, including a discrete optimization problem* (13)*. The theorem is established under the oracle scenario that these optimizations are always solved successfully. However, we should note that this cannot be achieved by polynomial-time algorithms due to the complexity of these optimizations.*

Remark 9 *Theorem 2 does not explicitly require Condition 2, because Condition 5 is a stronger condition that implies Condition 2, as shown in Lemma 4 in the Appendix S4. In fact, Condition 5 is sufficient for Condition 4, which further implies Condition 2. We need a stronger condition (i.e., Condition 5) here, for distinguishing between the loading structure and the unique variance at each stage of recursion. Similar to Condition 3, this condition imposes further requirements on the number of child factors and the number of descendant factors a factor can have. More specifically, for such a partition to exist, we need* $|v_i^*| \geq 2|D_i^*| + 3$*. Other than that, the full-rank requirement is easily satisfied by most hierarchical factor models. Similar to Condition 5, Condition 6 also requires* $|v_i^*| \geq 2|D_i^*|+3$*. This condition plays a central role in ensuring that Step 6 in Algorithm 2 is valid. Condition 7 is very mild. It is automatically satisfied when the sample covariance matrix is constructed using independent and identically distributed observations from the true model, and all the fourth-order moments of the i.i.d. data are finite. Condition 8 requires the true loading and unique variance parameters to satisfy the same boundedness constraints as in the ICB method in Algorithm 2. Theoretically, these constraints ensure that the parameter space is compact, which is needed for bounding the differences in the loss function of different models. Empirically, we notice that the ICB method works well even without these constraints, and thus omit these constraints in the computation. Condition 9 requires that* $c_{\max}$ *and* $d_{\max}$ *are chosen sufficiently large so that the search space covers the true model.*

3 Computation

As mentioned previously, the optimization problem in $\mathrm{IC}_k(c, d, \ldots, d)$ in Algorithm 2 can be cast into a continuous optimization problem and solved by an augmented Lagrangian method(ALM). In what follows, we provide the details.

With slight abuse of notation, we use the reparameterization of the unique variance matrix such that $\Psi_k = \mathrm{diag}(\boldsymbol{\psi}_k^2)$, where $\mathrm{diag}(\cdot)$ is a function that converts a vector to a diagonal matrix with the diagonal entries filled by the vector. Here, $\boldsymbol{\psi}_k^2 = \{\psi_{k1}^2, \ldots, \psi_{k,|\widehat{v}_k|}^2|\}$ is a $|\widehat{v}_k|$-dimensional vector for $\psi_{k1}, \ldots, \psi_{k,|\widehat{v}_k|} \in \mathbb{R}$. We further let $\mathcal{B}_s = \{2+(s-1)d, \ldots, 1+sd\}$ for $s = 1, \ldots, c$. We note that, up to a relabelling of the partition sets or, equivalently, dropping the label ordering constraint $\min\{v_1^k\} < \min\{v_2^k\} < \cdots < \min\{v_c^k\}$, the set $\mathcal{A}^k(c, d, \ldots, d)$ can be rewritten as

$$\{A = (a_{ij})_{|\widehat{v}_k| \times (1+cd)} : a_{ij}a_{ij'} = 0 \text{ for } i = 1, \ldots, |\widehat{v}_k|, j \in \mathcal{B}_s, j' \in \mathcal{B}_{s'}, s \neq s', |a_{ij}| \leq \tau\}.$$

Therefore, we can solve $\mathrm{IC}_k(c, d, \ldots, d)$ by solving the following continuous optimization problem with nonlinear zero constraints:

$$\begin{aligned} \bar{\Lambda}_{k,c}, \bar{\boldsymbol{\psi}}_{k,c} = \underset{\Lambda_k, \boldsymbol{\psi}_k}{\arg\min}\, l\left(\widetilde{\Sigma}_{k,0} + \Lambda_k(\Lambda_k)^\top + \mathrm{diag}(\boldsymbol{\psi}_k^2), S_k\right) \\ \text{s.t. } \lambda_{k,ij}\lambda_{k,ij'} = 0 \text{ for } i = 1, \ldots, |\widehat{v}_k|, j \in \mathcal{B}_s^k, j' \in \mathcal{B}_{s'}^k, s \neq s'. \end{aligned} \tag{18}$$

Here, the constraints on the loading and unique variance parameters are omitted for simplicity, as these constraints are always satisfied when we set τ and κ_2 to be sufficiently large and κ_1 to be sufficiently small. Once this optimization is solved, then for each i, there is one and only one $\mathcal{B}_s$ such that $(\bar{\Lambda}_{k,c})_{[\{i\},\mathcal{B}_s]} \neq \mathbf{0}$. Therefore, we obtain a partition of $1, \ldots, |\widehat{v}_k|$ by the sets

$$\{i : (\bar{\Lambda}_{k,c})_{[\{i\},\mathcal{B}_s]} \neq \mathbf{0}\}, s = 1, \ldots, c.$$

We obtain $v_1^{k,c}, \ldots, v_c^{k,c}$ by reordering $\{i : (\bar{\Lambda}_{k,c})_{[\{i\},\mathcal{B}_s]} \neq \mathbf{0}\}, s = 1, \ldots, c$ to satisfy the constraint on the labels of these sets.

We solve (18) by the ALM algorithm (see, e.g., Bertsekas, 2014), which is a standard approach to such problems. This method finds a solution to (18) by solving a sequence of unconstrained optimization problems. More specifically, in the tth iteration, $t = 1, 2, \ldots$, the ALM minimizes an augmented Lagrangian function that is constructed based on the result of the previous iteration. Details of the ALM are given in Algorithm 3 below, where the function $h(\cdot)$ returns the second largest values of a vector. The updating rule of $\beta_{jii'}^{(t)}$ and $c^{(t)}$ follows equations (1) and (47) in Chapter 2.2 of Bertsekas (2014), and the convergence of Algorithm 3 to a stationary point of (18) is guaranteed by Proposition 2.7 of Bertsekas (2014). We follow the recommended choices of $c_\theta = 0.25$ and $c_\sigma = 10$ in Bertsekas (2014) for the tuning parameters in Algorithm 3.

We remark on the stopping criterion in the implementation of Algorithm 3. We monitor the convergence of the algorithm based on two criteria: (1) the change in parameter values in two consecutive steps, measured by

$$\left(\|\Lambda_k^{(t)} - \Lambda_k^{(t-1)}\|_F^2 + \|\boldsymbol{\psi}_k^{(t)} - \boldsymbol{\psi}_k^{(t-1)}\|^2\right)^{1/2} / \left(|\widehat{v}_k|(2+d)\right)^{1/2},$$

and (2) the distance between the estimate and the space $\mathcal{A}^k(c, d, \ldots, d)$ measured by

$$\max_{i \in \{1,\ldots,|\widehat{v}_k|\}} h(\max_{j \in \mathcal{B}_1} |\lambda_{k,ij}^{(t)}|, \max_{j \in \mathcal{B}_2} |\lambda_{k,ij}^{(t)}| \ldots, \max_{j \in \mathcal{B}_c} |\lambda_{k,ij}^{(t)}|).$$

When both criteria are below their pre-specified thresholds, δ_1 and δ_2, respectively, we stop the algorithm. Let M be the last iteration number. Then the selected partition of $\{1, \ldots, \widehat{v}_k\}$, denoted by $v_1^{k,c}, \ldots, v_c^{k,c}$, is given by $v_s^{k,c} = \{j : |\lambda_{k,ij}^{(M)}| < \delta_2 \text{ for all } j \notin \mathcal{B}_s\}$. For the analyses in Sections 4 and 5, we choose $\delta_1 = \delta_2 = 0.01$.

Algorithm 3 An augmented Lagrangian method for solving $\mathrm{IC}_k(c, d, \ldots, d)$

Input: Initial value $\Lambda^{(0)}$ and $\boldsymbol{\psi}^{(0)}$, initial Lagrangian parameters $\beta_{ijj'}^{(0)}$ for $i = 1, \ldots, |\widehat{v}_k|$, $j \in \mathcal{B}_s$, $j' \in \mathcal{B}_{s'}$ and $s \neq s'$, initial penalty coefficient $c^{(0)} > 0$, constants $c_\theta \in (0, 1)$ and $c_\sigma > 1$, tolerances $\delta_1, \delta_2 > 0$, maximal iteration number $M_{\max}$.

1: **for** $t = 1, 2, \ldots, M_{\max}$ **do**

2: Solve the following problem:

$$\begin{aligned}\Lambda_k^{(t)}, \boldsymbol{\psi}_k^{(t)} = \arg\min_{\Lambda_k, \boldsymbol{\psi}_k} \quad & l\left(\widetilde{\Sigma}_{k,0} + \Lambda_k(\Lambda_k)^\top + \mathrm{diag}(\boldsymbol{\psi}_k), S_k\right) \\ & + \left(\sum_{i=1}^{|\widehat{v}_k|} \sum_{j \in \mathcal{B}_s, j' \in \mathcal{B}_{s'}, s \neq s'} \beta_{ijj'}^{(t)} \lambda_{k,ij} \lambda_{k,ij'}\right) \\ & + \frac{1}{2} c^{(t)} \left(\sum_{i=1}^{|\widehat{v}_k|} \sum_{j \in \mathcal{B}_s, j' \in \mathcal{B}_{s'}, s \neq s'} (\lambda_{k,ij} \lambda_{k,ij'})^2\right).\end{aligned}$$

3: Update $\beta_{ijj'}^{(t)}$ and $c^{(t)}$ according to equations (19) and (20)

$$\beta_{ijj'}^{(t)} = \beta_{ijj'}^{(t-1)} + c^{(t-1)} \lambda_{k,ij}^{(t)} \lambda_{k,ij'}^{(t)}, \tag{19}$$

and

$$c^{(t)} = \begin{cases} c_\sigma c^{(t-1)} & \text{if } \left(\sum_{i=1}^{|\widehat{v}_k|} \sum_{j \in \mathcal{B}_s, j' \in \mathcal{B}_{s'}, s \neq s'} (\lambda_{k,ij}^{(t)} \lambda_{k,ij'}^{(t)})^2\right)^{1/2} \\ & \quad > c_\theta \left(\sum_{i=1}^{|\widehat{v}_k|} \sum_{j \in \mathcal{B}_s, j' \in \mathcal{B}_{s'}, s \neq s'} (\lambda_{k,ij}^{(t-1)} \lambda_{k,ij'}^{(t-1)})^2\right)^{1/2}, \\ c^{(t-1)} & \text{otherwise.} \end{cases} \tag{20}$$

4: **if**

$$\left(\|\Lambda_k^{(t)} - \Lambda_k^{(t-1)}\|_F^2 + \|\boldsymbol{\psi}_k^{(t)} - \boldsymbol{\psi}_k^{(t-1)}\|^2\right)^{1/2} / \left(|\widehat{v}_k|(2+d)\right)^{1/2} < \delta_1$$

and

$$\max_{i \in \{1, \ldots, |\widehat{v}_k|\}} h(\max_{j \in \mathcal{B}_1} |\lambda_{k,ij}^{(t)}|, \max_{j \in \mathcal{B}_2} |\lambda_{k,ij}^{(t)}| \ldots, \max_{j \in \mathcal{B}_c} |\lambda_{k,ij}^{(t)}|) < \delta_2,$$

then

5: **return** $\Lambda_k^{(t)}, \boldsymbol{\psi}_k^{(t)}$.

6: **Break**

7: **end if**

8: **end for**

Output: $\Lambda_k^{(t)}, \boldsymbol{\psi}_k^{(t)}$.

Algorithm 3 can suffer from slow convergence when the penalty terms become large, resulting in an ill-conditioned optimization problem. When the algorithm does not converge within $M_{\max}$ iterations, we suggest restarting the algorithm, using the current parameter value as a warm start. We set $M_{\max} = 100$ in the simulation study in Section 4 and the real data analysis in Section 5 and keep the maximum number of restarting times to be five. In addition, since the optimization problem (18) is non-convex, Algorithm 3 may only converge to a local optimum and this local solution may not satisfy condition C4. Therefore, we recommend running it with multiple random starting points and then finding the best solution that satisfies condition C4. In our implementation, each time to solve (18), we start by running Algorithm 3 100 times, each with a random starting point. If more than 50 of the solutions satisfy C4, then we stop and proceed to Steps 5–8 in Algorithm 2. Otherwise, we rerun Algorithm 3 100 times with random starting points, until either 50 solutions satisfy C4 or the algorithm has been restarted five times.

4 Simulation Study

In this section, we examine the recovery of the hierarchical structure as well as the accuracy in estimating the loading matrix and the unique variance matrix of the proposed method. Suppose that $\widehat{v}_1, \ldots, \widehat{v}_{\widehat{K}}$ are the estimated sets of variables loading on each factor, where $\widehat{K}$ is the estimated number of factors, $\widehat{\Lambda}$ is the estimated loading matrix and $\widehat{\Psi}$ is the estimated unique variance matrix. To examine the recovery of the hierarchical factor structure, we measure the matching between the true sets of variables loading on each factor and the estimated sets of variables. More specifically, the following evaluation criteria are considered:

1. Exact Match Criterion (EMC): $\mathbf{1}(\widehat{K} = K) \prod_{k=1}^{\min(\widehat{K},K)} \mathbf{1}(\widehat{v}_k = v_k^*)$, which equals to 1 when the hierarchical structure is fully recovered and 0 otherwise.

2. Layer Match Criterion (LMC): $\mathbf{1}(\{\widehat{v}_k\}_{k\in\widehat{L}_t} = \{v_k^*\}_{k\in L_t})$, which is defined for each layer

t. It equals 1 if the sets of variables loading on the factors in the tth layer are correctly learned and 0 otherwise for $t = 1, \ldots, T$.

We then examine the accuracy in estimating the loading matrix and the unique variance matrix. We calculate the mean square error(MSE) for $\widehat{\Lambda}$ and $\widehat{\Psi}$, after adjusting for the sign indeterminacy shown in Theorem 1. More specifically, recall that $\mathcal{Q}$ is the set of sign flip matrices defined in Theorem 1. When the proposed method outputs a correct estimate of the hierarchical structure (i.e. EMC = 1), we define the MSEs for $\widehat{\Lambda}$ and $\widehat{\Psi}$ as $\text{MSE}_\Lambda = \min_{Q\in\mathcal{Q}} \|\widehat{\Lambda} - \Lambda^* Q\|_F^2/(JK)$, and $\text{MSE}_\Psi = \|\widehat{\Psi} - \Psi^*\|_F^2/J$.

We consider the following hierarchical factor structure shown in Figure 3 with the number of variables $J \in \{36, 54\}$, the number of layers $T = 4$, the number of factors $K = 10$, $L_1 = \{1\}$, $L_2 = \{2, 3\}$, $L_3 = \{4, \ldots, 8\}$, $L_4 = \{9, 10\}$ and $v_1^* = \{1, \ldots, J\}$, $v_2^* = \{1, \ldots, J/3\}$, $v_3^* = \{1 + J/3, \ldots, J\}$, $v_4^* = \{1, \ldots, J/6\}$, $v_5^* = \{1 + J/6, \ldots, J/3\}$, $v_6^* = \{1 + J/3, \ldots, 5J/9\}$, $v_7^* = \{1 + 5J/9, \ldots, 7J/9\}$, $v_8^* = \{1 + 7J/9, \ldots, J\}$, $v_9^* = \{1 + J/3, \ldots, 4J/9\}$, $v_{10}^* = \{1 + 4J/9, \ldots, 5J/9\}$. In the data generation model, Ψ^* is either a $J \times J$ identity matrix or $\Psi^* = \text{diag}(\psi_1^{*2}, \ldots, \psi_J^{*2})$ with $\psi_j^*, j = 1, \ldots, J$, i.i.d., following a Uniform$(0.5, 1.5)$ distribution, and Λ^* is generated by

$$\lambda_{jk}^* = \begin{cases} u_{jk} \text{ if } k = 1; \\ 0 \text{ if } k > 1, j \notin v_k^*; \\ (1 - 2x_{jk})u_{jk} \text{ if } k > 1, j \in v_k^*, \end{cases} \tag{21}$$

for $j = 1, \ldots, J$, and $k = 1, \ldots, K$. Here, u_{jk}s are i.i.d., following a Uniform$(0.5, 2)$ distribution and x_{jk}s are i.i.d., following a Bernoulli(0.5) distribution. For each value of J, we generate the true loading matrix Λ^* and the true unique variance matrix Ψ^* once and use it for all its simulations.

We consider 8 simulation settings, given by the combinations of $J = 36, 54$, two sample sizes, $N = 500, 2000$ and two generating processes of Ψ^*. For each setting, 100 independent simulations are generated. The results of learning the hierarchical factor structure and

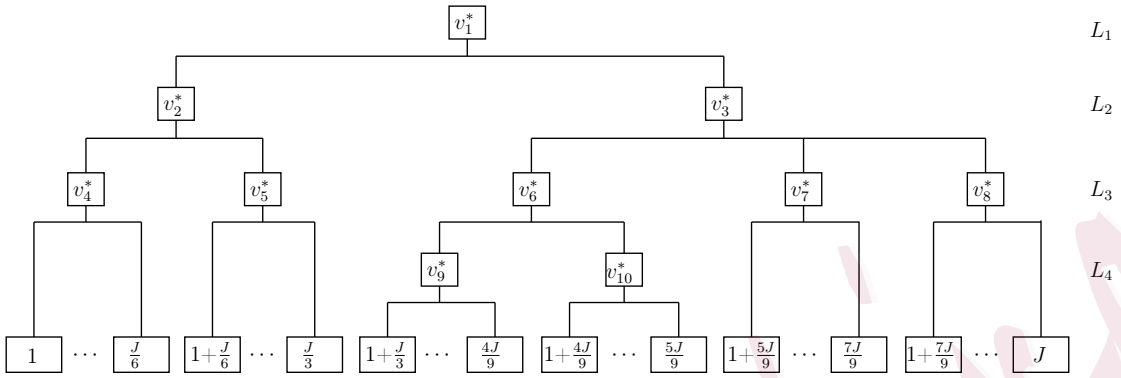


Figure 3: The hierarchical factor structure in the simulation study.

Table 1: The accuracy of the overall estimates of hierarchical structure and parameters.

Ψ	J	N	$\bar{K}$	$\bar{T}$	EMC	$\text{MSE}_{\widehat{\Lambda}}$	$\text{MSE}_{\widehat{\Psi}}$
Identity	36	500	10.01	4.00	0.98	2.90×10^{-3}	1.54×10^{-2}
		2000	10.04	4.00	0.97	0.74×10^{-3}	3.99×10^{-3}
	54	500	10.05	4.00	0.97	2.65×10^{-3}	6.45×10^{-3}
		2000	10.02	4.00	0.99	0.66×10^{-3}	1.63×10^{-3}
Heterogeneous	36	500	10.00	4.00	1.00	3.34×10^{-3}	1.45×10^{-2}
		2000	10.00	4.00	1.00	0.80×10^{-3}	3.15×10^{-3}
	54	500	10.01	4.00	0.99	2.69×10^{-3}	7.99×10^{-3}
		2000	10.04	4.00	0.98	0.67×10^{-3}	2.10×10^{-3}

estimating the model parameters are shown in Tables 1 and 2. In these tables, $\bar{K}$ and $\bar{T}$ report the average values of $\widehat{K}$ and $\widehat{T}$, respectively, and $|\widehat{L}_2|$, $|\widehat{L}_3|$ and $|\widehat{L}_4|$ report the average numbers of factors in $\widehat{L}_2$, $\widehat{L}_3$ and $\widehat{L}_4$, respectively. As shown in Table 1, the proposed method can accurately recover the true hierarchical factor structure more than 97% of the time under all the settings, with the highest accuracy of 100% achieved under the setting with $J = 36$ and heterogeneous diagonal entries in the unique variance matrix. The MSE of $\widehat{\Lambda}$ and $\widehat{\Psi}$ show that the loading matrix and the unique variance matrix are accurately estimated when the hierarchical structure is correctly learned.

Table 2: The accuracy of the estimated hierarchical structure on each layer.

Ψ	J	N	$\lvert\widehat{L}_2\rvert$	LMC_2	$\lvert\widehat{L}_3\rvert$	LMC_3	$\lvert\widehat{L}_4\rvert$	LMC_4
Identity	36	500	2.00	1.00	4.99	0.98	2.02	0.99
		2000	2.00	1.00	4.97	0.97	2.07	0.97
	54	500	2.00	1.00	4.97	0.97	2.08	0.97
		2000	2.00	1.00	4.99	0.99	2.03	0.99
Heterogeneous	36	500	2.00	1.00	5.00	1.00	2.00	1.00
		2000	2.00	1.00	5.00	1.00	2.00	1.00
	54	500	2.01	0.99	5.00	0.99	2.00	1.00
		2000	2.00	0.99	4.99	0.98	2.05	0.98

5 Real Data Analysis

We apply the exploratory hierarchical factor analysis to a personality assessment dataset based on the International Personality Item Pool (IPIP) NEO 120 personality inventory (Johnson, 2014). We investigate the structure of the Agreeableness scale based on a sample of 1655 UK participants aged between 30 and 40 years. This scale consists of 24 items, which are designed to measure six facets of Agreeableness, including Trust (A1), Morality (A2), Altruism (A3), Cooperation (A4), Modesty (A5), and Sympathy (A6). The responses to all the items are recorded on a 1-5 Likert scale and treated as continuous variables. The reversely worded items have been reversely scored so that a larger score always means a higher level of agreeableness. There is no missing data. Detailed descriptions of the items can be found in the Appendix S7. The learned hierarchical factor structure, which has four layers and ten factors, is shown in Figure 4, and the estimated loading matrix $\widehat{\Lambda}$ is shown in Table 3.

We now examine the learned model. We notice that the loadings on Factor 1 are all positive, except for item 18, which has a small negative loading. Thus, Factor 1 may be interpreted as a general Agreeableness factor. Factor 2 is loaded positively by all items designed to measure the Trust, Altruism, and Sympathy facets. Therefore, it may be inter-

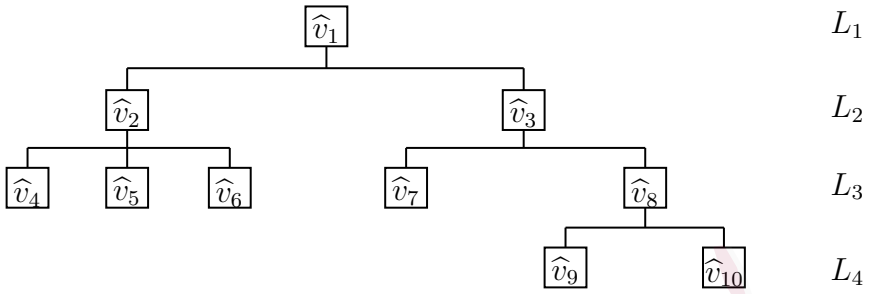


Figure 4: The hierarchical factor structure from the real data analysis

Table 3: The estimated loading matrix $\widehat{\Lambda}$ with four layers and ten factors.

Item	Facet	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
1	A1	0.47	0.14	0	0.70	0	0	0	0	0	0
2	A1	0.36	0.23	0	0.59	0	0	0	0	0	0
3	A1	0.32	0.22	0	0.69	0	0	0	0	0	0
4	A1	0.59	0.11	0	0.64	0	0	0	0	0	0
5	A2	0.44	0	0.55	0	0	0	0.61	0	0	0
6	A2	0.46	0	0.27	0	0	0	0.34	0	0	0
7	A2	0.56	0	0.42	0	0	0	0.61	0	0	0
8	A2	0.45	0	0.21	0	0	0	0	−0.10	0.05	0
9	A3	0.26	0.37	0	0	0.48	0	0	0	0	0
10	A3	0.26	0.54	0	0	0.16	0	0	0	0	0
11	A3	0.46	0.51	0	−0.11	0	0	0	0	0	0
12	A3	0.43	0.34	0	0	0.21	0	0	0	0	0
13	A4	0.21	0	0.48	0	0	0	0	−0.02	0	0.42
14	A4	0.46	0	0.14	0	0	0	0	−0.15	0	0.66
15	A4	0.63	0	0.21	0	0	0	0	−0.00	0	0.48
16	A4	0.57	0	0.34	0	0	0	0	−0.21	0	0.20
17	A5	0.36	0	0.43	0	0	0	0	0.68	−0.06	0
18	A5	−0.09	0	0.46	0	0	0	0	0.70	0.48	0
19	A5	0.06	0	0.49	0	0	0	0	0.86	0.41	0
20	A5	0.29	0	0.43	0	0	0	0	0.15	0	0.13
21	A6	0.23	0.44	0	0	0	0.75	0	0	0	0
22	A6	0.22	0.56	0	0	0	0.41	0	0	0	0
23	A6	0.41	0.57	0	−0.04	0	0	0	0	0	0
24	A6	0.34	0.40	0	0	0	0.38	0	0	0	0

preted as a higher-order factor of these facets. Factors 4–6 are child factors of Factor 2, and based on the loading patterns, they may be interpreted as the Trust, Altruism, and Sympathy factors, respectively. It is worth noting that items 11 (“Am indifferent to the feelings of others") and 23 (“Am not interested in other people’s problems"), which are designed to measure the Altruism and Sympathy facets, now load weakly and negatively on Factor 4 rather than their corresponding factors.

Factor 3 is another child factor of Factor 1. It is loaded with items designed to measure the facets of Morality, Cooperation, and Modesty. As all the nonzero loadings on Factor 3 are positive, it can be interpreted as a higher-order factor of morality, cooperation, and modesty. Factor 7 is the child factor of Factor 3. It is positively loaded by three items designed to measure the Morality facet, and can be interpreted accordingly. Factor 8 is another child factor of Factor 3. It is loaded positively by all the items designed to measure the Modesty facet and negatively, although relatively weakly, by all the items designed to measure the Cooperation facet, and item 8 ("Obstruct others' plans") that is designed to measure the Morality facet, but is closely related in concept to cooperation. Thus, we can treat Factor 8 as a higher-order factor of modesty and weak aggression (the opposite of cooperation). Finally, Factors 9 and 10 are child factors of Factor 8. Factor 10 may be interpreted as a cooperation factor, while Factor 9 seems to be a weak modesty factor.

Finally, we compare the learned hierarchical factor model with several alternative models based on the Bayesian Information Criterion (BIC; Schwarz, 1978), including

1. (CFA) A six-factor confirmatory factor analysis model with correlated factors. Each factor corresponds to a facet of Agreeableness, loaded by the four items designed to measure this facet.

2. (CBF) A confirmatory bi-factor model with one general factor and six group factors, where the group factors are allowed to be correlated. Each group factor corresponds to a facet of Agreeableness, loaded by the four items designed to measure this facet.

3. (EBF) An exploratory bi-factor model with one general factor and six group factors, where the group factors are allowed to be correlated. The bi-factor structure is learned using the method proposed in Qiao et al. (2025). Specifically, exploratory bi-factor models with $2, 3, \ldots, 12$ group factors are considered, among which the one with six group factors is selected based on the BIC.

Table 4 presents the BIC values of all the models, where the model labeled HF is the learned hierarchical factor model. From the results of BIC, the proposed hierarchical factor model fits the data best. Detailed results on the estimated loading matrix and the estimated correlation matrix of the competing models are shown in Appendix S8.

Table 4: The BICs of the hierarchical factor model and the competing models

	HF	CFA	CBF	EBF
BIC	102,987.54	103,841.48	103,200.42	103,026.10

6 Discussions

This paper proposes a divide-and-conquer method with theoretical guarantees for exploring the underlying hierarchical factor structure of the observed data. The method divides the problem into learning the factor structure from the general factor to finer-grained factors. It is computationally efficient, achieved through a greedy search algorithm and an augmented Lagrangian method. To our knowledge, this is the first statistically consistent method for exploratory hierarchical factor analysis that goes beyond the bifactor setting. Our simulation study shows that our method can accurately recover models with up to four factor layers, ten factors, and 54 items under practically reasonable sample sizes, suggesting that it may be suitable for various applications in psychology, education, and related fields. The proposed method is further applied to data from an Agreeableness personality scale, which yields a sensible model with four layers and ten factors that are all psychologically interpretable.

It is worth noting that the current work assumes that all the factors are orthogonal. Mathematically, it is possible to relax this assumption, though certain constraints are still needed. Specifically, two factors need to be orthogonal if one is a descendant of the other. Otherwise, the model is not identifiable due to rotational indeterminacy. For factors without

such a relationship, correlations may be allowed. For example, in the exploratory bi-factor analysis in Qiao et al. (2025), which concerns hierarchical factor models with two factor layers, factors within the second layer are allowed to be correlated. However, we should note that relaxing the orthogonality assumption can make the model less interpretable. Under the orthogonality assumption, the dependence between two variables is solely due to the shared factors. Such simple interpretations are important when the hierarchical factor model has a complex structure (e.g., with many factor layers), which is probably why all the existing hierarchical factor models, except for some special bi-factor models, adopt this orthogonality assumption. Therefore, it may not be worth extending the current theory and method to a more general setting with correlated factors, even though it is possible.

The current method also assumes that a general factor exists and includes it in the first factor layer. However, this may not always be the case. For example, in psychology, there is still a debate about whether a general factor of personality exists (see, e.g., Revelle and Wilt, 2013). In cases where we are unsure about the presence of a general factor, the current method can be easily modified to estimate a hierarchical factor model without a general factor, which can be achieved by modifying the first step of Algorithm 1.

The current method and asymptotic theory consider a relatively low-dimensional setting where the number of variables J is treated as a constant that does not grow with the sample size. However, in some large-scale settings, J can be on a scale of hundreds or even larger, so it may be better to treat it as a diverging term rather than a fixed constant. In that case, a larger penalty term may be required in the information criterion to account for the larger parameter space, and the asymptotic analysis needs to be modified accordingly.

Finally, the current work focuses on linear hierarchical factor models, which are suitable for continuous variables. In many applications of hierarchical factor models, we often encounter categorical data (e.g., binary, ordinal, and nominal) that may be better analyzed with non-linear factor models (see, e.g., Chen et al., 2020). We believe it is possible to extend

the current framework to the exploratory analysis of non-linear hierarchical factor models. In particular, building upon recent advances in the generalized latent factor model (e.g., Cui and Xu, 2025), our approach can be generalized to non-linear hierarchical factor models through likelihood-based estimation, subject to appropriate constraints on both factors and loadings.

Acknowledgement

The authors would like to thank the editor, the associate editor, and the reviewers for their constructive and valuable comments, which have substantially improved the manuscript.

References

Anderson, T. (2003). *An Introduction to Multivariate Statistical Analysis*. Wiley New York.

Anderson, T. and H. Rubin (1956). Statistical inference in factor analysis. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, pp. 111 – 150. University of California Press.

Bertsekas, D. P. (2014). *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press.

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research 36*(1), 111–150.

Brunner, M., G. Nagy, and O. Wilhelm (2012). A tutorial on hierarchically structured constructs. *Journal of Personality 80*(4), 796–846.

Carroll, J. B. (1993). *Human Cognitive Abilities: A survey of Factor-Analytic Studies*. Cambridge University Press.

Chen, F. F., S. G. West, and K. H. Sousa (2006). A comparison of bifactor and second-order models of quality of life. *Multivariate Behavioral Research 41*(2), 189–225.

Chen, Y., X. Li, and S. Zhang (2019). Joint maximum likelihood estimation for high-dimensional exploratory item factor analysis. *Psychometrika 84*(1), 124–146.

Chen, Y., X. Li, and S. Zhang (2020). Structured latent factor analysis for large-scale data: Identifiability, estimability, and their implications. *Journal of the American Statistical Association 115*(532), 1756–1770.

Cui, C. and G. Xu (2025). Identifiability and inference for generalized latent factor models. *arXiv preprint arXiv:2508.05866*.

DeYoung, C. G. (2006). Higher-order factors of the big five in a multi-informant sample. *Journal of Personality and Social Psychology 91*(6), 1138.

Fabrigar, L. R. and D. T. Wegener (2012). *Exploratory Factor Analysis*. Oxford University Press.

Fang, G., J. Guo, X. Xu, Z. Ying, and S. Zhang (2021). Identifiability of bifactor models. *Statistica Sinica 31*, 2309–2330.

Holzinger, K. J. and F. Swineford (1937). The bi-factor method. *Psychometrika 2*(1), 41–54.

Jennrich, R. I. and P. M. Bentler (2011). Exploratory bi-factor analysis. *Psychometrika 76*, 537–549.

Johnson, J. A. (2014). Measuring thirty facets of the five factor model with a 120-item public domain inventory: Development of the IPIP-NEO-120. *Journal of Research in Personality 51*, 78–89.

Kose, M. A., C. Otrok, and C. H. Whiteman (2008). Understanding the evolution of world business cycles. *Journal of International Economics 75*(1), 110–130.

Moench, E., S. Ng, and S. Potter (2013). Dynamic hierarchical factor models. *Review of Economics and Statistics 95*(5), 1811–1817.

Qiao, J., Y. Chen, and Z. Ying (2025). Exact exploratory bi-factor analysis: A constraint-based optimization approach. *Psychometrika 90*(3), 998–1013.

Reise, S. P., J. Morizot, and R. D. Hays (2007). The role of the bifactor model in resolving dimensionality issues in health outcomes measures. *Quality of Life Research 16*, 19–31.

Revelle, W. and J. Wilt (2013). The general factor of personality: A general critique. *Journal of Research in Personality 47*(5), 493–504.

Schmid, J. and J. M. Leiman (1957). The development of hierarchical factor solutions. *Psychometrika 22*(1), 53–61.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*, 461–464.

Sharma, P., B. Sivakumaran, and G. Mohan (2022). Using Schmid–Leiman solution with higher-order constructs in marketing research. *Marketing Intelligence & Planning 40*(4), 513–526.

Thomson, G. H. (1939). *The factorial analysis of human ability.* Houghton Mifflin.

Thurstone, L. L. (1947). *Multiple Factor Analysis.* University of Chicago Press.

Yung, Y.-F., D. Thissen, and L. D. McLeod (1999). On the relationship between the higher-order factor model and the hierarchical factor model. *Psychometrika 64*, 113–128.

Zheng, X., B. Aragam, P. K. Ravikumar, and E. P. Xing (2018). Dags with NO TEARS: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems 31*.

Author Information

[1]School of Mathematical Science, Fudan University, 20110180052@fudan.edu.cn

[2]Department of Statistics, London School of Economics and Political Science, y.chen186@lse.ac.uk

[3]Department of Statistics, Columbia University, zying@stat.columbia.edu