Factor Retention in Exploratory Multidimensional Item Response Theory

Changsheng Chen[12], Robbe D'hondt[23], Celine Vens[23] and Wim Van den Noortgate[12]

[1]*Faculty of Psychology and Educational Sciences, KU Leuven, Campus KULAK, Kortrijk, Belgium*

[2]*imec research group itec, KU Leuven, Kortrijk, Belgium*

[3]*Department of Public Health and Primary Care, KU Leuven, Campus KULAK, Kortrijk, Belgium*

Author Note

Corresponding author:

Changsheng Chen, Faculty of Psychology and Educational Science & imec research group itec, campus Kulak Kortrijk, KU Leuven, E. Sabbelaan 51, 8500 Kortrijk, Belgium**.** Email: changsheng.chen@kuleuven.be

Authors' information

Changsheng Chen (ORCID: https://orcid.org/0000-0001-6092-6655) is a Ph.D. student at the Faculty of Psychology and Educational Sciences, and the imec research group itec at the KU Leuven. His doctoral research focuses on learning analytics.

Robbe D'hondt (ORCID: https://orcid.org/0000-0001-7843-2178) is a Ph.D. student at the Faculty of Medicine, and the imec research group itec at the KU Leuven. His doctoral research focuses on the use of machine learning to model the prognosis of multiple sclerosis patients.

Celine Vens (ORCID: https://orcid.org/0000-0003-0983-256X) is a professor of machine learning at the Faculty of Medicine (Data-Driven Healthcare research group), and the imec research group itec at the KU Leuven. Her major interests include structured output learning and tree ensemble methods.

Wim Van den Noortgate (ORCID: https://orcid.org/0000-0003-4011-219X) is a professor of statistics at the Faculty of Psychology and Educational Sciences, and the imec research group itec at the KU Leuven. His major interests include learning analytics and meta-analysis.

Declaration

Funding

This work was partly supported by Research Fund Flanders (FWO) 1S38023N (Robbe D'hondt).

Conflicting interest

The authors declare that they have no conflicting interests.

Ethics approval

Not applicable

Consent to participate

Not applicable

Consent for publication

All authors approved the final manuscript and the submission to this journal.

Availability of data and materials

The data sets analyzed in this study were simulated in R provided by the VSC (Flemish Supercomputer Center). The simulation parameter settings and results from VSC can be obtained from https://osf.io/m23sg/.

Code availability

The relevant R and Python codes used in this study can be obtained from https://osf.io/m23sg/.

Authors' contributions

CC and WVDN conceptualized the study. CC prepared the initial draft of the manuscript and did the data simulation and analysis. RD developed the tuning and modeling machine learning pipeline under the supervision of CV, and performed additional analyses on the resulting models as shown in the Appendix. RD and CV revised the manuscript. WVDN supervised the study and revised the manuscript.

Acknowledgments

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation-Flanders (FWO) and the Flemish Government-department EWI.

Factor Retention in Exploratory Multidimensional Item Response Theory

Abstract

Multidimensional Item Response Theory (MIRT) is applied routinely in developing large-scale educational and psychological assessment tools, for instance, for exploring multidimensional structures of items using exploratory MIRT. A critical decision in exploratory MIRT analyses is the number of factors to retain. Unfortunately, the comparative properties of statistical factor retention methods and innovative Machine Learning (ML) methods for factor retention in exploratory MIRT analyses are still not clear. This study aims to fill this gap by comparing a selection of statistical and ML methods, including Kaiser Criterion (KC), Empirical Kaiser Criterion (EKC), Parallel Analysis (PA), scree plot (OC & AF), Very Simple Structure (VSS; C1 & C2), Minimum Average Partial (MAP), Exploratory Graph Analysis (EGA), Random Forest (RF), Histogram-based Gradient Boosted Decision Trees (HistGBDT), eXtreme Gradient Boosting (XGBoost), and Artificial Neural Network (ANN). The comparison was was performed using 720,000 dichotomous response datasets simulated by the MIRT, for various between-item and within-item structures and considering characteristics of large-scale assessments. The results show that MAP, RF, HistGBDT, XGBoost, and ANN tremendously outperform other methods. Among them, HistGBDT generally performs better than other methods. Furthermore, including statistical methods' results as training features improves the performance of ML methods. The methods' correct-factoring proportions decrease with an increase in missingness or a decrease in sample size. KC, PA, EKC, and scree plot (OC) are over-factoring, while EGA, scree plot (AF), and VSS (C1) are under-factoring. We recommend that practitioners use both MAP and HistGBDT to determine the number of factors when applying exploratory MIRT.

Keywords: exploratory multidimensional item response theory; MIRT; machine learning; factor retention; multidimensionality

Introduction

Exploring the latent dimensional structure (i.e., dimensionality) is an important step to develop assessment tools in educational and psychological measurement. The "dimensional structure" refers to the relationship between the designed test items and the conceptual psychological constructs that researchers intend to measure, which somehow reflects the validity information of designed tools (American Educational Research Association et al., 2014; Miller & Lovler, 2018). In practice, researchers usually conduct exploratory analyses to get insights into the dimensionality and subsequently further improve assessment tools. Technically, this kind of exploratory analyses can be decomposed into two steps, i.e., first determining the number of factors (or latent dimensions) to be retained (i.e., factor retention) and then defining the relationship between items and factors (i.e., exploring the factor structure). Thus, the estimated number of factors significantly affects the shape of dimensional structures and the final interpretations (Gorsuch, 2015). In the past decades, several statistical techniques have been developed to solve these problems. For example, Exploratory Factor Analysis (EFA) was designed for analyzing the dimensionality based on continuous items (Brown, 2015) and the exploratory Multidimensional Item Response Theory (MIRT) was introduced to analyze the case with dichotomous items (Bock et al., 1988; Cai, 2010; Chalmers, 2012; Mair, 2018; Wirth & Edwards, 2007). In addition to these general approaches, researchers have proposed specific factor retention methods, including the Kaiser Criterion (KC; Kaiser, 1960), Parallel Analysis (PA; Horn, 1965), and the use of a scree plot (Cattell, 1966).

However, most of these traditional statistical factor retention methods have only been evaluated and compared in the context of small-scale assessments. In the past years, large-scale online assessments with new data features have become more common. For example, online assessments, such as United States Medical Licensing Examination (USMLE, 2023)

and Graduate Record Examinations (Liu et al., 2018), usually are designed with hundreds and even thousands of items. A large number of items may challenge the performance of traditional methods, calling for a new comparison study. Moreover, compared to traditional testing, online assessments often administer items differently. In particular, every participant may get a tailored set of items where some items may be included in all sets, whereas some are not. It causes sparsity problems in the combined response matrix, which may further challenge the estimation stability for traditional methods that are usually used to analyze the response matrix with few missing values (Goretzko, 2022; Xia & Havan, 2024). In addition, previous studies mainly focus on the properties of factor retention methods for continuous items and less clarify their properties for dichotomous items. For analyzing the data of online educational assessments, for example, it is common to dichotomize students' responses as right or wrong, but it is not clear what effect this has on factor retention methods.

Although classical methods may somehow struggle with the features of newly prevailing assessments, recently developed techniques from Machine Learning (ML) offer possibilities to alleviate these shortcomings. Goretzko and Bühner (2020) found that using ML methods (i.e., Random Forest (RF) and eXtreme Gradient Boosting (XGBoost)) can predict the number of factors noticeably better than statistical methods based on continuous response data simulated by the EFA. However, in their study, the methods' performance was not investigated for dichotomous responses. Additionally, deep learning ML methods, such as Artificial Neural Network (ANN) that holds promising potential, were not yet examined. Moreover, Goretzko and Bühner (2020) mainly considered the features of relatively small-scale assessments (e.g., the number of items ranging from 4 to 42), which should be extended considering the current trends of online assessments.

In a nutshell, the performance of traditional statistical factor retention methods is doubtful in terms of analyzing large-scale dichotomous assessment data and the performance

of ML methods has not been studied comprehensively. Thus, in this study, we aim to fill this research gap by comparing selected statistical and ML methods based on dichotomous responses generated by the MIRT for certain multidimensional structures and give recommendations for practitioners regarding methods selection. More specifically, we want: 1) to explore and compare the performance of statistical methods and ML methods based on simulated dichotomous data (reflecting the features of large-scale assessment data) for several between-item and within-item structures, and 2) to analyze the effects of simulation features on the methods' performance.

The following part starts with a conceptual introduction to the exploratory MIRT with multidimensional structures and relevant factor retention methods. After that, details about data generation and analysis for methods' comparison are provided. Finally, results are described and discussed.

Exploratory Multidimensional Item Response Theory

As a common approach applied in the field of psychometrics and educational measurements, the MIRT is often used to develop test items and measure participants' latent skills or psychological tendencies. The principle behind the MIRT is that it models the probability of giving correct answers by considering the characteristics of both items and participants (Bonifay, 2020). More technically, the probability of giving correct answers is conditional on the item parameters (e.g., item slopes and item difficulties) and participants' ability levels. As an example, a multidimensional compensatory 2-parameter logistic model is given in Equation 1. $P\left(x_{ij}=1\middle|\boldsymbol{\theta}_i;\boldsymbol{\alpha}_j,d_j\right)$ is the probability of giving a correct answer to item *j* for person *i* conditional on the relevant parameters (i.e., $\boldsymbol{\theta}_i, \boldsymbol{\alpha}_j$, and d_j) where $\boldsymbol{\theta}_i = (\theta_{i1},\dots,\theta_{ik})$ are the ability parameters for person *i* on the dimensions 1 to *k*, $\boldsymbol{\alpha}_j = (\alpha_{j1},\dots,\alpha_{jk})$ are the slope parameters for item *j* on the dimensions 1 to *k*, d_j is the item intercept for item *j*, and *D* (usually set equal to 1.702) is the scaling parameter for making the

logistic model closer to the traditional the normal ogive model (Chalmers, 2012).

$$P(x_{ij} = 1|\boldsymbol{\theta}_i; \boldsymbol{\alpha}_j, d_j) = \frac{1}{1 + exp[-D(\boldsymbol{\alpha}_j^T \boldsymbol{\theta}_i + d_j)]} \quad \textbf{Equation 1}$$

Traditionally, in the unidimensional IRT, such as the Rasch model, we assume that all items are used to measure one intended factor, and item slopes are estimated to reflect the strength of the relationship between the items and that factor. However, in the MIRT, this unidimensional assumption is not made anymore. Akin to the EFA, an exploratory MIRT analysis implies doing a factor retention analysis first and estimating the parameters next. In particular, determining the number of factors influences how many parameters (such as item slopes and factor scores) should be estimated. For example, if 8 factors are determined, there will be 8 item slope estimates for each item. After that, all item slope estimates can be converted to factor loadings, and rotation methods can be performed to explore the final multidimensional structure (Cai, 2010; Chalmers, 2012; Wirth & Edwards, 2007).

When using the exploratory MIRT, one needs also to consider possible multidimensional structures. According to different assumptions of the relationship between items and factors, the multidimensional structures are generally classified into five categories, such as the between-item structure, the within-item structure, the bifactor structure, the two-tier structure, and the high-order structure (Bonifay, 2020).[1] This study is limited to the between-item and within-item structures because these are widely applied in practice and most of the available software supports these analyses (Chalmers, 2012). Technically, the between-item structure indicates that each item is solely associated with one factor, and factors are allowed to be correlated. In contrast, in the within-item structure, each item is allowed to be correlated with more than one factor, and factors can be correlated as well.

[1] Except for the consideration of the relationship between items and factors, the concern about the relationship between factors can also be considered, i.e., the compensatory and partial compensatory assumption. More details can be consulted in Bonifay (2020). In this study, the MIRT model for generating simulation data makes the compensatory assumption because of its wide application.

Factor Retention Methods

Table 1 presents most commonly used statistical or ML methods available for researchers regarding factor retention analysis. The statistical methods included some eigenvalue-related methods, such as KC, Empirical Kaiser Criterion (EKC; Braeken & Van Assen, 2017), PA, non-graphical scree plot with Optimal Coordinates (OC) or Acceleration Factor (AF; Raîche et al., 2013), Very Simple Structure (VSS; Revelle & Rocklin, 1979) with two variants (i.e., C1 & C2), Minimum Average Partial (MAP; Velicer, 1976), and Revised Parallel Analysis (RPA; Green et al., 2012, 2016a). Some of them have been criticized due to their inaccuracies (the tendency to over- or under-extract factors) in previous methods comparison studies (Cosemans et al., 2022; Golino & Epskamp, 2017). Additionally, there are some model-comparison-based statistical methods, including Comparison Data (CD; Ruscio & Roche, 2012), Chi-square test (Bollen, 1989), global model-fit indices (Mair, 2018), and Likelihood Ratio Test (LRT; Lawley, 1940). These approaches are usually computational-greedy and highly limited to the mechanism of the selected model itself, which is not suitable for analyzing the large-scale assessment data. In detail, by using the CD approach, the number of comparison pairs will explode (i.e., combinatorial explosion) when the number of factors is high.[2] The Chi-square test is sensitive to sample size and tends to reject the solutions with large sample size (Bentler & Bonett, 1980; Brown, 2015). The LRT only works for comparing nested models (Mair, 2018). As for the global model-fit indices, they all have respective assumptions or limitations, and therefore the methods sometimes produce inconsistent results, which makes it difficult to give an overall solution. Except for aforementioned statistical methods, researchers have recently developed Exploratory Graph Analysis (EGA) based on the graphical lasso with the regularization parameter specified by extended Bayesian

[2] For example, when 8 factors are assumed in the CD approach, the total number of comparison pairs will be 28 based on its comparison algorithm. More details can be found in Ruscio and Roche (2012).

information criterion, and they suggested that EGA has satisfying performance in terms of factor retention based on the data generated by the EFA (Cosemans et al., 2022; Golino & Epskamp, 2017).

Table 1

Factor Retention Methods for Dichotomous Responses

Method	Previous Research Simulation Model	
	EFA	**MIRT**
Statistical Method		
Kaiser Criterion (KC)	×	
Empirical Kaiser Criterion (EKC)	×	
Parallel Analysis (PA)	×	×
Scree Plot (OC & AF)	×	
Very Simple Structure (VSS; C1 & C2)	×	
Minimum Average Partial (MAP)	×	
Revised Parallel Analysis (RPA)	×	×
Comparison Data (CD)	×	
χ^2 test (Chi-square test)	×	
Global model-fit indices	×	
Likelihood Ratio Test (LRT)	×	
Exploratory Graph Analysis (EGA)	×	
Machine Learning Methods		
Random Forest (RF)		
Histogram-based Gradient Boosting (HistGBDT)		
eXtreme Gradient Boosting (XGBoost)		
Artificial Neural Network (ANN)		

Note. The right-side two columns indicates whether factor retention methods have been examined or compared regarding their properties based on the data (dichotomous items) generated by the EFA or MIRT in the previous literature. × means "Yes" and the blank means "No" (Cosemans et al., 2022; Golino & Epskamp, 2017; Green et al., 2016; Guo & Choi, 2023; Ruscio & Roche, 2012).

In addition to the statistical methods, it is possible to apply ML methods in this field. In general, the corresponding ML models are first trained to learn the patterns between a known target of interest and relevant known features in the training data (usually operated by cross validation to separate them into training information for training ML models and validation information for tuning hyperparameters). After that, they are evaluated based on

test data (by comparing their predictions and the targets of the test data). RF, XGBoost, Histogram-based Gradient Boosting Decision Trees (HistGBDT), and ANN are the commonly used ML methods for making predictions based on tabular data (Chen & Guestrin, 2016; Chollet, 2021; Ke et al., 2017). Based on the data type of targets, most of them can be used for three different tasks, including regression (for continuous targets), classification (for categorical targets), and ordinal classification (for ordinal categorical targets). These three variants can be applied on integer targets. For the factor retention analysis, two theoretical advantages of ML methods can be highlighted. First, from the perspective of prediction accuracy, the use of ML methods can avoid the over-factoring tendency, because their corresponding models are trained by simulation data and shaped by simulation settings. For example, if ML models are trained by simulation data generated within 5-factor settings, their predictions will not go over 5 factors. Second, they can combine the results of other approaches to improve their prediction performance. For example, the results of statistical methods can be used as extra features to train ML models and make predictions.

In this study, we want to compare selected methods, including KC, EKC, PA, non-graphical scree plot (OC & AF), VSS (C1 & C2), MAP, EGA, RF, XGBoost, HistGBDT, and ANN. Moreover, we will also investigate whether the performance of selected ML methods can be improved by using results of statistical methods as extra inputs. Therefore, two versions of ML models (corresponding to the four ML methods) are included, i.e., the ML models trained with or without using results of statistical methods as features. Furthermore, the aforementioned three variants (i.e., the regression, classifier and ordinal variants) for the ML methods were implemented because the prediction targets (i.e., the number of factors) are integers. The general reason for the selection is twofold. The first is that the properties of the selected methods are not yet clear for the data with dichotomous responses generated by the MIRT. Most of these methods were only studied with data generated by the EFA. We are

specifically interested in the performance of ML methods because previous studies suggested that RF and XGBoost reach surprising accuracy regarding predicting the number of factors (Goretzko & Bühner, 2020). Additionally, little research studies the performance of ANN and HistGBDT in this context, which arouses our interest in including them. The second reason for the selected methods is the consideration of computational power and practical convenience. Methods with an internal comparison design (comparing different solutions), such as RPA and CD, are usually computationally intensive, especially for the case with a large number of items, and therefore their use may not be feasible in large-scale assessments.

Method

A flowchart summarizing the steps taken to generate data and to implement methods is provided in the Supplemental Materials. Data were generated and analyzed in R 4.3.2 (R Core Team, 2023) and Python 3.9.18,[3] using the Flemish Supercomputer Center (Vlaams Supercomputer Centrum; VSC).

Data

The methods' comparison in this study was based on datasets generated by the MIRT, with dichotomous responses that mimic features of large-scale assessments. Table 2 presents the detailed settings of simulation features, which were both implemented for the between-item and within-item structures. These simulation features contained the number of items, the number of factors, the sample size, the percentage of missing entries in the response matrix, and the correlation between factors. To train ML models and compare all candidate methods, we simulated two sets of data, i.e., training data for training ML models and test data for comparing all candidate methods. The relevant values for generating training data were randomly selected from the designed range for each simulation feature, except for the number

[3] Specialized packages in R and Python were used for some specific methods. These R packages included 'sirt', 'psych', 'nFactors', and 'EGAnet'. The Python packages included 'numpy', 'scipy', 'scikit-learn', 'pandas', 'matplotlib', 'seaborn', 'xgboost', and 'tensorflow'.

of factors. The number of feature values selected for the training data was decided in order to balance the amount of information available to train ML models to cover possible scenarios and the computational power consumption. In total, for the training data, $20\ (the\ number\ of\ items) \times 8\ (the\ number\ of\ factors) \times 10\ (sample\ size) \times 10\ (missing\ percentage) \times 5\ (factor\ correlation) = 80{,}000$ scenarios were created. We simulated 4 datasets for each scenario, so 320,000 training datasets were generated and then summarized (by feature engineering) into one training feature matrix for training ML models. The test data were generated with fixed values, so the effects of simulation features on selected methods' performance could be further analyzed. Specifically, there were $6\ (the\ number\ of\ items) \times 8\ (the\ number\ of\ factors) \times 6\ (sample\ size) \times 5\ (missing\ percentage) \times 5\ (factor\ correlation) = 7{,}200$ scenarios. We simulated 100 datasets for each scenario, namely 720,000 test datasets generated for evaluating all methods.

Table 2

Simulation Features for Data Generation

Simulation features	Values for generating the training data	Values for generating the test data
The number of items	300-800 (randomly selected 20 values)	300, 400, 500, 600, 700, 800
The number of factors	1, 2, 3, 4, 5, 6, 7, 8	1, 2, 3, 4, 5, 6, 7, 8
Sample size	300-800 (randomly selected 10 values)	300, 400, 500, 600, 700, 800
Missing percentage	0-90% (randomly selected 10 values)	0, 25%, 50%, 75%, 90%
Factor correlation	0.1-0.5 (randomly selected 5 values)	0.1, 0.2, 0.3, 0.4, 0.5
Scenarios for training ML models:		
80,000 (each scenario with 4 repeated-simulated data, 320,000 datasets in total)		
Scenarios for evaluating the performance of all factor retention methods:		
7,200 (each scenario with 100 repeated-simulated data, 720,000 datasets in total)		

A multidimensional compensatory 2-parameter logistic model (See Equation 1) was used to generate data, using the "simdata" function of the R package "mirt" (Chalmers, 2012). To differentiate the between-item and within-item structure, different patterns of item slopes were set. For the between-item structure, the number of items was separated in a roughly even way based across the number of factors. For example, in the scenario with 3 factors and 600

items, 600 items were split into 196, 205, and 199 items corresponding to each factor. Each item was solely associated with one factor (by setting one of the item slopes α_{jk} to 1 and the rest to 0). For the within-item structure, in which each item is related to all factors, all α_{jk}values were randomly selected from a uniform distribution between -1 to 1. For example, in the scenario with 4 factors and 200 items, each item had 4 random item slopes between -1 and 1. The item intercepts d_j for both structures were randomly selected from a standard normal distribution with mean 0 and standard deviation 1. As for the ability parameter θ_{ik}, a standard normal distribution was used for both structures, when the number of factors was equal to 1. When the number of factors was equal or higher than 2, the ability parameters θ_{ik} followed a multivariate normal distribution with mean vector 0, standard deviation vector 1, and correlations decided by the simulated scenario (see Table 2). Based on the given values, Equation 1 was used to calculate the probability of giving a correct answer to item *j* for person *i*, and this probability was used to draw a dichotomous response (0 or 1) from a Bernoulli distribution.

Design and Analysis

Methods Implementation

As mentioned above, the MIRT was used to simulate the training data (for training ML models and tuning hyperparameters) and test data (for the evaluation and comparison of the statistical methods and ML methods). The statistical methods were implemented on both training and test data, based on estimated tetrachoric correlations that are commonly used for analyzing dichotomous responses. The estimation method for tetrachoric correlations proposed by Bonett and Price (2005) was adopted because of its stability and efficiency in the case of severe missingness. The prediction results of statistical methods based on test data were used for the final methods evaluation and comparison, while the results based on training data were used only as extra inputs for training ML models.

For the four ML methods, feature engineering was performed to convert the raw generated responses data into features (as information that can be processed by the ML). The extracted features were seen as inputs and the true number of factors (set in the simulation procedure) was regarded as targets. Three sources of features were extracted: 1) from the original response matrix: the sample size, the number of items, and the missing percentages; 2) from the tetrachoric correlation matrix: the determinant, the number of entries smaller or equal to 0.1, the number of eigenvalues higher than 0.7, the relative proportion of eigenvalues (e.g., the first, the first two, and the first three eigenvalues), the standard deviation of all eigenvalues, the number of eigenvalues explaining over 50% or 75% of the variance, the various kinds of matrix norms (e.g., the L_1-norm, Frobenius-norm, maximum-norm, and spectral-norm), the average of off-diagonal entries and the communality estimates, the sampling adequacy (Kaiser, 1970), the Gini-coefficient (Gini, 1921), the Kolm inequality (Kolm, 1999), the top 50 eigenvalue estimates (Goretzko & Bühner, 2020); and 3) from the results of other methods: KC, PA, EKC, scree plot (OC), scree plot (AF), VSS (C1), VSS(C2), and MAP. In addition to the feature engineering, choosing hyperparameters can noticeably affect the performance of the selected ML methods. For RF, XGBoost, and HistGBDT, they were trained with three variants and the relevant optimal hyperparameters were identified through the implementation of a 10-fold cross validation process on the training set. For ANN, hyperparameters were determined by a 5-fold cross-validation operation for computational reasons. The details of the hyperparameter search space and tuning results can be consulted in the Supplemental Materials.

Finally, the prediction results of the resulting ML models and the statistical methods based on the test data were further analyzed by a post-hoc analysis with evaluation metrics to compare their performance.

Evaluation Metrics

To evaluate the performance of candidate methods, several metrics were adopted to make the comparison fair and clear. The key metric was the deviation score, namely, the predicted number of factors minus the true number of factors. The deviation score was calculated for each test dataset under each method. To give a general impression of the methods' performance, some summarizing metrics based on deviation scores were used. The first was the accuracy, including the correct-factoring proportions (the number of correct-factoring datasets divided by the total number of datasets), the over-factoring proportions (the number of over-factoring datasets divided by the total number of datasets), and the under-factoring proportions (the number of under-factoring datasets divided by the total number of datasets). The second was the bias, i.e., the average deviation score. The third was the precision, i.e., the average absolute deviation score. The final was the agreement rates between each pair of methods (the number of datasets with the same deviation score divided by the total number of datasets for a pair of selected methods). Apart from the metrics related to deviation scores, the non-convergence proportions were used to show the methods' stability. Because some challenging scenarios were simulated, it could be expected that several methods might not always produce results.

Analysis Strategy

Two analysis strategies were used to provide the final results and conclusions. A descriptive analysis was conducted for the between-item structure and within-item structures. For the sake of parsimony, only the best performing variant of RF, HistGBDT and XGBoost was selected for further analyses. First, the results of accuracy, bias, and precision based on the deviation scores are discussed. Then, the agreement rates between methods are discussed. After that, the relationship between certain simulation features and correct-factoring proportions of selected representative methods is discussed. Apart from the descriptive

information, an inferential analysis was performed to examine whether the identified patterns were statistically supported by the relevant results with the Generalized Linear Mixed Model (GLMM; Molenberghs & Verbeke, 2005). Specifically, in the GLMM, the deviation scores were dichotomized into 1 (correct-factoring) and 0 (incorrect-factoring). The selected methods were used as a categorical predictor, and pairwise comparisons of their performance were made. Additionally, the simulation features were included as predictors to study their (differential) effects on the performance of methods. A random effect for the test dataset was included in the GLMM to account for the nesting of the factoring results for each method within datasets.

Results

Descriptive Analysis Results

For all training and test datasets for both the between-item and within-item structure, results were collected successfully. For RF, HistGBDT and XGBoost, the three variants produced relatively similar correct-factoring proportions across the two multidimensional structures. Yet, the classifier variant produced slightly better results for most cases, and therefore was selected for further study. Table 3 provides the descriptive analyses results for the test datasets for all selected statistical methods, the classifier variant of the three ML methods and ANN regarding, separately for the two structures (see the Supplemental Materials for the complete descriptive results including all variants of ML methods). Looking first at the results for the between-item structure, we note that the EGA had a severe estimation problem: for 94.54% of the datasets, the output indicated that the community memberships for each item were missing, meaning that there were no identifiable factors. In contrast, the rest of the methods successfully produced estimates for all collected datasets. From the perspective of correct-factoring proportions, the ML methods with statistical methods' results as extra features performed generally better than the corresponding versions

without extra features. The MAP and four ML methods (with or without extra training features) were distinctly better than other methods, with correct-factoring proportions higher than .7. The highest one was the ANN (extra) that reached .7827, slightly higher than the HistGBDT (Classifier-extra) with a proportion .7823. The MAP and four ML methods with extra training features gave similar results, with agreement rates ranging from .72 to .90 (see the Supplemental Materials). In addition, the VSS (C1), VSS (C2), and scree plot (AF) exhibited moderate performance, and their correct-factoring proportions ranged from .31 to .48. The correct-factoring proportions for the remaining methods (including the KC, PA, EKC, scree plot (OC), and EGA) were all under .12, and for the first four methods even under .02. The pattern with regard to precision was analogous to the one found for correct-factoring proportions, which also validated the favorable performance of the MAP and four ML methods.

Regarding the results of under-factoring or over-factoring proportions, the KC, PA, EKC, and scree plot (OC) strongly overestimated the number of factors, and their over-factoring proportions were higher than .98, while the corresponding proportions of the MAP and four ML methods (with extra training features) were lower than .19. This can also be observed in the distribution of deviation scores (see the Supplemental Materials), especially for the KC where the estimated number of factors was most often higher than 100. Compared to these over-factoring methods, the EGA, scree plot (AF) and VSS (C1) tended to be strongly under-factoring, worse than the VSS (C2) and MAP, while around 11% of the results from the four ML methods (with extra training features) were under-factoring. This was partially confirmed by the results of bias.

In terms of the results for the within-item structure, the EGA had a similar convergence problem, failing to give community estimates for 86.71% of the datasets. For the results of correct-factoring proportions, akin to those for the between-item structure, the ML

methods with extra features outperformed the versions without them. The MAP and four ML methods were clearly better than other methods, and their proportions were all higher than .80. Among them, the HistGBDT (Classifier-extra) reached the highest proportion (.8710), slightly better than the XGBoost (Classifier-extra) with a proportion of .8706. From the perspective of the agreement rates (see the Supplemental Materials), these five methods also had higher rates, ranging from .82 to .95. The VSS (C1), VSS (C2), and scree plot (AF) demonstrated moderate performance with the correct-factoring proportions ranging from .23 to .37. In contrast, the correct-factoring proportions of the remaining methods were around or lower than .10, indicating their poor performance. The analysis of precision showed similar patterns. Additionally, the KC, PA, EKC, and scree plot (OC) demonstrated a pronounced propensity to overestimate the number of factors, with over-factoring proportions exceeding .90. The scree plot (AF), VSS (C1), VSS (C2), and EGA tended to extract an insufficient number of factors with under-factoring proportions ranging from .42 to .70. This finding aligns with the results of bias. In contrast, the over-factoring and under-factoring proportions of the MAP and four ML methods (with extra training features) were lower than .08 and .12 respectively, which indicates their better performance.

Table 3

Descriptive Results

	Non-convergence Proportion		Correct-factoring Proportion		Under-factoring Proportion		Over-factoring Proportion		Bias		Precision	
	Between	Within	Between	Within	Between	Within	Between	Within	Between	Within	Between	Within
KC	0	0	0	0	0	0	1	1	162.1081	159.4733	162.1081	159.4733
PA	0	0	.0049	.0556	0	0	.9951	.9444	89.2115	84.2395	89.2115	84.2395
EKC	0	0	.0180	.0996	0	0	.9820	.9004	66.9803	65.2540	66.9803	65.2540
Scree Plot (OC)	0	0	.0101	.0597	.0060	.0032	.9839	.9370	28.7146	26.8213	28.7484	26.8408
Scree Plot (AF)	0	0	.3083	.3663	.6666	.5983	.0250	.0355	-2.8665	-2.6440	2.9166	2.7155
VSS (C1)	0	0	.3906	.2309	.4603	.7020	.1491	.0671	-1.6242	-2.6735	2.1610	2.8884
VSS (C2)	0	0	.4792	.3232	.2692	.4168	.2516	.2600	-0.5528	-0.4917	1.2881	1.7149
MAP	0	0	.7131	.8109	.1827	.1123	.1043	.0768	-0.5523	-0.2699	0.7609	0.4235
EGA	.9454	.8671	.1185	.1167	.8434	.4960	.0381	.3873	-3.2580	0.1307	3.4150	3.6408
RF (Classifier)	0	0	.7594	.8495	.0549	.1140	.1857	.0364	0.3193	-0.2581	0.5454	0.4221
HistGBDT (Classifier)	0	0	.7701	.8665	.0357	.1031	.1942	.0304	0.3473	-0.2533	0.5637	0.3935
XGBoost (Classifier)	0	0	.7557	.8650	.0244	.1034	.2199	.0315	0.4456	-0.2457	0.5936	0.3865
ANN	0	0	.7518	.8446	.0646	.0964	.1836	.0590	0.2415	-0.0839	0.4770	0.2866
RF (Classifier-extra)	0	0	.7777	.8533	.0844	.1184	.1379	.0283	0.0648	-0.3127	0.4677	0.3923
HistGBDT (Classifier-extra)	0	0	.7823	.8710	.0650	.1137	.1527	.0153	0.0983	-0.3390	0.5120	0.3819
XGBoost (Classifier-extra)	0	0	.7673	.8706	.0498	.1153	.1829	.0141	0.2155	-0.3410	0.5111	0.3841
ANN (extra)	0	0	.7827	.8471	.0977	.1166	.1312	.0364	0.0682	-0.1956	0.4004	0.3022

Note. ***Non-convergence proportion***: the number of non-convergence results divided by the total number of results; ***correct-factoring proportion***: the number of correct-factoring results divided by the total number of results; ***under-factoring proportion***: the number of under-factoring results divided by the total number of results; ***over-factoring proportion***: the number of over-factoring results divided the total number of results; ***bias***: the average deviation score (the number of predicted factors minus the true number of factors); ***precision***: the average absolute deviation score. The word "***extra***" within the parentheses refers to the model trained with the results of statistical methods as extra features.

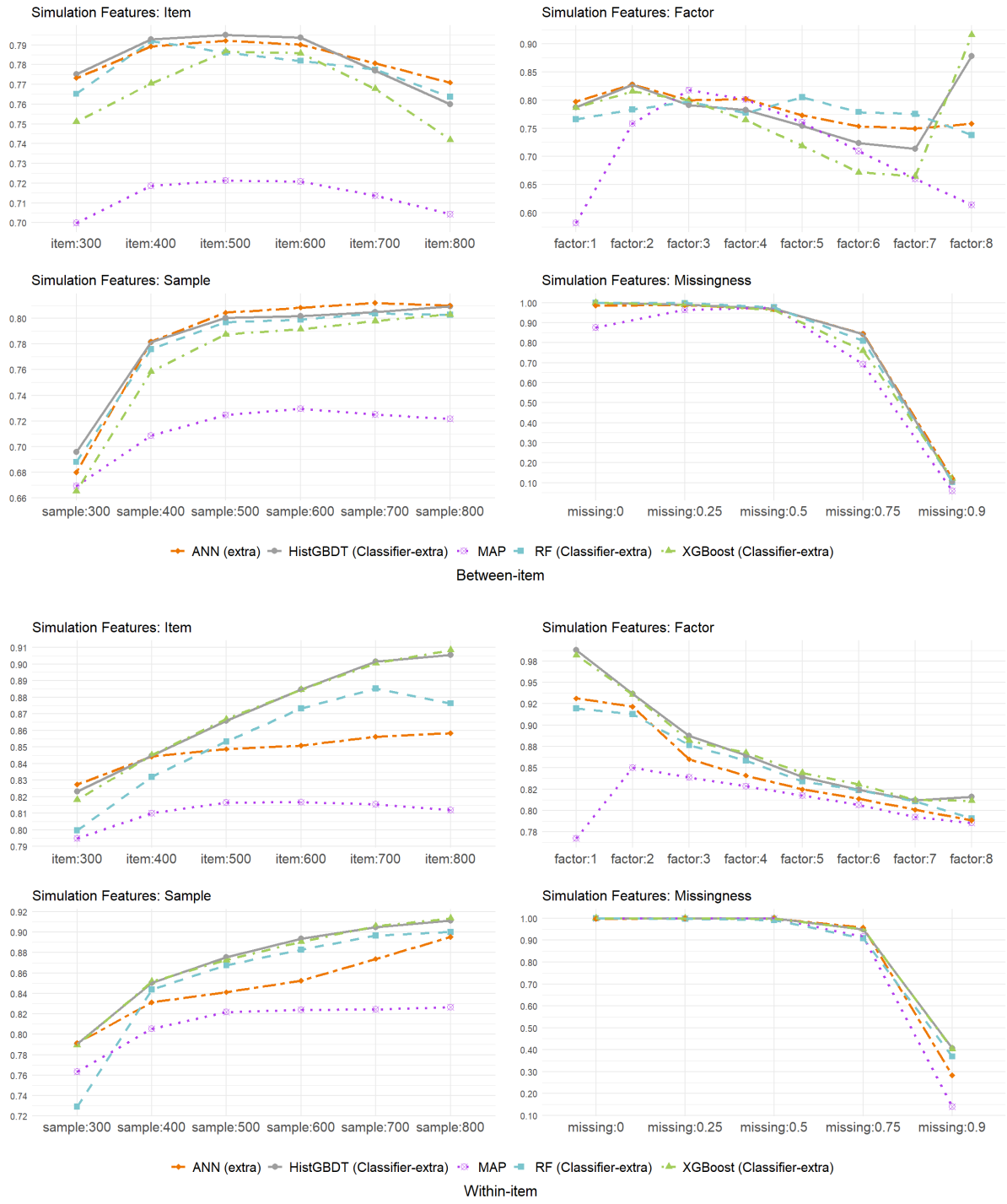
From the above results, it can be found that the MAP, RF (Classifier-extra), HistGBDT (Classifier-extra), XGBoost (Classifier-extra), and ANN (extra) were generally better than other methods across two multidimensional structures. We were further interested in how their correct-factoring proportions were affected by the simulation features. Figure 1 summarizes the results (detailed values provided by the Supplemental Materials). For the between-item structure, it is found that when the sample size increased, the correct-factoring proportions of the four ML methods rose sharply (especially for a sample size increasing from 300 to 500). The MAP followed a similar trend, but the proportions increased by a smaller size and went down slightly in the end. When the missingness became higher, the proportions of the four ML methods decreased tremendously from 1 to around .10. The proportion of the MAP increased mildly in the beginning and dropped noticeably if more than 50% of responses were missing. When the number of items increased from 300 to 800, the proportions of the five methods first went up and next down again. With regard to the number of factors, no discernible pattern emerged. The changes in proportions for the four ML methods exhibited a high degree of similarity, while the MAP displayed a somewhat disparate pattern in comparison. For example, the MAP exhibited a markedly lower initial proportion when the number of items was equivalent to 300 or the number of factors was equal to 1.

For the within-item structure, the patterns of missingness and sample size displayed similarities to those observed under the between-item structure. Specifically, when the sample size went up, the correct-factoring proportions of the five methods rose sharply (especially for the four ML methods). The MAP followed a similar trend, but its proportion remained similar when the sample size increased from 500 to 800. The increase in missingness from 50% to 90% was associated with a decrease in the correct-factoring proportions for all methods. It is worth noting that all methods reached a correct-factoring proportion around 1 when the missing percentages were lower than 50%. For other features, when the number of items

increased from 300 to 700, the proportions of the four ML methods rose by around .08. In contrast, the proportions of the MAP were almost similar. Regarding the number of factors, when this number increased from 2 to 8, a decrease in the proportions can be obviously detected for all methods.

Figure 1

Plots of Correct-Factoring Proportions for Simulation Features



Inferential Analysis Results

Table 4 presents the pairwise comparison results based on the GLMM. A significance level of .05 was used and the p-values were adjusted for multiple testing using Tukey's method (Lenth et al., 2023). For the between-item structure, most pairwise comparisons of selection methods showed a statistically significant difference. Specifically, the HistGBDT (Classifier-extra) and ANN (extra) statistically significantly outperformed other methods, leading to an increase in the log-odds ranging from 0.0958 to 1.3803. In other words, the probability of extracting the correct number of factors increased when one of them was applied compared to other methods. Additionally, there was no statistically significant difference between the HistGBDT (Classifier-extra) and ANN (extra). In contrast, the MAP led to a noticeable decrease in the log-odds ranging from -1.0576 to -1.3803, indicating its worse performance compared to the four ML methods. For the within-item structure, the general conclusions were similar to the between-item structure, except that the difference between the HistGBDT (Classifier-extra) and XGBoost (Classifier-extra) was not statistically significant. It indicated that they can both be considered better methods than the other methods. In addition, the MAP underperformed the four ML methods as well. Furthermore, we investigated the effect size of simulation features on the performance of those methods (see the Supplemental Materials) and found that the simulation features do not tremendously affect their performance.

Table 4

Pairwise Comparison Between Selected Methods Based on the GLMM

	Estimated difference in fixed effects	SE	Z-ratio	P-value
Between-item structure				
MAP - RF (Classifier-extra)	-1.2746	0.0076	-168.3122	< 0.001
MAP - HistGBDT (Classifier-extra)	-1.3704	0.0076	-180.0793	< 0.001
MAP - XGBoost (Classifier-extra)	-1.0576	0.0075	-141.2438	< 0.001
MAP - ANN (extra)	-1.3803	0.0076	-181.2820	< 0.001

RF (Classifier-extra) - HistGBDT (Classifier-extra)	-0.0958	0.0076	-12.5464	< 0.001
RF (Classifier-extra) - XGBoost (Classifier-extra)	0.2170	0.0076	28.5168	< 0.001
RF (Classifier-extra) - ANN (extra)	-0.1056	0.0076	-13.8336	< 0.001
HistGBDT (Classifier-extra) - XGBoost (Classifier-extra)	0.3129	0.0076	41.0282	< 0.001
HistGBDT (Classifier-extra) - ANN (extra)	-0.0098	0.0076	-1.2878	0.6988
XGBoost (Classifier-extra) - ANN (extra)	-0.3227	0.0076	-42.3106	< 0.001
Within-item structure				
MAP - RF (Classifier-extra)	-1.0584	0.0085	-124.6357	< 0.001
MAP - HistGBDT (Classifier-extra)	-1.5774	0.0089	-177.8243	< 0.001
MAP - XGBoost (Classifier-extra)	-1.5650	0.0089	-176.6118	< 0.001
MAP - ANN (extra)	-0.9009	0.0084	-107.4953	< 0.001
RF (Classifier-extra) - HistGBDT (Classifier-extra)	-0.5189	0.0091	-57.2394	< 0.001
RF (Classifier-extra) - XGBoost (Classifier-extra)	-0.5066	0.0091	-55.9120	< 0.001
RF (Classifier-extra) - ANN (extra)	0.1575	0.0088	17.9680	< 0.001
HistGBDT (Classifier-extra) - XGBoost (Classifier-extra)	0.0124	0.0092	1.3441	0.6635
HistGBDT (Classifier-extra) - ANN (extra)	0.6765	0.0090	74.9276	< 0.001
XGBoost (Classifier-extra) - ANN (extra)	0.6641	0.0090	73.6114	< 0.001

Discussion

In this study, several factor retention methods were compared for two multidimensional structures (i.e., the between-item and within-item structure) based on the dichotomous response datasets generated by the MIRT with simulation features reflecting the characteristics of large-scale assessments. The selected methods were compared by analyzing their performance in extracting the number of factors. The results of descriptive and inferential analysis showed that MAP and the four ML methods (including RF, HistGBDT, XGBoost, and ANN) were considerably better than other candidate methods across the two multidimensional structures. The performance of the five methods was relatively close to each other, although MAP performed slightly worse than the four ML methods. This general finding is partially consistent with the previous studies, although the simulation models and settings are different. For example, Goretzko and Bühner (2020) found that a gradient boosting model outperformed other methods regarding factor retention in the context of EFA. In a previous study of Cosemans et al. (2022) for binary items simulated by the EFA, the

average bias[4] of MAP was -0.955 that was much lower than KC and scree plot (AF) but higher than EGA, which is close to our estimates (-0.5528 for the between-item structure and -0.2699 for the within-item structure). Furthermore, the analysis results indicated that the incorporation of the results of statistical methods as training features improved the performance of ML methods, a finding that has not been previously documented in the literature.

In addition, our results revealed that EGA exhibited a comparatively inferior performance, a result that contrasts with the findings of previous studies. As a relatively newly developed method, several studies supported that EGA outperformed other statistical methods. For example, Cosemans et al. (2022) found that the expected bias of EGA was 0.065 across all conditions. Golino and Epskamp (2017) found that the expected correct-factoring proportion of EGA across all conditions was .96 (for the two-factor structure) and .89 (for the four-factor structure). In contrast, our bias estimate was -3.2580 (for the between-item structure) and 0.1307 (for the within-item structure), and the correct-factoring proportion was .1185 and .1167 for the two structures respectively. Moreover, our results showed EGA had a serious non-convergence issue, whereby EGA estimated zero factors for over 85% of datasets. This is a finding that has not been identified in the literature. Such a large difference could be due to the dissimilarities of simulation settings and data generation models. Especially, the effects of missingness were not considered in the previous studies.

In terms of other statistical methods, our results indicated that they were far more underperforming compared to previous studies. Guo and Choi (2023) found that the correct-factoring proportion of using traditional PA in the MIRT ranged from .83 to 1 which is completely different from our findings where the respective proportions of PA were .0049 for

[4] The expected bias of the mentioned methods from Cosemans et al. (2022) was calculated based on results for dichotomous data with a 50-50 split by tetrachoric correlations.

the between-item structure and .0556 for the within-item structure. This may be due to the different simulation settings. For example, in Guo and Choi (2023), missingness was not considered, the number of factors ranged from 1 to 3, and the number of items only included 30 and 60. In contrast, in our study, the missingness, the number of factors, and the number of items ranged extensively from 0 to 90%, 1 to 8, and 300 to 800 respectively. Nevertheless, this difference is still surprising. Cosemans et al. (2022) found that the average bias of KC was 2.435 which is tremendously different from our estimation (160.7907), although the bias when using the scree plot (AF) was -2.05, which is close to our finding (-2.7553). In the study of Goretzko and Bühner (2020) for the continuous items generated by the EFA, the statistical methods (e.g., PA, EKC, and KC) had adequate performance with the accuracy ranged from .7464 and .8842, which is different from our results. In this study, PA, EKC, and KC barely estimated the correct number of factors, and their correct-factoring proportions were lower than .10 for two structures, and KC did not even have one successful case.

Regarding the effects of simulation features, our results indicated that the simulation features do not tremendously affect the performance of MAP and four ML methods, but the general impacts of certain features on them were noticeable. For example, the correct-factoring proportions of the five methods noticeably dropped when the missing percentage increased by over 50%. Most of the previous methods comparison studies did not consider the effects of missingness on methods' performance, and our findings fill this gap. Another finding was that the increase in the sample size improved the methods' performance for the two structures. This finding is similar to the relevant conclusions in the study of Goretzko and Bühner (2020) where the accuracy of ML methods grew when the sample size increased from 250 to 1000. For the statistical methods, some studies found that providing a larger sample size may downgrade their accuracy. For example, Guo and Choi (2023) found that the correct-factoring proportion of using various PA methods with principal component analysis

under the tetrachoric correlation in the MIRT decreased by a range from .05 to .30 when the sample size increased from 500 to 1000.

Overall, in comparison to the previous studies based on the EFA, the performance of statistical factor retention methods exhibits completely dissimilar patterns for dichotomous data generated by the MIRT under the consideration of large-scale assessments. Additionally, we further confirm that ML methods can work effectively in the factor retention analysis and MAP (the only well-performing statistical method) should also be considered in the relevant practice. Applying ML methods brings about extra benefits, apart from their high prediction accuracy. First, users can select which features to be included in the model training for ML methods, which provides more flexibility to take extra special features into account. In contrast, most statistical methods are designed in a relatively fixed way, and it is less possible to adapt their mechanism for different datasets. Second, sometimes the application of ML methods is more efficient than the statistical methods. From our experience, some statistical methods (e.g., PA and EGA) need more computational power and time to finish the estimation, especially for high dimensional data. In contrast, the ML models trained without using results of statistical methods only need simple features from the original responses matrix and the estimated tetrachoric correlation matrix to make predictions, which can be time-efficient. However, using the ML models trained with statistical methods' results as extra features will take more time, even though they provide slightly better prediction accuracy. Nevertheless, we also confirm that several disadvantages of ML methods should be considered in practice. First, ML methods heavily rely on the training datasets and cannot extrapolate beyond the distribution of the training data. Specifically, in the real world, we never know the true models behind the data, and the number of factors is unobservable, which can only be estimated. We can only assume that the real data are depicted by certain models and use the assumed model to generate data to train ML models. For example, if the real data

is generated by an 8-factor model and the training datasets only cover 1 to 7 factors, the prediction of ML models will never produce an 8-factor solution. In a nutshell, there are still some cautious gaps between theoretical-assumed and real situations for using ML. The second is the black-box problem. We also notice that some ML methods, such as the ANN, are performed in a black-box way, which means that no clear explanations can be introduced to how the corresponding model makes predictions. This may arouse practitioners' confusion in the practice.

Based on the discussion above, we propose several recommendations for practitioners. First, if practitioners are not familiar with ML methods, MAP will be a recommended option for analyzing the large-scale dichotomous response assessment data, especially for the case with hundreds of items. Second, if practitioners are familiar with ML methods, we will recommend performing MAP and HistGBDT simultaneously and comparing their results. This is because ML methods are heavily restricted by the characteristics of the input training data, as mentioned before, and it would be better to consider estimates from the statistical methods which do not have this kind of problem at the same time. When the estimated results are consistent, practitioners can just simply trust the results. Otherwise, practitioners need to make further determinations based on their domain knowledge and the information provided by the model-fit indices within the MIRT. Third, using the results of statistical methods as training features is recommended because it can enhance the performance of ML methods, when there is no concern about the computation time and resources.

Some limitations of this study need to be highlighted. Analogous to all simulation studies, findings in this research are conditional on the simulation design. The generalization of any findings from this research should be cautious when the situations deviate from what the simulation settings tend to describe. Additionally, even though we consider various factor retention methods and simulation features, some other valuable methods and features can be

included further. For the statistical methods, several previous studies found that the RPA and CD demonstrated satisfying performance but were excluded in this study due to their computationally intensive properties (Guo & Choi, 2023; Ruscio & Roche, 2012). For ML methods, such as ANN, there are other kinds of architecture that can be considered. Other possible simulation features can also be included, such as the factor loadings, guessing and slip parameters, and ordinal items. The time usage of selected methods can be recorded to further investigate their performance regarding the time efficiency. Finally, one study mentioned that it was better to use Pearson correlation instead of tetrachoric correlation for binary items under certain circumstances (Cosemans et al., 2022), which can be investigated further.

Conclusion

This simulation study compares factor retention methods in the exploratory MIRT for dichotomous items with between-item and within-item structures. The methods selected include commonly used statistical methods as well as innovative ML methods. Previous research mainly concentrates on the properties of a subset of these methods, under data generated by the EFA rather than the MIRT, not accounting for dichotomous data and large-scale assessments. The findings of this study fill these knowledge gaps and show that the factor retention methods indeed show different patterns in exploratory MIRT analyses for large-scale dichotomous data. In particular, MAP, RF, HistGBDT, XGBoost, and ANN perform noticeably better than other methods across all simulation features and the two factor structures; among them, HistGBDT generally performs better. Performing ML methods with using the results of statistical methods as extra features can enhance their performance. Further analyses of the effects of simulation features show that an increased degree of data missingness downgrades methods' performance, whereas an increased sample size shows an opposite trend. We recommend that practitioners use MAP and HistGBDT at the same time to

compensate for the downsides of statistical and ML methods. Future studies can extend to other possible methods and simulation scenarios.

References

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. American Educational Research Association.

Bentler, P. M., & Bonett, D. G. (1980). Significance tests and goodness of fit in the analysis of covariance structures. *Psychological Bulletin*, *88*(3), 588–606. https://doi.org/10.1037/0033-2909.88.3.588

Bock, R. D., Gibbons, R., & Muraki, E. (1988). Full-information item factor analysis. *Applied Psychological Measurement*, *12*(3), 261–280. https://doi.org/10.1177/014662168801200305

Bollen, K. A. (1989). *Structural equations with latent variables*. John Wiley & Sons.

Bonett, D. G., & Price, R. M. (2005). Inferential methods for the tetrachoric correlation coefficient. *Journal of Educational and Behavioral Statistics*, *30*(2), 213–225. https://doi.org/10.3102/10769986030002213

Bonifay, W. (2020). *Multidimensional item response theory*. Sage.

Braeken, J., & Van Assen, M. A. L. M. (2017). An empirical Kaiser criterion. *Psychological Methods*, *22*(3), 450–466. https://doi.org/10.1037/met0000074

Brown, T. A. (2015). *Confirmatory factor analysis for applied research* (2nd ed.). The Guilford Press.

Cai, L. (2010). High-dimensional exploratory item factor analysis by a metropolis–hastings robbins–monro algorithm. *Psychometrika*, *75*(1), 33–57. https://doi.org/10.1007/s11336-009-9136-x

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, *1*(2), 245–276. https://doi.org/10.1207/s15327906mbr0102_10

Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, *48*(6). https://doi.org/10.18637/jss.v048.i06

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chollet, F. (2021). *Deep learning with python* (2nd ed.). Manning.

Cosemans, T., Rosseel, Y., & Gelper, S. (2022). Exploratory graph analysis for factor retention: Simulation results for continuous and binary data. *Educational and Psychological Measurement*, *82*(5), 880–910. https://doi.org/10.1177/00131644211059089

Gini, C. (1921). Measurement of inequality of incomes. *The Economic Journal*, *31*(121), 124. https://doi.org/10.2307/2223319

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PLOS ONE*, *12*(6), 1–26. https://doi.org/10.1371/journal.pone.0174035

Goretzko, D. (2022). Factor retention in exploratory factor analysis with missing data. *Educational and Psychological Measurement*, *82*(3), 444–464. https://doi.org/10.1177/00131644211022031

Goretzko, D., & Bühner, M. (2020). One model to rule them all? Using machine learning algorithms to determine the number of factors in exploratory factor analysis. *Psychological Methods*, *25*(6), 776–786. https://doi.org/10.1037/met0000262

Gorsuch, R. L. (2015). *Factor analysis*. Routledge.

Green, S. B., Levy, R., Thompson, M. S., Lu, M., & Lo, W.-J. (2012). A proposed solution to the problem with using completely random data to assess the number of factors with

parallel analysis. *Educational and Psychological Measurement*, *72*(3), 357–374. https://doi.org/10.1177/0013164411422252

Green, S. B., Redell, N., Thompson, M. S., & Levy, R. (2016). Accuracy of revised and traditional parallel analyses for assessing dimensionality with binary data. *Educational and Psychological Measurement*, *76*(1), 5–21. https://doi.org/10.1177/0013164415581898

Guo, W., & Choi, Y.-J. (2023). Assessing dimensionality of IRT models using traditional and revised parallel analyses. *Educational and Psychological Measurement*, *83*(3), 609–629. https://doi.org/10.1177/00131644221111838

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, *30*(2), 179–185. https://doi.org/10.1007/BF02289447

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*(1), 141–151. https://doi.org/10.1177/001316446002000116

Kaiser, H. F. (1970). A second generation little jiffy. *Psychometrika*, *35*(4), 401–415. https://doi.org/10.1007/BF02291817

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017, December). LightGBM: A highly efficient gradient boosting decision tree. *31st International Conference on Neural Information Processing Systems*. https://hal.science/hal-03953007

Kolm, S.-C. (1999). The rational foundations of income inequality measurement. In *Handbook of income inequality measurement* (pp. 19–100). Springer.

Lawley, D. N. (1940). VI.—The estimation of factor loadings by the method of maximum likelihood. *Proceedings of the Royal Society of Edinburgh*, *60*(1), 64–82. https://doi.org/10.1017/S037016460002006X

Lenth, R. V., Bolker, B., Buerkner, P., Giné-Vázquez, I., Herve, M., Jung, M., Love, J., Miguez, F., Riebl, H., & Singmann, H. (2023). *emmeans: Estimated marginal means, aka least-squares means* (Version 1.8.8) [Computer software]. https://cran.r-project.org/web/packages/emmeans/index.html

Liu, Y., Robin, F., Yoo, H., & Manna, V. (2018). Statistical properties of the GRE® psychology test subscores. *ETS Research Report Series*, *2018*(1), 1–13. https://doi.org/10.1002/ets2.12206

Mair, P. (2018). *Modern psychometrics with R*. Springer.

Miller, L. A., & Lovler, R. L. (2018). *Foundations of psychological testing*. SAGE Publications.

Molenberghs, G., & Verbeke, G. (2005). *Models for discrete longitudinal data*. Springer.

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Raîche, G., Walls, T. A., Magis, D., Riopel, M., & Blais, J.-G. (2013). Non-graphical solutions for cattell's scree test. *Methodology*, *9*(1), 23–29. https://doi.org/10.1027/1614-2241/a000051

Revelle, W., & Rocklin, T. (1979). Very simple structure: An alternative procedure for estimating the optimal number of interpretable factors. *Multivariate Behavioral Research*, *14*(4), 403–414. https://doi.org/10.1207/s15327906mbr1404_2

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. *Psychological Assessment*, *24*(2), 282–292. https://doi.org/10.1037/a0025697

USMLE. (2023). *2024 USMLE bulletin of information*. https://www.usmle.org/sites/default/files/2023-08/2024bulletin.pdf.pdf

Velicer, W. F. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika*, *41*(3), 321–327. https://doi.org/10.1007/BF02293557

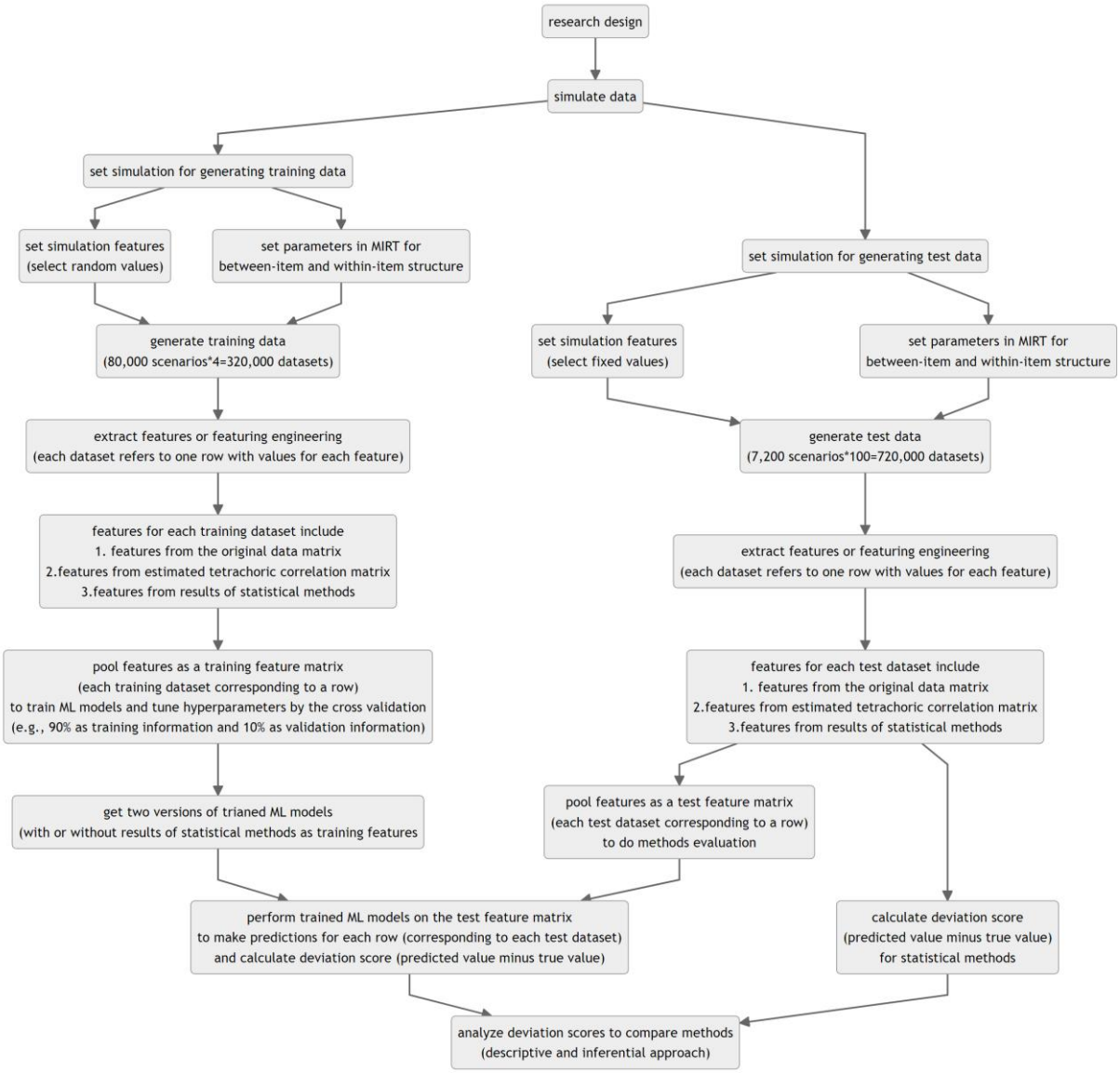
Wirth, R. J., & Edwards, M. C. (2007). Item factor analysis: Current approaches and future directions. *Psychological Methods*, *12*(1), 58–79. https://doi.org/10.1037/1082-989X.12.1.58

Xia, Y., & Havan, S. (2024). Using multiple imputation to account for the uncertainty due to missing data in the context of factor retention. *Educational and Psychological Measurement*, *84*(3), 577–593. https://doi.org/10.1177/00131644231178800

Factor Retention in Exploratory Multidimensional Item Response Theory - Supplemental Materials

Figure S1

Research Design Flowchart



Note. MIRT refers to Multidimensional Item Response Theory (MIRT). ML refers to Machine Learning.

Table S1

Descriptive Results (Complete Table)

	Non-convergence Proportion		Correct-factoring Proportion		Under-factoring Proportion		Over-factoring Proportion		Bias		Precision	
	Between	**Within**	**Between**	**Within**	**Between**	**Within**	**Between**	**Within**	**Between**	**Within**	**Between**	**Within**
KC	0	0	0	0	0	0	1	1	162.1081	159.4733	162.1081	159.4733
PA	0	0	.0049	.0556	0	0	.9951	.9444	89.2115	84.2395	89.2115	84.2395
EKC	0	0	.0180	.0996	0	0	.9820	.9004	66.9803	65.2540	66.9803	65.2540
Scree Plot (OC)	0	0	.0101	.0597	.0060	.0032	.9839	.9370	28.7146	26.8213	28.7484	26.8408
Scree Plot (AF)	0	0	.3083	.3663	.6666	.5983	.0250	.0355	-2.8665	-2.6440	2.9166	2.7155
VSS (C1)	0	0	.3906	.2309	.4603	.7020	.1491	.0671	-1.6242	-2.6735	2.1610	2.8884
VSS (C2)	0	0	.4792	.3232	.2692	.4168	.2516	.2600	-0.5528	-0.4917	1.2881	1.7149
MAP	0	0	.7131	.8109	.1827	.1123	.1043	.0768	-0.5523	-0.2699	0.7609	0.4235
EGA	.9454	.8671	.1185	.1167	.8434	.4960	.0381	.3873	-3.2580	0.1307	3.4150	3.6408
RF (Regression)	0	0	.7509	.8468	.0591	.1142	.1900	.0391	0.3644	-0.2514	0.5541	0.4003
RF (Classifier)	0	0	.7594	.8495	.0549	.1140	.1857	.0364	0.3193	-0.2581	0.5454	0.4221
RF (Ordinal)	0	0	.7578	.8507	.0558	.1123	.1864	.0370	0.3310	-0.2580	0.5388	0.4120
HistGBDT (Regression)	0	0	.7235	.8317	.0646	.1281	.2119	.0402	0.3165	-0.3069	0.5823	0.4528
HistGBDT (Classifier)	0	0	.7701	.8665	.0357	.1031	.1942	.0304	0.3473	-0.2533	0.5637	0.3935
HistGBDT (Ordinal)	0	0	.7707	.8504	.0517	.1119	.1776	.0377	0.3007	-0.2519	0.4992	0.3972
XGBoost (Regression)	0	0	.7263	.8350	.0410	.1183	.2327	.0467	0.5113	-0.2201	0.6406	0.3923
XGBoost (Classifier)	0	0	.7557	.8650	.0244	.1034	.2199	.0315	0.4456	-0.2457	0.5936	0.3865
XGBoost (Ordinal)	0	0	.6961	.8032	.1266	.1432	.1774	.0536	0.0939	-0.2992	0.6033	0.4670
ANN	0	0	.7518	.8446	.0646	.0964	.1836	.0590	0.2415	-0.0839	0.4770	0.2866
RF (Regression-extra)	0	0	.7699	.8472	.0722	.1203	.1580	.0325	0.2294	-0.3054	0.4827	0.3949
RF (Classifier-extra)	0	0	.7777	.8533	.0844	.1184	.1379	.0283	0.0648	-0.3127	0.4677	0.3923
RF (Ordinal-extra)	0	0	.7758	.8536	.0834	.1182	.1408	.0282	0.0865	-0.3137	0.4734	0.3945
HistGBDT (Regression-extra)	0	0	.7718	.8599	.0946	.1216	.1336	.0185	0.0535	-0.3527	0.5475	0.4078
HistGBDT (Classifier-extra)	0	0	.7823	.8710	.0650	.1137	.1527	.0153	0.0983	-0.3390	0.5120	0.3819
HistGBDT (Ordinal-extra)	0	0	.7779	.8500	.0689	.1142	.1532	.0358	0.1872	-0.2605	0.4466	0.3682

XGBoost (Regression-extra)	0	0	.7701	.8261	.0631	.1196	.1669	.0543	0.2799	-0.2312	0.4874	0.4196
XGBoost (Classifier-extra)	0	0	.7673	.8706	.0498	.1153	.1829	.0141	0.2155	-0.3410	0.5111	0.3841
XGBoost (Ordinal-extra)	0	0	.7470	.8254	.1365	.1343	.1164	.0402	-0.1395	-0.3234	0.4769	0.4323
ANN (extra)	0	0	.7827	.8471	.0977	.1166	.1312	.0364	0.0682	-0.1956	0.4004	0.3022

Note. ***Non-convergence proportion***: the number of non-convergence results divided by the total number of results; ***correct-factoring proportion***: the number of correct-factoring results divided by the total number of results; ***under-factoring proportion***: the number of under-factoring results divided by the total number of results; ***over-factoring proportion***: the number of over-factoring results divided the total number of results; ***bias***: the average deviation score (the number of predicted factors minus the true number of factors); ***precision***: the average absolute deviation score. The word "***extra***" within the parenthesis refers to the model trained with the results of statistical methods as extra features.

Hyperparameter Tuning and Results for RF and GBDT

Tuning procedure

In this section, we further detailed the hyperparameter tuning procedure of the RF and GBDT models. Specifically, in this study we considered the implementations of Random Forest and Histogram-based Gradient Boosting by the Python package "scikit-learn", and the implementation of XGBoost by the Python package "xgboost". Although very relevant, we do not consider the ordinal Random Forest implementation by the R package "OrdinalForest", as preliminary experiments indicate the performance is significantly below that of the other models and computational time makes the model infeasible for the scale of this study.

For each model we consider 3 variants: a regression variant (which predicts the number of factors as a continuous numerical response), a classification variant (which predicts the number of factors as a categorical response), and an ordinal variant (which transforms the problem into a multi-target binary classification problem). Although the classification variant seems like a natural choice, it does not take the ordinal nature of the response variable into account (which would require unequal misclassification costs, often not supported natively in these models). Therefore, we also consider an ordinal classification trick (Cheng, 2007)[5]. Instead of predicting whether the number of factors equals *{1, 2, ..., F}*, the ordinal variant predicts separately whether the number of factors is greater than 1, greater than 2, … greater than F-1. In this way, binary classification models can be trained to take the order of the response variable into account.

The selection of variant can be seen as a meta-hyperparameter. To fairly compare all three variants (e.g., on Mean Squared Error (MSE)), we post-processed the predictions of the classification and ordinal variants into a numerical outcome. Specifically, if a model in these

[5] Cheng, J. (2007, April 8). *A neural network approach to ordinal regression*. arXiv.org. https://arxiv.org/abs/0704.1028

variants predicted probabilities $\{p_1, p_2, \ldots p_F\}$ for a dataset to have $\{1,2,\ldots,F\}$ factors respectively, we used these probabilities as weights to give a continuous prediction of $\sum_{i=1}^{F} i \cdot p_i$.

The hyperparameter distributions under consideration are shown in Table S2. The range of these distributions was decided based on a preliminary exploration of univariate effects of each hyperparameter on performance (with all other hyperparameters on their default values). A randomized search was performed to optimize the hyperparameters for each model, where 60 hyperparameter combinations were sampled at random from the distributions in Table S2. Each hyperparameter combination was scored by its average MSE on the test sets in a stratified 10-fold cross-validation setup (stratified based on the label distribution). The combination with the lowest average MSE was retained.

Tuning results

After tuning the hyperparameters, the model with the lowest cross-validated MSE on both the between-item and the within-item structures was the classifier variant of HistGBDT. A confusion matrix for both structures on the test set is shown in Figure S2 (attained by rounding the predictions to the nearest integer). In particular, we noticed a tendency to over-factor for the between-item structure and to under-factor for the within-item structure.

An ablation study was also performed to assess the utility of tuning in this context. To this end, the test set performance of each tuned model was compared to the model with default hyperparameters defined by the used libraries (tuned and default models both trained on the full training set). The results for the two structures can be found in Table S3. For the between-item structure, tuning slightly improved predictive performance in general. For the within-item structure, tuning also had a positive effect in general, although the ordinal variant of XGBoost seems to perform noticeably worse after tuning.

Table S2

Hyperparameter Combinations Under Consideration for Tree Ensemble Methods

Parameter	Random forest	HistGBDT	XGBoost
Number of trees	100	100	100
Maximum depth per tree	10, 15, unconstrained	2, 3, 4, 5, 6, 7, 8	2, 3, 4, 5, 6
Number of features per split	U([10%, 100%])	NA	U([10%, 100%])
Number of samples per tree	100%	100%	U([60%, 100%])
Learning rate	NA	LU([10^{-2}, 10^{0}])	LU([10^{-2}, 10^{0}])
L2 regularization	NA	0, 0.1, 1	NA
Min split loss reduction	NA	NA	U([0, 0.5])
Splitting criterion or loss function	Regression: (Friedman) MSE, Poisson Classification: Gini, Entropy	Regression: squared error, gamma, poisson, quantile	Regression: squared loss Classification: Poisson

Note. Parameters without mentions are kept to their default values. U([a,b]) represents the uniform distribution on [a,b], LU([a,b]) the $\log_{10}$-uniform distribution on [a,b].

Table S3

Ablation Study: Test Set MSE of Tuned Model Versus Model With Default Hyperparameters

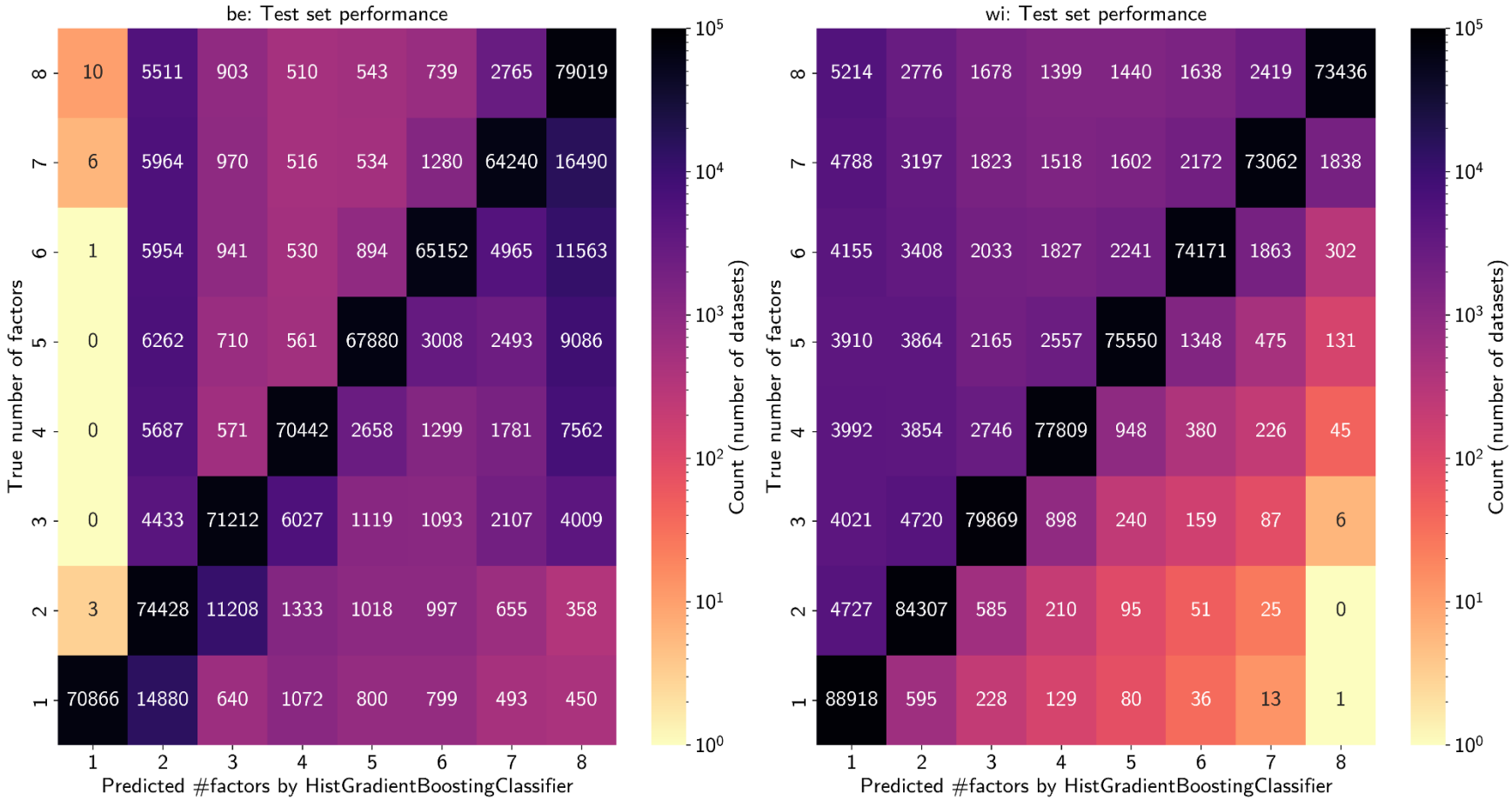
		Between-item		Within-item	
Model	**Variant**	**Default**	**Tuned**	**Default**	**Tuned**
Random forest	Regression	2.603	**1.246**	1.545	**1.405**
	Classification	1.491	**1.279**	**1.411**	1.431
	Ordinal	1.444	**1.241**	**1.440**	1.463

	Regression	1.805	1.805	**1.463**	1.618
HistGBDT	Classification	**1.562**	1.681	1.620	**1.563**
	Ordinal	**1.143**	1.167	1.521	**1.194**
	Regression	2.431	**1.311**	1.474	**1.407**
XGBoost	Classification	1.692	**1.526**	1.582	**1.558**
	Ordinal	**1.175**	1.237	**1.360**	1.456

Note. For each comparison, the lowest (i.e., best) score is marked in bold.

Figure S2

Confusion Matrices of Test Set Predictions for the Classification Variant of HistGBDT, Showing Predictions Versus True Values



Note. This model achieved the lowest cross-validated mean squared error on the training set. The left figure was under between-item structure; and the right one was under within-item structure.

Hyperparameter Tuning and Results for ANN

In this part, we further explain the details about the hyperparameter tuning and results of ANN. The Python package "TensorFlow" (version: 2.10.1) was applied for all analyses, following the instructions provided by Chollet (2021)[6].

Generally, we regarded this prediction task as a regression problem. For the between-item structure, first, we trained a baseline ANN model for the latter comparison with the ANN model with more complex architectures. The baseline model had a single neuron in a fully connected layer, and it was trained by using RMSprop optimizer, optimizing towards minimizing Mean Squared Error (MSE, as the loss function). The performance of the baseline model was evaluated by Mean Absolute Error (MAE). We set the epochs (iterations over the entire data) as 200 and the batch size (the number of per gradient update) as 2048, and 20% of the training data was used for validation during training. After that, we evaluated the trained baseline model based on the test data and the MAE for the baseline model was 0.9029.

Second, we arbitrarily defined a more complex ANN model to improve the baseline performance. The complex ANN model had six layers where each layer (from the first to the last) had 400, 200, 100, 50, 25, and 1 neuron respectively with Rectified Linear Unit (ReLU) as the activation function. It was trained by using RMSprop as optimizer, MSE as the loss function, and MAE as the performance metric based on a 5-fold cross validation. The epochs and the batch size were set as 200 and 2048 respectively during the training. After that, we further improved the model by examining the relationship between the epochs and the validation loss and decided to use 62 as the epochs. Based on the test dataset, the MAE for the complex model was 0.4957, which was noticeably better than the baseline model.

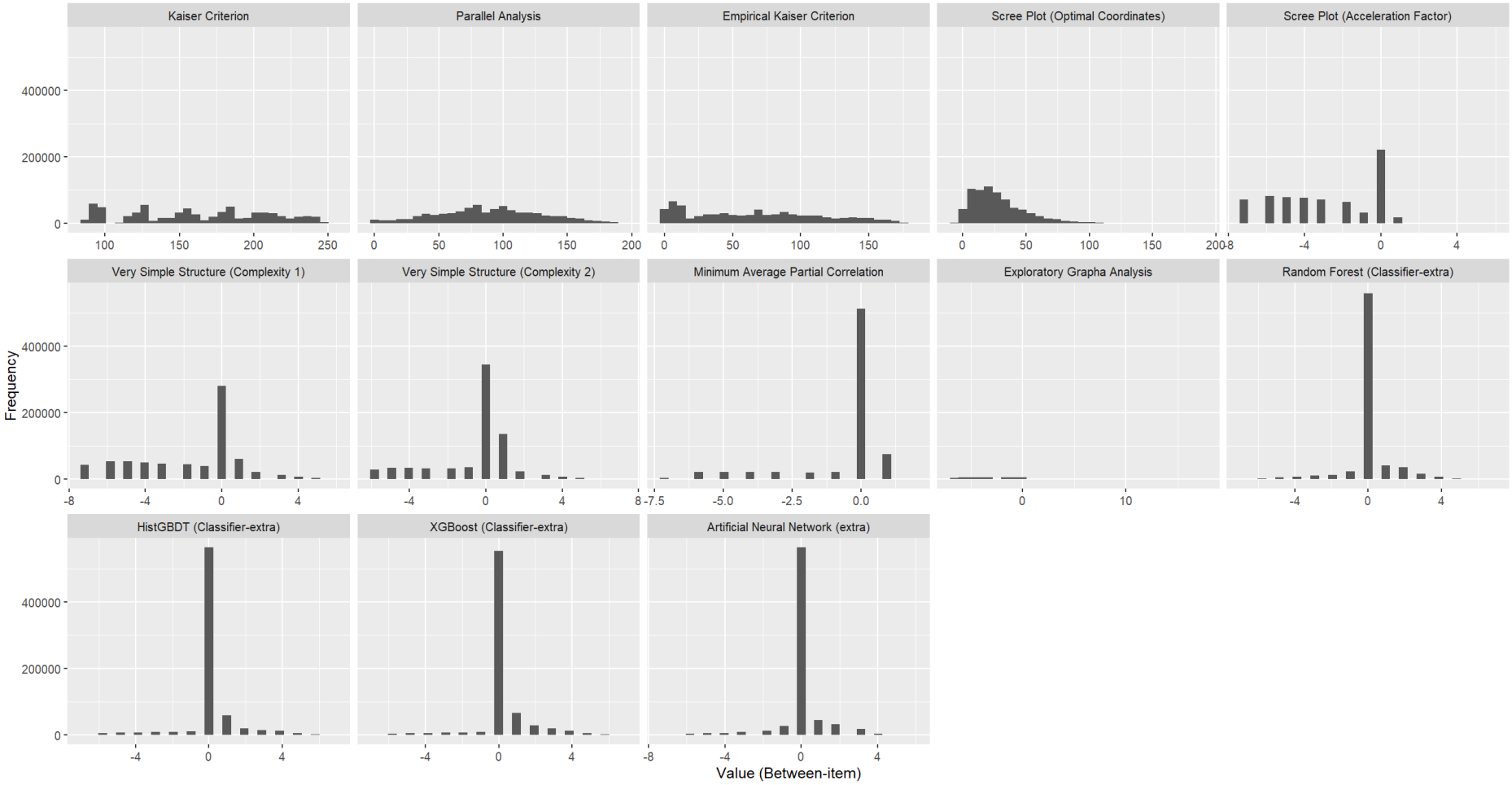
The analysis for the within-item structure followed similar operations. The settings of the baseline were the same as the between-item structure. The MAE of the baseline ANN model

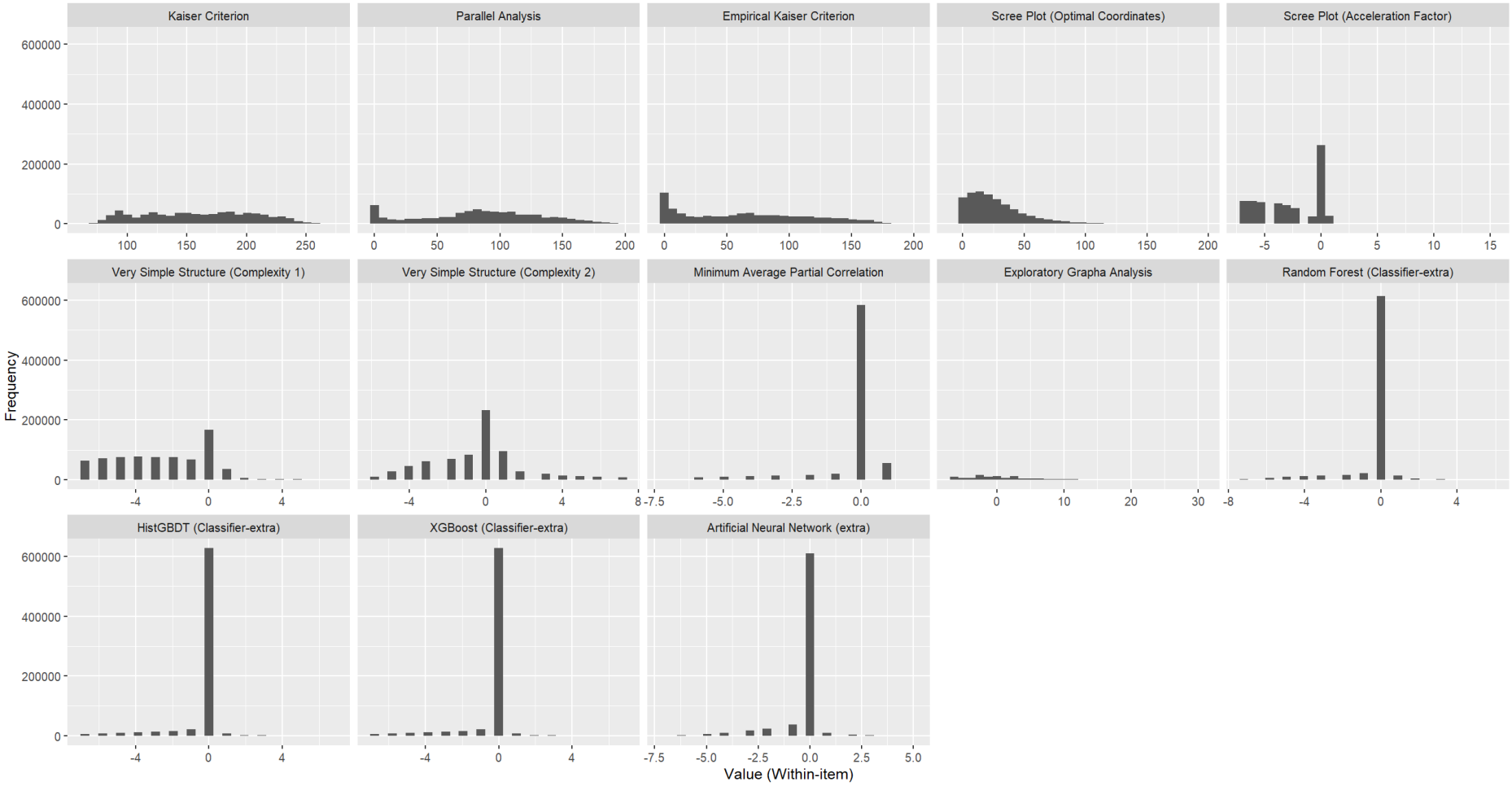
[6] Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Simon and Schuster.

based on the test data was 0.5410. The complex ANN model was designed differently. After several trials, it had three layers where each layer (from the first to the last) had 30, 15, and 1 neuron respectively with ReLU as the activation function. It was trained based on a 5-fold cross validation with RMSprop as optimizer, MSE as the loss function, and MAE as the performance metric. The epochs and the batch size were set as 200 and 2048 respectively during the training. The final epochs were set as 54 after the relationship between the epochs and the validation loss was examined. The MAE of the complex model for the test data was 0.5010, which was better than the baseline model.

Figure S3

Frequency Distribution of Deviation Scores

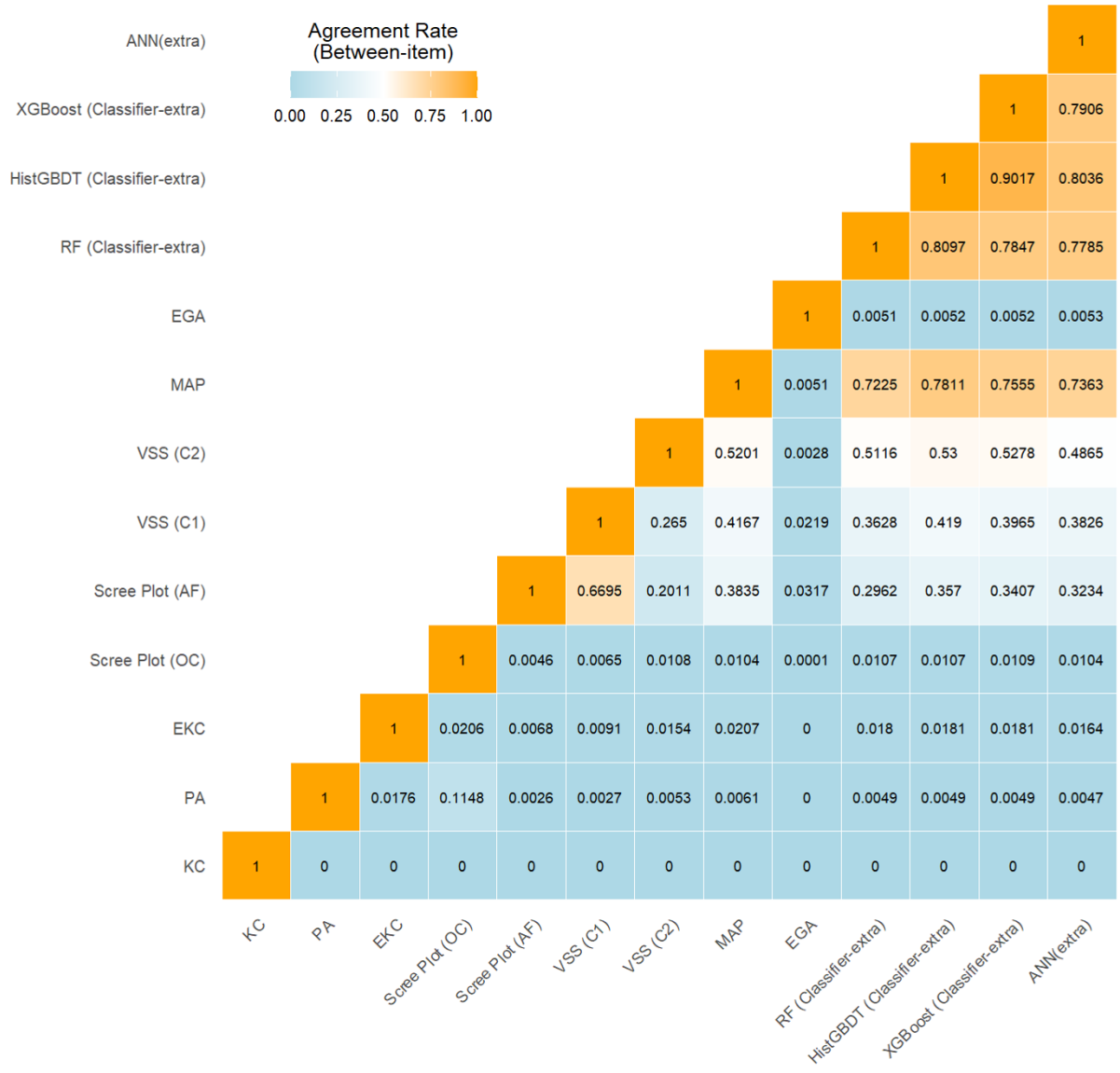




Note. The deviation score is equal to "the predicted number of factors minus the true number of factors".

Figure S4

Agreement Rate



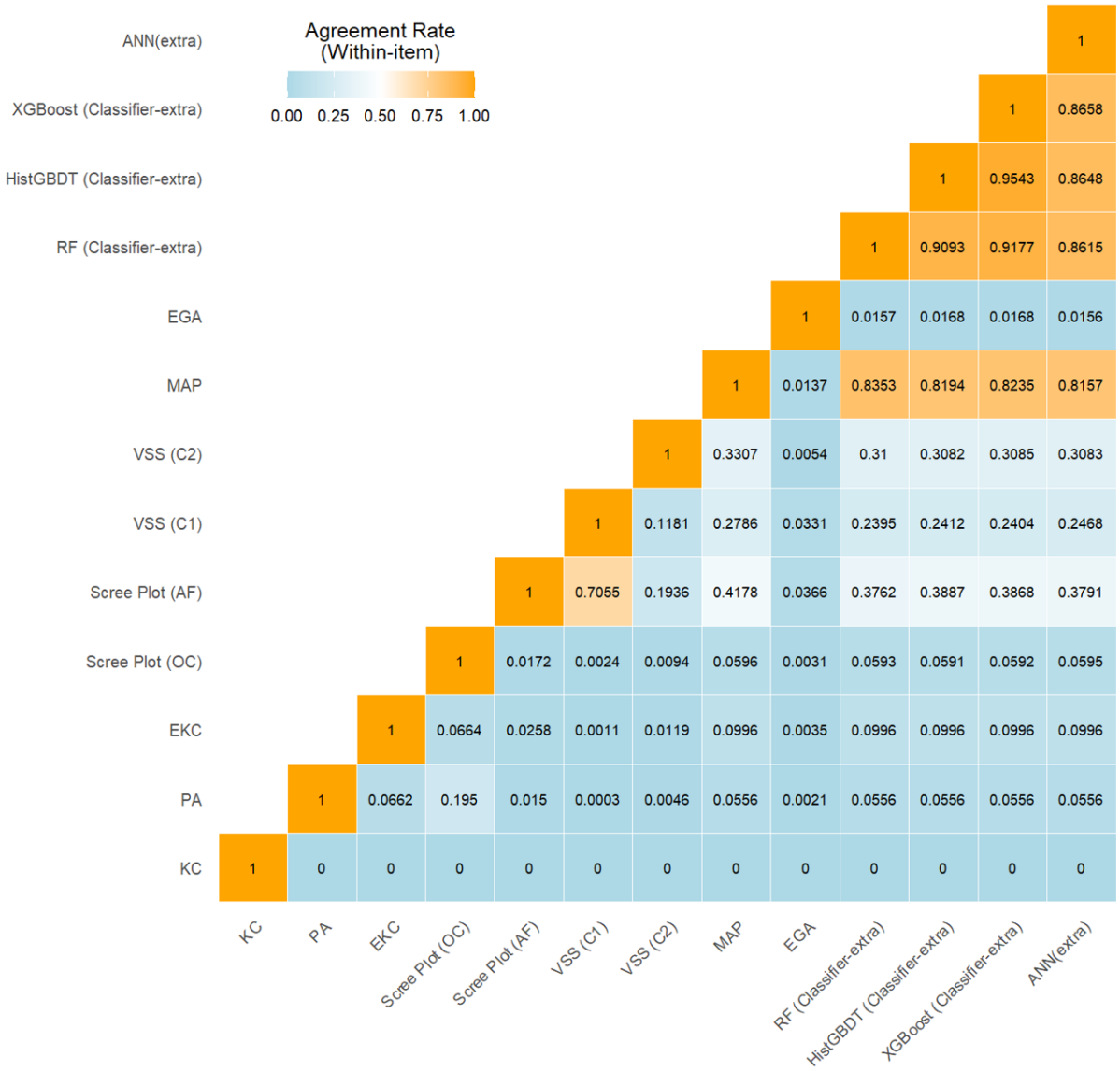
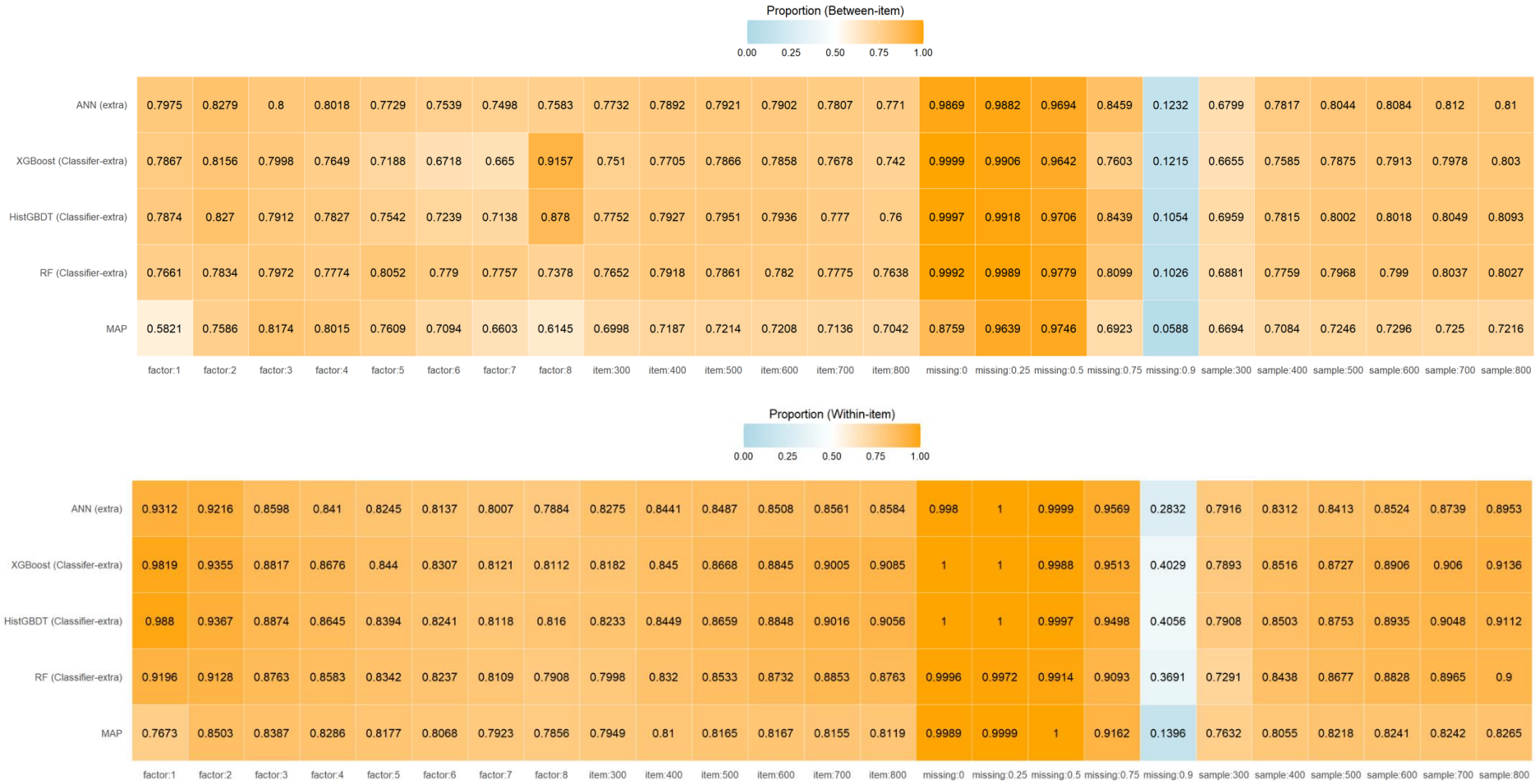

Agreement Rate
(Within-item)
0.00 0.25 0.50 0.75 1.00
ANN(extra)
XGBoost (Classifier-extra)
HistGBDT (Classifier-extra)
RF (Classifier-extra)
EGA
MAP
VSS (C2)
VSS (C1)
Scree Plot (AF)
Scree Plot (OC)
EKC
PA
KC
1 0.8658 0.8648 0.8615 0.0156 0.8157 0.3083 0.2468 0.3791 0.0595 0.0996 0.0556 0
1 0.9543 0.9177 0.0168 0.8194 0.3085 0.2404 0.3868 0.0592 0.0996 0.0556 0
1 0.9093 0.0168 0.8235 0.3082 0.2412 0.3887 0.0591 0.0996 0.0556 0
1 0.0157 0.8353 0.31 0.2395 0.3762 0.0593 0.0996 0.0556 0
1 0.0137 0.0054 0.0331 0.0366 0.0031 0.0035 0.0021 0
1 0.3307 0.2786 0.4178 0.0596 0.0996 0.0556 0
1 0.1181 0.0094 0.0119 0.0046 0
1 0.7055 0.1936 0.0024 0.0011 0.0003 0
1 0.0172 0.0258 0.015 0
1 0.0664 0.195 0
1 0.0662 0
1 0
1
KC
PA
EKC
Scree Plot (OC)
Scree Plot (AF)
VSS (C1)
VSS (C2)
MAP
EGA
RF (Classifier-extra)
HistGBDT (Classifier-extra)
XGBoost (Classifier-extra)
ANN(extra)

Figure S5

Correct-Factoring Proportions for Selected Simulation Features



Effect Size Analysis of Simulations Features

We further investigated the size of the effects of simulation features on methods' performance. Table S4 shows the estimated results of partial eta squared and Cohen's f for all interaction terms (the interaction between four methods and simulation features). For interpreting the two indexes, we use Cohen's (1988) rules of thumb: Cohens' f lower than 0.1 refers to a negligible effect; from 0.1 to 0.25 to a small effect; from 0.25 to 0.5 to a medium effect and higher than 0.5 to a large effect; a partial eta squared lower than 0.0099 refers to a negligible effect, from 0.0099 to 0.0588 to a small effect, from 0.0588 to 0.1379 to a medium effect and higher than 0.1379 to a large effect (Richardson, 2011)[7]. Therefore, we can say that for the between-item structure, sample size and missingness have small effects on the performance of selected methods; for the within-item structure, the number of items and factors, sample size, and missingness have small effects as well. Generally, the simulation features do not affect the performance of the methods tremendously to make them different from each other.

Table S4

Effect Size Analysis of Simulations Features

	Between-item			Within-item		
	η^2	η_p^2	Cohens' f	η^2	η_p^2	Cohens' f
Analysis methods × The number of items	0.001	0.001	0.028	0.015	0.016	0.129
Analysis methods × The number of factors	0.004	0.004	0.067	0.024	0.026	0.163
Analysis methods × Sample size	0.013	0.014	0.120	0.022	0.023	0.154
Analysis methods × Missing percentages	0.032	0.039	0.201	0.028	0.034	0.189

Note. η^2, and η_p^2 means eta-squared and partial eta-squared respectively. Technical details of three metrics can be consulted by Cohen (1988).[8]

[7] Richardson, J. T. E. (2011). Eta squared and partial eta squared as measures of effect size in educational research. *Educational Research Review*, 6(2), 135–147. https://doi.org/10.1016/j.edurev.2010.12.001

[8] Cohen, J. (1988). *Statistical power analysis for the behavioral sciences (2nd ed)*. L. Erlbaum Associates.