psych

MDPI

Article

Evaluating Cluster-Level Factor Models with `lavaan` and M*plus*

Suzanne Jak [1,*], Terrence D. Jorgensen [1] and Yves Rosseel [2]

[1] Research Institute of Child Development and Education, University of Amsterdam, 1012 WX Amsterdam, The Netherlands; T.D.Jorgensen@uva.nl
[2] Department of Data Analysis, Ghent University, B-9000 Ghent, Belgium; Yves.Rosseel@UGent.be
[*] Correspondence: S.Jak@uva.nl

Abstract: *Background*: Researchers frequently use the responses of individuals in clusters to measure cluster-level constructs. Examples are the use of student evaluations to measure teaching quality, or the use of employee ratings of organizational climate. In earlier research, Stapleton and Johnson (2019) provided advice for measuring cluster-level constructs based on a simulation study with inadvertently confounded design factors. We extended their simulation study using both M*plus* and `lavaan` to reveal how their conclusions were dependent on their study conditions. *Methods:* We generated data sets from the so-called configural model and the simultaneous shared-and-configural model, both with and without nonzero residual variances at the cluster level. We fitted models to these data sets using different maximum likelihood estimation algorithms. *Results:* Stapleton and Johnson's results were highly contingent on their confounded design factors. Convergence rates could be very different across algorithms, depending on whether between-level residual variances were zero in the population or in the fitted model. We discovered a worrying convergence issue with the default settings in M*plus*, resulting in seemingly converged solutions that are actually not. Rejection rates of the normal-theory test statistic were as expected, while rejection rates of the scaled test statistic were seriously inflated in several conditions. *Conclusions:* The defaults in M*plus* carry specific risks that are easily checked but not well advertised. Our results also shine a different light on earlier advice on the use of measurement models for shared factors.

Keywords: multilevel SEM; cluster-level constructs; maximum likelihood estimation

check for updates

Citation: Jak, S.; Jorgensen, T.D.; Rosseel, Y. Evaluating Cluster-Level Factor Models with `lavaan` and M*plus*. *Psych* **2021**, *3*, 134–152. https://doi.org/10.3390/psych3020012

Academic Editor: Alexander Robitzsch

Received: 29 April 2021
Accepted: 26 May 2021
Published: 31 May 2021

1. Introduction

To measure constructs at the cluster level—termed *shared constructs* [1,2]—researchers frequently use the responses of individuals in clusters. For example, students' evaluations may be used to measure the teaching quality of instructors, patient reports may be used to evaluate social skills of therapists, and residents' ratings may be used to evaluate neighborhood safety.

When multiple items are used to measure such cluster-level constructs, multilevel confirmatory factor analysis (CFA) models are useful. These models allow for the evaluation of the factor structure at the cluster level (modeling the (co)variances among item means across clusters), and at the individual level (modeling the (co)variances across individuals within clusters).

If the cluster-level construct, for example teacher quality, would be perfectly measured using the responses of students, all students evaluating the same teacher would agree, and provide exactly the same item scores. In that case, there will not be any systematic variance in the item scores within clusters (but there will still be variance due to sampling error).

In practice, individuals within a cluster do not all provide the same responses to the items, leading to systematic variance (and covariance) to be explained at the individual level. The question then arises how the variance within clusters should be modeled. Stapleton et al. [1] proposed a model for cluster-level constructs with a saturated model at the individual level. This work was updated by proposing and evaluating several two-level factor models in Stapleton and Johnson [2]. In this article, we will provide a simulation study to evaluate under what scenarios the proposed models are able to provide sensible results, partly by replicating the study by Stapleton and Johnson. A second aim is to investigate the types and frequency of estimation problems in the software packages M*plus* [3] and `lavaan` [4], which have different default settings that could have important consequences for convergence problems and the quality of obtained results.

1.1. Different Types of Two-Level Models

Stapleton and Johnson [2] evaluated three different models for cluster-level constructs: the 'configural model', the 'unconstrained model', and the 'simultaneous shared-and-configural model'. We will introduce these three models in the next section.

1.1.1. Configural Model

Configural models are factor models in which the same factor structure is applied to the within and the between level, and the factor loadings are constrained to be equal across levels. The configural model decomposes the common factor(s) into a within-cluster and a between-cluster part, meaning that the between- and within-components can be interpreted as stemming from the same latent variable. For example, one could have measured collaborative playing skills in children in different classrooms using several items, hypothesizing that the collaborative playing skills systematically vary within as well as across classrooms (for a teaching-quality example see [5]). The configural model would allow one to interpret the differences in the within-level common factor representing within-classroom differences in collaborative play, and the differences in the between-level common factor representing between-classroom differences in collaborative play.

The cross-level invariance of factor loadings is necessary for a meaningful interpretation of factors at the two levels [1,6–12]. The left panel of Figure 1 shows a population configural model in which each indicator's factor loading is equal across levels.

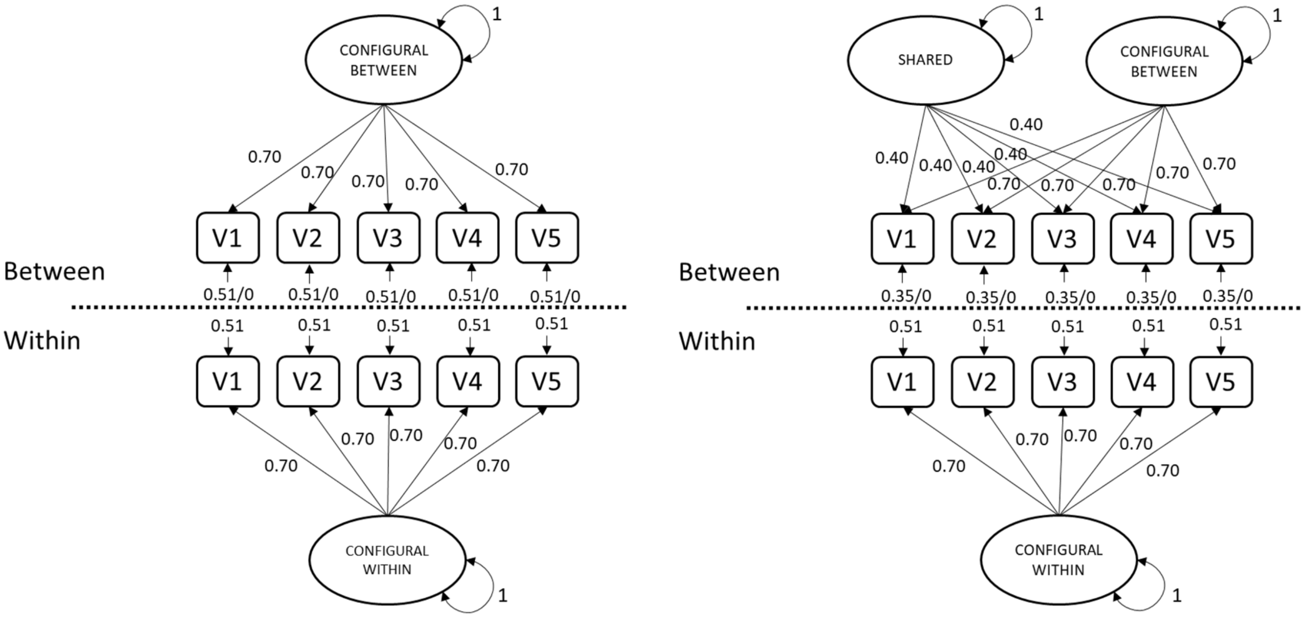

Figure 1. Data generating models with parameter values.

1.1.2. Unconstrained Model

Unconstrained two-level factor models do not have cross-level invariance of factor loadings, implying that different constructs may be measured in each cluster, or that different structures altogether dictate covariation at each level. As a result, the common factors at the two levels do not reflect the within- and between component of the same latent variable. Instead, the theoretical meaning of the latent variables is different at the two levels (as well as across clusters). An unconstrained model resembles the left panel of Figure 1, but without equality across levels of each indicator's loading.

In practice, it can be quite difficult to describe exactly how the interpretation of the factors varies across levels. For example, if the factor loading for a specific indicator is higher at the between level than at the within level, then the factor at the between level will represent more of the content of that specific indicator (and the other way around). If the pattern of higher and smaller factor loadings varies across items, it will become harder to appropriately label the common factor at each level. The unconstrained two-level model is therefore not a theoretically useful model. The model was still included in our (and Stapleton and Johnson's) study because it is regularly applied in practice, and because we argue that an unconstrained two-level model is capable of fitting data generated from a simultaneous shared-and-configural model.

1.1.3. Simultaneous Shared-and-Configural Model

The simultaneous shared-and-configural model was introduced by Stapleton et al. [1] and comprises a configural model with an added common factor at the between level (see the right panel of Figure 1). In this model, the configural factor represents a 'nuisance' factor, i.e., a common factor that is accidentally measured using the individual responses, and that also systematically varies over clusters. In the examples provided by Stapleton and Johnson [2], this nuisance factor could represent an individual's tendency to always agree with statements (acquiescence), which could also have a cluster component (e.g., some clusters exhibit more acquiescence than others). The additional between-level factor, which is uncorrelated with the configural factor, then represents the objective shared construct that was the focus of the research. This shared factor does not differ within clusters, and therefore has no within-level component.

Stapleton and Johnson [2] constrained the intraclass correlation (ICC) of the configural factor to a specific value, for example by constraining the variance of the between-level factor to (0.05/0.95) times the within-level factor, leading to an ICC of 0.05 for the configural factor. The authors argued that such a constraint was needed to identify the model. They seemed to have overlooked that the cross-level constraint on the factor loadings already identifies the factor variance of the configural factor on the between level. The shared factor can be identified by either fixing its variance to 1 or by fixing one of the factor loadings to 1. The additional constraint is therefore not necessary to identify the model.

1.2. Estimation of Two-Level Models

Different algorithms have been proposed to obtain maximum likelihood (ML) estimates, and their availability and defaults vary across software. In this section we discuss four algorithms available in M*plus*, two of which are also available in `lavaan`. We also discuss the estimation of between-level residual variances, which was confounded with model type in Stapleton and Johnson's [2] simulation study.

1.2.1. Maximum Likelihood Estimation Algorithms

Both `lavaan` [4] and M*plus* [3] use normal-theory ML estimation by default for continuous variables, using the observed information matrix to derive *SE*s. Although also available in `lavaan`, only M*plus* defaults to a χ^2 statistic and *SE*s that are robust to nonnormality. The robust χ^2 statistic provided by `lavaan` and M*plus* is asymptotically equivalent to Yuan and Bentler's T_2^* statistic [13]. This robust test statistic (with *SE*s) is requested in `lavaan` with the argument estimator = "MLR" (or equivalently, test = "yuan.bentler.mplus" and se

= “robust.sem”; see the ?lavOptions help page) and in M*plus* with the ANALYSIS option ESTIMATOR = MLR [3] (chapter 16).

By default, `lavaan` maximizes the sum of log-likelihoods of the clusters—which M*plus* refers to as the observed-data log-likelihood (ALGORITHM = ODLL)—using a quasi-Newton (QN) algorithm, which is the same algorithm used for ML estimation with single-level data. The expectation–maximization (EM) algorithm is also available in both (`lavaan` and M*plus* to obtain ML estimates, although the implementation in `lavaan` is notably slower. The EM algorithm can be requested by passing the argument optim.method = “em” to `lavaan`, or with the ANALYSIS option ALGORITHM = EM in M*plus* [3] (chapter 16). M*plus* also has an accelerated EM algorithm (ALGORITHM = EMA), achieved by switching to QN when EM does not optimize quickly enough (i.e., when relative or absolute changes in log-likelihood do not decrease enough between iterations). M*plus* can also switch between a Fisher-scoring (FS) algorithm and EM (with ALGORITHM = FS), but EMA is the default, and neither EMA nor FS are currently available in `lavaan`. Availability of current options and their default settings are listed for `lavaan` and M*plus* in Table 1.

Table 1. Availability of estimation options in `lavaan` and M*plus*.

Software Package	**Require $\theta_B \geq 0$?**	**QN**		**EM**		**EMA**		**FS**	
		ML	**MLR**	**ML**	**MLR**	**ML**	**MLR**	**ML**	**MLR**
`lavaan`		D	✔	✔	✔				
M*plus*	✔ (D = 0.1^4)	✔	✔	✔	✔	✔	D	✔	✔

Note: ✔ indicates availability (or requirement). θ_B = between-level residual variances. QN = quasi-Newton algorithm. EM = expectation–maximization algorithm. EMA = accelerated EM algorithm. FS = Fisher scoring algorithm. ML(R) = (robust) maximum likelihood. D = default setting.

Convergence of the algorithm is determined by tracking criteria at each iteration—namely, the log-likelihood function and its first derivative. After any of these algorithm’s convergence criteria (i.e., the rules for stopping the optimizer from iterating further) have been met, it must be verified that the optimizer in fact converged on a maximum.

In M*plus*, convergence for any optimization algorithm for ML estimation can be controlled with ANALYSIS options, such as the ODLL derivative using CONVERGENCE (for QN) or MCONVERGENCE (for EM or EMA), as well as LOGCRITERION (absolute change in log-likelihood from previous iteration) and RLOGCRITERION (relative change in log-likelihood from previous iteration). The maximum number of iterations is set by ITERATIONS for QN and MITERATIONS for EM(A). The default optimizer used by `lavaan` is nlminb from the stats package, whose options can be set by passing a named list to the control = argument (see the ?lavOptions help page). `lavaan`’s defaults are list (iter.max = 10,000, abs.tol = .Machine$double.eps × 10, rel.tol = 1×10^{-10}). When using EM, `lavaan` has its own parallel dedicated arguments, shown in the last example on the tutorial page: https://lavaan.ugent.be/tutorial/multilevel.html, accessed on 31 May 2021.

Verifying that the optimizer converged on a maximum involves checking the first and second derivatives of the log-likelihood function with respect to the estimated parameters, respectively called the “gradient” and “Hessian.” If ML estimates were obtained, each element of the gradient vector should be effectively zero (the “first-order condition”). Upon finding a nonzero gradient element, `lavaan` warns that the optimizer did not find a ML solution and “estimates below are most likely unreliable.” Likewise, M*plus* output will contain the message: “The model estimation did not terminate normally due to a non-zero derivative of the observed-data loglikelihood,” referring to the first-order condition. Any nonzero element of the gradient indicates the corresponding parameter estimate is not a ML estimate. But the reverse is not necessarily true: if the gradient consists only of zeros, it could be a minimum or a saddle point rather than a maximum. In order to verify the solution is a maximum, the Hessian should be negative definite. Because the Hessian is intensive to compute, this “second-order condition” is rarely checked to simply verify convergence. However, multiplying the Hessian by -1 yields the information matrix, the inverse of which is the asymptotic covariance matrix of the estimated parameters (the diagonal of which contains the squared *SE*s). Thus, if the information matrix is not positive definite (and so cannot be inverted), a warning is issued that *SE*s cannot be calculated.

Because EMA is the default algorithm in M*plus*, we describe one more computational detail about the acceleration aspect (which seems to be shared by the FS algorithm, but we did not focus on that). When the log-likelihood does not change fast enough between iterations, M*plus* attempts to accelerate optimization by switching from EM to QN. This could backfire if the QN step overshoots its target, instead decreasing the log-likelihood at the next iteration. In such circumstances, M*plus* will then restart the EM algorithm as though the user selected EM instead of the default EMA, because EM alone (without switching to QN) will always increase the log-likelihood between iterations. However, we discovered that apparent convergence with EM after EMA failed does not necessarily converge when explicitly setting ALGORITHM = EM. Causes and potential consequences are provided in the Results section.

1.2.2. Between-Level Residual Variances in Two-Level Models

When strong factorial invariance across clusters holds, then in a two-level model, not only will factor loadings be equal across levels, residual variances at the between level will be zero [11,14–16]. For configural constructs (i.e., with cross-level invariance of factor loadings), any residual variance at the between level (θ_B) can therefore be interpreted as differences in intercepts across clusters (measurement bias, also called cluster bias by Jak, Oort and Dolan [14]. Nonzero residual variance at the between level ($\theta_B > 0$) means that the cluster-level differences in the indicators are not all explained by cluster-level differences in the common factor. In other words, variables other than what was intended to be measured cause differences in the indicator scores across clusters. In practice, it may not be realistic to expect exactly zero cluster bias for all indicators, similar to how exact invariance of intercepts generally does not hold [17]. That is, cluster invariance may hold only approximately, implying small θ_B instead of zero θ_B. Moreover, some indicators may be subject to cluster bias while other indicators are not (representing partial invariance [18]).

Sample estimates of θ_B vary around the population values, so when they are (nearly) zero, estimates can frequently take negative values simply due to sampling error. Thus, if strong factorial invariance across clusters holds even approximately ($\theta_B \cong 0$), then estimating θ_B under non-negative constraints may lead to trouble with convergence in samples that would have contained at least one negative variance under unconstrained estimation. By default, `lavaan` does not restrict θ_B estimates to be positive when using QN, while M*plus* does. The EM algorithm requires $\theta_B > 0$ in both packages, because the EM algorithm requires the between-level model-implied covariance matrix to be positive definite. The minimum-variance requirement (set with the ANALYSIS option VARIANCE) must be between 0 and 1, so negative values for θ_B are not permitted in M*plus*. This requirement may therefore result in nonconvergence for populations with $\theta_B \cong 0$.

In applications of (shared-and-)configural-construct models, it can be valuable to assess cluster bias to establish whether Level-2 residual variances should be constrained to zero, which could avoid negative-variance estimates. However, nonconvergence under minimum-variance-constrained estimation when $\theta_B \cong 0$ would prevent the ability to compare that model to one with strong factorial invariance across clusters ($\theta_B = 0$). In our simulation study, we explicitly crossed these design factors: zero vs. nonzero θ_B in the population model and fixed vs. estimated θ_B in the fitted model. We focus on conditions where population θ_B is exactly zero, representing exact invariance, but we will show some results based on conditions with $\theta_B = 0.01$ and $\theta_B = 0.0001$ as well. Chen, Bollen, Paxton, Curran and Kirb [19] describe the possible causes of, consequences of, and possible strategies to handle inadmissible solutions in more detail. Negative variance estimates could result from either model misspecification or sampling error. So before fixing negative variance estimates to zero, one should first test the null hypothesis that the parameter is an admissible solution. For example, if the 95% confidence interval for a residual variance includes positive values, then one cannot reject the null hypothesis (using $\alpha = 0.05$) that the true population value is indeed positive; in this case, if the model fits well and there are no other signs of misspecification, one could conclude that the true parameter is simply close enough to zero that sampling error occasionally yields a negative estimate. For more discussion about negative variance parameters and estimates see [20–22].

1.3. Overview of the Study

In this article, we will provide an extensive simulation study to evaluate two-level models for measuring cluster-level constructs. We will replicate the analyses of Stapleton and Johnson [2] with slight alterations. The differences in the models to be evaluated are that we will evaluate a simultaneous shared-and-configural model with unconstrained ICC instead of unnecessarily fixing the ICC of the configural factor to various specific values. In addition, we will generate data with zero or nonzero between-level residual variances, and we fit models with freely estimated or with fixed (to zero) between-level residual variances. We justify our design choices and provide our expectations following the description of our design factors in Section 2.

2. Materials and Methods

We generated all data using R [15] (version 4.0.4). We used both `lavaan` [4] (Version 0.6-7 and M*plus* [3] (Version 8.5) to fit all models to those data. R syntax to generate and analyze the Monte Carlo results are available from the Open Science Framework: https://osf.io/sdwam/ (accessed on 31 May 2021).

2.1. Data Generating Models

Stapleton and Johnson [1] generated data from two population models, being a configural model with $\theta_B = 0$, and a shared model with $\theta_B > 0$. As this confounding of factor structure and presence of θ_B is not apparent from the article, their results are easily misinterpreted. We generated data from four different population models. These are models with or without the existence of an additional between-level construct, and with or without cluster bias (indicated by $\theta_B > 0$).

Where possible, we use the same population values for parameters as reported in Stapleton and Johnson [2]. Figure 1 shows the population models with (left panel) and without (right panel) the additional between-level factor. All population factor variances were 1. For the configural factor, all factor loadings at the within and between levels were 0.70. The factor loadings for the shared factor all were 0.40. The θ_B values were either fixed to zero or chosen to standardize the between-level factor loadings (e.g., $1 - (0.40^2 + 0.70^2) = 0.35$ in the model with the shared factor). Stapleton and Johnson only generated data from the shared-and-configural model with $\theta_B = 0$ and from the configural-only model with $\theta_B > 0$, so these factors were confounded in their study design.

These parameter values led to item intra class correlations (ICCs) of 0.50 in all conditions with $\theta_B > 0$. In conditions with $\theta_B = 0$, item ICCs were 0.32 for the configural model conditions, and item ICCs were 0.39 for the shared model conditions. The ICC of the configural factor was 0.50 in all conditions.

2.2. Sample Size Conditions

We generated data for 50, 100, and 200 clusters with a fixed size of 20 individuals per cluster. These are the same sample size conditions as Stapleton and Johnson [2], but without the 300-cluster condition.

2.3. Fitted Models

We fitted three different models (unconstrained, configural, shared-and-configural) to each simulated dataset, both with freely estimated θ_B and fixed $\theta_B = 0$. The unconstrained model is a two-level CFA model with one factor at each level and with freely estimated factor loadings at both levels. The factor variances of the unconstrained model are fixed at one at both levels. This model has *df* = 10 in the condition with freely estimated θ_B, and *df* = 15 in the condition with fixed $\theta_B = 0$.

The configural model adds cross-level invariance constraints to the unconstrained model. With factor loadings constrained to be equal across levels, the factor variance at the between level can be freely estimated. The configural model therefore has *df* = 14 in the condition with freely estimated θ_B, and *df* = 19 in the condition with fixed $\theta_B = 0$.

The shared-and-configural model adds to the configural model an orthogonal between-level common factor. In this model, the factor variance of the shared factor is fixed at 1, and the factor loadings of the shared factor are freely estimated. For the configural part of the model, the factor loadings are again constrained to be equal across levels, the variance of the within-level configural factor is fixed at one, and its variance at the between level is freely estimated. The shared-and-configural model has *df* = 9 in the conditions with freely estimated θ_B, and *df* = 14 in the condition with fixed $\theta_B = 0$.

2.4. Estimation Options

In all sample-size conditions, we fit all models to data from all populations using ML estimation with the QN algorithm and an observed information matrix in `lavaan` and M*plus*. In a follow-up study holding the number of clusters constant at 100, we also compared QN to EMA (the default algorithm in M*plus*) and EM in M*plus*. A third study used MLR (the default in M*plus*) to compare the rejection rates reported by Stapleton and Johnson [2].

2.5. Number of Conditions and Replications

The primary study design consisted of 2 (configural or shared-and-configural population model) × 2 ($\theta_B = 0$ or $\theta_B > 0$ in the population) × 3 (sample size) = 12 data conditions. We generated 1000 datasets per condition. For all 12,000 datasets, 3 (unconstrained, configural, or shared) × 2 ($\theta_B = 0$ or $\theta_B > 0$) = 6 models were fitted with ML estimation using the QN algorithm in both software packages. In the first follow-up study, the subset of 4000 datasets with 100 clusters were analyzed using ML estimation for the same six models in both software packages, but additionally using EMA and EM in M*plus*. The second follow-up study used MLR, again with only QN in `lavaan` but QN and EMA in M*plus*.

2.6. Expectations Regarding Convergence and Rejection Rates

Multilevel structural equation modeling is very susceptible to estimation problems. Especially with small numbers of clusters, and/or little variance at the between level, non-converged and inadmissible solutions are frequently observed [19–22]. Overall, we expected more convergence problems in conditions with fewer clusters [2,8]. Table 2 depicts the four data-generating models (irrespective of sample size) in the rows, and the six fitted models in the columns. The grey cells represent the six conditions that were evaluated by Stapleton and Johnson [2]. We expected that when the correct model was fitted (cells labeled with 'T'), models would converge for nearly 100% of samples (with lower convergence in smaller samples), and rejection rates would be close to the nominal α level. When $\theta_B > 0$ was not taken into account (i.e., by fixing $\theta_B = 0$ in the analysis), we expected high rejection rates (cells labeled 'R' in Table 2). In conditions with an overparameterized model (cells labeled 'O' in Table 2), such as when $\theta_B = 0$ but was freely estimated, we expected more convergence problems, but nominal rejection rates for the converged cases.

In the remaining cells (labeled 'S' in Table 2), the shared factor was omitted from the model. Because the ratio of between-level population cross-loadings was proportional across indicators (i.e., they were all 0.70 for the configural factor and all 0.40 for the shared factor), the two between-level factors were perfectly confounded in Stapleton and Johnson's [2] population (which we replicated). Shared-factor loadings were only identified because the configural loadings were constrained to equality across levels. As a result, the shared-factor variance would have been absorbed by the single factor at the between level (inflating its variance), so the unconstrained and configural models should have fit perfectly to data generated from the shared-and-configural model. Therefore, we expected that rejection rates in these conditions would still be nominal, in contrast to the 100% rejection rates reported by Stapleton and Johnson [2] (p. 319), which they attributed to omitting the shared factor rather than to fixing $\theta_B = 0$ when data were generated with $\theta_B > 0$.

Table 2. Overview of the four data generation conditions (rows), and the six fitted models (columns).

Data-Generating Model		**Fitted Model**					
		Uncon		**Conf**		**Shared**	
		$\theta_B > 0$	$\theta_B = 0$	$\theta_B > 0$	$\theta_B = 0$	$\theta_B > 0$	$\theta_B = 0$
Configural	$\theta_B > 0$	O	R	T	R	O	R
	$\theta_B = 0$	O	O	O	T	O [a]	O
Shared	$\theta_B > 0$	S	R	S	R	T [a]	R
	$\theta_B = 0$	S	S	S	S	O	T

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. T = True model fitted, R = Theta between not taken into account, O = Overparameterized model, S = Shared factor not explicitly modeled. Grey cells represent conditions evaluated by Stapleton and Johnson [1]. [a] In contrast to Stapleton and Johnson [1], we did not fix the ICC of the configural factor.

Stapleton and Johnson [2] (Table 1) also reported that when data were generated from a configural-model population, fitting the unconstrained model yielded around 50% rejection rates, which should not be the case because it is an overparameterized model, as our Table 2 shows. They also reported 0% convergence when fitting the shared-and-configural model to data generated from a configural-model population, which they attributed merely to "no between-cluster covariance to be modeled above that explained by the configural" factor [2] (p. 319). Because overparameterization did not prevent convergence in their unconstrained model, we expected their 0% convergence could be at least partly related to the unnecessary constraint on the configural construct's ICC in the shared-and-configural models. Since Stapleton and Johnson used the M*plus* defaults (MLR + EMA), we will only compare the rejection rates we find with their findings in a follow up study, and not in our primary study in which we apply ML with QN.

3. Results

3.1. Convergence Rates in the Primary Study

Nonconvergence in `lavaan` only occurred when fitting the shared-and-configural models to datasets. Regardless of θ_B in the data-generating model, convergence was consistently >95% when estimating θ_B. When $\theta_B = 0$ in the population, oddly, convergence problems only occurred in `lavaan` when appropriately fixing $\theta_B = 0$, but convergence was still approximately 80% and improved in larger samples, whereas convergence in the same conditions was notably lower in M*plus* and varied erratically across sample sizes.

Regardless of the fitted model (unconstrained, configural, or shared), M*plus* additionally had convergence problems (nearly 0% convergence) when estimating θ_B despite $\theta_B = 0$ in the population. In contrast, `lavaan` had 100% convergence in the same conditions (except the conditions described in the previous paragraph). In order to evaluate the effects of generating data with exact or approximate cluster invariance, we also evaluated the conditions with 100 clusters while fixing θ_B to 0.0001 (the minimum value for a variance parameter in M*plus*) or 0.01 (a small but realistic amount of variance). Table A1 shows that the convergence problems when estimating θ_B freely persisted in conditions with approximate instead of exact cluster invariance, with 0–0.5% convergence in conditions with generated θ_B = 0.0001, and 12.4–43.1% convergence in conditions with generated θ_B = 0.01.

As expected, nonconvergence was generally exacerbated by fewer clusters (except when M*plus* fitted shared models with θ_B = 0 in the population and analysis models). For all conditions in Table 3 `lavaan` either converged more often than M*plus* or both packages converged in 100% of samples.

Table 3. Convergence rates for the six models in the four data conditions and three sample size conditions.

				50 Clusters		100 Clusters		200 Clusters	
Data Model		**Fitted Model**		`lavaan`	**M*plus***	`lavaan`	**M*plus***	`lavaan`	**M*plus***
Config	$\theta_B > 0$	Uncon	$\theta_B > 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	*$\theta_B > 0$*	1.000	1.000	1.000	1.000	1.000	1.000
		Config	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Shared	$\theta_B > 0$	0.969	0.681	0.998	0.793	1.000	0.938
		Shared	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
Config	$\theta_B = 0$	Uncon	$\theta_B > 0$	1.000	0.002	1.000	0	1.000	0
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	$\theta_B > 0$	1.000	0.002	1.000	0.001	1.000	0.003
		Config	*$\theta_B = 0$*	1.000	1.000	1.000	1.000	1.000	1.000
		Shared	$\theta_B > 0$	0.955	0	0.970	0	0.989	0
		Shared	$\theta_B = 0$	0.787	0.199	0.863	0.227	0.968	0.218
Shared	$\theta_B > 0$	Uncon	$\theta_B > 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	$\theta_B > 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Shared	*$\theta_B > 0$*	0.957	0.699	0.991	0.772	1.000	0.918
		Shared	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
Shared	$\theta_B = 0$	Uncon	$\theta_B > 0$	1.000	0	1.000	0	1.000	0
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	$\theta_B > 0$	1.000	0	1.000	0	1.000	0.003
		Config	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Shared	$\theta_B > 0$	0.960	0	0.970	0	0.987	0
		Shared	*$\theta_B = 0$*	0.821	0.358	0.868	0.215	0.938	0.254

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. *Italicized* = conditions in which the true model was fitted.

3.2. Rejection Rates in the Primary Study

Rejection rates using α = 0.05 are provided in Table 4. In populations with θ_B > 0, rejection rates were as expected in both M*plus* and `lavaan`. In fact, the rates are mostly identical in conditions where convergence rates were 100% for both `lavaan` and M*plus*, reinforcing the expectation the two software packages provide the same results when fitting the same model to the same data, using the same estimation routine and calculating the normal-theory χ^2 test statistic. In the other conditions, differences in convergence rates cause small differences between the results obtained with `lavaan` and M*plus* with ML.

Fixing θ_B = 0 yielded 100% power to reject the model, and freely estimating θ_B yielded rejection rates that did not appreciably differ from the nominal 5% and were closer in larger samples. In populations with θ_B = 0, however, rejections rates were nearly 0% across conditions and software (except when they could not be calculated in conditions where M*plus* did not converge).

3.3. Follow-Up Study Comparing ML Algorithms

Table 5 includes the same convergence and rejection rates for the 100-cluster conditions reported in Tables 3 and 4 (i.e., using the QN algorithm), as well as the EM(A) algorithms in M*plus* . Convergence rates of EMA and QN were very similar in populations with $\theta_B >$ 0. When $\theta_B = 0$ in the population, EMA converged in all samples, including the conditions that fitted $\theta_B > 0$ (for which QN failed in all samples). Convergence rates for EM were consistently zero for fitted models with $\theta_B = 0$, regardless of the population model. This implies some counter-intuitive results with EM. For example, fitting the correct configural model with $\theta_B = 0$ leads to 0% convergence, while fitting the overparametarized configural model with $\theta_B > 0$ (while $\theta_B = 0$ in the population) converged in 90.4% of the replications. In addition, EM convergence rates were particularly low in conditions where the shared model with $\theta_B > 0$ was fitted to data generated with $\theta_B = 0$. For the converged cases, there were no notable differences in rejection rates across the different algorithms.

Table 4. Rejection rates of the χ^2 test at $\alpha = 0.05$ with ML estimation using QN algorithm.

				50 Clusters		**100 Clusters**		**200 Clusters**	
Data Model		**Fitted Model**		`lavaan`	**M*plus***	`lavaan`	**M*plus***	`lavaan`	**M*plus***
Config	$\theta_B > 0$	Uncon	$\theta_B > 0$	0.072	0.072	0.062	0.062	0.050	0.050
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	*$\theta_B > 0$*	0.072	0.072	0.057	0.057	0.058	0.058
		Config	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Shared	$\theta_B > 0$	0.041	0.034	0.037	0.026	0.056	0.046
		Shared	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
Config	$\theta_B = 0$	Uncon	$\theta_B > 0$	0.003	-	0.006	-	0.006	-
		Uncon	$\theta_B = 0$	0.001	0.001	0.001	0.001	0.006	0.004
		Config	$\theta_B > 0$	0.007	-	0.005	-	0.01	-
		Config	*$\theta_B = 0$*	0.003	0.003	0.003	0.003	0.009	0.009
		Shared	$\theta_B > 0$	0.002	-	0.003	-	0.008	-
		Shared	$\theta_B = 0$	0	0	0	0	0.002	0.005
Shared	$\theta_B > 0$	Uncon	$\theta_B > 0$	0.071	0.071	0.044	0.044	0.044	0.044
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Config	$\theta_B > 0$	0.068	0.068	0.049	0.049	0.050	0.050
		Config	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
		Shared	*$\theta_B > 0$*	0.024	0.016	0.038	0.026	0.058	0.051
		Shared	$\theta_B = 0$	1.000	1.000	1.000	1.000	1.000	1.000
Shared	$\theta_B = 0$	Uncon	$\theta_B > 0$	0.002	-	0.004	-	0.003	-
		Uncon	$\theta_B = 0$	0.003	0.003	0.003	0.003	0.002	0.002
		Config	$\theta_B > 0$	0.004	-	0.006	-	0.005	-
		Config	$\theta_B = 0$	0.005	0.004	0.004	0.004	0.001	0.001
		Shared	$\theta_B > 0$	0	-	0.003	-	0.004	-
		Shared	*$\theta_B = 0$*	0.001	0	0	0	0.001	0

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. *Italicized* = conditions in which the true model was fitted.

Nonconvergence Anomaly with M*plus*

Different M*plus* estimation algorithms had similar rejection rates but different convergence rates in Table 5. In order to further investigate this finding, we focus on one condition from Table 5, where the population model is a shared model with $\theta_B = 0$, but θ_B is freely estimated in the fitted model. In this condition, EM failed to converge in 98.1% of the replications, QN failed to converge in 100% of the replications, yet EMA converged for each sample. We inspected the TECH8 output, which prints the optimization history for each replication, and found that all 998 converged replications using EMA contained the message: "The optimization algorithm has changed to the em algorithm". As noted in Section 1.2.1, this occurs when a QN step (used to accelerate convergence with EM) fails to improve the log-likelihood.

We investigated further by fitting the model to the first generated dataset from this condition (i.e., as a single analysis, not using the MONTECARLO feature). The analysis yielded an apparently converged solution, with a scaled $\chi^2(9) = 2.817$, $p = 0.971$. But indeed the optimization history showed that the default EMA algorithm failed after 111 iterations, at which point M*plus* switched to EM and appeared to converge after 243 iterations. However, when we explicitly selected ALGORITHM = EM in the M*plus* input file, we saw that the MCONVERGENCE criterion of the EM algorithm was not fulfilled, despite having apparently converged when EM was used following the failure of EMA to converge. Close inspection of each interation in the optimization history for this data set reveals that Iteration 243 of explicitly requested EM had the same log-likelihood as the final (apparently converging) Iteration 243 of EM following failure of EMA; however, when explicitly requested, the EM continued iterating until the maximum number of m-iterations (500) was reached and eventually failed to converge. When we ran the MONTECARLO analysis on all 1000 datasets in this condition with ALGORITHM = EM, the model converged on a solution in none of the data sets.

Table 5. Convergence and rejection rates with 100 clusters across ML estimation algorithms.

				Convergence Rates				**Rejection Rates**			
		Algorithm:		**QN**	**QN**	**EMA**	**EM**	**QN**	**QN**	**EMA**	**EM**
		Software:		`lavaan`	**M*plus***	**M*plus***	**M*plus***	`lavaan`	**M*plus***	**M*plus***	**M*plus***
Data Model		**Fitted Model**									
Config	$\theta_B > 0$	Uncon	$\theta_B > 0$	1.000	1.000	1.000	1.000	0.062	0.062	0.062	0.062
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	0	1.000	1.000	1.000	-
		Config	$\mathit{\theta_B > 0}$	1.000	1.000	1.000	1.000	0.057	0.057	0.057	0.057
		Config	$\theta_B = 0$	1.000	1.000	1.000	0	1.000	1.000	1.000	-
		Shared	$\theta_B > 0$	0.998	0.793	0.741	0.730	0.037	0.026	0.022	0.025
		Shared	$\theta_B = 0$	1.000	1.000	1.000	0	1.000	1.000	1.000	-
Config	$\theta_B = 0$	Uncon	$\theta_B > 0$	1.000	0	1.000	0.897	0.006	-	0.009	0.008
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	0	0.001	0.001	0.001	-
		Config	$\theta_B > 0$	1.000	0.001	1.000	0.904	0.005	-	0.009	0.009
		Config	$\mathit{\theta_B = 0}$	1.000	1.000	1.000	0	0.003	0.003	0.003	-
		Shared	$\theta_B > 0$	0.970	0	1.000	0.028	0.003	-	0.007	-
		Shared	$\theta_B = 0$	0.863	0.227	1.000	0	0	0	0	-
Shared	$\theta_B > 0$	Uncon	$\theta_B > 0$	1.000	1.000	1.000	1.000	0.044	0.044	0.044	0.044
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	0	1.000	1.000	1.000	-
		Config	$\theta_B > 0$	1.000	1.000	1.000	1.000	0.049	0.049	0.049	0.049
		Config	$\theta_B = 0$	1.000	1.000	1.000	0	1.000	1.000	1.000	-
		Shared	$\mathit{\theta_B > 0}$	0.991	0.772	0.667	0.659	0.038	0.026	0.022	0.023
		Shared	$\theta_B = 0$	1.000	1.000	1.000	0	1.000	1.000	1.000	-
Shared	$\theta_B = 0$	Uncon	$\theta_B > 0$	1.000	0	1.000	0.907	0.004	-	0.012	0.009
		Uncon	$\theta_B = 0$	1.000	1.000	1.000	0	0.003	0.003	0.003	-
		Config	$\theta_B > 0$	1.000	0	1.000	0.909	0.006	-	0.012	0.009
		Config	$\theta_B = 0$	1.000	1.000	1.000	0	0.004	0.004	0.004	-
		Shared	$\theta_B > 0$	0.970	0	0.998	0.019	0.003	-	0.010	-
		Shared	$\mathit{\theta_B = 0}$	0.868	0.215	1.000	0	0	0	0.003	-

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. QN = quasi-Newton algorithm. EM = expectation–maximization algorithm. EMA = accelerated EM algorithm. ML(R) = (robust) maximum likelihood. *Italicized* = conditions in which the true model was fitted.

After requesting help from the M*plus* support team about this anomaly, we learned that the employed convergence criteria for EM after failure of EMA differ from the EM defaults. Specifically, for EM after EMA, M*plus* does not check the first order condition (i.e., whether the gradient consists of zero's). As a result, any model seems to converge, including those that do not with EM (or ODLL), after which the gradient is checked. Effectively, this means that for the results obtained with EM-after-EMA, it is not verified that the obtained parameter estimates are actual ML estimates. It is important to note that this issue is not detectable from the M*plus* output file. First, there is no indication of the change in convergence criteria. The M*plus* output will list a minimum derivative value of 0.0001 as one of the convergence criteria, while in reality this criterium is ignored. Users may verify this by setting the mconvergence criterium to a large number (like 1000) in an explicit EM analysis. This will lead to the same results as obtained with an EMA analysis in which M*plus* switched to EM. Second, M*plus* does not provide any warning about the switch from EMA to EM. Table A2 (first column) shows the number of replications across conditions for which the algoritm changed from EMA to EM.

3.4. Follow-Up Study Comparing ML Algorithms with Robust Corrections

The final follow-up study was conducted to reveal under what conditions Stapleton and Johnson's [2] inflated rejection rates could be replicated, given that they used the default estimation options (MLR with EMA) in M*plus* . Because MLR simply begins with ML estimation, the convergence rates were the same for MLR as reported for ML in Table 5. Robust corrections are calculated after convergence of the algorithm on ML estimates. Table 6 reports rejection rates of the scaled χ^2 statistic in the 100-cluster conditions using the QN and EMA algorithms in M*plus* and QN in `lavaan`. The results obtained with MLR show that the rejections rates of the default test statistic in M*plus* can be seriously inflated under certain conditions that we examined in this simulation study—namely, in populations for which there is strong invariance across clusters ($\theta_B = 0$). The same inflation is not apparent when using MLR (with QN) in `lavaan`. However, the same inflation is apparent when using QN in M*plus*, except in conditions where models did not converge, in which case rejection rates cannot be calculated. Because interpreting these differential results involves quite some detail across software packages, we focus first on how our results compare to the results presented by Stapleton and Johnson [2], and then discuss the differences between software packages.

3.4.1. Comparison with Stapleton and Johnson (2019)

The last column of Table 6 shows the rejection rates reported by Stapleton and Johnson [2]. Their inflated rejection rates with 100 clusters were replicated with M*plus* using EMA (the default, used in their simulation), but only in the conditions where population $\theta_B = 0$ in the configural model. That is, when the correct model was fitted to data generated under the configural model with no residual variance, rejection rates were 11%. Also, when fitting the overparameterized unconstrained model to the same data, rejection rates were as highly inflated as Stapleton and Johnson reported, but only when the model was additionally overparameterized by freely estimating θ_B. When appropriately fixing $\theta_B = 0$, the rejection rates were not nearly as inflated (i.e., 13% rather than 51%). Furthermore, the same pattern of results just described (for fitting the unconstrained model to configural-model data) can be seen not only when fitting the configural or shared models to the same data, but also when fitting any model to data from populations with a shared construct (but again, only when population $\theta_B = 0$).

The pattern of M*plus* EMA results in Table 6 challenges the conclusions offered by Stapleton and Johnson [2]. First, the highly inflated rejection rates of the unconstrained model (fitted to configural-model data) should not have been attributed to overparameterization. Stapleton and Johnson [2] (p. 319) contended that unnecessarily estimating additional parameters "used unnecessary degrees of freedom, and the χ^2 test criterion was lowered," perhaps forgetting that the χ^2 test statistic itself would also lower in this case (on average, as much as the df decrease). Table 6 shows that the unconstrained model's rejection rates were not substantially larger than $\alpha = 5\%$ when appropriately estimating $\theta_B > 0$, matching Stapleton and Johnson's results when fitting an unconstrained model to data from a population with a shared construct. Note that this is also consistent with our prediction that even the configural model (which is similar to but more constrained than the unconstrained model) should have rejection rates near the α level. When fitting the unconstrained model to data from a population without a shared construct, Stapleton and Johnson's high rejection rates were instead due to unnecessarily fixing $\theta_B = 0$.

Finally, Stapleton and Johnson [2] observed 100% power to reject a configural model when the population includes a shared factor, which we predicted should yield rejection rates close to the α level. However, Table 6 shows that result is only consistent with fixing $\theta_B = 0$ when the population $\theta_B > 0$. Appropriately estimating θ_B yielded 5% rejection rates, supporting our claim that the shared and configural factors would be confounded due to proportionally equivalent loadings across indicators. All in all, Stapleton and Johnson's results are not generalizable because they did not hold constant whether $\theta_B = 0$ either in the population or in the fitted model. Note that all of these inflated error rates occurred only when using the M*plus* default setting (i.e., a scaled test statistic) because the same patterns were not found in Table 5. We will elaborate more on the inflated error rates found with the scaled test statistic in the discussion.

Table 6. Rejection rates of scaled χ^2 statistic (MLR) with 100 clusters across estimation algorithms.

				Algorithm			
				QN	**QN**	**EMA**	
		Software		`lavaan`	**M*plus***	**M*plus***	**S&J**
Data Model		**Fitted Model**					
Config	$\theta_B > 0$	Uncon	$\theta_B > 0$	0.070	0.070	0.070	
		Uncon	$\theta_B = 0$	NA	1.000	1.000	
		Config	$\mathit{\theta_B > 0}$	0.065	0.065	0.065	
		Config	$\theta_B = 0$	NA	1.000	1.000	
		Shared	$\theta_B > 0$	0.041	0.049	0.049	
		Shared	$\theta_B = 0$	NA	1.000	1.000	
Config	$\theta_B = 0$	Uncon	$\theta_B > 0$	0.006	-	0.512	0.54
		Uncon	$\theta_B = 0$	0.004	0.131	0.131	
		Config	$\theta_B > 0$	0.006	-	0.365	
		Config	$\mathit{\theta_B = 0}$	0.004	0.111	0.111	0.11
		Shared	$\theta_B > 0$	0.014	-	0.445	-
		Shared	$\theta_B = 0$	0.011	0.132	0.190	
Shared	$\theta_B > 0$	Uncon	$\theta_B > 0$	0.052	0.052	0.052	0.09
		Uncon	$\theta_B = 0$	NA	1.000	1.000	
		Config	$\theta_B > 0$	0.058	0.058	0.058	
		Config	$\theta_B = 0$	NA	1.000	1.000	1.000
		Shared	$\mathit{\theta_B > 0}$	0.043	0.040	0.037	0.09
		Shared	$\theta_B = 0$	NA	1.000	1.000	
Shared	$\theta_B = 0$	Uncon	$\theta_B > 0$	0.005	-	0.515	
		Uncon	$\theta_B = 0$	0.007	0.122	0.122	
		Config	$\theta_B > 0$	0.006	-	0.357	
		Config	$\theta_B = 0$	0.008	0.104	0.104	
		Shared	$\theta_B > 0$	0.006	-	0.430	
		Shared	$\mathit{\theta_B = 0}$	0.018	0.126	0.170	

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. QN = quasi-Newton algorithm. EM = expectation–maximization algorithm. EMA = accelerated EM algorithm. MLR = robust maximum likelihood. S&J = Rejection rates reported by Stapleton & Johnson [1]. *Italicized* = conditions in which the true model was fitted. NA = Test statistic not available.

3.4.2. Comparison of MLR Results with `lavaan` and M*plus*

Looking at the results of `lavaan` with MLR in Table 6, six conditions stand out where models were seriously misspecified because population $\theta_B > 0$ but fitted $\theta_B = 0$. In these conditions, `lavaan` did not provide a test statistic. Closer inspection of these results indicated that the scaled test statistis was actually not defined due to a negative trace involved in calculating the scaling factor (i.e., the trace of $\mathbf{U\Gamma}$ [23]). Users can obtain this quantity from `lavaan` using the function lavInspect(), with the second argument as

"UGamma", as well as the **U** or **Γ** matrix separately using "UfromUGamma" or "gamma". Naturally, the same issue occurs for M*plus* . However, instead of providing a warning, M*plus* reports the unscaled test statistic and indicates 'Undefined' for the scaling correction factor. The MONTECARLO output of M*plus* does not contain any information pointing to the scaled test statistic being undefined, so we only discovered this was happening because of our comparison of results reported by `lavaan`. Given that the models are severely misspecified in these conditions, the uncorrected test statistics are very high and will not lead to wrong conclusions in practice. The second column in Table A1 indicates the number of samples for which `lavaan` indicated that the test statistic was not available per condition.

In the conditions where $\theta_B > 0$ and in the fitted model $\theta_B > 0$, the results obtained with `lavaan` (with QN) and M*plus* were identical. For population conditions with $\theta_B = 0$ and fitted $\theta_B = 0$, the rejection rates obtained with M*plus* with QN are somewhat larger than the rejection rates obtained with `lavaan`. For example, the rejection rate was 0.004 in `lavaan` and 0.111 in M*plus* when fitting the correct configural model with $\theta_B = 0$. Since the rejection rates did not differ across `lavaan` and M*plus* with QN or EMA for the uncorrected test statistic (reported in Table 5), the difference in results must be rooted in how the scaled test statistics are calculated. In this condition there were 3 samples for which the scaling correction factor was not defined, implying that the results for M*plus* are partly based on unscaled test statistics. This may explain a small part of the difference across packages. Another source of differences may be found in how the packages proceed when the (augmented) observed information matrix is near-singular. Near-singularity of the observed information matrix often happens, and usually this it not a reason for concern. Both `lavaan` and M*plus* do not print out a warning when (equality or inequality) constraints are part of the model, and both programs probably use a different approach to handle these near-singular cases. While `lavaan` uses a generalized inverse, the solution of M*plus* is less clear. In some cases M*plus* gives the warning: "An adjustment to the estimatioon of the information matrix has been made". The last column in Table A3 shows the number of replications per condition for which M*plus* provided this warning.

In the conditions where population $\theta_B = 0$ and in the fitted model $\theta_B > 0$, `lavaan` showed low rejection rates as expected. The 100% nonconvergence of M*plus* with QN prevents comparison of results across software packages using the same QN algorithm. M*plus* with EMA however resulted in severely inflated Type 1 error rates, ranging from 0.365 to 0.515. In Table A3 one can see that the rejection rates are also inflated in conditions where θ_B is freely estimated while population θ_B is not exactly zero, but 0.0001 or 0.01, although the inflation is less severe (but still around 12%) in the conditions with $\theta_B = 0.01$.

4. Discussion

4.1. Summary of the Results

For all conditions in the primary study (comparing ML with QN across packages), `lavaan` either converged more often than M*plus* or both packages converged in 100% of samples. M*plus* never converged in conditions with population $\theta_B = 0$ but fitted $\theta_B > 0$. Rejection rates of the normal-theory χ^2 test statistic were identical across packages. Our comparison of ML algorithms in M*plus* showed that using EM did not converge in any condition for which fitted $\theta_B = 0$. With the default M*plus* settings for two-level models (MLR + EMA), M*plus* often switches to the EM algorithm. When this switch is made, M*plus* ignores one of the main convergence criteria (i.e., whether the algorithm in fact converged on a ML estimate, as revealed by the first derivative), meaning that the obtained results may be based on a non-converged solution. Users are not notified that convergence criteria are ignored, nor are they notified of this switch being made (unless they specifically request and pay attention to the TECH8 output, which seems unlikely to be common practice). In our second follow up study, we found seriously inflated scaled test statistics in M*plus* in populations with $\theta_B = 0$.

Based on the comparison of our results with those of Stapleton and Johnson [2], we showed that their advice is not generalizable because they did not independently vary whether $\theta_B = 0$ either in the population or in the fitted model. In all conditions in which they reported high rejection rates, this was the result of incorrectly fixing $\theta_B = 0$, or of inflated scaled test statistics obtained by using the default settings in M*plus*.

4.2. Recommendation for Practice

Our investigation reveals that the defaults in the most popular multilevel SEM software (M*plus*) carry specific risks that are easily checked but not well advertised. When using M*plus* with the default settings for two-level models, it is strongly recommended to check M*plus*' TECH8 output to verify convergence of the solution. Specifically, when the algorithm switched from EMA to EM, we recommend to verify whether the derivative criterion in the TECH8 output is sufficiently close to zero; one can also run the model again, explicitly requesting ALGORITHM = EM. If running the model with EM and the default convergence criteria does not lead to a converged solution, then one should be suspicious of the output obtained using EMA that switches to EM. Using the default settings in `lavaan` does not carry the same risks, given that the convergence rates were good, rejection rates appropriate, and one can be sure that there are no hidden changes of convergence checks or test statistics. Therefore, when M*plus* users find evidence that their apparently converged model might not have actually converged on a maximum, we recommend fitting their model (when possible) with `lavaan`, whose defaults do not cause the same rates of convergence problems, nor are `lavaan`'s test statistics (and Type I error rates) inflated in the very conditions when M*plus* results are dubious (Table 6; see also the Appendix A).

Researchers should be aware that the scaled χ^2 statistic can be seriously inflated. This is in line with earlier findings [24,25]. Moreover, there exist many computational options for (scaled) test statistics, and more research is needed to evaluate which of those work best in which conditions. Also, the default implementations differ across packages. Savalei and Rosseel [23] provide an overview of computational variations and how to apply them using `lavaan`.

4.3. Future Research

In this study we only focused on the statistical performance (convergence rates and rejection rates of standard and scaled test statistics) of the models proposed and evaluated in Stapleton and Johnson [2]. From a theoretical perspective, we also have some concerns with respect to the simultaneous shared-and-configural model. The goal of the proposed model is to disentangle the 'objective' shared construct from the shared part of what Stapleton and Johnson described as a nuisance factor. We find it hard to imagine a realistic situation in which one would want to measure an objective shared construct using individual responses. If an objective property of a cluster (for example a neighborhood or a school class) should be operationalized for a study, we think one should look at objective variables at the cluster level. For example, the crime rate per neighborhood, the size of the school class, or the gender of the teacher—Stapleton and Johnson similarly proposed measuring strictly Level-2 indicators in addition to the Level-1 indicators, but to be used as additional indicators of the shared factor (p. 325). From our perspective, as soon as one asks individuals in the cluster to rate a cluster-level object, one will most probably be measuring subjective perceptions of the construct (for example, perceived neighborhood safety, perceived teaching quality) for which within-cluster differences would be expected. Future research may therefore focus on the theoretical meaning and practical applications of the models evaluated in the current research.

It was known from earlier research that the scaled χ^2 as implemented in M*plus* overrejects models when the sample size is not large enough, especially with large models [25]. In line with earlier findings based on two-level models [14,26], our simulation study shows that over-rejection by this test statistic is exacerbated when between-level residual variances are zero (or small, see Table A2) in the population. Because $\theta_B = 0$ is to be expected when strong factorial invariance across clusters holds (i.e., when there is no cluster bias, even approximately), it is reasonable to assume that $\theta_B \cong 0$, at least for some of the indicators in a substantial part of research settings. In our study, we did not evaluate conditions with partial cluster invariance. Future research may investigate the performance of the scaled χ^2 as implemented in M*plus* when population residual variances are zero for some indicators but nonzero for others.

Several Bayesian approaches have been proposed and evaluated for estimating two-level factor models [21,22,27,28]. Bayesian methods with (weakly) informative priors may be able to avoid some of the estimation problems that occur with ML estimation, specifically when sample sizes are relatively small [28]. Alternatively, factor score regression methods [29,30] could be a solution to avoid certain estimation problems For future research, it would be interesting to evaluate the performance of these alternative methods under the conditions of our simulation study.

Author Contributions: Conceptualization, S.J. and T.D.J.; methodology, S.J., T.D.J. and Y.R.; simulation study, S.J.; interpretation of results, S.J., T.D.J., and Y.R.; data curation, S.J.; writing—original draft preparation, S.J. and T.D.J.; writing—review and editing, Y.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Dutch Research Council, grant numbers NWO-VENI-451-16-001 and 016.Veni.195.457.

Data Availability Statement: R syntax to generate and analyze the Monte Carlo results, as well as the M*plus* output files, are available from the Open Science Framework: https://osf.io/sdwam/. Accessed on 31 May 2021.

Acknowledgments: We thank Laura Stapleton and Tessa Johnson for providing additional information about their simulation study. We also thank the M*plus* support team for their clarification about the employed convergence criteria when using EM after EMA.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Convergence rates with 100 clusters for ML estimation with QN in M*plus* with population θ_B being zero, 0.0001, and 0.01.

			Population Value θ_B		
Data Model	**Fitted Model**		**0**	**0.0001**	**0.01**
Config	Uncon	$\theta_B > 0$	0	0.005	0.401
	Uncon	$\theta_B = 0$	1.000	1.000	1.000
	Config	$\theta_B > 0$	0.001	0.004	0.431
	Config	$\theta_B = 0$	1.000	1.000	1.000
	Shared	$\theta_B > 0$	0	0.001	0.245
	Shared	$\theta_B = 0$	0.227	0.237	0.722
Shared	Uncon	$\theta_B > 0$	0	0.005	0.200
	Uncon	$\theta_B = 0$	1.000	1.000	1.000
	Config	$\theta_B > 0$	0	0.002	0.383
	Config	$\theta_B = 0$	1.000	1.000	1.000
	Shared	$\theta_B > 0$	0	0	0.124
	Shared	$\theta_B = 0$	0.215	0.446	0.952

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances.

Table A2. Frequency (in 1000 samples) of M*plus* switch from EMA to EM, undefined scaling corrections and saddle point warnings.

Data Model		Fitted Model		Switch EMAto EM	Scaling Factor Undefined	Saddle Point
Config	$\theta_B > 0$	Uncon	$\theta_B > 0$	0	0	0
		Uncon	$\theta_B = 0$	0	1000	0
		Config	$\theta_B > 0$	0	0	0
		Config	$\theta_B = 0$	0	1000	0
		Shared	$\theta_B > 0$	0	9	11
		Shared	$\theta_B = 0$	0	1000	0
Config	$\theta_B = 0$	Uncon	$\theta_B > 0$	985	4	0
		Uncon	$\theta_B = 0$	0	3	0
		Config	$\theta_B > 0$	981	3	0
		Config	$\theta_B = 0$	0	3	0
		Shared	$\theta_B > 0$	998	11	0
		Shared	$\theta_B = 0$	518	7	0
Shared	$\theta_B > 0$	Uncon	$\theta_B > 0$	0	0	0
		Uncon	$\theta_B = 0$	0	1000	0
		Config	$\theta_B > 0$	0	0	0
		Config	$\theta_B = 0$	0	1000	0
		Shared	$\theta_B > 0$	3	10	13
		Shared	$\theta_B = 0$	0	1000	0
Shared	$\theta_B = 0$	Uncon	$\theta_B > 0$	987	2	0
		Uncon	$\theta_B = 0$	0	2	0
		Config	$\theta_B > 0$	984	2	0
		Config	$\theta_B = 0$	0	1	0
		Shared	$\theta_B > 0$	998	10	407
		Shared	$\theta_B = 0$	518	11	0

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. Saddle point = M*plus* with EMA warns that "The model estimation has reached a saddle point or a point where the observed and the expected information matrices do not match. An adjustment to the estimation of the information matrix has been made".

Table A3. Rejection rates with 100 clusters for MLR estimation with EMA in M*plus* with population θ_B being zero, 0.0001, and 0.01.

			Population Value θ_B		
Data Model	**Fitted Model**		**0**	**0.0001**	**0.01**
Config	Uncon	$\theta_B > 0$	0.512	0.522	0.124
	Uncon	$\theta_B = 0$	0.131	0.127	0.940
	Config	$\theta_B > 0$	0.365	0.338	0.118
	Config	$\theta_B = 0$	0.111	0.108	0.905
	Shared	$\theta_B > 0$	0.445	0.450	0.123
	Shared	$\theta_B = 0$	0.190	0.161	0.719
Shared	Uncon	$\theta_B > 0$	0.515	0.498	0.119
	Uncon	$\theta_B = 0$	0.122	0.135	0.927
	Config	$\theta_B > 0$	0.357	0.353	0.103
	Config	$\theta_B = 0$	0.104	0.112	0.899
	Shared	$\theta_B > 0$	0.430	0.403	0.127
	Shared	$\theta_B = 0$	0.170	0.160	0.713

Note: Uncon = Unconstrained model, Conf = Configural model. Shared = Simultaneous shared-and-configural model. θ_B = between-level residual variances. Reported convergence rates for these conditions were all higher than 0.973, but across θ_B conditions, the frequencies of M*plus* switching to the EM algorithm (and ignoring one of the convergence criteria) are very similar to the frequencies reported in the first column of Table A2, so convergence status is effectively unknown.

References

1. Stapleton, L.M.; Yang, J.S.; Hancock, G.R. Construct Meaning in Multilevel Settings. *J. Educ. Behav. Stat.* **2016**, *41*, 481–520. [CrossRef]
2. Stapleton, L.M.; Johnson, T.L. Models to Examine the Validity of Cluster-Level Factor Structure Using Individual-Level Data. *Adv. Methods Pract. Psychol. Sci.* **2019**, *2*, 312–329. [CrossRef]
3. Muthén, B.O.; Muthén, L.K. *Mplus User's Guide*, 8th ed.; Muthén & Muthén: Los Angeles, CA, USA, 1998.
4. Rosseel, Y. `lavaan`: An R Package for Structural Equation Modeling. *J. Stat. Softw.* **2012**, *48*, 1–36. [CrossRef]
5. van Schaik, S.D.M.; Leseman, P.P.M.; Haan, M. de Using a Group-Centered Approach to Observe Interactions in Early Childhood Education. *Child Dev.* **2018**, *89*, 897–913. [CrossRef] [PubMed]
6. Asparouhov, T.; Muthén, B. General Random Effect Latent Variable Modeling: Random Subjects, Items, Contexts, and Parameters. In Proceedings of the Annual Meeting of the National Council on Measurement in Education, Vancouver, BC, Canada, 13–17 April 2012.
7. Hox, J.J.; Moerbeek, M.; van de Schoot, R. *Multilevel Analysis: Techniques and Applications*, 3rd ed.; Routledge: London, UK, 2017; ISBN 978-1-317-30868-3.
8. Jak, S. Cross-Level Invariance in Multilevel Factor Models. *Struct. Equ. Model. Multidiscip. J.* **2019**, *26*, 607–622. [CrossRef]
9. Kim, E.S.; Dedrick, R.F.; Cao, C.; Ferron, J.M. Multilevel Factor Analysis: Reporting Guidelines and a Review of Reporting Practices. *Multivar. Behav. Res.* **2016**, *51*, 881–898. [CrossRef]
10. Mehta, P.D.; Neale, M.C. People Are Variables Too: Multilevel Structural Equations Modeling. *Psychol. Methods* **2005**, *10*, 259–284. [CrossRef]
11. Rabe-Hesketh, S.; Skrondal, A.; Pickles, A. Generalized Multilevel Structural Equation Modeling. *Psychometrika* **2004**, *69*, 167–190. [CrossRef]
12. Muthén, B.O. Mean and Covariance Structure Analysis of Hierarchical Data. UCLA Statistics Series #62, August 1990. Available online: https://escholarship.org/uc/item/1vp6w4sr (accessed on 31 May 2021).
13. Yuan, K.-H.; Bentler, P.M. 5. Three Likelihood-Based Methods for Mean and Covariance Structure Analysis with Nonnormal Missing Data. *Sociol. Methodol.* **2000**, *30*, 165–200. [CrossRef]
14. Jak, S.; Oort, F.J.; Dolan, C.V. A Test for Cluster Bias: Detecting Violations of Measurement Invariance Across Clusters in Multilevel Data. *Struct. Equ. Model. Multidiscip. J.* **2013**, *20*, 265–282. [CrossRef]
15. Muthén, B.; Asparouhov, T. Recent Methods for the Study of Measurement Invariance With Many Groups: Alignment and Random Effects. *Sociol. Methods Res.* **2018**, *47*, 637–664. [CrossRef]
16. Jak, S.; Jorgensen, T.D. Relating Measurement Invariance, Cross-Level Invariance, and Multilevel Reliability. *Front. Psychol.* **2017**, *8*. [CrossRef]
17. Muthén, B.; Asparouhov, T. Bayesian SEM: A More Flexible Representation of Substantive Theory. *Psychol. Methods* **2012**, 313–335. [CrossRef] [PubMed]
18. Byrne, B.M.; Shavelson, R.J.; Muthén, B. Testing for the Equivalence of Factor Covariance and Mean Structures: The Issue of Partial Measurement Invariance. *Psychol. Bull.* **1989**, *105*, 456–466. [CrossRef]
19. Furlow, C.F.; Beretvas, S.N. Meta-Analytic Methods of Pooling Correlation Matrices for Structural Equation Modeling Under Different Patterns of Missing Data. *Psychol. Methods* **2005**, *10*, 227–254. [CrossRef] [PubMed]
20. Lüdtke, O.; Marsh, H.W.; Robitzsch, A.; Trautwein, U. A 2 × 2 Taxonomy of Multilevel Latent Contextual Models: Accuracy–Bias Trade-Offs in Full and Partial Error Correction Models. *Psychol. Methods* **2011**, *16*, 444–467. [CrossRef] [PubMed]
21. Zitzmann, S.; Lüdtke, O.; Robitzsch, A.; Marsh, H.W. A Bayesian Approach for Estimating Multilevel Latent Contextual Models. *Struct. Equ. Model. Multidiscip. J.* **2016**, *23*, 661–679. [CrossRef]
22. Depaoli, S.; Clifton, J.P. A Bayesian Approach to Multilevel Structural Equation Modeling With Continuous and Dichotomous Outcomes. *Struct. Equ. Model. Multidiscip. J.* **2015**, *22*, 327–351. [CrossRef]
23. Savalei, V.; Rosseel, Y. Computational Options for Standard Errors and Test Statistics with Incomplete Normal and Nonnormal Data in SEM. **2021**. [CrossRef]
24. Savalei, V. Expected versus Observed Information in SEM with Incomplete Normal and Nonnormal Data. *Psychol. Methods* **2010**, *15*, 352–367. [CrossRef]
25. Maydeu-Olivares, A. Maximum Likelihood Estimation of Structural Equation Models for Continuous Data: Standard Errors and Goodness of Fit. *Struct. Equ. Model. Multidiscip. J.* **2017**, *24*, 383–394. [CrossRef]
26. Jak, S.; Oort, F.J.; Dolan, C.V. Using Two-Level Factor Analysis to Test for Cluster Bias in Ordinal Data. *Multivar. Behav. Res.* **2014**, *49*, 544–553. [CrossRef] [PubMed]
27. Holtmann, J.; Koch, T.; Lochner, K.; Eid, M. A Comparison of ML, WLSMV, and Bayesian Methods for Multilevel Structural Equation Models in Small Samples: A Simulation Study. *Multivar. Behav. Res.* **2016**, *51*, 661–680. [CrossRef] [PubMed]
28. Lüdtke, O.; Robitzsch, A.; Wagner, J. More Stable Estimation of the STARTS Model: A Bayesian Approach Using Markov Chain Monte Carlo Techniques. *Psychol. Methods* **2018**, *23*, 570–593. [CrossRef]

29. Devlieger, I.; Rosseel, Y. Multilevel Factor Score Regression. *Multivar. Behav. Res.* **2020**, *55*, 600–624. [CrossRef] [PubMed]
30. Zitzmann, S.; Helm, C. Multilevel Analysis of Mediation, Moderation, and Nonlinear Effects in Small Samples, Using Expected a Posteriori Estimates of Factor Scores. *Struct. Equ. Model. Multidiscip. J.* **2021**, 1–18. [CrossRef]