




Article

Item Response Analysis of a Structured Mixture Item Response Model with *mirt* Package in R

Minho Lee ^{1,†} , Yon Soo Suh ^{2,†}  and Minjeong Jeon ^{1,*} 

¹ Department of Education, University of California, 457 Portola Avenue, Los Angeles, CA 90024, USA; leemino72@ucla.edu

² NWEA, 121 NW Everett Street, Portland, OR 97209, USA; yon.soo.suh@nwea.org

* Correspondence: mjjeon@ucla.edu

† These authors contributed equally to this work.

Abstract: Structured mixture item response models (StrMixIRMs) are a special type of constrained confirmatory mixture item response theory (IRT) model for detecting latent performance differences in a measurement instrument by characteristic item groups, and classifying respondents according to these differences. In light of limited software options for estimating StrMixIRMs under existing frameworks, this paper proposes reparameterizing it as a confirmatory mixture IRT model using interaction effects between latent classes and item groups. The reparameterization allows for easier implementation of StrMixIRMs with multiple software programs that have mixture modeling capabilities, including open-source ones. This widens the accessibility to these models to a broad range of users and thus can facilitate research and applications of StrMixIRMs. This paper serves two main goals: First, we introduce StrMixIRMs, focusing on the proposed reparameterization based on interaction effects and its various extensions. Second, we illustrate use cases of this novel reparameterization within the *mirt* 1.41 package in R by employing two empirical datasets. Detailed R code with notes are provided for the applications along with an interpretation of the outputs.

Keywords: structured mixture item response model; mixture item response model; item response data; latent classes; differentiation parameters; item groups; confirmatory method; interaction effects; *mirt* package



Citation: Lee, M.; Suh, Y.; Jeon, M. Item Response Analysis of a Structured Mixture Item Response Model with *mirt* Package in R. *Psych* 2024, 6, 377–400. <https://doi.org/10.3390/psych6010023>

Academic Editor: Shenghai Dai

Received: 11 January 2024

Revised: 23 February 2024

Accepted: 4 March 2024

Published: 8 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mixture item response theory (IRT) models are among a family of models that combine latent class analysis (LCA) with IRT [1–3] in order to account for systematic heterogeneity in item response behaviors to measurement instruments. They assume that a population being measured is comprised of discrete latent subpopulations or latent classes, with each of them responding to an instrument in a qualitatively different way. This is realized by allowing each latent class of respondents to have different item parameters as well as ability distributions. That is, an IRT model is assumed to hold for each latent class, but the specific IRT model varies across the classes. Mixture IRT models have both the advantages of LCA and IRT, where individuals are not only classified into different latent groups with their respective item parameters, but can also be located on a continuous latent ability scale within a class. Thus, they can serve the dual purpose of diagnosis and comparison across individuals [4–8].

Mixture IRT models have been utilized in psychological and educational settings, predominantly in an exploratory fashion with no a priori assumptions about the source or type of discrepancies in individuals' responses. The purpose is the detection of latent classes, typically by starting with a single-class solution to which more classes are sequentially added until the best-fitting model is determined using various fit indices [6,9]. Such exploratory models only provide the most suitable (i.e., most different) set of latent subpopulations, and thus, require the latent classes to be characterized after their extraction

(e.g., [7]). Furthermore, in the process, exploratory mixture IRT models allow all item parameters to differ across classes, leading to potential difficulties in model estimation and interpretation. To address this, techniques such as regularization have been proposed in order to induce some structure to the models and alleviate the burden (e.g., [10,11]).

1.1. Confirmatory Mixture Item Response Models

Contrarily, when the number and nature of latent classes can be pre-specified according to theory or hypotheses (e.g., [5,12–14]), mixture IRTs can be used in a confirmatory manner to verify their existence and characteristics. Often, these confirmatory models are accompanied by additional constraints befitting the research question(s) at hand prior to data analysis. Although the confirmatory approach is less common in mixture IRT modeling, we point out that confirmatory applications are actually very commonplace, as seen in confirmatory factor analysis and diagnostic classification models (DCMs) [15]. The confirmatory nature coupled with added constraints makes these mixture IRT models more interpretable than their exploratory counterparts that are solely data dependent [16].

In this paper, we focus on a specific type of constrained confirmatory mixture IRT models [17–20], originating in the Saltus model by Wilson [19]. This model was initially developed to detect discontinuities in performance depending on Piagetian developmental stages [19,21,22]. The Saltus model is confirmatory in that it has a predetermined number of latent classes and utilizes preset item groups designed to differentiate the latent classes in line with theory. The latent classes are identified by performance differences on item groups that are modeled to shift together, and in equal amounts, for each latent class. This is achieved through Saltus or leap parameters, which impose additional constraints on item parameters based on prior information on items and item–class relationships. In this way, the Saltus model can provide nuanced, fine-grained information about respondents that goes beyond their latent trait levels. The theory-driven approach in a Saltus model assumes that the latent subpopulations are structured in some way, which are realized by the use of corresponding item groups that place a specific structure on the model. Additional constraints to item parameters on the basis of this structure across latent classes lead to reduction in the number of parameters to be estimated compared to traditional mixture IRT models. These two features make the Saltus model distinct from other existing confirmatory mixture IRT models.

The theory-based nature and parsimony afforded by the Saltus model allows for testing various hypotheses regarding structural differences between subpopulations of individuals, which need not be limited to cognitive developmental settings [23]. The Saltus model can be utilized to analyze any assessment data where a set of items (or item groups) are constructed to elicit discrepant behaviors that illuminate structural differences between presumed latent classes. The model can be useful, for example, for diagnostic classifications and differential learning progression of individuals in cross-sectional and longitudinal contexts. In addition, since its conception, various extensions have been proposed, including non-Rasch models, polytomous item types, inclusion of person predictors and multiple latent traits of interest [17,18], which have further widened the potential of the Saltus model. However, the term “Saltus model” carries with it the connotation of cognitive growth. To highlight the generalizability of Saltus models and their distinctive features, we instead refer to the models from here on as structured mixture item response models (StrMixIRMs). In addition, we refer to the Saltus parameters as differentiation parameters since the latent classes are differentiated by the parameter values.

1.2. Implementation Challenges

Despite the potential utilities and benefits, StrMixIRMs have been underutilized in applied research, exacerbated by difficulties in model estimation and implementation. Researchers have shown that the StrMixIRM can be estimated without specialized software but with general latent variable modeling programs [17,18,24]. For example, Jeon [17,18] showed how StrMixIRMs and their various extensions can be fit using Mplus [25]. The

main idea was to formulate the StrMixIRM as a confirmatory mixture IRT model with linear constraints where differentiation parameters are defined by using equality constraints on the differences in item intercept parameters across latent classes. This approach opened up a new opportunity to employ StrMixIRMs and their extensions via a widely utilized, general-purpose software program. Nonetheless, it is not without limitations. For example, using equality constraints becomes somewhat cumbersome when a large number of items and latent classes are involved. In addition, not many software programs allow for linear equality constraints on item parameters, limiting its accessibility to a broader user group.

The current study aims to introduce an alternative parameterization of the StrMixIRM that overcomes the software limitations by defining the differentiation parameters as interaction effects in a two-way ANOVA context, where latent classes are the first factor and the item groups are the second factor. We note that class memberships are unknown and thus must be estimated in the model. More details of the proposed parameterization are provided in the subsequent sections. This alternative parameterization is implementable in many software programs that support mixture IRT models. In addition, conceptualizing the StrMixIRM as a logistic linear regression with interaction effects may help readers who are familiar with regression and ANOVA understand the model better. In order to promote the use of StrMixIRMs, we demonstrate how the novel parameterization of the StrMixIRM can be easily estimated in R [26], a freely available, open-source environment for statistical computing and graphics. Specifically, we utilize the R package *mirt* [27] in this paper.

The rest of the paper is organized as follows: We begin with an introduction to StrMixIRMs with focus on the proposed reparameterization based on interaction effects, followed by its demonstration using *mirt* [27] in R. Two empirical datasets are used with the dual purpose of elaborating on the specifics of the StrMixIRM as well as showcasing the different uses of the StrMixIRM. We provide detailed R codes centered around unidimensional StrMixIRMs. Attention is also paid to hypothesis setting and interpretation of results using the empirical datasets. We conclude the paper with a summary and discussion on contributions, limitations and possible extensions. We also provide additional R code (e.g., multidimensional applications) in the Supplemental Materials and the GitHub repository (<https://github.com/ysuh09/StrMixIRM>, accessed on 10 January 2024). The latter also includes *Mplus* implementations following the reparameterization as well as Jeon's [17,18] parameterization using linear equality constraints. They are used to confirm the *mirt* results and verify the equivalence of our reparameterization with Jeon's [17,18].

2. Theoretical Background

In this section, we introduce the StrMixIRM and its parameterization using linear constraints. We then propose the new parameterization using interaction effects and show its equivalence to the existing parameterization, along with potential extensions of the StrMixIRM afforded by this new parameterization. For consistency of illustration, we focus on the case with three latent classes and three item groups. However, the model can be applied to more or less and/or non-equal number of latent classes and item groups.

2.1. Introduction to Structured Mixture Item Response Models

2.1.1. Mixture Item Response Models

The original StrMixIRM, proposed by Wilson [19], is based on the Rasch model, which we also follow for the sake of simplicity. Let us begin with a mixture Rasch model [3] with H latent classes. A mixture model consists of a measurement model, pertaining to the item response probabilities, and a structural model, consisting of latent classes probabilities. The measurement model for a mixture Rasch model is given by

$$\text{logit}(\mathcal{P}(Y_{ij} = 1|\theta_{ih})) = \theta_{ih} - \beta_{jh}, \quad (1)$$

where Y_{ij} is a dichotomous response of respondent i ($i = 1, \dots, N$) to item j ($j = 1, \dots, J$). $\mathcal{P}(Y_{ij} = 1|\theta_{ih})$ is the conditional probability of correct response by an individual i with ability θ_{ih} to item j . θ_{ih} represents the latent trait of respondent i nested within latent class h

($h = 1, \dots, H$) with $\theta_{ih} \sim N(\mu_h, \sigma_h^2)$ and β_{jh} is the item i 's difficulty in latent class h . For model identification, all μ_h s are fixed to 0. When $H = 3$, the linear predictors for classes (dictating item response probabilities) are $\theta_{i,h=1} - \beta_{j,h=1}$ for Class 1 ($h = 1$), $\theta_{i,h=2} - \beta_{j,h=2}$ for Class 2 ($h = 2$), and $\theta_{i,h=3} - \beta_{j,h=3}$ for Class 3 ($h = 3$), respectively.

Equation (1) is a conditional model given a respondent's latent class. Since respondent i 's latent class membership is unknown, the marginal probability of respondent i 's item responses is given by

$$\mathcal{P}(\mathbf{y}_i) = \sum_{h=1}^H \pi_h \left[\int_{\theta_{ih}} \prod_{j=1}^J \mathcal{P}(Y_{ij} = 1 | \theta_{ih})^{y_{ij}} (1 - \mathcal{P}(Y_{ij} = 1 | \theta_{ih}))^{1-y_{ij}} N(\mu_h, \sigma_h^2) d\theta_{ih} \right], \quad (2)$$

where π_h is the latent class probability, y_{ij} is the realization of an item response, and $\mathbf{y}_i = (y_{i,j=1}, \dots, y_{i,j=J})'$ is the item response vector of respondent i .

2.1.2. Structured Mixture Item Response Models

Compared to the mixture IRT model above, the StrMixIRM actively utilizes pre-specified K item groups that are expected to differentiate respondents into H latent classes. The item groups can be knowledge domains, cognitive skills, or item design factors for which unobserved sub-populations of individuals are hypothesized to show differential performance beyond their general proficiency on a test.

We assume item j belongs to one of K item groups. The StrMixIRM can then be formulated as follows:

$$\text{logit}(\mathcal{P}(Y_{ijk} = 1 | \theta_{ih})) = \theta_{ih} - \beta_{jk} + \tau_{hk}, \quad (3)$$

where Y_{ijk} is an item response of respondent i to item j ($j = 1, \dots, J_k$) within item group k ($k = 1, \dots, K$), and β_{jk} is the item location or difficulty of item j within item group k . As before, θ_{ih} represents a latent trait of respondent i in latent class h . However, this time, only the first latent class' mean (i.e., the reference class) is fixed at 0 (i.e., $\mu_{h=1} = 0$), while the other μ_h s are freely estimated.

Here τ_{hk} is the differentiation parameter, which is the key parameter of the StrMixIRM. The differentiation parameter represents the differential performance of latent class h (i.e., a focal latent class) from the reference class on item group k given latent trait θ_{ih} . For model identification, τ_{hk} s for the first latent class ($h = 1$) and the first item group ($k = 1$) are fixed at 0: $\tau_{h=1,k} = \tau_{h,k=1} = 0$. In the case of $H = 3$ and $K = 3$, we have a total of four differentiation parameters that are estimated. The linear predictors in Equation (3) can be written for each latent class and item group as in Table 1. Based on the resulting latent trait distributions and the differentiation parameters, researchers can test their hypotheses about the subpopulations of interest. For example, if $\tau_{h=2,k=2}$ in Table 1 is positive, it indicates Class 2 has additional advantages in endorsing items in the second item group compared to Class 1 beyond the general proficiency θ_{ih} . Specific use case scenarios and detailed interpretations are provided in Section 3.

Table 1. Linear predictors in the StrMixIRM for $H = 3$ and $K = 3$.

	Item Group 1 ($k = 1$)	Item Group 2 ($k = 2$)	Item Group 3 ($k = 3$)
Class 1 ($h = 1$)	$\theta_{i,h=1} - \beta_{j,k=1}$	$\theta_{i,h=1} - \beta_{j,k=2}$	$\theta_{i,h=1} - \beta_{j,k=3}$
Class 2 ($h = 2$)	$\theta_{i,h=2} - \beta_{j,k=1}$	$\theta_{i,h=2} - \beta_{j,k=2} + \tau_{h=2,k=2}$	$\theta_{i,h=2} - \beta_{j,k=3} + \tau_{h=2,k=3}$
Class 3 ($h = 3$)	$\theta_{i,h=3} - \beta_{j,k=1}$	$\theta_{i,h=3} - \beta_{j,k=2} + \tau_{h=3,k=2}$	$\theta_{i,h=3} - \beta_{j,k=3} + \tau_{h=3,k=3}$

Note. All τ_{hk} s relating the first row and column are fixed to 0 for model identification.

Jeon [17] described τ_{hk} as the difference in the item group k 's item difficulty between a focal latent class and the reference latent class. Here item difficulties are parameterized using linear constraints, such that $\tau_{hk} = \beta_{jk(h)} - \beta_{jk(h=1)}$ where $\beta_{jk(h)}$ is the item j 's difficulty for latent class h and $h = 1$ is the reference class. The differences are constrained to be

equal across all j items in item group k , in order to estimate the differentiation parameters. Jeon [17] demonstrated how this parameterization can be implemented in `Mplus`.

This approach, however, has the limitations mentioned previously. For example, in `mirt`, one needs to specify custom codes within the package to enable constrained optimization with equality constraints to define the differentiation parameters as constant differences in item difficulties between item groups across latent classes. Such a set-up can be challenging [28], especially when a large number of items and item groups are involved. Even then, standard errors are not estimated for all model parameters in `mirt`.

2.2. Reparameterization: *StrMixIRMs Using Interaction Effects between Latent Classes and Item Groups*

We propose a new parameterization of the `StrMixIRM` where the differentiation parameters (τ_{hk} s) are treated as interaction effects between latent classes and item groups. To do this, additional discrete latent variables for the interactions are introduced, and the corresponding regression coefficient parameters are defined as the differentiation parameters. This new parameterization can be best understood in a two-way ANOVA context, where latent classes are the first factor and the item groups are the second factor. Dummy variables are used to indicate latent class and item group pairs, and the differentiation parameters (regression coefficients) indicate their interaction effects. The main difference with a conventional two-way ANOVA is that the dummy variables are latent because class memberships are unknown and must be estimated with the model.

2.2.1. Mixture Item Response Models

Let us define binary latent variables a_{ih} such that $a_{ih} = 1$ if respondent i is nested within latent class h , and $a_{ih} = 0$ otherwise. With this, we first reformulate the traditional mixture Rasch model (i.e., Equation (1)) as,

$$\text{logit}(\mathcal{P}(Y_{ij} = 1 | \theta_{ih}, a_{ih})) = \theta_{ih} - \beta_j + \lambda_{jh}a_{ih}, \quad (4)$$

where λ_{jh} is the regression coefficient of the binary latent variable a_{ih} . For model identification, we set $a_{i,h=1} = 0$ and $\lambda_{j,h=1} = 0$. By setting $\beta_{jh} = \beta_j - \lambda_{jh}a_{ih}$, Equation (4) is equivalent to the mixture Rasch model in Equation (1).

To explain this equivalence, with $H = 3$, the linear predictors for each class are $\theta_{i,h=1} - \beta_j$, $\theta_{i,h=2} - \beta_j + \lambda_{j,h=2}$, and $\theta_{i,h=3} - \beta_j + \lambda_{j,h=3}$ for Classes 1, 2, and 3, respectively. First, we can see that θ_{ih} is equal in both equations for all classes. Second, for Class 1 ($h = 1$), $\beta_{j,h=1}$ in Equation (1) equals β_j in Equation (4). Third, for other classes ($h \neq 1$), β_{jh} in Equation (1) equals $\beta_j - \lambda_{jh}$ in Equation (4).

Readers familiar with multiple linear regression may notice that a_{ih} is the indicator of the interaction effect (represented by λ_{jh}) between latent class h and item j , and $-\beta_j$ is the intercept for item j . Assuming latent class membership as known for all respondents when $H = 3$, Table 2 shows the values of a_{ih} with the linear predictors. As defined in Table 2, a_{ih} can only be 1 when involved with latent classes that are not Class 1 (i.e., $h \neq 1$). Thus, $\lambda_{j,h=2}$ and $\lambda_{j,h=3}$ are the interaction effects between latent classes and item j . The interpretation is consistent with that of linear regression. β_j is the item difficulty of Class 1 (intercept in the conventional linear regression), and λ_{jh} is the additional difference of Class h ($h \neq 1$) from Class 1 over the main effects of class and item. As class membership is unknown in advance, however, $a_{i,h=2}$ and $a_{i,h=3}$ (more generally a_{ih} for $h \neq 1$) are binary latent variables that need to be estimated from the data.

Table 2. Dummy variables (a_{ih}) for the linear predictors in the mixture Rasch model for Item j .

Latent Class	Logit	a_{ih} for λ_{jh}	
		$\lambda_{j,h=2}$	$\lambda_{j,h=3}$
$h = 1$	$\theta_{i,h=1} - \beta_j$	0	0
$h = 2$	$\theta_{i,h=2} - \beta_j + \lambda_{j,h=2}$	1	0
$h = 3$	$\theta_{i,h=3} - \beta_j + \lambda_{j,h=3}$	0	1

2.2.2. Structured Mixture Item Response Models

With this in mind, we can derive the StrMixIRM based on the parameterization in Equation (4) by introducing K item groups and reformulating Equation (3). We reiterate that the parameters of interest are interaction effects between latent class h and item group k (i.e., τ_{hk} s). The measurement model of Equation (3) is reformulated as

$$\text{logit}(\mathcal{P}(Y_{ijk} = 1 | \theta_{ih}, a_{ihk})) = \theta_{ih} - \beta_{jk} + \lambda_{jhk} a_{ihk}, \tag{5}$$

where a_{ihk} is a binary latent variable of respondent i within latent class h on item group k . $a_{ihk} = 1$ if respondent i is nested within latent class h and responds to an item within item group k , and $a_{ihk} = 0$ otherwise. λ_{jhk} is the regression coefficient of item j on a_{ihk} with an equality constraint such that $\lambda_{hk} = \lambda_{jhk} = \lambda_{j' hk}$ for all items j and j' within item group k ($j \neq j'$). a_{ihk} and λ_{jhk} regarding $h = 1$ and $k = 1$ are fixed to 0 for model identification. The equally constrained λ_{hk} is equivalent to the differentiation parameter τ_{hk} in Equation (3).

Table 3 presents the linear predictors for $H = 3$ and $K = 3$ of the StrMixIRM in Equation (5). By comparing it with Table 1, we can see that $\lambda_{hk} = \tau_{hk}$.

Table 3. Linear predictors in the StrMixIRM for $H = 3$ and $K = 3$ with the new parameterization.

	Item Group 1 ($k = 1$)	Item Group 2 ($k = 2$)	Item Group 3 ($k = 3$)
Class 1 ($h = 1$)	$\theta_{i,h=1} - \beta_{j,k=1}$	$\theta_{i,h=1} - \beta_{j,k=2}$	$\theta_{i,h=1} - \beta_{j,k=3}$
Class 2 ($h = 2$)	$\theta_{i,h=2} - \beta_{j,k=1}$	$\theta_{i,h=2} - \beta_{j,k=2} + \lambda_{h=2,k=2}$	$\theta_{i,h=2} - \beta_{j,k=3} + \lambda_{h=2,k=3}$
Class 3 ($h = 3$)	$\theta_{i,h=3} - \beta_{j,k=1}$	$\theta_{i,h=3} - \beta_{j,k=2} + \lambda_{h=3,k=2}$	$\theta_{i,h=3} - \beta_{j,k=3} + \lambda_{h=3,k=3}$

Note. All λ_{hk} s relating the first row and column are fixed to 0 for model identification. $\lambda_{hk} = \tau_{hk}$.

Presenting Equation (5) by means of linear regression helps us understand the differentiation parameter, τ_{hk} , as the interaction effect between latent class h and item group k . Assuming class membership are known, Table 4 presents linear predictors of latent class and item group pairs when $H = 3$ and $K = 3$. The last four columns in the table indicate the values of a_{ihk} , which is essentially the same as the dummy variables of the interaction effects in linear regressions for a two-way ANOVA. For instance, the column $\lambda_{h=2,k=3}$ only consists of 1 for the second latent class ($h = 2$, the first factor) and the third item group ($k = 3$, the second factor). As a result, θ_{ih} s relate the sum of the main effect and residual of latent class h (more specifically $\theta_{ih} = \mu_h + \text{residual}$), β_{jk} s relate the main effect of item j in item group k , and $\lambda_{hk} = \tau_{hk}$ s are the interaction effects between latent class h and item group k . Again, a_{ihk} s are treated as latent because class membership is unknown.

Employing the new parameterization in existing software programs or packages requires the following steps:

- First, introduce additional binary latent variables corresponding to item group k (a_{ihk}). In `mirt` and `Mplus`, it is possible to set the Gaussian quadrature points to 1 for all a_{ihk} . In `Mplus`, this can be achieved by constraining the mean and variance of a factor to be 1 and 0, respectively. Example syntax for `Mplus` is provided in the Supplemental Document (Listing S1).
- Second, impose equality constraints on the regression coefficients corresponding to each a_{ihk} . Equality constraints on the regression coefficients are more commonly

available in existing software programs or packages, compared to equality constraints on item difficulty differences.

- Third, set μ_h for $h \neq 1$ to be freely estimated. In Section 3, we provide a step-by-step illustration using `mirt` with detailed R codes and descriptions about the codes.

The steps above might seem complicated, but this approach is more flexible than the linear constraint approach proposed previously. At a minimum, this new approach can be employed with the `mirt` package in R, free of charge. We will also show below how many extensions of the StrMixIRM can be estimated with the proposed implementation with the `mirt` package.

Table 4. Dummy variables (a_{ihk}) for the linear predictors in the StrMixIRM for $H = 3$ and $K = 3$.

Latent Class	Item Group	Logit	a_{ihk} for λ_{hk}			
			$\lambda_{h=2,k=2}$	$\lambda_{h=2,k=3}$	$\lambda_{h=3,k=2}$	$\lambda_{h=3,k=3}$
$h = 1$	$k = 1$	$\theta_{i,h=1} - \beta_{j,k=1}$	0	0	0	0
	$k = 2$	$\theta_{i,h=1} - \beta_{j,k=2}$	0	0	0	0
	$k = 3$	$\theta_{i,h=1} - \beta_{j,k=3}$	0	0	0	0
$h = 2$	$k = 1$	$\theta_{i,h=2} - \beta_{j,k=1}$	0	0	0	0
	$k = 2$	$\theta_{i,h=2} - \beta_{j,k=2} + \lambda_{h=2,k=2}$	1	0	0	0
	$k = 3$	$\theta_{i,h=2} - \beta_{j,k=3} + \lambda_{h=2,k=3}$	0	1	0	0
$h = 3$	$k = 1$	$\theta_{i,h=3} - \beta_{j,k=1}$	0	0	0	0
	$k = 2$	$\theta_{i,h=3} - \beta_{j,k=2} + \lambda_{h=3,k=2}$	0	0	1	0
	$k = 3$	$\theta_{i,h=3} - \beta_{j,k=3} + \lambda_{h=3,k=3}$	0	0	0	1

Note. All $\lambda_{hk} = \tau_{hk}$.

2.3. Extensions and Coverage in `mirt`

Several extensions of the StrMixIRM have been proposed since Wilson [19] first introduced the Rasch form of the model with dichotomously scored items. Extensions include (a) 2PL IRT models (e.g., with slope differentiation parameters) [17], (b) models for polytomous item responses [17,22] and (c) multidimensional IRT models [18]. We briefly describe such extensions and then discuss the coverage of the various StrMixIRMs in the `mirt` package. The model presentation makes use of the new parameterization proposed in Section 2.2.

2.3.1. Two-Parameter Logistic IRT Models

Jeon [17] suggested a two-parameter logistic (2PL) version of the StrMixIRM, where item slopes on θ_{ih} are introduced. The measurement model is given by

$$\text{logit}(\mathcal{P}(Y_{ijk} = 1 | \theta_{ih}, a_{ihk})) = \alpha_{jk}\theta_{ih} - \beta_{jk} + \tau_{hk}a_{ihk}, \tag{6}$$

where α_{jk} is the item slope of item j within item group k . As item slopes are introduced, θ_{ih} for the first latent class follows a standard normal distribution, $N(0, 1^2)$. Means and variances for other classes are freely estimated as in Equation (5). This model applies in cases where researchers hypothesize items have different weights.

Jeon further extended Equation (6) by introducing slope differentiation parameters that allow latent classes to differ in item slopes depending on item groups in addition to item difficulty parameters. The measurement model is

$$\text{logit}(\mathcal{P}(Y_{ijk} = 1 | \theta_{ih}, a_{ihk})) = (\alpha_{jk} + s\tau_{hk})\theta_{ih} - \beta_{jk} + \tau_{hk}a_{ihk}, \tag{7}$$

where $s\tau_{hk}$ is the slope differentiation parameter of item group k on latent class h . This model will be useful if researchers hypothesize, for example, that some item groups would differentiate (better or worse) respondents with a higher latent trait from those with a lower

latent trait. An alternative form of a Rasch version with slope differentiation parameters can be written as

$$\text{logit}(\mathcal{P}(Y_{ijk} = 1|\theta_{ih}, a_{ihk})) = (1 + s\tau_{hk})\theta_{ih} - \beta_{jk} + \tau_{hk}a_{ihk}, \quad (8)$$

where α_{jk} is replaced with 1 because it is the Rasch StrMixIRM. Like Equation (5), the only constraint is regarding the mean of the reference latent class (i.e., $\mu_1 = 0$), while others are freely estimated. Hereafter, we call Equations (5)–(8) Rasch StrMixIRM, 2PL StrMixIRM, 2PL StrMixIRM with $s\tau_{hk}$, and Rasch StrMixIRM with $s\tau_{hk}$, respectively.

2.3.2. Polytomous Item Responses

Draney [22] and Jeon [17] expanded StrMixIRMs to their polytomous forms using partial credit models (PCM) [29] and graded response models (GRM) [30]. Our novel parameterization can also accommodate polytomous item types. For example, the GRM with the Rasch StrMixIRM is given by

$$\text{logit}(\mathcal{P}(Y_{ijk} \geq m|\theta_{ih}, a_{ihk})) = \theta_{ih} - \beta_{jkm} + \tau_{hk}a_{ihk}, \quad (9)$$

where m is the $m + 1$ th category ($m = 0, \dots, M - 1$), and β_{jkm} is the $m + 1$ th category's threshold parameter in the cumulative logit function ($\beta_{jk0} = -\infty$).

2.3.3. Multidimensionality

Jeon [18] also extended the StrMixIRM to incorporate multidimensional latent traits with dimension-specific differentiation parameters, where the classification of respondents can be made based on (a) the dimensions as a whole (single-membership) or (b) each dimension (mixed-membership). For example, the measurement model of single-membership models with D dimensions is given by

$$\text{logit}(\mathcal{P}(Y_{ijk} = 1|\theta_{idh}, a_{idhk})) = \theta_{idh} - \beta_{jk} + \tau_{dhk}a_{idhk}, \quad (10)$$

where θ_{idh} is the latent trait of respondent i within latent class h on dimension d ($d = 1, \dots, D$), τ_{dhk} is the differentiation parameter of latent class h and item group k on dimension d , and a_{idhk} is the discrete latent indicator variable of respondent i within latent class h for item group k on dimension d . The latent trait vector $\theta_{ih} = (\theta_{i1h}, \dots, \theta_{idh}, \dots, \theta_{iDh})'$ follows a multivariate normal distribution, $MVN(\boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)$, where $\boldsymbol{\mu}_h$ and $\boldsymbol{\Sigma}_h$ are the mean vector and covariance matrix of θ_{ih} , respectively. For model identification, $\boldsymbol{\mu}_h = \mathbf{0}$. Interested readers can refer to Jeon [18] for more in-depth discussion and Jeon, Draney, Wilson, and Sun [31] for an application to adolescents' developmental stages in deductive reasoning.

2.3.4. Coverage of StrMixIRM Extensions in mirt

The *mirt* package can handle a myriad of variations of the StrMixIRM using the new parameterization in Section 2.2. Table 5 summarizes the package's capabilities for handling the aforementioned extended models. We note that the 2PL StrMixIRM with $s\tau_{hk}$ (Equation (7)) can be fit in *mirt*, but the slope differentiation parameters have to be defined as constant differences between item slopes across latent classes. This requires additional codes similar to the parameterization using linear equality constraints. Therefore, we limit the scope of our demonstration to the remaining three models: Rasch StrMixIRM (Equation (5)), Rasch StrMixIRM with $s\tau_{hk}$ (Equation (8)), and 2PL StrMixIRM (Equation (6)). Interested readers can refer to the GitHub repository (<https://github.com/yusuh09/StrMixIRM>), accessed on 10 January 2024) for codes and analysis results.

Table 5. Possible StrMixIRMs in the *mirt* package.

Basis Model	Response Type		Multidimensionality	
	Dichotomous	Polytomous	Single	Mixed
Rasch	Yes	GRM and PCM	Yes	Yes
Rasch with $s\tau_{hk}$	Yes	GRM and PCM	Yes	Yes
2PL	Yes	GRM and PCM	Yes	Yes
2PL with $s\tau_{hk}$	Yes	GRM and PCM	Yes	Yes

Note. We removed StrMixIRM for simplicity. For the 2PL with $s\tau_{hk}$, additional custom codes are required, and standard errors are not calculated. "Yes" indicates that fitting the models is available, and "GRM and PCM" indicates both GRM and PCM type models are available for polytomous items.

3. StrMixIRM in R Using *mirt*

In this section, we illustrate how to fit various StrMixIRMs using the *mirt* package. For simplicity and clarity, we focus on the unidimensional counterparts of the first three models in Table 5: Rasch, Rasch with $s\tau_{hk}$, and 2PL StrMixIRMs. For multidimensional models, interested readers can refer to the Supplemental Document (Listings S2–S4).

In *mirt*, estimation of mixture IRT models uses the main function `multipleGroup` with "mixture-H" in the argument, where H is the number of latent classes. It is important to take heed that since mixture IRT models are prone to local maxima, multiple random starting values are highly recommended to ensure convergence at the global maximum [32]. In this paper, we run each model with 15 randomly generated multiple starting values and choose the replication with the largest log-likelihood as our final result.

Furthermore, obtaining standard errors from mixture models with more than two latent classes may result in a fatal error in the current version of *mirt* 1.41. Although this issue has been resolved [33] (see Issue #247), users must directly install the updated version from the *mirt* GitHub website [33]. Without standard errors, the model is readily estimated with more than two latent classes in the current version. The default argument in the `multipleGroup` function is to not calculate standard errors.

We use two datasets for demonstration: the Examination for the Certificate of Proficiency in English (ECPE) data [34] from the *GDINA* [35] package and the verbal aggression data [36] from the *lme4* [37] package. The two datasets serve to show different contexts to which the StrMixIRM can be applied. The ECPE data are used to fit the unidimensional models for dichotomous item responses with three latent classes. The verbal aggression data are used to fit the unidimensional models for polytomous item responses with two latent classes.

3.1. Dichotomous Responses: ECPE Data Analysis

The ECPE data, widely used in DCMs [15,38–40], consist of item responses from 2922 respondents to 28 items measuring three skills or attributes: morphosyntactic, cohesive, and lexical rules. Templin and Bradshaw [41] used the ECPE data to model attribute hierarchies. More specifically, they hypothesized a linear attribute hierarchy where lexical rules must be mastered to master cohesive rules, which in turn must be mastered before mastering morphosyntactic rules.

To define item groups, we follow this argument about the hierarchical nature of the attributes and refer to the Q-matrix used in Templin and Bradshaw [41]. The Q-matrix is a matrix indicating item-attribute incidence; if an item measures certain attributes, corresponding cells are indicated by 1, and 0 otherwise [15]. We define three item groups: the lexical rules as the reference item group ($k = 1$), and the others as focal item groups in which the cohesive rules constitute the second item group ($k = 2$) and the morphosyntactic rules form the third item group ($k = 3$). In the case when an item measures multiple attributes, the item group is set to the attribute furthest down the hierarchy. For example, Item 1 measures both cohesive and morphosyntactic rules and is thus placed within the morphosyntactic item group.

The results from Templin and Bradshaw imply that solving cohesive and morphosyntactic rules requires respondents to have additional cognitive skills beyond their overall English proficiency. For the StrMixIRM analyses, we hypothesize there are three sub-groups of respondents: (a) the highest performing class where respondents have all additional required skills relative to the other two classes, (b) moderate performing class with respondents lacking one or more additional skills (i.e., cohesive and/or morphosyntactic rules) compared to the highest performing class but having more of one or more of these skills relative to the lowest performing class, and (c) the lowest performing class where respondents are lacking in all additional required skills compared to the other two classes. Beyond overall English proficiency, we hypothesize that performance on cohesive and morphosyntactic rules successfully differentiates respondents into these three latent classes. Examining the latent trait distribution and differentiation parameters will help us understand the extent to which lower-performing respondents show additionally lower performance on the additional skill domains when compared to higher-performing respondents.

The below Listing 1 shows data loading and item group information.

Listing 1. ECPE: load data.

```

1 library(GDINA) # From GDINA package
2
3 ecpe <- GDINA::ecpe$dat # loading ECPE data
4
5 # Item group
6 # 1 - lexical: items 4, 5, 6, 9, 15, 18, 19, 22, 26, and 28
7 # 2 - cohesive: items 2, 8, 17, 23, and 24
8 # 3 - morphosyntactic: items 1, 3, 7, 10-14, 16, 20, 21, 25, and 27

```

3.1.1. Model Specification Using `mirt.model` Syntax

As StrMixIRMs are confirmatory mixture IRT models, more detailed model specification is required using the `mirt.model` syntax. We first present the model syntax for the Rasch StrMixIRM with detailed notes on what each component refers to. For subsequent model syntaxes of the Rasch StrMixIRM with $s\tau_{hk}$ and 2PL StrMixIRM, we provide the model syntax with notes focused on lines that differ from the Rasch StrMixIRM. Model syntaxes are realizations of Equations (5), (6) and (8). Listing 2 presents the input model syntax of the Rasch StrMixIRM.

- Lines 5 to 7 indicate items loaded on the latent variables. F_1 , T_2 , and T_3 indicate θ_{ih} , a_{ih2} , and a_{ih3} , respectively. The corresponding slopes can be found in subsequent lines as a_1 , a_2 , and a_3 , following the order of the input latent variables.
- Line 9 constrains item intercepts to be equal across latent classes (β_{jk}).
- Lines 11 and 12 fix the slope of F_1 to be 1 as defined in the Rasch StrMixIRM. In the case of the two other models, this part will be replaced (see below). `MIXTURE_1` indicates the first latent class (reference class), and `MIXTURE_2` and `MIXTURE_3` are the respective focal latent classes.
- Lines 14 and 15 fix the slopes of T_2 and T_3 for the reference latent class to be 0 for model identification, leading to their differentiation parameters being 0.
- Lines 17 to 18 constrain the slopes of T_2 and T_3 for the focal classes to be equal across the loaded items within latent classes. This leads to the differentiation parameters for the focal classes.
- Lines 20 to 22 model the latent trait distributions. In `mirt`, the default setting is a standard normal distribution for each latent variable with covariances between latent variables set to 0. Per order of input latent variables, the mean and variance are represented by `MEAN_F` and `COV_FF` where F indicates the F th latent variable. Covariances between the F th and F' th latent variables are represented by `COV_FF'`. In the context of this paper, mean and variance regarding θ_{ih} are only specified (i.e., F_1). For the discrete latent variables, we can leave them at the default setting and only need to define quadrature points in Section 3.1.2.

- Line 20 specifies the latent trait distribution of the reference class. For model identification, $\mu_1 = 0$ so that only the variance is freely estimated: `FREE [MIXTURE_1] = (GROUP, COV_11)`. In the case of the 2PL StrMixIRM, this line is removed.
- Lines 21 and 22 specify the distribution of the focal classes. Both mean and variance are freely estimated.

Listing 2. ECPE: Rasch StrMixIRM syntax.

```

1 # load the package
2 library(mirt)
3
4 # mirt model syntax for Rasch model
5 mod.syn <- "F1 = 1-28
6 T2 = 2,8,17,23,24
7 T3 = 1,3,7,10,11,12,13,14,16,20,21,25,27
8
9 CONSTRAINB = (1-28, d)
10
11 START [MIXTURE_1,MIXTURE_2,MIXTURE_3] = (1-28, a1, 1)
12 FIXED [MIXTURE_1,MIXTURE_2,MIXTURE_3] = (1-28, a1)
13
14 START [MIXTURE_1] = (2,8,17,23,24, a2, 0),
    (1,3,7,10,11,12,13,14,16,20,21,25,27, a3, 0)
15 FIXED [MIXTURE_1] = (2,8,17,23,24, a2),
    (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
16
17 CONSTRAIN [MIXTURE_2] = (2,8,17,23,24, a2),
    (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
18 CONSTRAIN [MIXTURE_3] = (2,8,17,23,24, a2),
    (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
19
20 FREE [MIXTURE_1] = (GROUP, COV_11)
21 FREE [MIXTURE_2] = (GROUP, MEAN_1), (GROUP, COV_11)
22 FREE [MIXTURE_3] = (GROUP, MEAN_1), (GROUP, COV_11)"

```

For the Rasch StrMixIRM with $s\tau_{hk}$ (Equation (8)), Lines 11 and 12 should be replaced as in Listing 3 since the model introduces slope differentiation parameters.

Compared to Lines 11 and 12 in Listing 2, Lines 8 to 15 in Listing 3 show that

- Slope parameters are fixed to 1 for the reference item group for all latent classes (Lines 8 and 9),
- Slope parameters are fixed to 1 for the focal item groups in the reference latent class (Lines 11 and 12),
- Slope parameters are freely estimated for the focal item groups in the focal latent classes but constrained to be equal within latent classes and item groups (Lines 14 and 15) to produce the slope differentiation parameters.

The 2PL StrMixIRM (Equation (6)) introduces item slopes and constrains them to be equal across latent classes, so the first latent class' latent distribution must be $N(0, 1^2)$ for model identification. For this, Lines 11–12 in Listing 2 become Line 7, and Line 20 is removed as in Listing 4.

Elaborating, compared to Lines 11, 12, and 20 in Listing 2, Listing 4 shows that

- Slope parameters are freely estimated but constrained to be equal between latent classes (Line 7),
- Line 20 of Listing 2 is removed, meaning that the latent trait distribution of the reference latent class is $N(0, 1^2)$.

Listing 3. ECPE: Rasch StrMixIRM with $s\tau_{hk}$ syntax.

```

1 # mirt model syntax for Rasch model with slope differentiation parameters
2 mod.syn <- "F1 = 1-28
3 T2 = 2,8,17,23,24
4 T3 = 1,3,7,10,11,12,13,14,16,20,21,25,27
5
6 CONSTRAINB = (1-28, d)
7
8 START [MIXTURE_1,MIXTURE_2,MIXTURE_3] = (4,5,6,9,15,18,19,22,26,28, a1, 1)
9 FIXED [MIXTURE_1,MIXTURE_2,MIXTURE_3] = (4,5,6,9,15,18,19,22,26,28, a1)
10
11 START [MIXTURE_1] = (2,8,17,23,24,1,3,7,10,11,12,13,14,16,20,21,25,27, a1,
12 1)
13 FIXED [MIXTURE_1] = (2,8,17,23,24,1,3,7,10,11,12,13,14,16,20,21,25,27, a1)
14
15 CONSTRAIN [MIXTURE_2] = (2,8,17,23,24, a1),
16 (1,3,7,10,11,12,13,14,16,20,21,25,27, a1)
17 CONSTRAIN [MIXTURE_3] = (2,8,17,23,24, a1),
18 (1,3,7,10,11,12,13,14,16,20,21,25,27, a1)
19
20 START [MIXTURE_1] = (2,8,17,23,24, a2, 0),
21 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3, 0)
22 FIXED [MIXTURE_1] = (2,8,17,23,24, a2),
23 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
24
25 CONSTRAIN [MIXTURE_2] = (2,8,17,23,24, a2),
26 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
27 CONSTRAIN [MIXTURE_3] = (2,8,17,23,24, a2),
28 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
29
30 FREE [MIXTURE_1] = (GROUP, COV_11)
31 FREE [MIXTURE_2] = (GROUP, MEAN_1), (GROUP, COV_11)
32 FREE [MIXTURE_3] = (GROUP, MEAN_1), (GROUP, COV_11)"

```

Listing 4. ECPE: 2PL StrMixIRM syntax.

```

1 # mirt model syntax for 2PL model
2 mod.syn <- "F1 = 1-28
3 T2 = 2,8,17,23,24
4 T3 = 1,3,7,10,11,12,13,14,16,20,21,25,27
5
6 CONSTRAINB = (1-28, d)
7 CONSTRAINB = (1-28, a1)
8
9 START [MIXTURE_1] = (2,8,17,23,24, a2, 0),
10 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3, 0)
11 FIXED [MIXTURE_1] = (2,8,17,23,24, a2),
12 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
13
14 CONSTRAIN [MIXTURE_2] = (2,8,17,23,24, a2),
15 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
16 CONSTRAIN [MIXTURE_3] = (2,8,17,23,24, a2),
17 (1,3,7,10,11,12,13,14,16,20,21,25,27, a3)
18
19 FREE [MIXTURE_2] = (GROUP, MEAN_1), (GROUP, COV_11)
20 FREE [MIXTURE_3] = (GROUP, MEAN_1), (GROUP, COV_11)"

```

3.1.2. Model Estimation

Listing 5 shows code for running the StrMixIRMs with 15 random starting points. These codes do not differ depending on the StrMixIRMs defined above.

Listing 5. ECPE: Calibration.

```

1 # Quadrature points: 145s for the latent trait [-8, 8] / 1 for two discrete
  latent variables
2 quad <- cbind(seq(-8, 8, length.out = 145), 1, 1)
3
4 # Analysis with multiple random starting points (15)
5 ecpe.out <- vector(mode = "list", length = 15)
6 ecpe.ll <- NULL
7 set.seed(2023)
8 for(r in 1:15){
9   # running model with mirt
10  ecpe.out[[r]] <-
11  mirt::multipleGroup(data = ecpe, # response data object
12                      model = mod.syn, # mirt model syntax
13                      SE=TRUE, # calculate standard errors
14                      dentype = "mixture-3", # mixture IRT for 3 classes
15                      technical = list(customTheta = quad, # specified
16                                     quadratures
17                                     iterations
18                                     NCYCLES = 2000), # max number of
19                                     GenRandomPars = TRUE) # random starting values
20  # save log-likelihood at convergence
21  ecpe.ll <- c(ecpe.ll, mirt::extract.mirt(ecpe.out[[r]], what="logLik"))
22 }

```

- Line 2 manually defines the quadrature points for the primary latent trait θ_{ih} (column 1) and discrete latent variables a_{ihk} (columns 2 and 3). Because mirt does not allow adaptive quadrature by iterations, we set a wide range and large number of quadrature points to guarantee a sufficient level of precision for estimating means and variances, in particular for the case that focal classes' mean and variance parameters are large. Users can change the range and number of quadrature points as needed. By setting the second and third columns (i.e., a_{ih2} and a_{ih3}) to have only one quadrature point, the standard EM algorithm [42] only evaluates the one quadrature point, resulting in the discrete latent variables.
- Lines 5 to 20 estimate the StrMixIRMs using the `mirt::multipleGroup` function. Because of the possibility of local maxima, we fit the model with 15 starting points.
- Lines 11 to 17 include the function arguments for the `mirt::multipleGroup` function. `data = ecpe` is the item response data (Line 11). `model = mod.syn` is the mirt model syntax (Line 12). `SE=TRUE` is to obtain standard errors of estimates (Line 13). The default is Oakes' method [43,44]. `dentype = "mixture-3"` indicates the estimated model is a mixture IRT model with three latent classes (Line 14).
- Lines 15 to 16 include the `technical` argument. `customTheta = quad` commands mirt to use the quadrature points specified in Line 2 (Line 15). `NCYCLES = 2000` sets the number of iterations to be 2000 (Line 16); the default setting is 500, but we increase it to promote convergence.
- Line 17, `GenRandomPars = TRUE`, refers to randomly generated starting values.
- Line 20 stores the log-likelihood value at convergence for each replication. The replication with the highest log-likelihood is selected as the output.

Once model estimation is completed, the results can be printed using Listing 6.

Listing 6. ECPE: Model output.

```

1 # result with the highest log-likelihood
2 highest <- which(ecpe.ll == max(ecpe.ll))
3
4 # select the best output
5 out <- ecpe.out[[highest]]
6
7 # model information
8 mirt::extract.mirt(out, what="nest") # the number of unique parameters
9 mirt::extract.mirt(out, what="logLik") # log-likelihood
10 mirt::extract.mirt(out, what="AIC") # AIC
11 mirt::extract.mirt(out, what="BIC") # BIC
12 mirt::extract.mirt(out, what="SABIC") # SABIC
13
14 # Estimates and SE
15 out.coef <- mirt::coef(out, printSE=T)
16
17 # item slope (in the case of 2PL)
18 slope <- NULL
19 for(i in 1:28){
20 slope <- rbind(slope, out.coef[["MIXTURE_1"]][[i]][, "a1"])
21 }
22
23 # item intercept (a negative of item difficulty)
24 int <- NULL
25 for(i in 1:28){
26 int <- rbind(int, out.coef[["MIXTURE_1"]][[i]][, "d"])
27 }
28
29 # Intercept Tau parameter
30 out.coef[["MIXTURE_2"]][[2]][, "a2"] # tau22
31 out.coef[["MIXTURE_2"]][[1]][, "a3"] # tau23
32 out.coef[["MIXTURE_3"]][[2]][, "a2"] # tau32
33 out.coef[["MIXTURE_3"]][[1]][, "a3"] # tau33
34
35 # Slope Tau parameter (in the case of slope differentiation parameters)
36 out.coef[["MIXTURE_2"]][[2]][, "a1"] # stau22
37 out.coef[["MIXTURE_2"]][[1]][, "a1"] # stau23
38 out.coef[["MIXTURE_3"]][[2]][, "a1"] # stau32
39 out.coef[["MIXTURE_3"]][[1]][, "a1"] # stau33
40
41 # Latent trait distribution
42 out.coef[["MIXTURE_1"]][[28+1]] # for the class 1 (reference)
43 out.coef[["MIXTURE_2"]][[28+1]] # for the class 2 (focal 1)
44 out.coef[["MIXTURE_3"]][[28+1]] # for the class 3 (focal 2)
45
46 # class probability
47 exp(c(out.coef[["MIXTURE_1"]][[28+1]][1, "PI"],
48       out.coef[["MIXTURE_2"]][[28+1]][1, "PI"],
49       out.coef[["MIXTURE_3"]][[28+1]][1, "PI"]))/
50 sum(exp(c(out.coef[["MIXTURE_1"]][[28+1]][1, "PI"],
51           out.coef[["MIXTURE_2"]][[28+1]][1, "PI"],
52           out.coef[["MIXTURE_3"]][[28+1]][1, "PI"])))

```

- Lines 2 and 5 select the output of the iteration with the highest log-likelihood.
- Lines 8 to 12 print the number of estimated parameters and model fit indices (e.g., AIC, BIC, and SBIC).
- Line 15 provides the model parameter estimates with standard errors.
- Lines 18 to 21 give the slope estimates. This is meaningful when the 2PL StrMixIRM is fit.
- Lines 24 to 27 produce the intercept estimates (i.e., $-\beta_{jk}$).
- Lines 30 to 33 give the differentiation parameters. Item groups 2 and 3 are related to the second latent variable ("a2") and the third latent variable ("a3"), respectively.

- Lines 36 to 39 provide the slope differentiation parameters. “est – 1” results in the slope differentiation parameters (see Equation (8)). Standard errors are not affected by subtraction.
- Lines 42 to 44 are to obtain the estimates for the latent trait distributions. The estimates are stored in the last element (the number of items + 1). The only parameters of interest are MEAN_1, COV_11, and PI. PI is the coefficient for the class probability on the multinomial logit model.
- Lines 46 to 52 give the latent class probabilities, π_{hs} .

Table 6 summarizes the final estimated results of the three models. For simplicity, only structural parameters (i.e., latent class probabilities, latent trait distribution parameters, and differentiation parameters) are included. Item parameter estimates are located in the Supplemental Document (Tables S2 and S5).

Table 6. ECPE: Model parameter estimates.

		Rasch		Rasch with $s\tau_{hk}$		2PL	
1. Model fit		Value		Value		Value	
N.par		39		43		66	
Log-lik		−42,635.12		−42,629.41		−42,463.00	
AIC		85,348.25		85,344.83		85,058.00	
BIC		85,581.47		85,601.97		85,452.68	
SBIC		85,457.55		85,465.34		85,242.97	
2. Model parameters		EST	SE	EST	SE	EST	SE
Class probabilities	π_1	0.56	-	0.54	-	0.49	-
	π_2	0.32	-	0.30	-	0.33	-
	π_3	0.12	-	0.17	-	0.19	-
Latent trait distributions	μ_1	0.00	-	0.00	-	0.00	-
	σ_1^2	0.34	0.01	0.57	0.02	1.00	-
	μ_2	0.43	0.04	−0.66	0.03	1.04	0.10
	σ_2^2	1.09	0.05	0.85	0.03	3.85	0.23
	μ_3	−1.60	0.02	0.34	0.07	−3.49	0.05
Differentiation parameters	σ_3^2	0.19	0.01	1.71	0.11	0.81	0.03
	τ_{22}	0.03	0.15	0.52	0.20	−0.40	0.17
	τ_{23}	0.88	0.08	−0.55	0.11	0.64	0.11
	τ_{32}	0.99	0.13	0.89	0.19	1.28	0.17
	τ_{33}	0.89	0.08	1.04	0.14	1.23	0.13
	$s\tau_{22}$	-	-	−0.27	0.10	-	-
	$s\tau_{23}$	-	-	−0.57	0.05	-	-
	$s\tau_{32}$	-	-	−0.42	0.13	-	-
$s\tau_{33}$	-	-	−0.07	0.11	-	-	

Note. For the 2PL StrMixIRM, the original analysis seemed to converge at a local maximum; the largest log-likelihood is substantially smaller than that of Mplus. Therefore, we re-analyzed the data by utilizing the output from the Rasch StrMixIRM as the starting values (see the GitHub repository). The presented output are the results of re-analysis.

Interpretation of results is discussed using the Rasch StrMixIRM result as an example. To determine the meaning of the latent classes, we should look at the latent trait distributions and differentiation parameters together rather than examining each parameter separately. Examining the means of the latent classes, Class 2 shows the highest overall performance ($\hat{\mu}_2 = 0.43$), while Class 3 shows the lowest overall performance ($\hat{\mu}_3 = -1.60$) on the test. Differentiation parameters for cohesive rules item group ($k = 2$) suggest that given θ_{ih} , Class 2 performs similarly to Class 1 ($\hat{\tau}_{22} = 0.03$) whereas Class 3 has better performance than Class 1 ($\hat{\tau}_{32} = 0.99$). Differentiation parameters on morphosyntactic rules ($k = 3$) suggest that both focal classes (i.e., Class 2 and Class 3) demonstrate better performance on corresponding items ($\hat{\tau}_{23} = 0.88$ and $\hat{\tau}_{33} = 0.89$) given θ_{ih} .

Solely relying on the size and direction of the differentiation parameters leads to incorrect results because it ignores the overall performance represented by μ_h . For instance, Class 3 has positive differentiation parameters on both domains but also has a negative latent mean. Therefore, it is necessary to compare the sum of μ_h and τ_{hk} so that the latent classes are accurately evaluated. Table 7 and Figure 1 present each latent class' average performance on each of the three item groups. The ordering of latent classes from highest to lowest performance is equal in every item group: Class 2, followed by Class 1, and lastly, Class 3. Comparing Class 1 to Class 2, the relative outperformance of Class 2 over Class 1 is similar in the lexical and cohesive rules (on average, 0.43 and 0.46, respectively) and remarkable in morphosyntactic rules (1.31). This implies that Class 1 respondents mastered the lexical and cohesive rules but did not master morphosyntactic rules, which are indicated to be the most difficult skill to master.

Table 7. ECPE: Average performance of latent classes on each domain, $\hat{\mu}_h + \hat{\tau}_{hk}$.

	Item Group		
	Lexical	Cohesive	Morphosyntactic
Class 1	0.00	0.00	0.00
Class 2	0.43	0.46	1.31
Class 3	-1.60	-0.61	-0.71

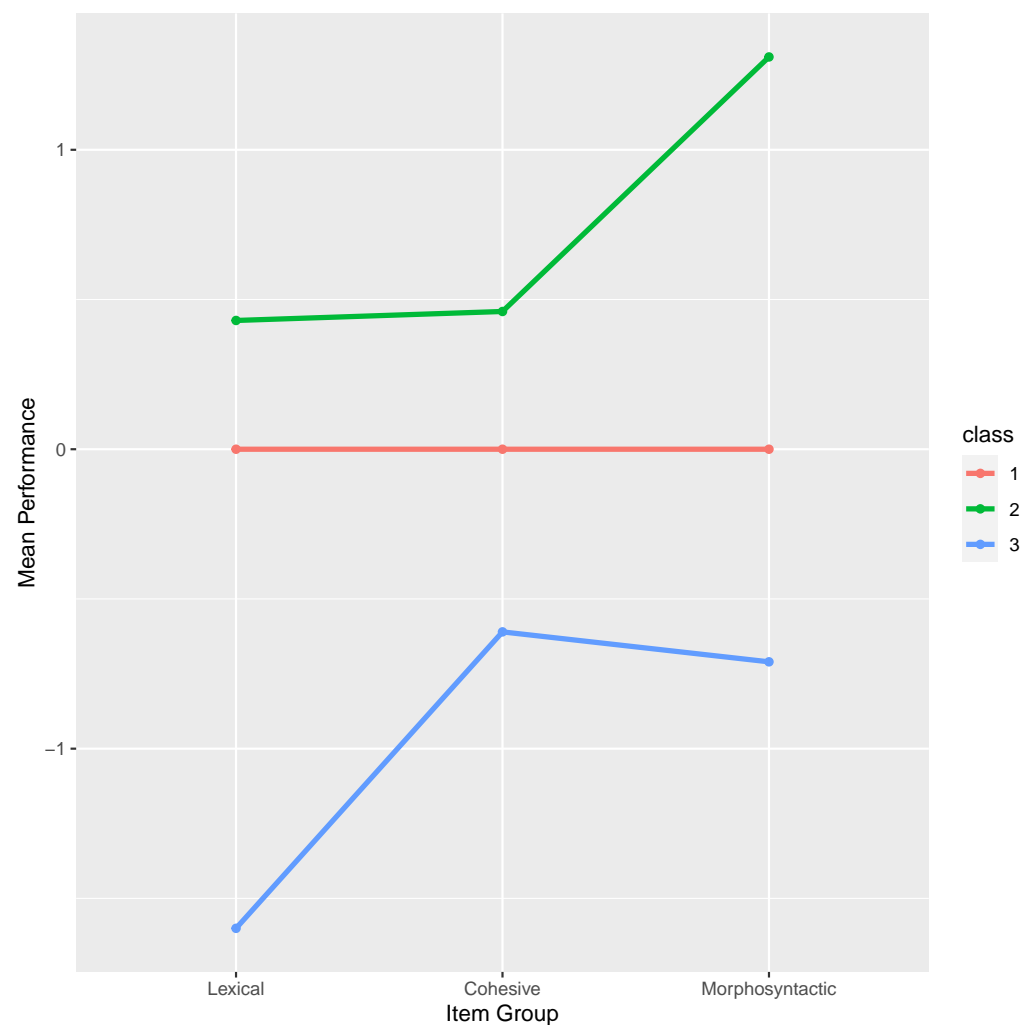


Figure 1. ECPE: Average performance of latent classes on each domain, $\hat{\mu}_h + \hat{\tau}_{hk}$. Red, green, and blue lines are of Classes 1, 2, and 3, respectively.

Inspecting the model parameter results, our hypotheses are confirmed. Respondents are clearly separated into the three latent classes based on their performance on cohesive and morphosyntactic rules: the highest (Class 2), moderate (Class 1), and lowest (Class 3) performing classes. Respondents from Classes 2 and 3 show consistent performance across all domains; they seem to have mastered (Class 2) or not mastered (Class 3) all the additional skills. Respondents from Class 1 show high performance in the lexical and cohesive rules compared to Class 3 but somewhat low performance in the morphosyntactic rules in comparison to Class 2.

3.2. Polytomous Responses: Verbal Aggression Data

The verbal aggression data has widely been used in IRT model applications [36,45,46] that employ mixture IRT models [5,17,18]. A total of 316 individuals responded to 24 items nested within three crossed factors: situation types ('Self-to-lame' and 'Other-to-blame') in four situations ('Bus', 'Train', 'Store', and 'Operator'); behavior types ('Curse', 'Scold', and 'Shout'); and behavior modes ('Want' and 'Do'). Each item has three categories: 'No' (0), 'Perhaps' (1), and 'Yes' (2). Among these factors, Jeon [17] hypothesizes 'Do' behavior would highlight additional differences in verbal aggression between respondents beyond their overall tendency because 'Do' behavior can cause more serious damage than 'Want' behavior. We follow Jeon's [17] hypothesis that the focal item group of 'Do' items can differentiate latent classes and use their polytomous item responses for illustration of the aforementioned three models using GRMs (Equation (9)) in mirt. Listing 7 presents the loading data and item information.

Listing 7. Verbal Aggression: load data.

```

1 library(lme4)
2 data(VerbAgg)
3
4 colnames(VerbAgg)
5 # Anger: Anger score -> person covariate
6 # Gender: Gender -> person covariate
7 # item: item name
8 # resp: subject response -> polytomous no (0) < perhaps (1) < yes (2)
9 # id: respondent id
10 # btype: behavior type -> item covariate (curse, scold, shout)
11 # situ: situation type -> item covariate (other, self)
12 # mode: behavior mode -> item covariate (want, do)
13 # r2: dichotomous item responses -> N (0), Y(1)
14
15 # use only polytomous item responses
16 verb <- VerbAgg[, c("id", "item", "resp")]
17
18 # recode polytomous item responses
19 verb$resp <- ifelse(verb$resp == "no", 0,
20                   ifelse(verb$resp == "perhaps", 1, 2))
21
22 # long to wide
23 verb.wide <- reshape(verb, idvar = "id", timevar="item", direction="wide")
24 verb.wide <- verb.wide[,-1] # remove id column
25 colnames(verb.wide)
26
27 # Item group
28 # ig1 (want items): 1-12
29 # ig2 (do items): 13-24

```

3.2.1. Model Specification

The `mirt.model` syntax for model specification and running models resembles that of the ECPE data example (Section 3.1). Therefore, we underscore changes for the polytomous item response analyses by focusing on Listing 8 that presents model syntax code for the Rasch StrMixIRM.

Listing 8. Verbal aggression: Rasch StrMixIRM syntax.

```

1 # mirt model syntax for Rasch model
2 mod.syn <- "F1 = 1-24
3 T2 = 13-24
4
5 CONSTRAINB = (1-24, d1)
6 CONSTRAINB = (1-24, d2)
7
8 START [MIXTURE_1, MIXTURE_2] = (1-24, a1, 1)
9 FIXED [MIXTURE_1, MIXTURE_2] = (1-24, a1)
10
11 START [MIXTURE_1] = (13-24, a2, 0)
12 FIXED [MIXTURE_1] = (13-24, a2)
13
14 CONSTRAIN [MIXTURE_2] = (13-24, a2)
15
16 FREE [MIXTURE_1] = (GROUP, COV_11)
17 FREE [MIXTURE_2] = (GROUP, MEAN_1), (GROUP, COV_11)"

```

- Lines 2 and 3 indicate the name of the latent variable and items loaded on that latent variable.
- Lines 5 and 6 constrain item intercepts to be equal across latent classes. We note that for both PCMs and GRMs, `mirt` denotes intercepts corresponding to the second and third categories as `d1` and `d2`, respectively.
- The rest of the lines are similar to that of ECPE model syntax (Listing 2). As this analysis assumes two latent classes, we only have `MIXTURE_1` and `MIXTURE_2`.
- For the Rasch with $s\tau_{hk}$ and 2PL StrMixIRMs, the model syntaxes mimic those of the ECPE example (refer to Listings 3 and 4).

3.2.2. Model Estimation

Listing 9 displays how to run models with a specified model syntax (e.g., Listing 8).

Listing 9. Verbal aggression: calibration.

```

1 # Quadrature points: 145s for the latent trait [-8, 8] / 1 for one discrete
  latent variable
2 quad <- cbind(seq(-8, 8, length.out = 145), 1)
3
4 # Analysis with multiple random starting points (15)
5 verb.out <- vector(mode = "list", length = 15)
6 verb.ll <- NULL
7 set.seed(2023)
8 for(r in 1:15){
9   # running model with mirt
10  verb.out[[r]] <-
11  mirt::multipleGroup(data = verb.wide, # response data object
12                      model = mod.syn, # mirt model syntax
13                      SE=TRUE, # calculate standard errors
14                      #itemtype = "gpcm", # for partial credit models
15                      dentype = "mixture-2", # mixture IRT for 2 classes
16                      technical = list(customTheta = quad, # specified
17                                     quadratures
18                                     iterations
19                                     NCYCLES = 1000), # max number of
20                                     GenRandomPars = TRUE) # random starting values
21  # save log-likelihood at convergence
22  verb.ll <- c(verb.ll, mirt::extract.mirt(verb.out[[r]], what="logLik"))
23 }

```

We again use 15 random starting points and select the replication with the highest log-likelihood. The codes are mostly similar to those of the ECPE example, except for the following:

- There is one discrete latent variable (Line 2).
- `data` has been changed (Line 11).
- The default is a GRM, but `itemtype = "gpcm"` can be added for the PCM (Line 14).
- `dentype` equals "mixture-2" because we only assume two latent classes (Line 15).
- `NCYCLES = 1000` in this analysis as the data size is smaller than that of the ECPE (Line 17).

Users can refer to Listing 10 to obtain outputs.

Listing 10. Verbal aggression: model output.

```

1 # result with the highest log-likelihood
2 highest <- which(verb.ll == max(verb.ll))
3
4 # select the best output
5 out <- verb.out[[highest]]
6
7 # model information
8 mirt::extract.mirt(out, what="nest") # the number of unique parameters
9 mirt::extract.mirt(out, what="logLik") # log-likelihood
10 mirt::extract.mirt(out, what="AIC") # AIC
11 mirt::extract.mirt(out, what="BIC") # BIC
12 mirt::extract.mirt(out, what="SABIC") # SABIC
13
14 # Estimates and SE
15 out.coef <- mirt::coef(out, printSE=T)
16
17 # item slope (in the case of 2PL)
18 slope <- NULL
19 for(i in 1:24){
20 slope <- rbind(slope, out.coef[["MIXTURE_1"]][[i]][, "a1"])
21 }
22
23 # item intercepts (a negative of item difficulty)
24 int1 <- NULL # for the second category
25 int2 <- NULL # for the third category
26 for(i in 1:24){
27 int1 <- rbind(int1, out.coef[["MIXTURE_1"]][[i]][, "d1"])
28 int2 <- rbind(int2, out.coef[["MIXTURE_1"]][[i]][, "d2"])
29 }
30
31 # Intercept Tau parameter
32 out.coef[["MIXTURE_2"]][[13]][, "a2"] # tau22
33
34 # Slope Tau parameter (in the case of Rasch with slope differentiation
35 # parameters)
36 out.coef[["MIXTURE_2"]][[13]][, "a1"] # stau22
37
38 # Latent trait distribution
39 out.coef[["MIXTURE_1"]][[24+1]] # for the class 1 (reference)
40 out.coef[["MIXTURE_2"]][[24+1]] # for the class 2 (focal)
41
42 # class probability
43 exp(c(out.coef[["MIXTURE_1"]][[24+1]][1, "PI"],
44 out.coef[["MIXTURE_2"]][[24+1]][1, "PI"])) /
45 sum(exp(c(out.coef[["MIXTURE_1"]][[24+1]][1, "PI"],
46 out.coef[["MIXTURE_2"]][[24+1]][1, "PI"])))

```

Table 8 presents the final estimated results for each of the three models (i.e., Rasch, Rasch with $s\tau_{hk}$, and 2PL StrMixIRMs). Again, only structural parameter estimates are included for simplicity. Item parameter estimates can be found in the supplementary document (Tables S4 and S6).

Once more, we only examine the Rasch StrMixIRM results. Considering the means of θ_{ih} do not have a statistically significant difference ($\hat{\mu}_2 = -0.11$), the positive differentiation parameter ($\hat{\tau}_{22} = 1.74$) clearly shows that respondents in Class 2 are more likely to respond to higher response categories for ‘Do’ items compared to those in Class 1. In other words, Class 2 respondents are more likely to exhibit ‘Do’ verbally aggressive behaviors when confronted with frustrating situations. Like this, ‘Do’ items can clearly differentiate respondents prone to displaying more serious behaviors (Class 2).

Table 8. Verbal aggression: model parameter estimates.

		Rasch		Rasch with $s\tau_{hk}$		2PL	
1. Model fit		Value		Value		Value	
N.par		53		54		66	
Log-lik		−6255.40		−6250.63		−6228.37	
AIC		12,616.81		12,609.27		12,608.74	
BIC		12,815.86		12,812.08		12,894.18	
SBIC		12,647.76		12,640.80		12,653.12	
2. Model parameters		EST	SE	EST	SE	EST	SE
Class probabilities	π_1	0.48	-	0.52	-	0.65	-
	π_2	0.52	-	0.48	-	0.35	-
Latent trait distributions	μ_1	0.00	-	0.00	-	0.00	-
	σ_1^2	3.02	0.42	2.69	0.31	1.00	-
	μ_2	−0.11	0.07	0.05	0.08	−0.57	0.03
	σ_2^2	0.84	0.06	1.19	0.08	0.17	0.01
Differentiation parameters	τ_{22}	1.74	0.13	1.75	0.12	2.13	0.15
	$s\tau_{22}$	-	-	−0.34	0.08	-	-

Therefore, we confirm our hypothesis that there is a subpopulation of respondents (Class 2) exhibiting excess expressions of anger above simply wanting to express it. The proportion of these respondents ($\hat{\pi}_2 = 0.52$) is not negligible, and the amount of additional anger ($\hat{\tau}_{22} = 1.74$) is substantially large.

4. Discussion

In light of limited software options to conduct StrMixIRM analyses, the present study proposes reparameterizing it as a confirmatory mixture IRT model using interaction effects of latent classes and item groups. Along with increased efficiency and new opportunities for previously unexplored extensions of StrMixIRMs, this parameterization makes the StrMixIRM readily implementable in multiple software packages with mixture modeling capabilities.

We first presented the framework and its flexibility to handle many variants of the StrMixIRM (e.g., dichotomous and polytomous forms and multidimensional extensions). We followed with in-depth illustrations on how to estimate these models using the freely available `mirt` package in R, as well as how to interpret the ensuing results. In order to emphasize the applicability of the StrMixIRM outside its origins in modeling developmental stages, we used two empirical datasets to describe two other scenarios where the StrMixIRM may be useful. We presented detailed R code with notes as well as in depth interpretation of the output. We found that the results were very similar between `mirt` and `Mplus` for the two applications, as shown Section S1. Additionally, we conducted a small simulation study to evaluate the performance of `mirt` and `Mplus`. Both programs performed well and

the differences in the estimated results were negligible. Detailed results are provided in Section S4.

The availability of an open-source tool coupled with a systematic introduction to using it for various StrMixIRM estimations is expected to pave the way for more researchers to realize the potential of StrMixIRMs. In addition, we anticipate our parameterization formulating the StrMixIRM as basically a logistic linear regression with interaction effects can promote the understanding of StrMixIRMs by providing an alternative conceptualization in addition to existing frameworks.

Caveats of the procedures discussed in this paper based on the `mirt` package include specifying multiple random starts to solve the local maxima issue prevalent in LCA. Moreover, it is recommended to check that the maximum likelihood can be replicated [47]. Whereas many software programs for LCA or mixture modeling have commands to streamline this process with detailed procedures, mixture modeling in its current form in `mirt` requires the user to set up and run repeated iterations of the model themselves, and then manually select the optimal final output. Thus, estimating the StrMixIRM with a large number of multiple starting points may require much more time than existing commercial software. For example, the average running time over 30 simulated datasets (see Section S4) was 2.73 min with `mirt` (including the computation of standard errors), while 0.45 min with `Mplus` on a laptop computer with 1.80 GHz Intel Core i5-8265U CPU processor and 8 GB of RAM. Note that these differences in the computing time might be due to the different estimation settings of the two programs.

A more prominent issue is with estimating the standard errors of model parameters. In addition to the fact that adding equality (and inequality) constraints in itself does not provide standard errors (e.g., StrMixIRM using linear equality constraints on item difficulties or the 2PL StrMixIRM with $s\tau_{hk}$), we noticed that the standard errors for all parameters from the `mirt` package were smaller than those from `Mplus` (see Section S1 in the Supplemental Document). We speculate this is due to different methods for standard error calculations, but additional investigation is needed to identify the source of the discrepancies as well as which standard error method might be more appropriate for the StrMixIRM.

Notwithstanding the multiple extensions of the StrMixIRM organized in this paper, they are by no means exhaustive. Other extensions are possible such as the inclusion of person predictors, additional item guessing parameters [17] and the StrMixIRM in multi-level [16] or longitudinal contexts [23,48]. A key advantage of our novel reparameterization is the flexibility to incorporate all of these suggestions, some of which are not possible in the original parameterization. For example, our new parameterization enables a researcher's hypothesis to incorporate an investigation of the variation in differentiation parameters across schools.

In fact, our parameterization using interaction effects of discrete latent variables need not be limited to the family of StrMixIRMs but can be applied many other models involving discrete latent variables, such as DCMs. Nonetheless, whether existing software, preferably an open-source one like the `mirt` package, can accommodate all aforementioned StrMixIRM extensions requires further investigation. It would be useful to explore the possibility of using other statistical programs such as SAS or Stata and Bayesian estimation platforms like Stan using the reparameterization, along with a comparison between the capabilities and advantages of each program. This would offer more options for users to help them conduct StrMixIRM analyses.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/psych6010023/s1>. The Supplemental Document consists of Section S1: Equivalence Between Model Parameterizations in mirt and Mplus, Section S2: Verbal Aggression Data: Multidimensional Rasch StrMixIRM in mirt, Section S3: Item Parameters in ECPE and Verbal Aggression Data Analyses, and Section S4: Simulation Study. Tables and code listings corresponding to each section are prefixed by S.

Author Contributions: Conceptualization, M.L., Y.S.S. and M.J.; methodology, M.L. and M.J.; software, M.L. and Y.S.S.; validation, M.L. and Y.S.S.; formal analysis, M.L. and Y.S.S.; writing—original draft preparation, M.L. and Y.S.S.; writing—review and editing, M.L., Y.S.S. and M.J.; visualization, M.L.; supervision, M.J.; project administration, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets used in this paper are publicly available in the R packages of GDINA and lme4 as stated in the paper. All R code and Mplus input and output files mentioned in the paper are available at <https://github.com/ysuh09/StrMixIRM>, accessed on 10 January 2024.

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

Abbreviations

The following abbreviations are used in this manuscript:

StrMixIRM	Structured mixture item response model
IRT	Item response theory
LCA	Latent class analysis
2PL	Two-parameter logistic
PCM	Partial credit model
GRM	Graded response model
DCM	Diagnostic classification model

References

- Formann, A.K. Linear logistic latent class analysis. *Biom. J.* **1982**, *24*, 171–190. [[CrossRef](#)]
- Formann, A.K. Linear logistic latent class analysis for polytomous data. *J. Am. Stat. Assoc.* **1992**, *87*, 476–486. [[CrossRef](#)]
- Rost, J. Rasch models in latent classes: An integration of two approaches to item analysis. *Appl. Psychol. Meas.* **1990**, *14*, 271–282. [[CrossRef](#)]
- Bolt, D.M.; Cohen, A.S.; Wollack, J.A. A mixture item response model for multiple-choice data. *J. Educ. Behav. Stat.* **2001**, *26*, 381–409. [[CrossRef](#)]
- Choi, I.H.; Wilson, M. Multidimensional classification of examinees using the mixture random weights linear logistic test model. *Educ. Psychol. Meas.* **2015**, *75*, 78–101. [[CrossRef](#)]
- Cohen, A.S.; Bolt, D.M. A mixture model analysis of differential item functioning. *J. Educ. Meas.* **2005**, *42*, 133–148. [[CrossRef](#)]
- Embretson, S.E. Mixed Rasch models for measurement in cognitive psychology. In *Multivariate and Mixture Distribution Rasch Models*; von Davier, M., Carstensen, C.H., Eds.; Springer: New York, NY, USA, 2007; pp. 235–253. [[CrossRef](#)]
- Mislevy, R.J.; Verhelst, N. Modeling item responses when different subjects employ different solution strategies. *Psychometrika* **1990**, *55*, 195–215. [[CrossRef](#)]
- Preinerstorfer, D.; Formann, A.K. Parameter recovery and model selection in mixed Rasch models. *Br. J. Math. Stat. Psychol.* **2012**, *65*, 251–262. [[CrossRef](#)]
- Robitzsch, A. Regularized mixture Rasch model. *Information* **2022**, *13*, 534. [[CrossRef](#)]
- Wallin, G.; Chen, Y.; Moustaki, I. DIF Analysis with Unknown Groups and Anchor Items. *Psychometrika* **2024**. [[CrossRef](#)]
- Bolt, D.M.; Cohen, A.S.; Wollack, J.A. Item parameter estimation under conditions of test speededness: Application of a mixture Rasch model with ordinal constraints. *J. Educ. Meas.* **2002**, *39*, 331–348. [[CrossRef](#)]
- Boughton, K.A.; Yamamoto, K. A HYBRID model for test speededness. In *Multivariate and Mixture Distribution Rasch Models*; von Davier, M., Carstensen, C.H., Eds.; Springer: New York, NY, USA, 2007; pp. 147–156. [[CrossRef](#)]
- Kim, N.; Bolt, D.M. A mixture IRTree model for extreme response style: Accounting for response process uncertainty. *Educ. Psychol. Meas.* **2021**, *81*, 131–154. [[CrossRef](#)]

15. Rupp, A.A.; Templin, J.; Henson, R.A. *Diagnostic Measurement: Theory, Methods, and Applications*; Guilford Press: New York, NY, USA, 2010.
16. Langi, M.; Jeon, M. Identifying and supporting academically low-performing schools in a developing country: An application of a specialized multilevel IRT Model to PISA-D assessment data. *Psychometrika* **2023**, *88*, 332–356. [CrossRef]
17. Jeon, M. A constrained confirmatory mixture IRT model: Extensions and estimation of the Saltus model using Mplus. *Quant. Method. Psychol.* **2018**, *14*, 120–136. [CrossRef]
18. Jeon, M. A specialized confirmatory mixture IRT modeling approach for multidimensional tests. *Psychol. Test. Assess. Model.* **2019**, *61*, 91–123. Available online: https://www.psychologie-aktuell.com/fileadmin/Redaktion/Journale/ptam-2019-1/06_Jeon.pdf (accessed on 5 January 2024).
19. Wilson, M. Saltus: A psychometric model of discontinuity in cognitive development. *Psychol. Bull.* **1989**, *105*, 276–289. [CrossRef]
20. Mislevy, R.J.; Wilson, M. Marginal maximum likelihood estimation for a psychometric model of discontinuous development. *Psychometrika* **1996**, *61*, 41–71. [CrossRef]
21. Dawson-Tunik, T.L.; Goodheart, E.A.; Draney, K.; Wilson, M.; Commons, M.L. Concrete, abstract, formal, and systematic operations as observed in a “Piagetian” balance-beam task series. *J. Appl. Meas.* **2010**, *11*, 1. Available online: <https://pubmed.ncbi.nlm.nih.gov/20351445/> (accessed on 5 January 2024).
22. Draney, K. The Polytomous Saltus Model: A Mixture Model Approach to the Diagnosis of Developmental Differences. Ph.D. Thesis, University of California, Berkeley, Berkeley, CA, USA, 1996. Available online: <https://www.proquest.com/openview/5ce8ad064e810d2ec9fbc6c1bd06530a/1?pq-origsite=gscholar&cbl=18750&diss=y> (accessed on 5 January 2024).
23. Jeon, M.; Draney, K.; Wilson, M. A general saltus LLTM-R for cognitive assessments. In *Quantitative Psychology Research: The 78th Annual Meeting of the Psychometric Society*; Millsap, R.E., Bolt, D.M., van der Ark, L.A., Wang, W.C., Eds.; Springer: New York, NY, USA, 2015; pp. 73–90. [CrossRef]
24. Fieuw, S.; Spiessens, B.; Draney, K. Mixture models. In *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*; de Boeck, P., Wilson, M., Eds.; Springer: New York, NY, USA, 2004; Chapter 11, pp. 317–340. [CrossRef]
25. Muthén, B.; Muthén, L. *Mplus User’s Guide*; Eighth Edition; Muthén & Muthén: Los Angeles, CA, USA, 2017. Available online: https://www.statmodel.com/download/usersguide/MplusUserGuideVer_8.pdf (accessed on 5 January 2024).
26. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2023.
27. Chalmers, R.P. mirt: A multidimensional item response theory package for the R environment. *J. Stat. Softw.* **2012**, *48*, 1–29. [CrossRef]
28. Chalmers, R.P. Three Parameterizations of Rasch Model. Available online: <https://philchalmers.github.io/mirt/html/Three-Rasch.html> (accessed on 25 December 2023).
29. Masters, G.N. A Rasch model for partial credit scoring. *Psychometrika* **1982**, *47*, 149–174. [CrossRef]
30. Samejima, F. Estimation of latent ability using a response pattern of graded scores. *Psychometrika* **1969**, *34*, 100. Available online: <https://psycnet.apa.org/record/1972-04809-001> (accessed on 5 January 2024). [CrossRef]
31. Jeon, M.; Draney, K.; Wilson, M.; Sun, Y. Investigation of adolescents’ developmental stages in deductive reasoning: An application of a specialized confirmatory mixture IRT approach. *Behav. Res. Methods* **2020**, *52*, 224–235. [CrossRef]
32. Dean, N.; Raftery, A.E. Latent class analysis variable selection. *Ann. Inst. Stat. Math.* **2010**, *62*, 11–35. [CrossRef]
33. Chalmers, R.P. Philchalmers/Mirt. Available online: <https://github.com/philchalmers/mirt> (accessed on 22 December 2023).
34. Templin, J.; Hoffman, L. Obtaining diagnostic classification model estimates using Mplus. *Educ. Meas.* **2013**, *32*, 37–50. [CrossRef]
35. Ma, W.; de la Torre, J. GDINA: An R package for cognitive diagnosis modeling. *J. Stat. Softw.* **2020**, *93*, 1–26. [CrossRef]
36. de Boeck, P.; Wilson, M. *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*; Springer: New York, NY, USA, 2004. [CrossRef]
37. Bates, D.; Mächler, M.; Bolker, B.; Walker, S. Fitting linear mixed-effects models using lme4. *J. Stat. Softw.* **2015**, *67*, 1–48. [CrossRef]
38. De La Torre, J. The generalized DINA model framework. *Psychometrika* **2011**, *76*, 179–199. [CrossRef]
39. Henson, R.A.; Templin, J.L.; Willse, J.T. Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika* **2009**, *74*, 191–210. [CrossRef]
40. Von Davier, M. A general diagnostic model applied to language testing data. *Br. J. Math. Stat. Psychol.* **2008**, *61*, 287–307. [CrossRef]
41. Templin, J.; Bradshaw, L. Hierarchical diagnostic classification models: A family of models for estimating and testing attribute hierarchies. *Psychometrika* **2014**, *79*, 317–339. [CrossRef] [PubMed]
42. Bock, R.D.; Aitkin, M. Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika* **1981**, *46*, 443–459. [CrossRef]
43. Chalmers, R.P. Numerical approximation of the observed information matrix with Oakes’ identity. *Br. J. Math. Stat. Psychol.* **2018**, *71*, 415–436. [CrossRef] [PubMed]
44. Oakes, D. Direct calculation of the information matrix via the EM. *J. R. Stat. Soc. Series. B Stat. Methodol.* **1999**, *61*, 479–482. [CrossRef]
45. Braeken, J.; Tuerlinckx, F.; De Boeck, P. Copula functions for residual dependency. *Psychometrika* **2007**, *72*, 393–411. [CrossRef]

46. Luo, J.; De Carolis, L.; Zeng, B.; Jeon, M. Bayesian estimation of latent space item response models with JAGS, Stan, and NIMBLE in R. *Psych* **2023**, *5*, 396–415. [[CrossRef](#)]
47. Sinha, P.; Calfee, C.S.; Delucchi, K.L. Practitioner’s guide to latent class analysis: Methodological considerations and common pitfalls. *Crit. Care Med.* **2021**, *49*, 63–79. [[CrossRef](#)]
48. Von Davier, M.; Xu, X.; Carstensen, C.H. Measuring growth in a longitudinal large-scale assessment with a general latent variable model. *Psychometrika* **2011**, *76*, 318–336. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.