*Article*

*Journal of Educational and Behavioral Statistics*
*Vol. 00, No. 0, pp. 1–34*
*DOI: 10.3102/10769986241289399*
*Article reuse guidelines: sagepub.com/journals-permissions*


# A Position-Sensitive Mixture Item Response Model

**Klint Kanopka**
*New York University*

**Benjamin W. Domingue**
*Stanford University Graduate School of Education*

*Standard item response theory (IRT) models are ill-equipped for when the probability of a correct response depends on the location in the test where an item is encountered—a phenomenon we refer to as position effects. Unmodeled position effects complicate comparing respondents taking the same test. We propose a position-sensitive IRT model that is a mixture of two item response functions, capturing the difference in response probability when the item is encountered early versus late in the test. The mixing proportion depends on item location and latent person-level characteristics, separating person and item contributions to position effects. We present simulation studies outlining various features of model performance and end with an application to a large-scale admissions test with observed position effects.*

Keywords: *psychometrics; item response theory; machine learning*

In typical item response theory (IRT) applications (e.g., the standard one-parameter logistic [1PL], two-parameter logistic [2PL], and three-parameter logistic [3PL]), two core assumptions make estimation possible. First, respondent abilities and item parameters are stable over the course of a single test. Second, item parameters do not vary from person to person. While the differential item functioning literature focuses on addressing between group bias in item parameters (Camilli et al., 1994), standard IRT models are ill-equipped for scenarios wherein the probability of a correct response exhibits a dependence on the location in the test where an item is encountered—a phenomenon that we broadly refer to as *position effects*. In situations where items do not change position between respondents, position effects are not a concern, as there is no variation in item position to induce differences in item behavior for different respondents. This can be a scenario where a test consists of only a single form or the items shared between forms appear in the same location. This breaks down when tests use multiple forms that share items or with the use of computer adaptive testing (CAT) designs where items are selected from a bank on the fly. Unmodeled position effects make observed performance contingent upon the

specific test form a respondent is exposed to, inducing bias in individual ability estimates. This introduces a potential liability in our attempt to measure ability, directly compare students taking the same test, and make consequential inferences from test scores.

Position effects are, by no means, a new concern. While Zeller et al. (2017) speculate that position effects can be attributed to ordering items by increasing difficulty, others have found that item parameter estimates are not always stable across item positions (Leary & Dorans, 1982; Meyers et al., 2008). Kingston and Dorans (1984) also observe a dependence of estimated item parameters on item position in GRE items. Their recommendation—based on the assumption that observed position effects are due to practice effects or fatigue—is for more flexible models with parameters that incorporate item familiarity or position. Debeer and Janssen (2013) present a framework by which IRT models can begin to account for position effects. An additional complication, noted by Weirich et al. (2017), is that item-induced position effects also interact with individual effort during a test, implying some change in response behavior. Speaking to changes in response behavior, Domingue et al. (2021) observe a change in the relationship between an individual's response speed and response accuracy over the course of the large computer adaptive test that persists when controlling for the effects of specific items. Both of these point to the idea that position effects are not a phenomenon localized entirely within items or persons, despite much of the previous attempts at modeling position effects treating them as such.

Previous item-focused modeling work has attempted to address this through the use of more flexible item response models, including the linear logistic test model (Hohensinn et al., 2008) and explanatory item response models (De Boeck, 2004). The tacit assumption in these approaches, however, is that position effects are born from an interaction between an item and its position, and all respondents experience the same degree of parameter displacement. While this approach preserves the assumption that individual ability is stable over the course of a single test, it runs contrary to the idea that individuals may vary in their willingness and ability to persist in test-taking and provide consistent effort, as noted by Feather (1962). As each different test form is both a sample and permutation of possible items, the presence of individual differences complicates the counterfactual reasoning that underlies the equating procedure (Tucker-Drob, 2011).

In parallel to the research on modeling item position effects, a separate line of work allows for modeling between-person variability in position effects (Albano, 2013; Trendtel & Robitzsch, 2018; Weirich et al., 2014). These models impose the assumption that individual differences in position effects are linear functions of item position. Another approach is to treat person ability as dynamic and model it over time (Shanker Tripathi & Domingue, 2019), but the data burden at each time point is more suited to modeling within-person ability

changes across multiple tests as opposed to ability changes within a single test. A more tractable model estimates a person-specific ability decay, which is computationally similar to the explanatory IRT approach suggested above (De Boeck, 2004). There are two trade-offs with the general person-side approach: First, it assumes that position effects are uniform in the same direction, that is to say, that items all get harder or easier as the test progresses. Second, it implies that the magnitude of a position effect does not depend on features of the item (i.e., item content), contrary to some empirical observations (Kingston & Dorans, 1984).

The one-sided nature (i.e., position effects are entirely due to items or people) of these solutions does not comport with observed effort moderation of position effects (Weirich et al., 2017), nor with evidence of within-person variation in response time (Domingue et al., 2021). Thus, encountering an item early as opposed to late may have a significant impact on how the item is perceived by the test taker, manifesting as a dependence of estimated item parameters on the location in which the item was encountered. This dependence comes hand in hand with significant between-person variability in the rate at which these response processes evolve. Such heterogeneity would be especially troublesome in CAT settings, where items are selected and delivered based on pre-calibrated item parameters. These facts suggest that solutions allowing for variation of functioning across both items and people may be required.

It is important to recognize that the presence of position effects does not always necessitate the modeling of position effects, but there are downstream consequences. In the case of a few test booklets taken by multiple students that exhibit position effects, differences in estimated ability distributions can, in practice, be reconciled using equating. If there are individual differences in item position effects, however, differences in item position may distort estimated ability scales enough to make between booklet comparisons and rank ordering impossible. Computer adaptive tests only exacerbate this problem, as large item banks and variable item positions can result in only a single student encountering a specific arrangement of test items. In this case, the use of pre-calibrated item parameters can induce potentially large amounts of error in the measurement of person ability, comprising a significant threat to validity in high-stakes testing scenarios. As such, explicit modeling of position effects may be required to appropriately compare test takers who are exposed to different items in different locations.

To address these shortcomings and reconcile observed position effects with the notion of an evolving response process, we propose a novel position-sensitive IRT model that is a mixture of item response models. Our goal is to construct an interpretable model of position effects that produces comparable ability estimates while also allowing for item- and person-level variation in position effects. We first describe the model and its estimation. Next, we

present a series of simulations to demonstrate both parameter recovery and the behavior of the model in the absence of position effects. We then turn to an empirical application using a large college admissions test from Brazil. Here, we demonstrate how the model can be applied to new data and show that it can (a) allow for a single IRT model to be estimated that accounts for differences in test difficulty due to item ordering and (b) shed light on the behavior of items across different positions in a test booklet.

## Model

As our approach is based upon IRT, we begin with a brief overview of the key constituent models that will also assist in the introduction of notation. As used here, IRT models the probability that an individual $j$ responds correctly to an item $i$ as a monotonically increasing function of a person-side latent ability, $\theta_j$, and a set of item-side parameters. In its simplest form, IRT uses a single item parameter, $b_i$, and a logistic link function to describe the probability of correct response. In this 1PL model, the probability depends solely on the difference between the individual's latent ability and the item's difficulty, $b_i$. Taking $\sigma(z)$ to be the standard logistic sigmoid function,

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

We can simplify the notation for the 1PL item response function (IRF):

$$P\left(X_{ji} = 1|\theta_j, b_i\right) = \sigma(\theta_j - b_i). \tag{1}$$

More complex versions of the IRF exist, including formulations that adjust the rate at which the function increases or modifies the lower and upper asymptotes. We call specific attention to the version that estimates the rate at which the response probability increases: the 2PL) model, which adds a discrimination parameter, $\alpha_i$, that describes how well an individual item discerns between high ability and low ability respondents:

$$P\left(X_{ji} = 1|\theta_j, \alpha_i, b_i\right) = \sigma\left(\alpha_i\left(\theta_j - b_i\right)\right). \tag{2}$$

### *Mixture Item Response Models*

Mixture models are not new to IRT applications (Sen & Cohen, 2019) but have increased in popularity and usage as access to computing resources has become more common and less expensive (von Davier & Rost, 2018). Rost (1990) originally proposed the IRT mixture model as a combination of latent class analysis and the Rasch model. Work by Yamamoto (1995) recognized that

two different item response models may be required in the mixture. This model, called the HYBRID model, has been used in tests with time pressure where a respondent may switch from an effortful response process to a random response process as time runs down. Importantly, this work models a specific type of position effect as a change in the respondent's cognitive process. One particular extension, the speededeness model Yamamoto and Everson (1995), estimates points in the item sequence where respondents switch from a Rasch IRF to a random guessing IRF and has been used by others, including Bolt et al. (2002).

Our model follows this tradition by using a mixture of two IRFs meant to separately capture early and late test response processes. Most mixture IRT models (including the HYBRID model and its extensions) are *discrete* mixtures, meaning that individual item responses are assumed to belong to a specific group, and the mixing parameters estimate a probability of group membership. We instead conceptualize the overall IRF as a continuous combination of two response modes, as in the continuous HYBRID (C-HYBRID) extension (Nagy & Robitzsch, 2021). We utilize the same functional form imposed on the mixing parameters that Nagy and Robitzsch (2021) use, but instead conceptualize the two types of response behavior the model combines as both dependent upon the respondent's latent ability. These response behaviors are described as early- and late-test IRFs. This allows for a person to experience variation in response probability that depends on where an item is encountered. The mixing proportions of the two models, as in the C-HYBRID model, are functions of both the location of the item in the test booklet and person-level characteristics. As such, our method diverges from the discrete mixture tradition and may be considered an extension of the C-HYBRID model. Our work diverges from the C-HYBRID model in that our second response behavior also depends upon person ability, which may be more reasonable in high-stakes testing situations or places where test speededness is not a significant constraint on individual performance.

### *Incorporating Position Sensitivity*

We allow for person- and item-level heterogeneity in position effects via a model that presupposes position effects have both person-side and item-side components. We do this by constructing an IRF that is a mixture of two kernel IRFs, one modeling the person-item interaction had it occurred *early* in the test (indexed by a subscript $\alpha$, the first letter of the Greek alphabet), while the other models the person-item interaction had it occurred *late* in the test (indexed by a subscript $\omega$, the last letter of the Greek alphabet). Each of these IRFs is allowed to have its own set of estimated item parameters (denoted $\mathbf{b}_{\alpha i}$ and $\mathbf{b}_{\omega i}$). We use a mixing parameter, $\pi_{ji}$, to allow the mixture of these two kernel IRFs, denoted $\psi$, to vary smoothly as a function of the difference between an item's position and an estimated person-side parameter

$$P\left(X_{ji} = 1|\theta_j, \mathbf{b}_{\alpha i}, \mathbf{b}_{\omega i}, \pi_{ji}\right) = \pi_{ji}\psi_{\alpha}(\theta_j, \mathbf{b}_{\alpha i}) + \left(1 - \pi_{ji}\right)\psi_{\omega}(\theta_j, \mathbf{b}_{\omega i}). \quad (3)$$

The simplest practical version of Equation 3 uses the 1PL IRF in Equation 1 as the kernel. In that case, $\psi_{\alpha}\theta_i, \mathbf{b}_{\alpha i}) = \sigma\theta_j - b_{\alpha j})$ and $\psi_{\omega}\theta_i, \mathbf{b}_{\omega i}) = \sigma\theta_j - b_{\omega j})$. We write the full probability that student $j$ responds correctly to item $i$ as:

$$P\left(X_{ji} = 1|\theta_j, b_{\alpha i}, b_{\omega i}, \pi_{ji}\right) = \pi_{ji}\sigma(\theta_j - b_{\alpha i}) + \left(1 - \pi_{ji}\right)\sigma(\theta_j - b_{\omega i}), \quad (4)$$

where $X_{ji} = 1$ if student $j$ responds correctly to item $i$ (otherwise $X_{ji} = 0$) and $\theta_j$ is student $j$'s latent ability. Note that there are two item difficulty parameters, $b_{\alpha i}$ and $b_{\omega i}$. In the first term on the right-hand side, $b_{\alpha i}$ is item $i$'s difficulty when encountered at the beginning of the test. In the second term, $b_{\omega i}$ is item $i$'s difficulty when encountered at the end of the test.

Note the treatment of the dependence of response probability on position: we continue to assume person ability is stable over the course of a single test. Thus, an individual does not have corresponding early and late abilities, evidenced by the presence of only a single $\theta_j$. Between-person differences in the transition from this early test state to late test state are captured in $\pi_{ji}$ and are modeled to not exhibit a dependence on the person ability, $\theta_j$. We specify $\pi_{ji} \in (0, 1)$ so that it is high when the respondent is in an early test response state and low when in a late test response state. To achieve this, we impose a theory-driven functional form on $\pi_{ji}$. First, we require that the value of $\pi_{ji}$ is monotonically decreasing for an individual as they progress through the test (response process progression does not backtrack). Inspired by the 2PL IRF, we write $\pi_{ji}$ as

$$\pi_{ji} = \sigma\left(c\left(k_j - s_{ji}\right)\right), \quad (5)$$

where $s_{ji} \in [0, 1]$ is the *sequence number*, or the position in the test where student $j$ encountered item $i$. If the response from student $j$ to item $i$ is the $n$-th recorded response for that student, we set $s_{ji} = (n - 1)/\left(N_j - 1\right)$ where $N_j$ is the total number of responses for the student. Thus, the first item they encounter occurs at $s_{ji} = 0$, and the final item they encounter occurs at $s_{ji} = 1$, allowing for variation in the number of items each respondent is exposed to.

Next, $k_j$ is a person-side parameter corresponding to the position in the test student $j$ is equally likely to be exhibiting early test and late test response behavior. The $c$ parameter then governs the rate of transition from early test to late test response behavior and may be interpreted as the change in the log odds of exhibiting early test behavior over the course of the entire test (recall that $s_{ji}$ is rescaled to the interval $[0, 1]$). This is identical to the ''process discrimination'' parameter, $\lambda$, from the C-HYBRID model of Nagy and Robitzsch (2021). Importantly, these parameters can only be interpreted in the context of differences in item parameters. In cases where items become *more* difficult the later they are encountered, $k_j$ can be interpreted as related to endurance or cognitive stamina. In cases where items become *less* difficult, $k_j$ corresponds to how

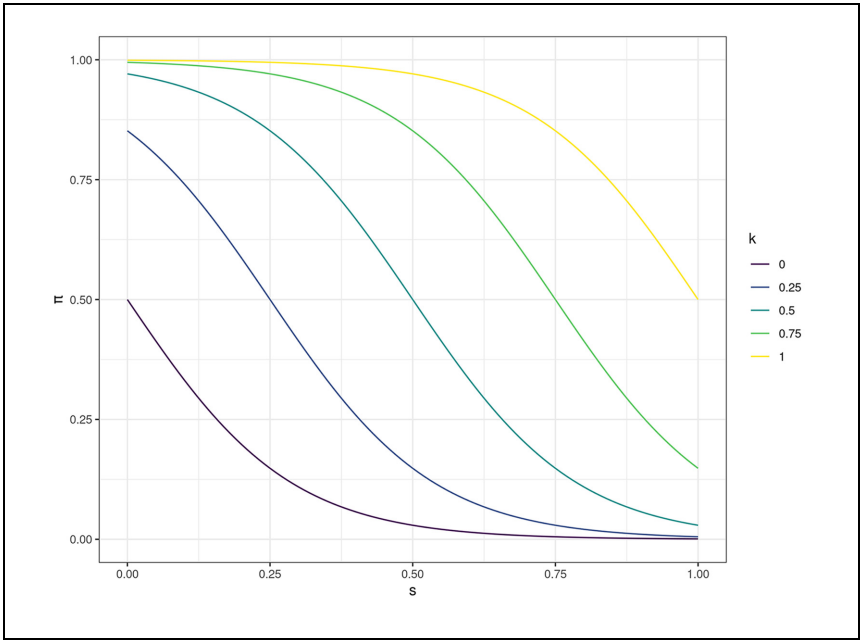


FIGURE 1. *The evolution of* $\pi_{ji}$ *as a function of item position for a variety of* $k_j$ *values. Recall that* $\pi_{ji}$ *is interpreted as the proportion of early test IRF in the mixture and is, by construction, monotonically decreasing as a function of item position.*
*Note.* IRF = item response function.

quickly individuals acclimate to the items or the rate at which practice effects accumulate. If early and late test discriminations are estimated (with a 2 + parameter kernel IRF), these parameters can also relate to how quickly respondents ''get in the groove'' or disengage from the test. Figure 1 shows how $\pi_{ji}$ can vary as a function of item position. Here, each line has the same $c$, but a separate $k_j$ value. Recalling that higher values of $\pi_{ji}$ correspond to higher proportions of early test response behavior in the mixture, we observe for higher values of the person parameter $k_j$, the respondent preserves primarily early test behavior further into the test. Here, we also emphasize that the magnitude of the position effects are captured in the difference between early and late test difficulties, $\delta = b_\omega - b_\alpha$. Figure 2 shows the predicted response probability for three different items with different $\delta$ values, encountered in different test positions, $(s_{ji})$. The different lines demonstrate how Equation 4 responds to changes in item position, with the vertical spread from $(s_{ji}) = 0$ to $(s_{ji}) = 1$, demonstrating the maximum magnitude of observed position effects. Looking at separations of $\delta \in \{0.25, 0.5, 1\}$ across the three panels, we see that about

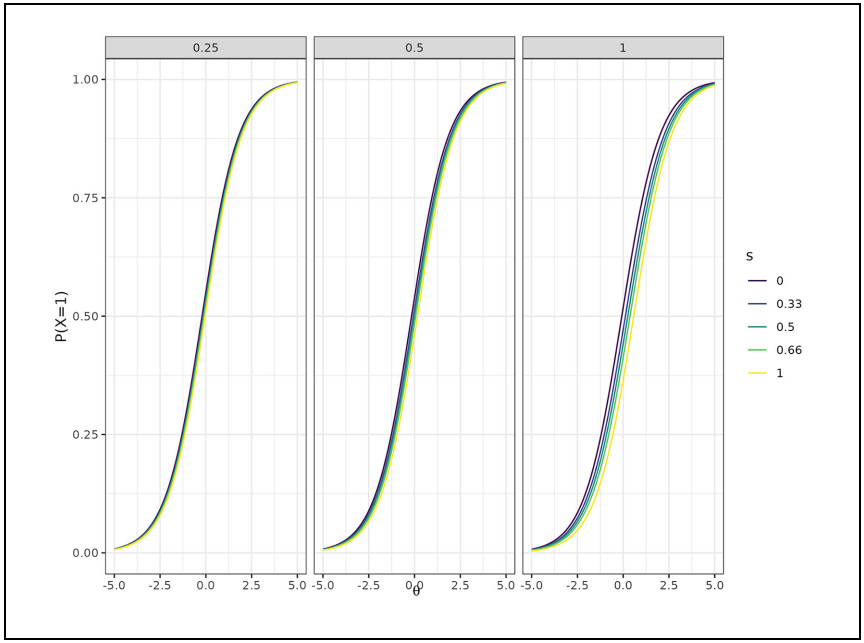


FIGURE 2. *Example IRF for an item with early test difficulty* $b_\alpha = -0.25$ *and late test difficulty* $b_\omega \in \{0, 0.25, 0.75\}$. *Each panel is labeled by the difference in difficulties,* $b_\omega - b_\alpha$. *All panels show IRFs generated with* $c = 3$ *and* $k_j = 0.5$. *Each line shows the probability of correct response as a function of the item position,* $s_{ji}$, *giving a sense of the degree to which the difficulty parameter separation impacts the probability of correct response. Note that because* $b_\omega > b_\alpha$, *the item is perceived as harder the further into the test it is encountered.*
*Note.* IRF = item response function.

$\theta = 0$, the differences in the probability of correct response from the first item to the final item are $-0.04, -0.08, -0.15$, respectively. These differences show how the model parameters encode the magnitude of the observed position effects for each item and demonstrate that for $\delta = 0$, items exhibit no position effects.

## *Estimation*

The estimation of more flexible models with more parameters naturally comes at a computational cost. We begin by benchmarking the estimation challenge for the position-sensitive model relative to conventional alternatives. Recall that this approach estimates two full sets of item parameters for each item

and two parameters per person. As such, the estimation of a unidimensional position-sensitive IRT model with a $K$-PL kernel for a test with $N$ items and $M$ respondents involves, at most, $2(K \times N + M) + 1$ parameters. Importantly, we note that the overall complexity of $O(KN + M)$ and the number of parameters estimated is linear in the number of items and persons, making it comparable to the number of parameters estimated by unidimensional and multidimensional IRT models.

Traditionally, IRT software developers tend to prefer a marginal maximum likelihood (MML) estimation procedure, often using the expectation maximization (EM) Algorithm (Bock & Aitkin, 1981) with quadrature points, as it produces asymptotically consistent estimates of item parameters. Markov Chain Monte Carlo (MCMC) estimation is also a common and flexible alternative, though run times can be much longer than EM-based approaches. As we suggest, the use of our model for adaptive tests that may have large numbers of respondents and huge item banks, we propose the use of advances in optimization driven by the deep learning literature. While framing the problem in terms of joint maximum likelihood (JML) estimation is not guaranteed to produce asymptotically consistent estimates of item parameters for all IRT models, it is computationally efficient to implement, converges faster than MCMC, and produces good results even in high dimensional scenarios (Chen et al., 2019). Similar to the constrained JML procedure implemented by Chen et al. (2019), we impose an $\ell_2$ regularization penalty (also known as a ridge penalty; see Hastie, 2020) on the likelihood when estimating person parameters. This is analogous to, though slightly different from, treating person parameters as random effects (see De Boeck, 2004, for more information on person-side random effects in the estimation of IRT models). Importantly, this helps combat the divergence of estimated parameters in the presence of respondents of extremely high or low ability and can work to stabilize the estimation of item parameters. After item parameters are estimated, person parameters are then estimated using unpenalized maximum likelihood estimation, as is done in MML implementations. Given this, our software implementation uses a JML approach implemented in Python (Van Rossum & Drake, 2009) and PyTorch (Paszke et al., 2017). This allows for multiple benefits. First, we take advantage of PyTorch's automatic differentiation. This allows the same basic optimization framework to be quickly applied to increasingly baroque kernel IRFs. Additionally, we take advantage of the Adam optimizer[1] (Kingma & Ba, 2014). Adam is an implementation of stochastic gradient descent (SGD), where optimization steps are taken along gradients with respect to individual observations (as opposed to the full dataset, as in vanilla gradient descent). In theory, this allows for more frequent, but noisier, parameter updates that may decrease convergence time. In practice, we implement a *minibatched* version of Adam, where optimization steps are taken along gradients with respect to a subset of the data, increasing

the number of parameter updates per iteration through the full training data relative to full batch gradient descent, while also reducing the variance in each individual update relative to SGD. Adam also uses a variable learning rate, meaning that the algorithm makes large updates early in fitting and reduces the size of individual steps as it approaches convergence. This adaptively balances fitting speed with numerical precision. Regularization is also built into the PyTorch implementation. The key advantage of Adam is the use of momentum (see Qian [1999] to further smooth update steps and accelerate model fitting. This is done using moving averages of the gradient to adjust individual parameter updates (for convergence proofs and remarks on limitations, see Défossez et al., 2020; Reddi et al., 2019). Finally, our PyTorch implementation allows computation to be offloaded to a graphical processing unit (GPU) if available, which more efficiently handles the repeated matrix multiplication operations used in gradient-based optimization.

To ensure model identification, we assume that all person parameters, namely $\theta_j, k_j$, are normally distributed. This is enforced, computationally, by the introduction of $\ell_2$ regularization during the person parameter update step. This is similar to treating person parameters as random effects, in that person parameters are shrunk toward their means, but different in how much shrinkage is applied. Random effects use the strength of evidence, with more evidence inducing less shrinkage, to determine the amount of shrinkage. Regularization, on the other hand, allows users to set the amount of shrinkage they desire. Additionally, we also implement item centering by enforcing that early test difficulties are mean zero $\left(\bar{b}_\alpha = 0\right)$.

We additionally make two reparametrizations at the software level. In the first, we replace the late test difficulty with the early test difficulty plus the offset,

$$b_{\omega i} = b_{\alpha i} + \delta_i, \tag{6}$$

as in testing, we found this change to result in more stable parameter recovery and faster convergence. The second change we make, borrowed from Nagy and Robitzsch (2021), involves the reparametrization of the functional form of $\pi_{ji}$. Specifically, we modify Equation 5 by rescaling $k_j$ as follows:

$$\pi_{ji} = \sigma\left(c\left(k_j - s_{ji}\right)\right) = \sigma\left(\bar{k} + k_j^\star - c \cdot s_{ji}\right).$$

This allows for two main benefits. First, $\bar{k}$, the mean of the $k_j$ distribution, and $c$ are estimated alongside item parameters. This leaves only the individual-level variation from the whole-test behavior, $k_j^\star$, to be estimated alongside person parameters. Since $\bar{k}$ is the mean of the rescaled $k_j$ distribution, $k_j^\star$ is now mean zero, allowing it to be properly shrunk by $\ell_2$ regularization in the person-side update step. Additionally, this also leads to more stable parameter recovery and faster convergence.

One key challenge with estimation comes with the selection of starting parameters. Specifically, if initial item parameters are not offset (i.e., such that $|b_{\alpha i} - b_{\omega i}| > \delta$ for some $\delta$) during early rounds of model fitting, the model may not learn to separate the parameters unless there are a large number of respondents, creating a situation analogous to the cold-start problem in recommender systems (Lam et al., 2008). We solve this problem by using a fixed offset of $\delta_i = 0.5$ for the starting value of parameters across all items. While not the most computationally efficient solution, merely specifying *some* initial separation of item parameters performs well even in the case when $\delta_i \leqslant 0$. For other initializations, we specify software defaults to be

$$\theta_j \sim \mathcal{N}(0, 1)$$
$$k_j^{\star} = b_{\alpha i} = 0$$
$$\delta_i = \tau = 0.5$$
$$c = 1$$

The Python software implementation we have developed is invoked from the command line and allows users to specify kernel IRF, maximum number of parameter update step alternations, maximum number of iterations within each parameter update step, batch size, learning rates for item and person parameter update steps separately, amount of regularization for person parameter update steps, and convergence thresholds. Additionally, it will autodetect the availability of a GPU for use in estimation.

## Method

Here, we present two simulation studies and one empirical example. The simulation studies probe two specific aspects of model behavior: parameter recovery under a known data-generating model and model behavior in the absence of position effects. The empirical example illustrates how the model can be applied to analyze data with observed position effects but an unknown data-generating process.

### *Study One*

We first begin by simulating data from the position-sensitive model with a 1PL kernel to demonstrate parameter recovery under a variety of conditions. We simulate data with $N_{\text{persons}} \in \{50, 100, 500, 1{,}000\}$ respondents and $M_{\text{items}} \in \{20, 50, 100\}$ items. All conditions are fully crossed with $N_{\text{replications}} = 100$ replications within each condition. As such, each plot below contains the results of $N_{\text{persons}} \times M_{\text{items}} \times N_{\text{replications}} = 1{,}200$ simulations. Each is fit using a convergence threshold of $\varepsilon = 0.001$, a fixed learning rate of 0.05,

batch size of 64, and a regularization parameter of $\lambda = 10^{-5}$. Note that these will impact both convergence speed and precision and, in practice, ought to be tuned to a specific application.

Within each simulation condition, we first generate item parameters. Referencing Equation 6, we first construct test and item parameters by drawing the response behavior transition, early item difficulties, and difficulty offsets such that

$$
\begin{aligned}
c &\sim \mathcal{N}\left(1, 0.4^2\right) \\
b_{\alpha i} &\sim \mathcal{N}(0, 1) \\
\delta_i &\sim \mathcal{N}\left(0.5, 0.5^2\right).
\end{aligned}
$$

We then draw person parameters as

$$
\begin{aligned}
\theta_j &\sim \mathcal{N}(0, 1) \\
k_j &\sim \mathcal{N}\left(0, 0.2^2\right).
\end{aligned}
$$

Additionally, we apply the restrictions that $c \in [0.25, 1.75]$ and $k_j \in [0, 1]$. Next, we generate the sequence numbers, $s_{ji}$, by assuming each respondent responds to each item but is exposed to it in a random order. We do this by shuffling the sequence from $\{1, \ldots, M_{\text{items}}\}$ and then rescaling them to be on the unit interval, $[0, 1]$. Finally, we generate individual responses by making individual Bernoulli draws with the probability of correct response, $P\left(X_{ji} = 1\right)$, according to Equation 4.

*Item Parameter Recovery.* We first look at the ability of the model to recover the item parameters, $b_{\alpha i}, b_{\omega_i}$. Note that this recovery is in line with the specification of Equation 4, not the way the model is estimated in software, nor how the simulated data are generated. We first compute the root mean squared error (RMSE) within each replication across all estimated item difficulties (both early and late, indexed by $t$), computed as

$$
\text{RMSE} = \sqrt{\frac{1}{2M} \sum_{t \in \{\alpha, \omega\}} \sum_{i=1}^{M} (\hat{b}_{ti} - b_{ti})^2}, \tag{7}
$$

where $M$ is the number of items. Note that because each item has two difficulties, we average over $2M$ parameters. In Figure 3, we show the average RMSE from the 100 replications with a 95% confidence interval along the $y$-axis. Along the $x$-axis, we show the number of respondents. Along the panels, we vary the number of items. In general, we see that as the number of respondents increases, the RMSE in the recovery of item parameters decreases, with the exception of the $N = 1{,}000, M = 20$ cases This discrepancy is not necessarily

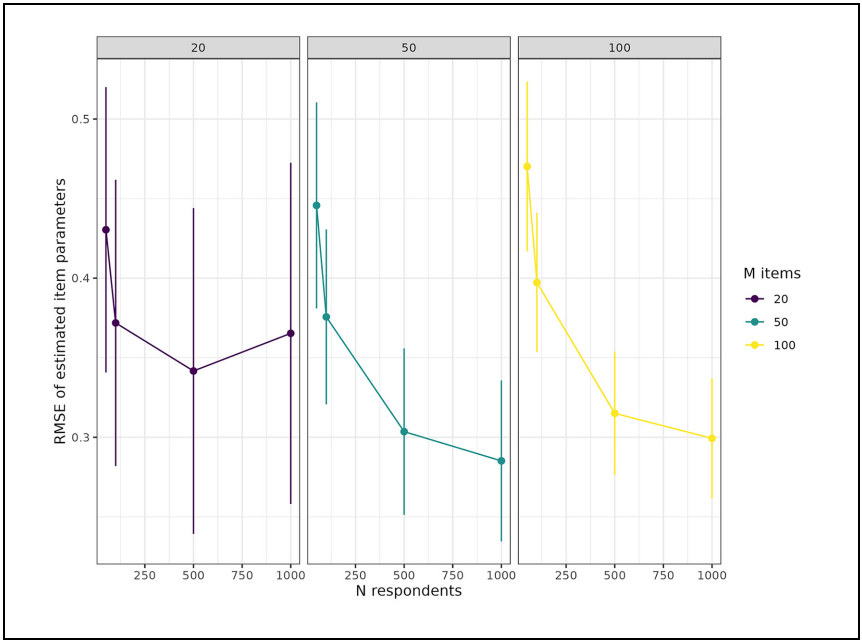


FIGURE 3. *RMSE for item parameter recovery. RMSE is displayed on the* y*-axis, number of respondents along the* x*-axis, and number of items vary across panels and colors. Each point is the mean and 95% confidence interval derived from 100 replications. We see a general relationship whereby more respondents produce lower RMSE.*
*Note.* RMSE = root mean squared error.

problematic, as the RMSE can be driven down further using a different convergence threshold and degree of person parameter regularization for each simulation condition. Additionally, we see that RMSE is lower for $M = 50$ than it is for $M = 100$. As good item parameter estimation requires observing items in a variety of positions, this is the first indication that the number of respondents plays an extremely important role in parameter estimation. In general, more items require an increasing number of respondents to estimate an appropriate separation between early and late test difficulties, a consideration we will explore shortly.

Figure 4 is the same as above, but now with the number of items along the *x*-axis and number of respondents varied along panels and colors. Here, we see that as the number of respondents increases, RMSE decreases. Despite this, however, for a fixed number of respondents, the RMSE is not monotonically

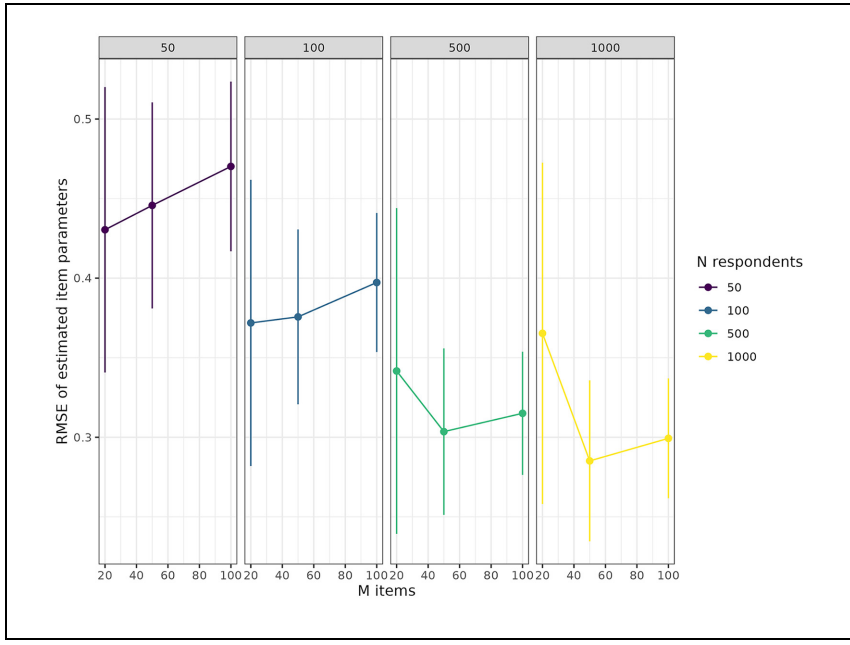


FIGURE 4. *RMSE for item parameter recovery. RMSE is displayed on the* y*-axis, number of items along the* x*-axis, and the number of respondents vary across panels and colors. Each point is the mean and 95% confidence interval derived from 100 replications. We see a general relationship whereby more respondents produce lower RMSE.*

*Note.* RMSE = root mean squared error.

decreasing with the number of items, pointing to a more complex dependency between hyperparameter selection, test length, and the number of respondents for parameter recovery.

Figure 5 shows bias in parameter recovery along the *y*-axis, computed within each replication as

$$\text{Bias} = \frac{1}{2M} \sum_{t \in \{\alpha, \omega\}} \sum_{i=1}^{M} \hat{b}_{ti} - b_{ti}, \tag{8}$$

where $M$ is the number of items and we again average over $2M$ parameters. The number of respondents varies along the *x*-axis and the number of items varies across panels and colors. Here, we see a story similar to the one above. For small numbers of items $(M = 20)$, especially with large numbers of respondents, bias

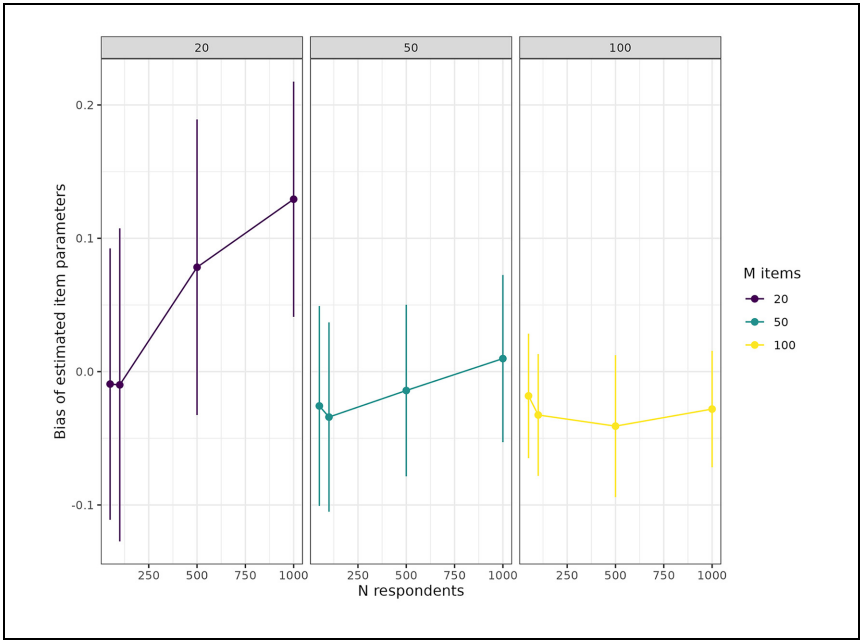


FIGURE 5. *Bias for item parameter recovery. Bias is displayed on the* y*-axis, the number of respondents along the* x*-axis, and the number of items vary across panels and colors. Each point is the mean and 95% confidence interval derived from 100 replications. We see that, in most cases, bias is near zero.*

is unacceptable. For larger numbers of items, however, bias is generally near zero. Figure 6 shows bias with the number of items along the *x*-axis and the number of respondents varying across panels and colors. Here the relationship is less clear-cut, though the magnitude of overall bias is small and nearly all of the 95% confidence intervals still cover zero.

A natural question is to wonder what is the source of the errors in parameter recovery may be. Figures 7 and 8 show RMSE and bias on the *y*-axis, respectively, with the number of items along the *x*-axis. Note that RMSE and bias are averaged over *M* parameters within each replication, as the summation over *t* in Equations 7 and 8 is no longer present. Now, the number of respondents varies along horizontal faceting and colors, while the vertical faceting splits between early and late test difficulty. When comparing RMSE for early and late test difficulties in Figure 7, we see that the RMSE is lower for early test parameters than for late test parameters. This points to issues with parameter recovery being more closely tied to the model being unable to properly separate the late test

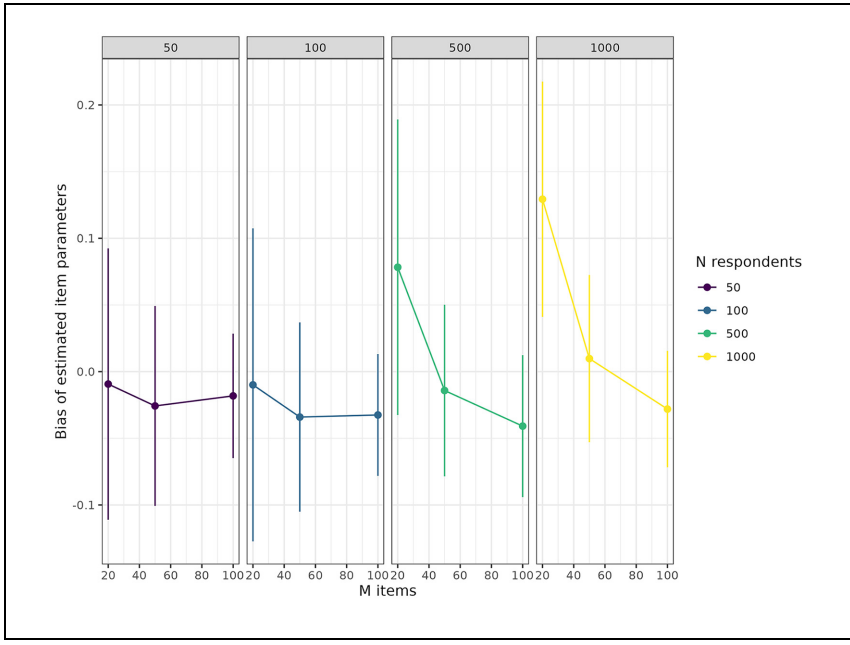


FIGURE 6. *Bias for item parameter recovery. Bias is displayed on the* y*-axis, the number of items along the* x*-axis, and the number of respondents vary across panels and colors. Each point is the mean and 95% confidence interval derived from 100 replications. We see that, in most cases, bias is near zero.*

difficulty parameters from the early test difficulties, as opposed to the location of items on the scale. This is especially problematic for the $N = 1{,}000$ respondents $M = 20$ items condition, where the late test difficulty RMSEs are extremely high. This problem can be reduced by reducing convergence thresholds and increasing the amount of regularization in the person parameters. Figure 8 shows bias in the same way. First, note that universally, bias in early test difficulties is near zero. For smaller numbers of items and higher numbers of respondents, late test difficulties are overestimated. That is, the model overseparates difficulties, which can be reduced by increasing the amount of regularization as the number of respondents increases. For higher numbers of items or lower numbers of respondents, the model tends to underestimate late test difficulties, underseparating difficulty parameters.

*Person Parameter Recovery.* Next, we turn to the recovery of the person parameters, $\theta_j$ and $k_j$. Recall that during model fitting, item parameters are

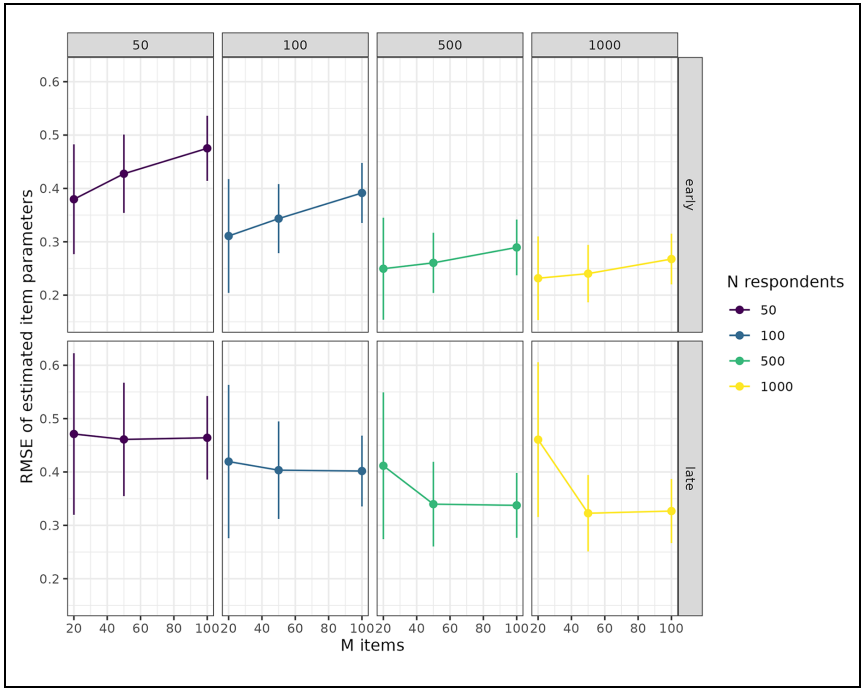


FIGURE 7. *RMSE for item parameter recovery, split by early and late test parameters. RMSE is displayed on the* y*-axis, the number of items along the* x*-axis, and the number of respondents vary horizontal facets and colors. The top row is early test difficulties, while the bottom row is late test difficulties. Each point is the mean and 95% confidence interval derived from 100 replications. In general, RMSE is lower for early test parameters than late test parameters, indicating that while estimated difficulties are correctly located, separation is a larger source of previously observed error.*
*Note.* RMSE = root mean squared error.

estimated directly while person parameters are subject to $\ell_2$ regularization, with person parameters being re-estimated at the end without regularization. This is because while regularization will stabilize the estimation of the item parameters, it will necessarily bias the estimates of person parameters. During this final stage of estimation, we found that 140 of 1,200 replications were subject to diverging estimates of $k_j$ for some respondents, classified by replications where $\text{RMSE}_k > \sqrt{10}$. As such, we have excluded these 140 replications from the following analysis. Table 1 shows the distribution of excluded replications. Notice that these occur almost exclusively for simulation conditions with large numbers of respondents. This may be due to a higher likelihood of observing extreme

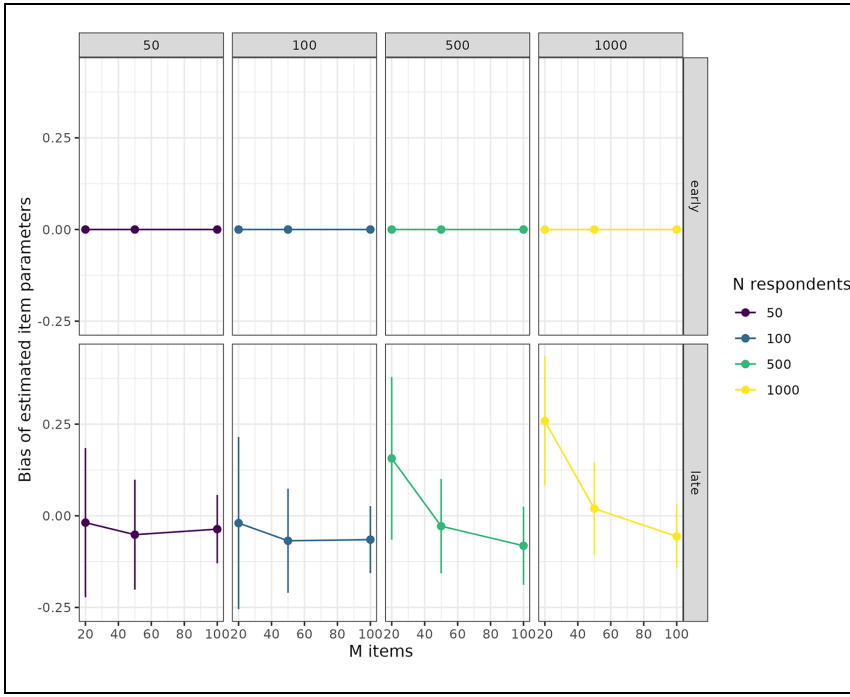


FIGURE 8. *Bias for item parameter recovery, split by early and late test parameters. Bias is displayed on the* y*-axis, the number of items along the* x*-axis, and the number of respondents vary horizontal facets and colors. The top row is early test difficulties, while the bottom row is late test difficulties. Each point is the mean and 95% confidence interval derived from 100 replications. For early test parameters, bias is essentially zero. For late test parameters, however, estimates tend to be downwardly biased unless there are smaller numbers of items and high numbers of respondents, resulting in difficulties that are underseparated.*

values of $k_j$ with more respondents. As such, putting software thresholds on values of $k_j$ is likely required to solve this behavior.

First, we look to the recovery of the person ability, $\theta_j$. Figure 9 shows RMSE on the *y*-axis, the number of respondents on the *x*-axis, and the number of items varying across panels and colors. Here we see that the primary driver of recovery error is the number of items, with RMSE for ability recovery being relatively stable for a given number of items. We see the variation in RMSE shrinking as the number of respondents increases, likely due to improved item parameter estimation. Next, we look at Figure 10, which shows bias on the *y*-axis. Here, we see that while bias is generally low, abilities are typically

TABLE 1.
*Number of Excluded Simulations by Condition*

| $M_{\text{items}}$ | $N_{\text{persons}}$ | $N$ excluded |
|---|---|---|
| 20 | 50 | 1 |
| 20 | 100 | 3 |
| 20 | 500 | 38 |
| 20 | 1,000 | 23 |
| 50 | 50 | 0 |
| 50 | 100 | 0 |
| 50 | 500 | 25 |
| 50 | 1,000 | 13 |
| 100 | 50 | 0 |
| 100 | 100 | 0 |
| 100 | 500 | 23 |
| 100 | 1,000 | 14 |

*Note*. A simulation was excluded if the estimation of $k_j$ was unstable, defined as $\text{RMSE}_k > \sqrt{10}$. Note that while a total of 140 simulations (11.7%) are excluded, these are localized within trials that had high numbers of respondents.

overestimated. Additionally, the bias moves toward zero as the number of respondents increases.

Next, we look at the estimation of the individual switching point, $k_j$. Recall that this is the position where the mixture is equal parts early test response process and late test response process. Figure 11 shows RMSE, while Figure 12 shows bias. Note that the magnitude of these errors is quite large, especially with larger numbers of items. Even with rejecting some replications, a more aggressive constraint of $k_j$ must be implemented at the software level. For this, we advise increasing the amount of regularization used during person parameter estimation as the number of respondents increases. Failure to accurately recover $k_j$ parameters causes the software to fail to properly separate early and late test difficulties. While stricter exclusion criteria could have been implemented here, it is important to demonstrate the sensitivity of $k_j$ to variation in sample size.

It is also important to note that, like most other optimization software, our software implementation controls many of the features of model fitting with hyperparameters tunable by the end user and can be optimized to provide better or worse performance in individual scenarios. We have implemented what we believe to be sensible defaults that would apply to many scenarios, but better efficiency and parameter recovery may be obtained with additional tuning. For best performance, we advise reducing the convergence threshold as the number of estimated person and item parameters increases, as well as increasing the amount of regularization as the number of respondents increases.

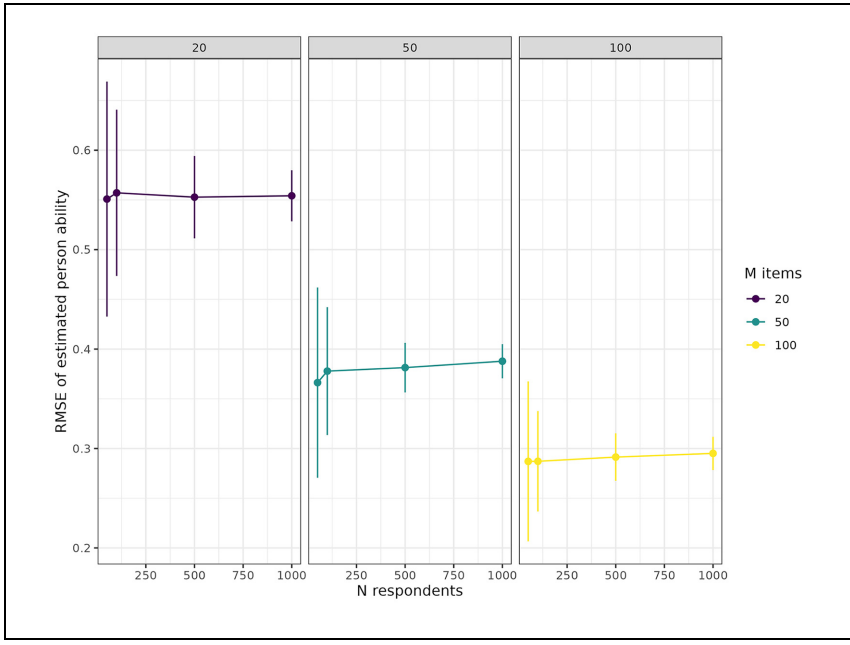


FIGURE 9. *RMSE for recovery of person ability, $\theta_j$. RMSE is displayed on the* y*-axis, the number of respondents along the* x*-axis, and the number of items is varied across panels and color. We observe a strong dependence of RMSE on the number of items, with more items producing lower RMSE.*
*Note.* RMSE = root mean squared error.

### *Study Two*

One worry when implementing a more flexible model is that it will overfit to small variations in the data used to estimate its parameters. As such, we want to understand how our model performs in a context where position effects are not present. To do this, we present a short simulation whereby we simulate item responses using a 1PL model and then fit a position-sensitive model using a 1PL kernel. Knowing the true data-generating process, we recognize fitting a model with one extra parameter per person and item is inefficient. Still, we hope to observe two features in the position-sensitive model. First, when there are no position effects, we look to see that early and late test item difficulties are largely identical. Second, we look for a strong correlation (i.e., near unity) between both early and late item difficulties in the position-sensitive model and the data-generating model.

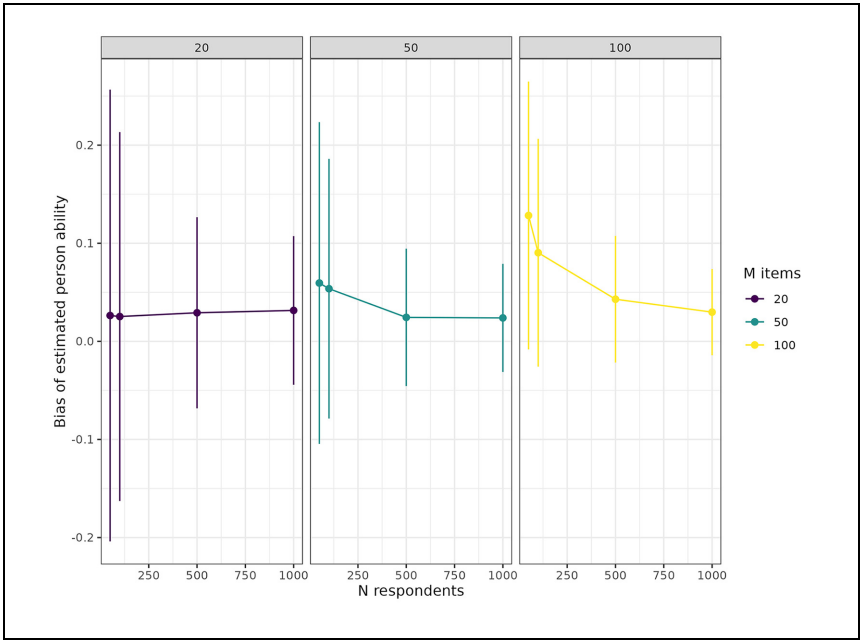


FIGURE 10. *Bias for recovery of person ability, $\theta_j$. Bias is displayed on the* y*-axis, the number of respondents along the* x*-axis, and the number of items is varied across panels and color. As the number of respondents increases, bias in ability estimation tends to approach zero.*

We simulate data using a 1PL model with $N = 1,000$ respondents with abilities drawn from a standard normal, $M = 50$ dichotomous items with difficulties drawn from a standard normal, and no position effects. We fit two models to the data: a standard 1PL model estimated in R using mirt (Chalmers, 2012) and a position-sensitive IRT model using a 1PL kernel. The left panel of Figure 13 shows the relationship between early and late test difficulty when estimated using the position-sensitive model. Since the probability of correct response was simulated with no dependence on item position, we expect $b_{\alpha i} = b_{\omega i}$. In general, this is what we observe. Next, we want to see how estimated item difficulties from the position-sensitive model compare to those from a 1PL model. The right panel of Figure 13 plots estimated 1PL difficulty against the early test difficulty $(b_{\alpha i})$ for the position-sensitive model. We selected early test difficulty somewhat arbitrarily, as early and late test difficulties were nearly identical. Again, we see that estimated difficulties are extremely highly correlated and nearly identical, aligning tightly along the $y = x$ line.

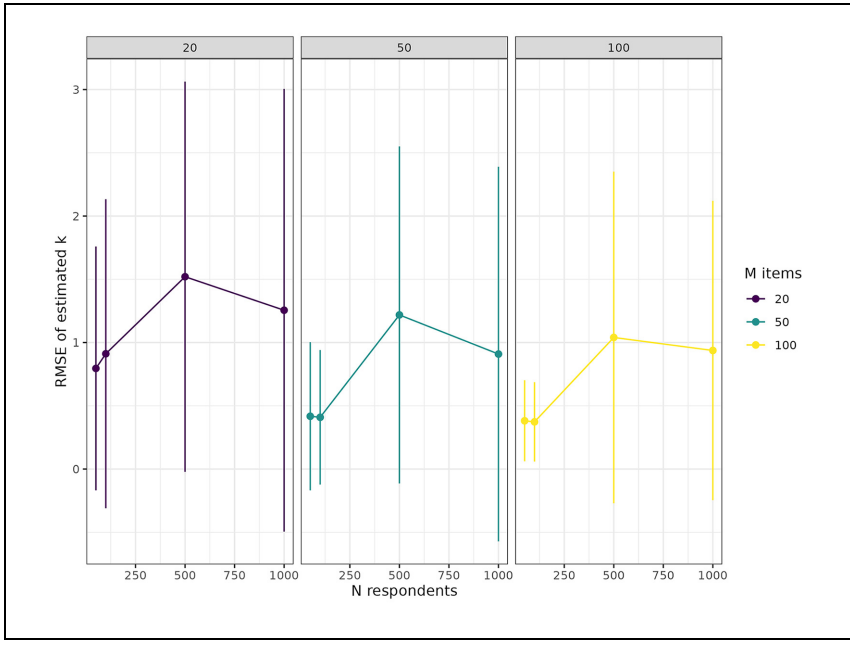


FIGURE 11. *RMSE for recovery of person switching point,* $k_j$. *RMSE is displayed on the* y*-axis, the number of respondents along the* x*-axis, and the number of items is varied across panels and colors. Note that the final estimation of* $k_j$ *can be unstable for some replications, leading to extremely high magnitudes for RMSE in replications with larger numbers of respondents.*
*Note.* RMSE = root mean squared error.

Finally, we look at ability estimates. Figure 14 plots person ability estimated from a 1PL model against person ability estimated from the position-sensitive model. Here, we see two key features of the plot. First, notice that there are horizontal bands of points. This occurs because the sum score is the sufficient statistic for the 1PL model, and as such, it produces a discretized ability distribution. The position-sensitive model breaks this property by having subtle variations in early and late item parameters and requiring information about item position to construct an ability estimate. Additionally, these abilities are estimated differently. For the 1PL model, mirt first estimates item parameters using an MML approach and then uses maximum likelihood estimation to estimate person parameters. The position-sensitive model uses a JML approach to estimate both simultaneously. While the distributions of estimated abilities are ordered by the sum score (see Sijtsma et al., 2024), the sufficient statistic for the ability

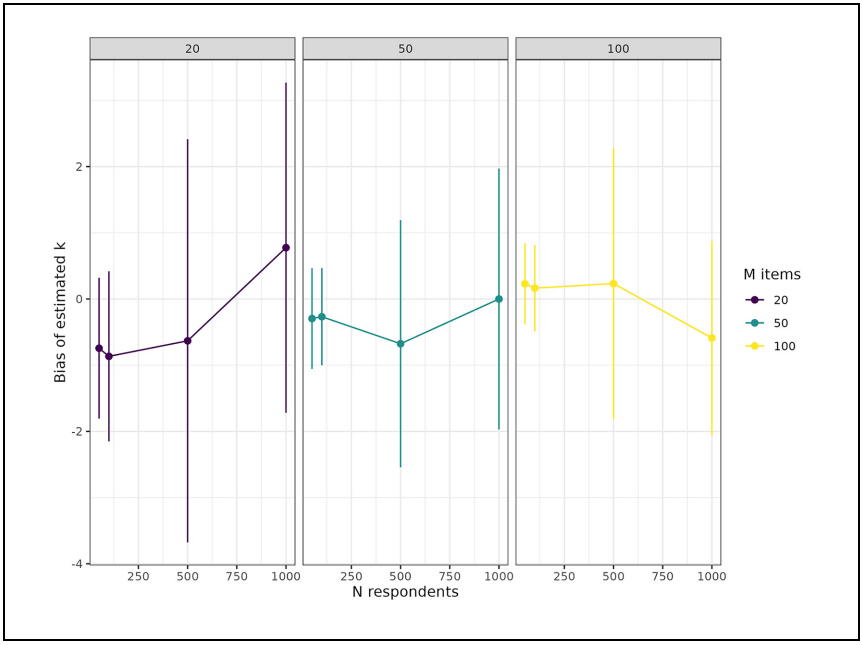


FIGURE 12. *Bias for recovery of person switching point,* $k_j$. *Bias is displayed on the* y*-axis, the number of respondents along the* x*-axis, and the number of items is varied across panels and colors. Note that the final estimation of* $k_j$ *can be unstable for some replications, leading to extreme observed bias in replications with larger numbers of respondents.*

estimate in the position-sensitive model, even with a 1PL kernel, is clearly not the sum score. While the position-sensitive model will not separate early and late item parameters when position effects do not exist, the additional parameters will introduce measurement error to the ability estimation procedure that may not be appropriate for high-stakes decisionmaking.

### *Empirical Application: Brazilian National College Entrance Exam*

Finally, we turn to an empirical application with observed position effects. In the 2014 administration of the math portion of the Brazilian national college entrance exam (ENEM), students were randomly assigned one of four booklets. These booklets all contained the same items, with the only variation being item ordering. Despite this, score distributions for each booklet were different. Given the way ENEM is administered, this plausibly attributes differences in observed score distributions to position effects. Below, we fit the position-sensitive model

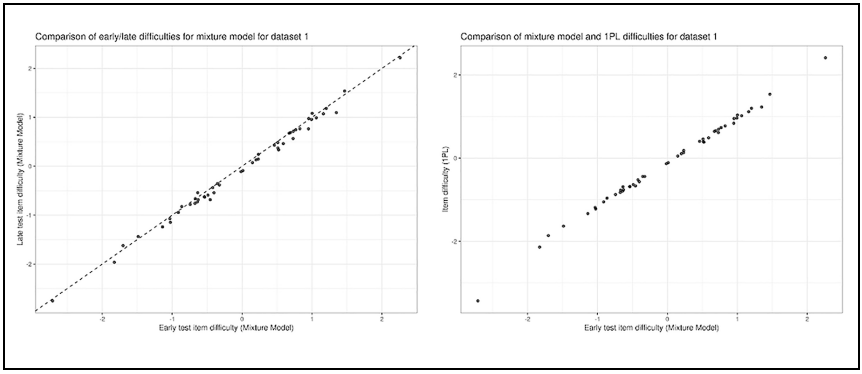


FIGURE 13. *Comparison of estimated item difficulties from the 1PL model and the position-sensitive mixture model. The left panel shows position-sensitive early and late test difficulties plotted against each other. The right panel shows 1PL difficulty along the* y*-axis while early test difficulty is plotted along the* x*-axis. As early and late test difficulties are approximately equivalent to each other and 1PL difficulties, we conclude the position-sensitive model correctly recovers the 1PL structure.*
*Note.* 1PL = one parameter logistic.

to the ENEM 2014 math data. From this, we examine the item parameters and how they relate to observed position effects and show that the ability distributions across booklets are aligned.

*Background.* ENEM is the Brazilian national college entrance exam (INEP, 2009). Each year, students take 180 multiple-choice items spread across four domains (language, social science, natural science, and mathematics) on two consecutive Sundays. Administration is simultaneous across the country, so students have a start time that ranges from 10:30 a·m to 1:30 p·m., depending on where they live. Because all students are exposed to items simultaneously, ENEM can use a smaller item bank than a large-scale test with multiple administrations. This makes item integrity much less of a security concern, though it increases the possibility of within-classroom cheating. ENEM's approach is to administer the same 180 items to all students (minimizing costs associated with item development) but randomize the order of the items (to discourage cheating). This is done by developing four separate booklets, each labeled by cover color. These booklets are assigned to students based on their seating position so that students with the same color booklet do not sit next to each other. Booklets are then scored using a 3PL model with pre-calibrated item parameters.

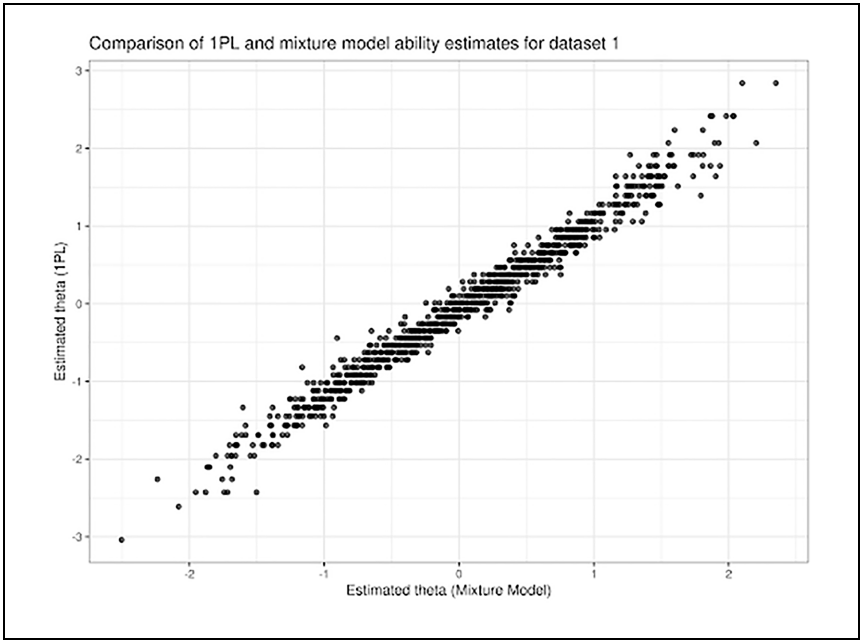


FIGURE 14. *Comparison of estimated abilities from the 1PL model and the position-sensitive mixture model. The structure of the model allows for respondents with the same sum score to receive different ability estimates.*
*Note.* 1PL = one parameter logistic.

In 2014, in the math section, students who received the blue book answered fewer questions correctly and received lower scores than students on other booklets (see Table 2). To get a better sense of how the score distributions differ between booklets, the left panel of Figure 17 highlights the highest and lowest performing booklets (gray and blue, respectively). The *x*-axis shows ability percentiles and the position of each line on the *y*-axis shows the proportion of respondents taking that booklet at or above that ability percentile. The observed gap between the ability distributions on the two booklets is problematic for two reasons. First, ability distributions within each booklet ought to be equivalent in expectation, as booklets are assigned at random. They do not appear so, with gaps as large as 10% of respondents in some positions. Because there are so many respondents, sampling variation is unlikely to be the cause of these observed differences. As such, differences are likely due to either a massive failure of randomization or, more likely given the scale of the test, non-equivalence of booklets. Second, ENEM makes no attempt to do post-administration

TABLE 2.
*Distribution of Scores Across ENEM Booklets*

| Booklet | Mean *N* correct | Mean score |
|---|---|---|
| Blue | 11.1 | 469.8 |
| Gray | 11.3 | 481.1 |
| Pink | 11.3 | 478.6 |
| Yellow | 11.4 | 477.6 |

*Note*. ENEM = Brazilian National College entrance exam.

equating of scores across booklets. As such, if pre-calibrated item parameters are used and they are not stable across booklets, the resultant scores are not comparable. Given the scale of ENEM, this could amount to hundreds of thousands of respondents incorrectly sorted around a given cut point. The importance of ENEM, combined with the observed difference in scores where it should not have existed, gives rise to a threat to the validity of score interpretations.

Considering that the only difference in the test between different booklets was the item ordering, position effects could be at play. Figure 15 shows the results of a regression of item response on item position for a subset of $N = 10{,}000$ test takers sampled evenly across booklets for all 45 items. Items are arranged along the $x$-axis by their position in the yellow booklet. Along the $y$-axis is the point estimate for the coefficient on item position with a Bonferroni corrected 95% confidence interval. Here, we see evidence of position effects in 34 of the 45 items. Items appearing below the $y = 0$ line in this plot exhibit a probability of correct response that decreases as the item is encountered later in the test, which occurs in 16 of the 34 items. There is a standard psychometric solution to this problem, where a separate scoring model can be fit to each booklet and the ability distributions can be aligned using some equating method. We instead model position effects directly in the scoring model.

*Data and Method.* We used the 45 items from the math section of the 2014 administration of ENEM. All students were presented with the same 45 items simultaneously in one of four colored booklets: blue, yellow, pink, and gray. The order of the items within booklets was randomized by rearranging complete pages. From the full pool of respondents, we selected a stratified random sample of $N = 10{,}000$ respondents across the four booklet colors who were preparing to graduate from high school.

We fit a single version of Equation 3 to our sample, using a 1PL kernel. After model fitting, we have two sets of item parameters per item, one latent ability per student, and one ‘‘endurance’’ parameter, $k_j$, per student.

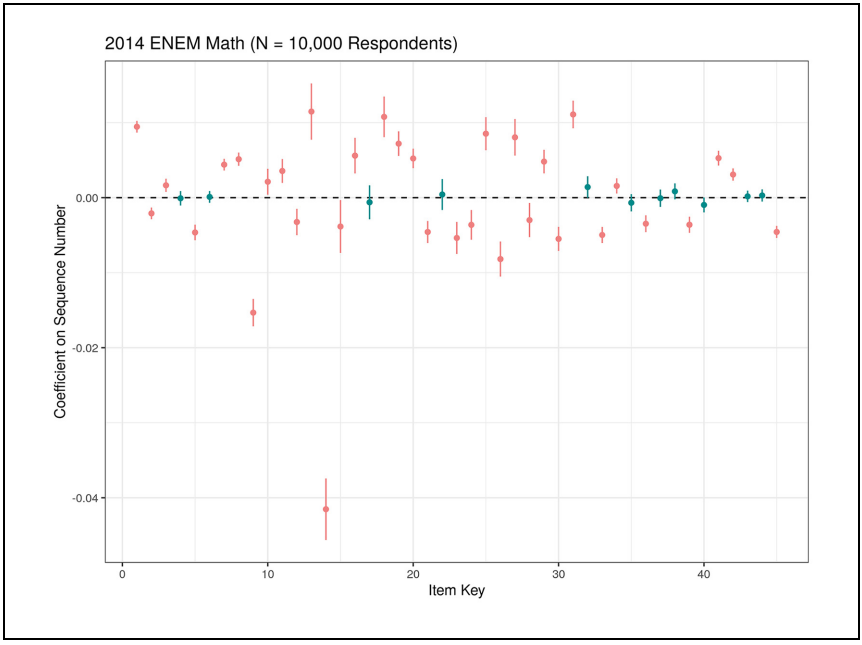


FIGURE 15. *Regressions of correct/incorrect responses on item position for all 45 items in the math section of the 2014 administration of ENEM for a subset of* $N = 10{,}000$ *respondents sampled evenly across booklet colors. Points colored pink have a significant (Bonferroni corrected) coefficient on item position, implying that item responses are dependent on the position where the item is encountered.*
*Note.* ENEM = Brazilian National College entrance exam.

*Results.* We show results using a 1PL kernel. First, we look for an indication that our model reasonably detects item position effects. To do so, we first compute the difference in difficulty between the early and late item parameters, $\delta_i = b_{i\omega} - b_{i\alpha}$, with the expectations that items with larger magnitude position effects ought to display a larger $\delta_i$. As such, we plot $\delta_i$ against the coefficients from the response regressions on item position from Figure 15. Figure 16 shows this relationship. One concern might be that the variation in the position in which an item occurs may be a driving force behind the observed magnitude of position effects. To investigate this, we also color points by the standard deviation of the four positions the item occurs in. Importantly, we observe that the observed variation in item position does not seem to be associated with either the magnitude of position effects or the degree to which $\delta_i$ aligns with expectations.

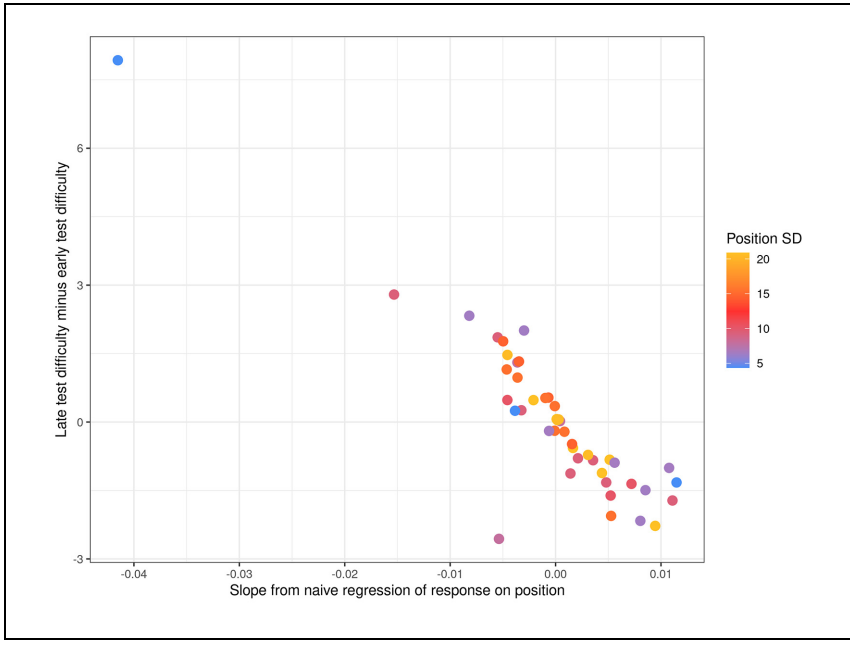


FIGURE 16. *Difference between early and late test difficulty, $\delta_i$, plotted against slopes on item position from naive regressions to estimate position effects. Note that the mixture model estimates position effects of a direction and magnitude that agrees with observed trends.*
*Note.* ENEM = Brazilian National College entrance exam.

Next, we look to see the degree to which estimated ability distributions align after the application of the position-sensitive model. The right panel of Figure 17 shows the proportion of students performing above a certain cut score by booklet, projected onto an ability percentile scale for comparison with the left panel. The overlapping lines in the right panel of Figure 17 show that the application of our position-sensitive model immediately closes the gap between what were previously the highest and lowest ability booklets.

If high-stakes tests like ENEM are used for centralized admissions criteria, multiple cut points are selected to sort respondents into groups for college admission. As the 2014 administration of ENEM had nearly six million respondents, cut-point selection without some method of ability distribution alignment can result in the misclassification of hundreds of thousands of respondents simply based on booklet assignment at testing time.

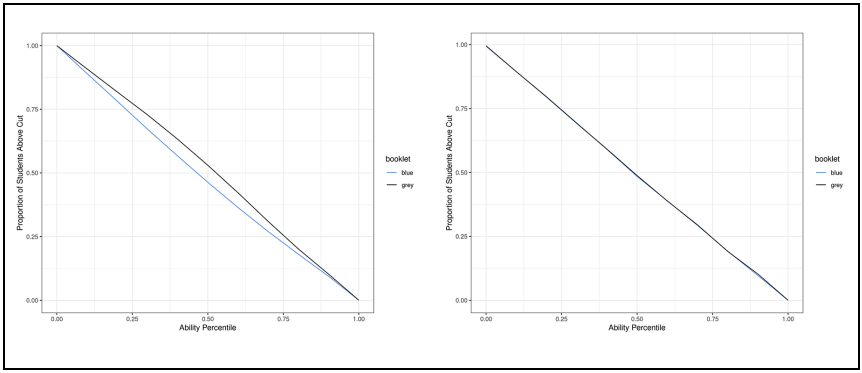


FIGURE 17. *Proportion of students within a booklet at or above a cut point, by booklet. The left panel uses ENEM reported scores and the right panel uses scores derived from a position-sensitive model. Only the highest and lowest-performing booklets are shown.*
*Note.* ENEM = Brazilian National College entrance exam.

## Discussion

In this article, we develop a mixture-based approach for dealing with position effects. The approach allows for variation in both the degree to which items vary in functioning across the assessment *and* the relative amount of the test that respondents spend in different (i.e., early/late) responding modes. Central to this method is the imposition of a functional form on the mixing parameter akin to that of the C-HYBRID model (Nagy & Robitzsch, 2021). The two key contributions of this study are the mixing of early and late response processes with distinct item parameters to study position effects and the introduction of the Adam optimization algorithm to psychometric work (Kingma & Ba, 2014). We conduct simulation studies to show suitable parameter recovery when the mixture model is the data-generating model, as well as model behavior in the absence of position effects. In general, we find that the model does not find position effects when they do not exist, making it safe to use when only a subset of items exhibit position effects. Estimation may be sensitive to the sample sizes of both persons and items, as well as specific choices of hyperparameters like learning rate, the amount of regularization, batch size, and convergence thresholds. We then use this model to better understand position effects on a large-stakes standardized assessment. This assessment has millions of respondents and is administered in such a way that differences in ability distributions across booklets can be plausibly attributed to only position effects. Despite not knowing the data-generating process for this assessment, we demonstrate that the model produces interpretable item parameters that allow us to better understand position effects and

distributions of estimated person abilities that are comparable across different item orderings.

Many large-scale and high-stakes testing programs use pre-calibrated item parameters. This practice allows for theoretical flexibility when constructing test forms from an item bank. However, when multiple test forms with variable item positions are used, the assumption of stable item parameters may not hold. This could pose a substantial threat to test validity, as booklet assignment confounds score interpretations that assume observed score differences are based largely on latent ability. While many approaches to modeling position effects exist (Debeer & Janssen, 2013), explicitly modeling position effects in a way that is both flexible and identifiable offers test developers and users a way to not only gather evidence about the potential magnitude of position effects but also to investigate position effects at the level of individual items or persons.

One key use case for this model is in CAT. In testing scenarios where there are only a few forms with permutations of items, between-form differences in observed ability distributions can be aligned through an equating procedure. With CAT, large item banks and flexibility in item positioning can result in an enormous number of effective test forms, each exposed to as few as a single respondent. In this case, position effects may introduce a new dimension of comparability issues between different adaptive test takers and pencil-and-paper respondents (Van der Linden & Glas, 2000; Wang & Kolen, 2001). In part, this is because there is no true random assignment of item content. If there is a suspicion of position effects, using a position-sensitive model can account for differences in individual response processes that evolve throughout the test. Additionally, the software developed to fit this model is designed to be used with a variety of flexible kernel IRFs, has tunable convergence criteria, and can run on GPU to aid in model fitting for large data sets.

Our mixture formulation of a position-sensitive IRT model has theoretical advantages and is safe to use when position effects are not present, but also has limitations. First, model fitting may be computationally intensive depending on both the amount of data collected and the booklet design. While drawing on fitting techniques from the deep learning community and moving computation to a GPU can provide a significant speed boost, more parameters necessarily come with a computational burden (Kingma & Ba, 2014). Additionally, while we have shown in simulation that the model tends not to overfit and agrees with vanilla IRT models when position effects are not present, idiosyncrasies in individual response patterns always have the potential to provide misleading results, though this is not unique to our model. In simulation, we see that parameters that are tightly bound to position effects can be extremely difficult to recover accurately without large numbers of respondents, a problem that may be addressed with new advances in estimation approaches (Belov et al., 2024; Welling et al., 2024; Zhang & Chen, 2024). Additionally, model selection is always a key concern,

and the ability of the model to, for example, produce aligned ability distributions requires the selection of an appropriate kernel. Merely including position effects won't make the need for modeling item discrimination obsolete, for example.

There are many avenues for future work expanding on this study. The most clear-cut is the implementation of more complex IRF kernels. Our use of only the 1PL kernel is certainly a limitation of the present work. Another route that we view as promising is the flexibility afforded in the specification of the functional form of the mixing parameter. It is conceivable that this is an avenue in which to include process data directly within a measurement model. This follows previous work by Molenaar et al. (2016) using hidden Markov models with response time evidence to model distinct response states but does push the model outside of the C-HYBRID framework. Given sufficient evidence that a change in an individual's response process can be described by some observation of their behavior on a test, this could be a principled way to include information like response time, keystroke logs, or other data collected in a computerized testing environment directly in the model.

## Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Jacobs Foundation.

## ORCID iD

Klint Kanopka https://orcid.org/0000-0003-3196-9538

## Note

1. The name is derived from adaptive moment estimation.

## References

Albano, A. D. (2013). Multilevel modeling of item position effects. *Journal of Educational Measurement*, *50*(4), 408–426.

Belov, D. I., Lüdtke, O., & Ulitzsch, E. (2024). Likelihood-free estimation of IRT models in small samples: A neural networks approach. *PsyArXiv*. https://doi.org/10.31234/osf.io/w3cyq

Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, *46*(4), 443–459.

Bolt, D. M., Cohen, A. S., & Wollack, J. A. (2002). Item parameter estimation under conditions of test speededness: Application of a mixture Rasch model with ordinal constraints. *Journal of Educational Measurement*, *39*(4), 331–348.

Camilli, G., Shepard, L. A., & Shepard, L. (1994). *Methods for identifying biased test items* (Vol. 4). Sage.

Chalmers, R. P. (2012). MIRT: A multidimensional item response theory package for the r environment. *Journal of Statistical Software*, *48*, 1–29.

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

Debeer, D., & Janssen, R. (2013). Modeling item-position effects within an IRT framework. *Journal of Educational Measurement*, *50*(2), 164–185.

De Boeck, P. (2004). *Explanatory item response models: A generalized linear and nonlinear approach*. Springer Science & Business Media.

Défossez, A., Bottou, L., Bach, F., & Usunier, N. (2020). *A simple convergence proof of Adam and Adagrad*. arXiv preprint arXiv:2003.02395.

Domingue, B. W., Kanopka, K., Stenhaug, B., Soland, J., Kuhfeld, M., Wise, S., & Piech, C. (2021). Variation in respondent speed and its implications: Evidence from an adaptive testing scenario. *Journal of Educational Measurement*, *58*(3), 335–363.

Feather, N. T. (1962). The study of persistence. *Psychological Bulletin*, *59*(2), 94.

Hastie, T. (2020). Ridge regularization: An essential concept in data science. *Technometrics*, *62*(4), 426–433.

Hohensinn, C., Kubinger, K. D., Reif, M., Holocher-Ertl, S., Khorramdel, L., & Frebort, M. (2008). Examining item-position effects in large-scale assessment using the linear logistic test model. *Psychology Science*, *50*(3), 391.

Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. (2009). *National High School Exam (ENEM): Theoretical and methodological texts*. MEC/INEP.

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980.

Kingston, N. M., & Dorans, N. J. (1984). Item location effects and their implications for irt equating and adaptive testing. *Applied Psychological Measurement*, *8*(2), 147–154.

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). *Addressing cold-start problem in recommendation systems*. [Conference session] Proceedings of the 2nd international conference on Ubiquitous information management and communication (pp. 208–211). Association for Computing Machinery.

Leary, L. F., & Dorans, N. J. (1982). The effects of item rearrangement on test performance: A review of the literature. *ETS Research Report Series*, *1982*(2), i-23.

Meyers, J. L., Miller, G. E., & Way, W. D. (2008). Item position and item difficulty change in an irt-based common item equating design. *Applied Measurement in Education*, *22*(1), 38–60.

Molenaar, D., Oberski, D., Vermunt, J., & De Boeck, P. (2016). Hidden Markov item response theory models for responses and response times. *Multivariate Behavioral Research*, *51*(5), 606–626.

Nagy, G., & Robitzsch, A. (2021). A continuous hybrid IRT model for modeling changes in guessing behavior in proficiency tests. *Psychological Test and Assessment Modeling*, *63*(3), 361–395.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). *Automatic differentiation in pytorch* [Conference session]. 31st Conference on Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA.

Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural Networks*, *12*(1), 145–151.

Reddi, S. J., Kale, S., & Kumar, S. (2019). *On the convergence of Adam and beyond*. arXiv preprint arXiv:1904.09237.

Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement*, *14*(3), 271–282.

Sen, S., & Cohen, A. S. (2019). Applications of mixture IRT models: A literature review. *Measurement: Interdisciplinary Research and Perspectives*, *17*(4), 177–191.

Shanker Tripathi, A., & Domingue, B. W. (2019). *Curve fitting from probabilistic emissions and applications to dynamic item response theory*. arXiv preprint arXiv:1909.03586.

Sijtsma, K., Ellis, J. L., & Borsboom, D. (2024). Recognize the value of the sum score, psychometrics' greatest accomplishment. *Psychometrika*, *89*(1), 84–117.

Trendtel, M., & Robitzsch, A. (2018). Modeling item position effects with a bayesian item response model applied to PISA 2009–2015 data. *Psychological Test and Assessment Modeling*, *60*(2), 241–263.

Tucker-Drob, E. M. (2011). Individual differences methods for randomized experiments. *Psychological Methods*, *16*(3), 298.

Van der Linden, W. J., & Glas, C. A. (2000). *Computerized adaptive testing: Theory and practice*. Springer.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

von Davier, M., & Rost, J. (2018). Logistic mixture-distribution response models. In W. J. Van der Linden (Ed.), *Handbook of item response theory* (pp. 393–406). Chapman and Hall/CRC.

Wang, T., & Kolen, M. J. (2001). Evaluating comparability in computerized adaptive testing: Issues, criteria and an example. *Journal of Educational Measurement*, *38*(1), 19–49.

Weirich, S., Hecht, M., & Böhme, K. (2014). Modeling item position effects using generalized linear mixed models. *Applied Psychological Measurement*, *38*(7), 535–548.

Weirich, S., Hecht, M., Penk, C., Roppelt, A., & Böhme, K. (2017). Item position effects are moderated by changes in test-taking effort. *Applied Psychological Measurement*, *41*(2), 115–129.

Welling, W. S., Sheng, Y., & Zhu, M. M. (2024). Cuda-aware MPI implementation of Gibbs sampling for an IRT model. *Cluster Computing*, *27*(2), 1821–1830.

Yamamoto, K. (1995). Estimating the effects of test length and test time on parameter estimation using the hybrid model. *ETS Research Report Series*, *1995*(1), i-39.

Yamamoto, K., & Everson, H. T. (1995). Modeling the mixture of IRT and pattern responses by a modified hybrid model1. *ETS Research Report Series*, *1995*(1), i-26.

Zeller, F., Reiß, S., & Schweizer, K. (2017). Is the item-position effect in achievement measures induced by increasing item difficulty? *Structural Equation Modeling: A Multidisciplinary Journal*, *24*(5), 745–754.

Zhang, L., & Chen, P. (2024). A neural network paradigm for modeling psychometric data and estimating IRT model parameters: Cross estimation network. *Behavior Research Methods*, *56*, 7026–7058

## Authors

KLINT KANOPKA is an assistant professor at New York University; e-mail: klint.kanopka@nyu.edu. He is interested in applications of machine learning to psychometrics.

BENJAMIN W. DOMINGUE is an associate professor at the Stanford Graduate School of Education; e-mail: bdomingue@stanford.edu. He is interested in psychometrics and quantitative methods.

Manuscript received November 17, 2023
Revision received August 28, 2024
Accepted September 4, 2024