ADAM estimation for the hierarchical joint model of response accuracy and response time

Yingshi Huang

School of Education & Information Studies
University of California, Los Angeles

1 Introduction

With the widespread adoption of computer-based testing, researchers now have access to increasingly rich data sources. Among these, response time has become one of the most commonly collected measures.

Response time offers a valuable, complementary window into examinees' cognitive processes, providing a fine-grained view of problem solving that accuracy alone cannot capture (Van der Linden, 2006). Such timing information has proven useful in several ways. First, researchers have used response time to improve the precision of ability estimates. By borrowing information from response latencies, joint models achieve higher accuracy in measuring examinee proficiency than models based solely on correctness (Bolsinova and Tijmstra, 2018; De Boeck and Jeon, 2019). Moreover, in time-limited, high-stakes assessments, such as pilot certification exams and medical licensure tests, examinees must demonstrate both correctness and speed under pressure (Maris and Van der Maas, 2012). Jointly analyzing response accuracy and response time therefore helps to identify candidates who satisfy both performance and timing requirements. Second, response time can serve as a diagnostic tool for data quality by detecting aberrant response behaviors that accuracy data alone would miss. For example, in low-stakes testing environments some examinees engage in rapid guessing, producing response patterns characterized by very short latencies and low correctness rates (Cheng and Shao, 2022). Because low-proficiency examinees may also yield low accuracy, response times are essential for distinguishing random guessing from genuine attempts, thereby protecting the validity of the assessment. These studies underscore the power and growing prevalence of join analysis using both response time and response accuracy. As a result, joint modeling of response time and accuracy has grown explosively in recent years (Fox and Marianti, 2016; Guo et al., 2022; Wang et al., 2018).

Among the various approaches, the two-level hierarchical joint model introduced by Van der Linden (2007) is one of the most widely used frameworks. It is popular for its flexible, "plug-and-play" design: at the first level, researchers are relatively free for the choices of response accuracy and response time models, and at the second level, person and item latent parameters are modeled jointly. This elegant two-tier structure effectively captures dependencies between accuracy and time, making it an appealing choice for both methodological research and applied assessment contexts.

Despite its widespread use, estimating the hierarchical joint model remains challenging due to its inherent complexity and heterogeneous curvature. Specifically, the two-level struc-

ture is conceptually straightforward but yields a large parameter space. For a combination of a 2-parameter item response theory model for accuracy and a log-normal model for response time, one must estimate: Level 1: $2 \times N + 2 \times J + 2 \times J$ parameters (N person-level speed and ability; J item-level discrimination, difficulty, time-intensity, and time-variance) and Level 2: 2 + 4+ 10 hyperparameters (means, variances, and covariances among person and item). Moreover, the information contributed by each response is uneven across parameters, leading to heterogeneous curvature in the joint likelihood surface. Some parameters are informed primarily by accuracy data, while others rely on timing information. Such imbalance produces highly irregular likelihood landscapes. Currently, only the Markov chain Monte Carlo (MCMC) approach is available for the hierarchical join model, typically via Gibbs sampling, which iteratively draws from the full joint posterior distribution. MCMC's strength lies in its ability to accommodate complex dependencies and high-dimensional parameter spaces while providing full posterior summaries (Gelman & Rubin, 1992; Liu, 2008). However, it suffers from slow mixing and long run times as dimensionality grows, and requires careful tuning of sampling schemes and convergence diagnostics. These drawbacks motivate the search for alternative methods.

The Adaptive Moment Estimation (ADAM) algorithm presents a promising approach to overcoming the estimation challenges of hierarchical joint models. Originally introduced by Kingma and Ba (2014), ADAM has become a standard optimizer within popular machine-learning frameworks such as TensorFlow and PyTorch (Paszke, 2019). Its key innovation lies in adaptively scaling the learning rate for each parameter according to the history of past updates, thereby preventing excessive adjustments in regions of high curvature. By maintaining first and second moment estimates of the gradients, ADAM leverages historical direction information to produce more informed and stable parameter updates. Additionally, ADAM has demonstrated robust performance even on non-convex loss surfaces.

Building on these strengths, this project develops an ADAM-based estimation pipeline tailored to the hierarchical joint model. The following sections first review the specification of the hierarchical joint model. We then describe the implementation of ADAM within this context. A simulation study follows, comparing ADAM's efficiency and estimation accuracy against traditional MCMC approaches. Then conclude with a discussion of the findings and limitations.

2 The Hierarchical Framework

2.1 Model specification

The hierarchical framework uses a two-level structure to jointly model the response accuracy and response time.

Response accuracy (level 1). Following the two-parameter item response theory model, the response accuracy Y_{ij} of examinee $i \in \{1, \ldots, N\}$ on item $j \in \{1, \ldots, J\}$ can be determined as:

$$Y_{ij} \sim f(y_{ij}; \theta_i, a_j, b_j), \tag{1}$$

with the Bernoulli distribution $f(y_{ij};\theta_i,a_j,b_j)$. Then, the probability of giving a correct answer as:

$$P(Y_{ij}=1)=\frac{1}{1+\exp[-a_j(\theta_i-b_j)]}, \tag{2}$$

where θ_i is the ability of examinee i; b_j is the item difficulty parameter of item j, while a_j is the discrimination parameter.

Response time (level 1). Using the log-normal theory of response time, the response time T_{ij} is defined as:

$$T_{ij}\sim f(t_{ij};\tau_i,\beta_j,\alpha_j), \tag{3}$$

with the log normal density function as:

$$f(t_{ij};\tau_i,\beta_j,\alpha_j)=\frac{\alpha_j}{t_{ij}\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left[\alpha_j(\ln t_{ij}-(\beta_j-\tau_i))\right]^2\right\}, \tag{4}$$

where β_j is the time density parameter for item j, representing the averaged time required for solving the item, and α_j is the time discriminating parameter. τ_i is the latent speed of person i.

Multivariate normal distribution (level 2). Combining two first-level models, we can have the person parameter vector as $\boldsymbol{\xi}_i=(\theta_i,\tau_i)$, and the item parameter vector as $\boldsymbol{\psi}_j=(a_j,b_j,\beta_j,\alpha_j)$. $\boldsymbol{\xi}_i$ follows a multivariate normal distribution as:

$$\boldsymbol{\xi}_i\sim f(\boldsymbol{\xi}_i;\boldsymbol{\mu}_{\mathcal{P}},\boldsymbol{\Sigma}_{\mathcal{P}}), \tag{5}$$

with the density function as:

$$f(\boldsymbol{\xi}_i;\boldsymbol{\mu}_{\mathcal{P}},\boldsymbol{\Sigma}_{\mathcal{P}})=\frac{|\boldsymbol{\Sigma}_{\mathcal{P}}^{-1}|^{1/2}}{2\pi}\exp\left[-\frac{1}{2}\left(\boldsymbol{\xi}_i-\boldsymbol{\mu}_{\mathcal{P}}\right)^\top\boldsymbol{\Sigma}_{\mathcal{P}}^{-1}\left(\boldsymbol{\xi}_i-\boldsymbol{\mu}_{\mathcal{P}}\right)\right], \tag{6}$$

where the $\boldsymbol{\mu}_{\mathcal{P}}=(\mu_\theta,\mu_\tau)$ is the mean vector, and $\boldsymbol{\Sigma}_{\mathcal{P}}=[(\sigma_\theta^2,\sigma_{\tau\theta})^\top,(\sigma_{\theta\tau},\sigma_\tau^2)^\top]$ is the covariance matrix. For the identification purpose, we have constraints as $\boldsymbol{\mu}_{\mathcal{P}}=(\mu_\theta,\mu_\tau)=(0,0)$, and $\sigma_\theta^2=1$.

Similarly, we have $\boldsymbol{\psi}_j\sim f(\boldsymbol{\psi}_j;\boldsymbol{\mu}_{\mathcal{I}},\boldsymbol{\Sigma}_{\mathcal{I}})$ with the density function as:

$$f(\boldsymbol{\psi}_j;\boldsymbol{\mu}_{\mathcal{I}},\boldsymbol{\Sigma}_{\mathcal{I}})=\frac{|\boldsymbol{\Sigma}_{\mathcal{I}}^{-1}|^{1/2}}{(2\pi)^2}\exp\left[-\frac{1}{2}\left(\boldsymbol{\psi}_j-\boldsymbol{\mu}_{\mathcal{I}}\right)^\top\boldsymbol{\Sigma}_{\mathcal{I}}^{-1}\left(\boldsymbol{\psi}_j-\boldsymbol{\mu}_{\mathcal{I}}\right)\right], \tag{7}$$

where the $\boldsymbol{\mu}_{\mathcal{I}}=(\mu_a,\mu_b,\mu_\beta,\mu_\alpha)$ is the mean vector, and $\boldsymbol{\Sigma}_{\mathcal{I}}$ is the covariance matrix being defined as:

$$\boldsymbol{\Sigma}_{\mathcal{I}}=\begin{bmatrix}\sigma_a^2 & \sigma_{ab} & \sigma_{a\beta} & \sigma_{a\alpha}\\ \sigma_{ba} & \sigma_b^2 & \sigma_{b\beta} & \sigma_{b\alpha}\\ \sigma_{\beta a} & \sigma_{\beta b} & \sigma_\beta^2 & \sigma_{\beta\alpha}\\ \sigma_{\alpha a} & \sigma_{\alpha b} & \sigma_{\alpha\beta} & \sigma_\alpha^2\end{bmatrix}.$$

Hence, the likelihood function can be written as:

$$\mathcal{L} = \prod_{j=1}^{J} \prod_{i=1}^{N} f(y_{ij}; \theta_i, a_j, b_j) f(t_{ij}; \tau_i, \beta_j, \alpha_j) f(\boldsymbol{\xi}_i; \boldsymbol{\mu}_{\mathcal{P}}, \boldsymbol{\Sigma}_{\mathcal{P}}) f(\boldsymbol{\psi}_j; \boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\Sigma}_{\mathcal{I}}). \tag{8}$$

Then, defining $P(Y_{ij} = 1) = \pi_{ij}$, $\ln t_{ij} - (\beta_j - \tau_i) = r_{ij}$, $\boldsymbol{\Sigma}_{\mathcal{P}}^{-1} = \boldsymbol{\Omega}_{\mathcal{P}}$, and $\boldsymbol{\Sigma}_{\mathcal{I}}^{-1} = \boldsymbol{\Omega}_{\mathcal{I}}$, we can have the log-likelihood function as:

$$\begin{aligned}
\ell &= \sum_{j=1}^{J} \sum_{i=1}^{N} \left[\log f(y_{ij}; \theta_i, a_j, b_j) + \log f(t_{ij}; \tau_i, \beta_j, \alpha_j) + \log f(\boldsymbol{\xi}_i; \boldsymbol{\mu}_{\mathcal{P}}, \boldsymbol{\Sigma}_{\mathcal{P}}) + \log f(\boldsymbol{\psi}_j; \boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\Sigma}_{\mathcal{I}}) \right] \\
&= \sum_{j=1}^{J} \sum_{i=1}^{N} \left[y_{ij} \log \pi_{ij} + (1 - y_{ij}) \log(1 - \pi_{ij}) \right] && (9) \\
&+ \sum_{j=1}^{J} \sum_{i=1}^{N} \left[\log \alpha_j - \log t_{ij} - \frac{1}{2} \log(2\pi) - \frac{1}{2} (\alpha_j r_{ij})^2 \right] && (10) \\
&+ \sum_{i=1}^{N} \left[\frac{1}{2} \log |\boldsymbol{\Omega}_{\mathcal{P}}| - \log(2\pi) - \frac{1}{2} (\boldsymbol{\xi}_i - \boldsymbol{\mu}_{\mathcal{P}})^{\top} \boldsymbol{\Omega}_{\mathcal{P}} (\boldsymbol{\xi}_i - \boldsymbol{\mu}_{\mathcal{P}}) \right] && (11) \\
&+ \sum_{j=1}^{J} \left[\frac{1}{2} \log |\boldsymbol{\Omega}_{\mathcal{I}}| - 2 \log(2\pi) - \frac{1}{2} (\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})^{\top} \boldsymbol{\Omega}_{\mathcal{I}} (\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}}) \right]. && (12)
\end{aligned}$$

3 The Adaptive Moment Estimation

The core idea of the Adaptive Moment (ADAM) algorithm is to use the first moment of the gradient as momentum to track the average direction of past gradients, and the second moment to adapt the learning rate based on the magnitude of historical gradients. This design accelerates convergence, helps avoid getting stuck in flat regions, and provides an adaptive step size that adjusts according to the frequency of updates for each parameter.

3.1 The gradient of each parameter

To realize the ADAM algorithm, the first step is to obtain the gradient of each parameter.

Considering the ith examinee, for the person parameters, based on Equations 9 and 11, and Equations 10 and 11, we can have:

$$\frac{\partial \ell}{\partial \theta_i} = \sum_{j=1}^{J} a_j (y_{ij} - \pi_{ij}) - \left[\boldsymbol{\Omega}_{\mathcal{P}} (\boldsymbol{\xi}_i - \boldsymbol{\mu}_{\mathcal{P}}) \right]_1 = \sum_{j=1}^{J} a_j (y_{ij} - \pi_{ij}) - \left[\boldsymbol{\Omega}_{\mathcal{P}} \boldsymbol{\xi}_i \right]_1, \tag{13}$$

$$\frac{\partial \ell}{\partial \tau_i} = -\sum_{j=1}^{J} \alpha_j^2 r_{ij} - \left[\boldsymbol{\Omega}_{\mathcal{P}} (\boldsymbol{\xi}_i - \boldsymbol{\mu}_{\mathcal{P}}) \right]_2 = -\sum_{j=1}^{J} \alpha_j^2 r_{ij} - \left[\boldsymbol{\Omega}_{\mathcal{P}} \boldsymbol{\xi}_i \right]_2 \tag{14}$$

where $[\cdot]_p$ represents the pth element in the vector. Similarly, considering the jth item, based on Equations 9 and 12, and Equations 10 and 12, we can have:

$$\frac{\partial \ell}{\partial a_j} = \sum_{i=1}^{N}(\theta_i - b_j)(y_{ij} - \pi_{ij}) - \left[\boldsymbol{\Omega}_{\mathcal{I}}(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})\right]_1, \tag{15}$$

$$\frac{\partial \ell}{\partial b_j} = -\sum_{i=1}^{N} a_j(y_{ij} - \pi_{ij}) - \left[\boldsymbol{\Omega}_{\mathcal{I}}(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})\right]_2, \tag{16}$$

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{N} \alpha_j^2 r_{ij} - \left[\boldsymbol{\Omega}_{\mathcal{I}}(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})\right]_3, \tag{17}$$

$$\frac{\partial \ell}{\partial \alpha_j} = \sum_{i=1}^{N}\left(\frac{1}{\alpha_j} - \alpha_j r_{ij}^2\right) - \left[\boldsymbol{\Omega}_{\mathcal{I}}(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})\right]_4. \tag{18}$$

Further, to ensure the identifiability of the item response model and the positive property of a_j and α_j, the optimization process for these two parameters focuses on the log scale. Then, we have the gradient defined as:

$$\frac{\partial \ell}{\partial \log a_j} = \frac{\partial \ell}{\partial a_j}\frac{\partial a_j}{\partial \log a_j} = a_j \frac{\partial \ell}{\partial a_j}, \tag{19}$$

$$\frac{\partial \ell}{\partial \log \alpha_j} = \frac{\partial \ell}{\partial \alpha_j}\frac{\partial \alpha_j}{\partial \log \alpha_j} = \alpha_j \frac{\partial \ell}{\partial \alpha_j}. \tag{20}$$

For the second level parameters, we can have:

$$\frac{\partial \ell}{\partial \boldsymbol{\mu}_{\mathcal{I}}} = \boldsymbol{\Omega}_{\mathcal{I}} \sum_{j=1}^{J}(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}}), \tag{21}$$

$$\frac{\partial \ell}{\partial \boldsymbol{\Sigma}_{\mathcal{P}}} = \frac{1}{2}\left[\boldsymbol{\Omega}_{\mathcal{P}}\left(\sum_{i=1}^{N}(\boldsymbol{\xi}_i \boldsymbol{\xi}_i^{\top}) - N\boldsymbol{\Sigma}_{\mathcal{P}}\right)\boldsymbol{\Omega}_{\mathcal{P}}\right], \tag{22}$$

$$\frac{\partial \ell}{\partial \boldsymbol{\Sigma}_{\mathcal{I}}} = \frac{1}{2}\left[\boldsymbol{\Omega}_{\mathcal{I}}\left(\sum_{j=1}^{J}(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})(\boldsymbol{\psi}_j - \boldsymbol{\mu}_{\mathcal{I}})^{\top} - J\boldsymbol{\Sigma}_{\mathcal{I}}\right)\boldsymbol{\Omega}_{\mathcal{I}}\right]. \tag{23}$$

In terms of the person and item covariance matrices, the Cholesky factorization is employed to ensure the symmetric positive definite property, that is, $\boldsymbol{\Sigma} = LL^{\top}$. L is a real lower triangular matrix L_{low} with positive diagonal entries L_{diag}. Then, the gradients of the person and item L matrices are as follows:

$$\frac{\partial \boldsymbol{\Sigma}_{\mathcal{P}}}{\partial L_{\mathcal{P}}} = 2\frac{\partial \ell}{\partial \boldsymbol{\Sigma}_{\mathcal{P}}} L_{\mathcal{P}}, \tag{24}$$

$$\frac{\partial \boldsymbol{\Sigma}_{\mathcal{I}}}{\partial L_{\mathcal{I}}} = 2\frac{\partial \ell}{\partial \boldsymbol{\Sigma}_{\mathcal{I}}} L_{\mathcal{I}}. \tag{25}$$

Similarly, to ensure the positive property of the variance, the log scale of the diagonal elements are computed. For the item diagonal elements, we can have:

$$\frac{\partial \boldsymbol{\Sigma}_{\mathcal{I}}}{\partial \log L_{\mathcal{I}\_diag}} = \frac{\partial \boldsymbol{\Sigma}_{\mathcal{I}}}{\partial L_{\mathcal{I}\_diag}} \frac{\partial L_{\mathcal{I}\_diag}}{\partial \log L_{\mathcal{I}\_diag}} = \operatorname{diag}\left(\frac{\partial \ell}{\partial \boldsymbol{\Sigma}_{\mathcal{I}}}\right) L_{\mathcal{I}\_diag}. \tag{26}$$

The computation of the person variance follows the same logic.

3.2 The ADAM algorithm

After having the gradients, the estimation with ADAM can be straightforward. Let $\boldsymbol{X} = (\boldsymbol{\theta}, \boldsymbol{\tau}, \log \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\beta}, \log \boldsymbol{\alpha}, \boldsymbol{\mu}_{\mathcal{I}}, L_{\mathcal{P}\_low}, \log L_{\mathcal{P}\_diag}, L_{\mathcal{I}\_low}, \log L_{\mathcal{I}\_diag})$ represents the vector of all the parameters of interest. Then, the corresponding gradient vector is denoted as $\nabla f(\boldsymbol{x})$. The update rule of the parameters at kth iterate is as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta(\tilde{z}_k \odot v_k), \tag{27}$$

where $\odot$ is the Hadamard product; η is a fixed step size. $\tilde{z}_k$ is the adaptive term that carries previous history of the gradient with the pth element being defined as:

$$\begin{aligned} \tilde{z}_k(p) &= 1/\sqrt{z_k(p)}, \\ z_k &= \operatorname{pmax}\left(z_{k-1}, w_1 z_{k-1} + \nabla f(\boldsymbol{x}_k) \odot \nabla f(\boldsymbol{x}_k)\right). \end{aligned} \tag{28}$$

Here, $w_1 \in (0, 1)$ is the weighting parameter. The $\operatorname{pmax}(\cdot)$ function provides the maximum values of a vector. Further, we have the momentum term v_k being defined as:

$$v_k = w_2 v_{k-1} + (1 - w_2)\nabla f(\boldsymbol{x}_k), \tag{29}$$

where $w_2 \in (0, w_1)$ is the weighting parameter for the momentum.

4 Simulation

The simulation study aims to evaluate the parameter recovery and computation efficiency of the ADAM fashion estimation process. The traditional MCMC approach is used as the baseline.

4.1 Method

Based on specifications from widely administered computerized licensure and certification examinations, the simulation study examines two test-length conditions: 60 and 80 items. For medical assessment, we reference the Next Generation NCLEX-RN, which delivers between 85 and 150 items per exam (Ignatavicius, 2021). In the realm of pilot certification, the Federal Aviation Administration's Airman Knowledge Testing Matrix specifies a 60-item exam for the Airline Transport Pilot Multiengine Airplane Canadian conversion and an 80-item Flight Navigator knowledge test (Administration, 2023). Two levels of sample size are

considered: 3,000 and 5,000. By mirroring the real-world conditions, the design ensures that the evaluation of the ADAM estimation pipeline reflects practical testing environments.

The hierarchical join model was used for data generation. The setting of parameters followed the reported estimates in Van der Linden (2007). Specifically, for the person parameters, $\sigma_{\theta,\tau} = 0.3$ and $\sigma_\tau^2 = 0.6$. The $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$ parameters follow the multivariate normal distribution with the mean vector as $\boldsymbol{\mu}_\mathcal{P} = (0, 0)$. For the item parameters, the covariance vector was set as $(\sigma_{b,a}, \sigma_{\beta,a}, \sigma_{\beta,b}, \sigma_{\alpha,a}, \sigma_{\alpha,b}, \sigma_{\alpha,\beta}) = (0, -0.11, 0.30, 0, 0.23, 0.18)$. The variance vector was set as $(\sigma_a^2, \sigma_b^2, \sigma_\beta^2, \sigma_\alpha^2) = (0.16, 0.5, 1, 0.20)$. The $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\beta}, \boldsymbol{\alpha}$ item parameters were simulated from the multivariate normal distribution with the mean vector as $\boldsymbol{\mu}_\mathcal{I} = (1, 0, \log(55), 0.7)$.

The MCMC approach served as the baseline. We adopted the prior distributions recommended by Van der Linden (2007) and implemented the model in the *nimble* R package (de Valpine et al., 2017), which compiles user-specified models to C++ for enhanced performance. Three independent Markov chains were run for 10,000 iterations each, discarding the first 5,000 as burn-in and applying a thinning interval of five to reduce autocorrelation. The ADAM estimation algorithm was coded in R, using entirely user-defined functions. The stopping rule was designed as the sum of absolute change in the parameter vector between successive iterations falling below a prespecified tolerance (here, $1e-6$). The step size was set as 0.01, and $w_1 = 0.95$, $w_2 = 0.9$.

To evaluate performance, we recorded the running time as a measure of computational efficiency, and we compared bias and root mean square error (RMSE) for each parameter under both estimation methods.

4.2 Results

The running time for both MCMC and ADAM estimation methods are summarized in Table 1. It shows that the ADAM estimator completes parameter estimation in over ten times less than the MCMC approach, indicating a marked gain in computational efficiency.

Table 1: The running time in minutes of MCMC and ADAM estimations.

N	J	MCMC	ADAM
3000	60	111.85	11.67
	80	191.77	15.68
5000	60	172.47	14.81
	80	274.75	16.99

Figures 1 and 2 display the bias and RMSE for first-level person and item parameters. Only first-level parameters are presented, as they serve as a baseline estimate, that is, if these are inaccurate, second-level estimates are unlikely to be reliable. For the discrimination (a), difficulty (b), and time-intensity (β) parameters, ADAM consistently outperforms MCMC, as evidenced by its lower bias and RMSE. In contrast, MCMC yields more accurate estimates for the time-variance (α) parameter. Estimates of person parameters show similar accuracy between the two methods. Although the bias of θ estimated by ADAM is larger than MCMC under the $N = 5000, J = 80$ condition, it is still acceptable. The RMSE of θ estimated by MCMC is higher at $N = 5000, J = 60$, but it is within acceptable bounds.

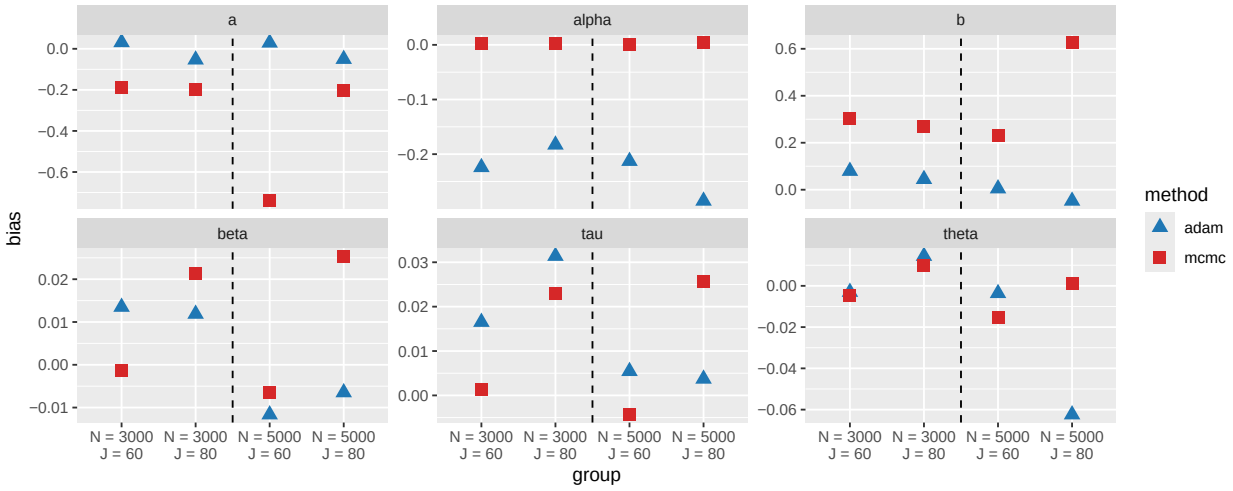

Figure 1: The bias of each first level parameter across conditions.

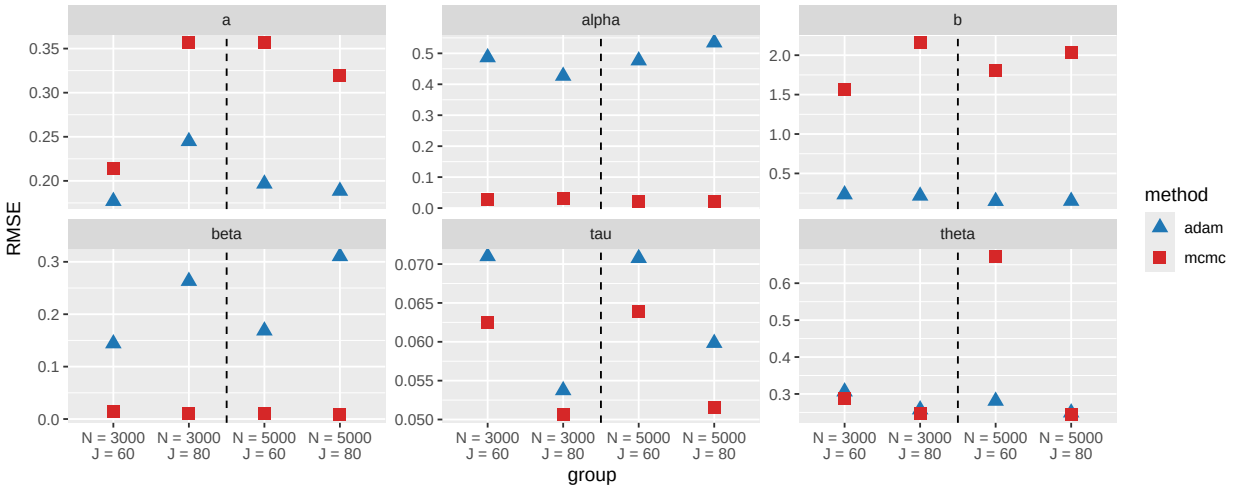

Figure 2: The RMSE of each first level parameter across conditions.

5 Discussion

The incorporation of response time into assessment models provides substantial insights into examinees' problem-solving processes and has driven a marked shift toward jointly modeling response accuracy and latency (De Boeck and Jeon, 2019; Van der Linden, 2006). Among the available approaches, the two-level hierarchical joint model stands out for its straightforward, intuitive design. Despite its appeal, parameter estimation for this model remains challenging: the estimation depends totally on MCMC, which can demand extensive computation time and may encounter convergence difficulties when applied to large-scale data. To address these issues, our project proposes an ADAM-based optimization algorithm specifically adapted to the hierarchical joint framework. Simulation study reveals that the ADAM estimator has over ten times efficiency higher than MCMC while achieving comparable accuracy in parameter recovery.

This project demonstrates the promising potential of applying the ADAM algorithm to estimate complex latent variable models. However, several important considerations remain. First, while ADAM offers faster computation and comparable accuracy, it should not be viewed as a complete replacement for MCMC. MCMC not only provides point estimates but also yields full posterior distributions, which can be valuable in many contexts. Future work is needed to develop methods for deriving standard errors within the ADAM framework. Second, the current implementation of ADAM constrains all discrimination parameters to be positive. In practice, however, negative discrimination estimates do occur and can indicate issues with item quality. In a pilot attempt to relax this constraint by fixing only item 1's discrimination parameter to 1, estimation accuracy remained unsatisfactory. Third, this study focuses on large-scale applications. It remains unclear how ADAM performs under conditions of small sample sizes or limited test lengths, which warrants further investigation.

References

Administration, F. A. (2023). *Pilot's Handbook of Aeronautical Knowledge (2025): FAA-H-8083-25C*. Simon and Schuster.

Bolsinova, M. and Tijmstra, J. (2018). Improving precision of ability estimation: Getting more from response times. *British Journal of Mathematical and Statistical Psychology*, 71(1):13–38.

Cheng, Y. and Shao, C. (2022). Application of change point analysis of response time data to detect test speededness. *Educational and psychological measurement*, 82(5):1031–1062.

De Boeck, P. and Jeon, M. (2019). An overview of models for response times and processes in cognitive tests. *Frontiers in psychology*, 10:102.

de Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Lang, D. T., and Bodik, R. (2017). Programming with models: writing statistical algorithms for general model structures with nimble. *Journal of Computational and Graphical Statistics*, 26(2):403–413.

Fox, J.-P. and Marianti, S. (2016). Joint modeling of ability and differential speed using responses and response times. *Multivariate behavioral research*, 51(4):540–553.

Guo, X., Jiao, Y., Huang, Z., and Liu, T. (2022). Joint modeling of response accuracy and time in between-item multidimensional tests based on bi-factor model. *Frontiers in Psychology*, 13:763959.

Ignatavicius, D. D. (2021). Preparing for the new nursing licensure exam: the next-generation nclex. *Nursing2024*, 51(5):34–41.

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

Maris, G. and Van der Maas, H. (2012). Speed-accuracy response models: Scoring rules based on response time and accuracy. *Psychometrika*, 77(4):615–633.

Paszke, A. (2019). Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Van der Linden, W. J. (2006). A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics*, 31(2):181–204.

Van der Linden, W. J. (2007). A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika*, 72(3):287–308.

Wang, S., Zhang, S., Douglas, J., and Culpepper, S. (2018). Using response times to assess learning progress: A joint model for responses and response times. *Measurement: Interdisciplinary Research and Perspectives*, 16(1):45–58.