

LLM-Based Pairwise Judgment for Math Item Parameter Modeling using Workflows and Agents

Suhwa Han and Frank Rijmen

Cambium Assessment

Correspondence: suhwa.han@cambiumassessment.com

Abstract

This study examines the feasibility of using large language models (LLMs) as judges for pairwise comparisons of item difficulty and to the extent which the resulting comparison outcomes recover banked difficulty parameters. The study in particular fine-tunes an instruction-tuned LLM to evaluate the impact of task-specific fine-tuning on the parameter prediction accuracy. The study also demonstrates an agentic approach to hyperparameter tuning of LLM-training task.

1 Introduction

Item parameter modeling in educational measurement refers to the prediction of item response theory (IRT) parameters using item content available—such as item text—prior to operational administration (Alkhuzaey et al., 2024). With recent advances in large language models (LLMs), researchers have increasingly leveraged LLMs to develop models that can predict item parameters using item content. One prominent line of work involves fine-tuning of encoder-based LLMs with regression heads using banked item parameters as labels (Maeda and Lu, 2026; Han et al., 2025; Yaneva et al., 2024). Other approaches exploit the advanced capabilities of foundation models, such as GPT-4 (OpenAI, 2023), by prompting them to directly elicit numerical ratings of item difficulty (Razavi and Powers, 2025). While each of these approaches has reported promising results, both have limitations. The encoder-based fine-tuning approaches are generally limited by smaller model sizes (hundreds of millions of parameters) compared to generative LLMs with multi-billion parameters. For prompting-based approaches with foundational models, prior work has shown that scalar ratings elicited from LLMs are prone to biased or compressed score distributions (Licht et al., 2025; Stureborg et al., 2024).

More recently, interest has emerged in using LLMs for pairwise comparison methods (Han and Rijmen, 2026; Li and Gushta, 2026; Smith and Rewley, 2026). Rather than eliciting absolute difficulty ratings from LLMs, these methods require only a series of relative judgments—which item in a pair is more difficult—a task generally considered more straightforward (Thurstone, 2017). The resulting comparison outcomes are then aggregated using a measurement model such as the Bradley-Terry model (Bradley and Terry, 1952) to recover a continuous latent difficulty scale. While pairwise comparison methods with human judges have previously been applied for item parameter prediction (Australian Council for Educational Research, 2022), applying these methods to large item pools with human judges remain practically challenging due to the quadratically growing number of required judgments. LLM-based pairwise judgments offer a scalable solution by automating the repetitive comparisons. Moreover, LLM-based pairwise methods may address aforementioned limitations of the existing LLM-based methods. First, they are well suited to large generative models because pairwise comparisons can be naturally formulated as next-token prediction tasks. Second, they can bypass the problem of compressed or biased score distributions associated with direct scalar ratings from LLMs as the latent scale is instead derived through a measurement model.

1.1 Study Purpose and Contributions

The purpose of the current study is to examine the feasibility of using LLMs as judges to automate pairwise comparison tasks and the extent to which item parameters can be recovered by aggregating LLM-generated pairwise outcomes using the Bradley-Terry model. Specifically, we fine-tune an open-source generative LLM for the pairwise comparison task using a dedicated training set with which the true pairwise outcomes from banked item

parameters serve as training labels. We then evaluate the fine-tuned model’s performance on the pairwise task by applying it to a separate development set and assessing the closeness of the latent scale derived from pairwise outcomes to the banked parameter values. Additionally, we compare the performance of the fine-tuned model to that of the off-the-shelf counterpart applied to the same development set via inference only (i.e., no task-specific fine-tuning).

To the authors’ knowledge, no prior study has fine-tuned an open-source generative LLM to develop LLM judges for the item parameter modeling. This distinction is particularly relevant for this pairwise comparison methods where computational overhead grows quadratically with the number of items to be compared. While foundational models could be theoretically applied, doing so at scale would result in substantial token costs, rendering the fine-tuning open-source models a more scalable alternative.

A further contribution of this work is the explicit characterization of the research pipeline as an *agentic system*. Following the taxonomy of (Anthropic, 2024), we distinguish *workflows*—in which LLMs are orchestrated through predefined paths—and *agents*—in which LLMs dynamically direct their own processes and tool usage—and describe relevant components of the pipeline according to this distinction. We intend this characterization to serve as early guidance for the systematic and efficient use of agentic systems in LLM-based research avenue in the field.

2 Using LLMs as Pairwise Judges for Item Parameter Modeling

Substituting human judges with LLM judges warrants special consideration. This section discusses two such considerations that informed the design of the current study’s methodology on item parameter modeling.

2.1 Positional Bias in LLMs as Pairwise Judges and the Swap Test

Prior studies on LLMs as pairwise judges have repeatedly reported systematic positional bias, whereby the model’s judgment is swayed by the order in which items are presented rather than by their substantive content (Wang et al., 2024; Shi et al., 2025). In the context of item parameter modeling, Han and Rijmen (2026) showed that this

positional bias persisted even when a foundational model was used for the comparison, and that it can be mitigated by prompting the model to generate explanatory texts before the verdict (Wang et al., 2024). As a metric to quantify a model’s positional bias, Shi et al. (2025) proposed a swap test in which every pair is evaluated twice with item presentation order reversed between passes. A comparison passes the swap test if the model selects the same item regardless of presentation order while inconsistent decisions indicate a positionally biased judgment.

2.2 Generative LLMs in Pairwise Judgments

Generative LLMs are trained autoregressively, meaning each token in a sequence is predicted conditioned on all preceding tokens. This unidirectional architecture naturally makes these models well-suited for text generation tasks, which can be elicited by wrapping inputs in structured, chat-formatted prompts. Pairwise comparison tasks, which require only a single discrete output—a token indicating which item is more difficult (e.g., "A" or "B")—can be naturally formulated as a next-token prediction task without any architectural modification required. At inference, the model’s judgment is obtained by selecting the token with the highest probability at the designated output position, if greedy decoding is chosen. Fine-tuning can be formulated in the same manner by using the true pairwise outcome as the training target for that same token position.

3 Methods

This section details the methods used in this study’s item parameter modeling pipeline, from data preparation, prompt design, model training and inference to Bradley-Terry aggregation of the resulting pairwise outcomes.

3.1 Data and Sampling

This study utilized 1,247 of 1,386 Grade 6 mathematics items drawn from a large-scale K-12 assessment. Each item was associated with banked IRT parameters calibrated from operational administration. The 1,247 items were partitioned into three subsets: a training set (975 items), an inner validation set used to select the best checkpoint during fine-tuning (133 items), and a development set used for model selection and for reporting the results (139 items). The remaining 139 items constitute a test partition that was reserved and not evaluated

in the present study (see limitations in Section 6). The majority of items were dichotomous (1,213 items) with a small subset scored polytomously (34 items). Dichotomous items were calibrated using two-parameter logistic (2PL) model (Birnbaum, 1968) while polytomous items were calibrated with generalized partial credit model (GPCM) (Muraki, 1992). Appendix A shows the descriptive statistics of the banked difficulty parameters.

Evaluation Target: Banked IRT Difficulty.

The evaluation target in this study was the banked IRT difficulty parameter. Because the 2PL model is a special case of GPCM, 2PL-calibrated parameters were treated within the same GPCM parameterization in this study. GPCM describes the probability of an examinee with a latent trait level θ to obtain a score of $v \in \{0, 1, \dots, m_i\}$ for item i as:

$$p_{iv} = \frac{\exp(\sum_{r=0}^v Da_i(\theta - b_i + d_{ir}))}{\sum_{c=0}^{m_i} \exp(\sum_{r=0}^c Da_i(\theta - b_i + d_{ir}))},$$

where a_i and b_i respectively denote the overall discrimination and difficulty parameters for item i , and d_{ir} represents the step parameter for the category r for the item. The GPCM parameters used in this study were estimated under the constraints $d_{i0} = 0$ and $\sum_r d_{ir} = 0$. For dichotomous items, the model reduces to 2PL. Accordingly, the item-level difficulty parameter b_i was used as the evaluation target for both dichotomous and polytomous items.

Augmented Item Text. As the input to feed to the LLM prompt for the pairwise comparison task, we constructed a single augmented item text per item by concatenating three distinct text sources: (i) *item text*, the raw text of the question and stimulus presented to students, (ii) *accessibility context*, additional item context from accommodation features that provides information often not captured in item text alone (e.g., text descriptions of images for Braille users), and (iii) *scoring rubric* that contains the answer key or the human-readable scoring rubric. Appendix B shows an illustrative example of this augmented item text for a synthetic item.

3.2 Language Model

Throughout this study, we used Gemma-4-12B (Gemma Team, 2026), an open-source, instruction-tuned generative language model with approximately 12 billion parameters. As an instruction-tuned model, the model takes inputs formatted as a structured chat conversation consisting of system,

user and assistant turns. We used this chat format to structure every generation step in the pipeline.

3.3 LLM Difficulty Breakdown for Bias-free Pairwise Comparison

Motivated by the finding of Wang et al. (2024) that positional bias can be mitigated by eliciting explanations from LLMs before a verdict, we adopted a variant of chain-of-thought (CoT) prompting (Zhou et al., 2022) where the model produces a structured *difficulty breakdown* for each item prior to comparison. The breakdown is generated through a sequence of prompts, each addressing a distinct perspective on the factors that may contribute to item difficulty. The final breakdown contains a sequence of the model’s responses from each step, incorporating the responses of preceding prompts as context. This breakdown structure not only ensures that explanatory reasoning precedes the verdict, but may also yield a more comprehensive difficulty breakdown based on multiple perspectives.

Specifically, our CoT prompting strategy consisted of three stages: (i) *expert perspective* where the model takes the perspective of a math content expert evaluating the item’s content and cognitive demand, (ii) *student perspective* where the model takes the perspective of a typical student and examines the required prerequisite and the effect of response format, and (iii) *synthesis* where the model integrates the preceding responses into an overall difficulty level and drivers. Appendix C and D respectively show examples of our CoT prompts for a synthetic item and the resulting final breakdown.

3.4 Pairwise Judgment, Swap Test, and Tie-breaking

Once structured breakdowns had been generated for individual items, pairwise judgments were elicited by presenting the model with the breakdowns of two items and asking it to select the more difficult item based on the breakdowns. Each pair was judged twice with the presentation order reversed. A swap test was then applied to assess the model’s predisposition to positional bias, checking whether the model selected the same item as more difficult across both passes despite the reversed order. When the two passes disagreed, the pair was flagged as positionally biased and resolved through a dedicated tie-breaking prompt. Appendix E and F respectively show examples of the pairwise comparison prompt and the tie-breaking prompt used when a pair fails the swap test.

3.5 Estimating Latent Difficulty via Bradley-Terry Model

Once pairwise judgments had been obtained for all evaluated item pairs, the Bradley-Terry model (Bradley and Terry, 1952) was used to aggregate the outcomes to place the items on a continuous latent difficulty scale. The model assumes that each item i has a latent strength parameter β_i and describes the probability that item i is judged more difficult than item j as

$$P(i > j) = \frac{\exp(\beta_i)}{\exp(\beta_i) + \exp(\beta_j)}.$$

The model parameters were estimated using the `choix` Python package, yielding each item’s position on a continuous latent difficulty scale. Because the Bradley-Terry estimates are identified only up to an additive constant, the parameters were linearly rescaled to match the mean and standard deviation of the Bradley-Terry estimates to those of the banked difficulty parameters.

3.6 Fine-tuning a Generative Model for Pairwise Comparison

The difficulty breakdown, pairwise judgment and the Bradley-Terry latent difficulty estimation described in Sections 3.3 to 3.5 together constitute the *inference* pipeline. For training, we targeted a specific component of the pipeline—the pairwise judgment step in 3.4—as the trainable signal in scaling the difficulty parameters essentially come from the pairwise comparison task.

Training Objective. The training examples from the designated training data were prepared using the same chat format used at inference (see Appendix E): a user turn containing the pairwise comparison prompt followed by an assistant turn containing the target response. For training, the verdict token in the target response was derived from the banked difficulty parameters. The assistant turn, on the other hand, included a fixed, templated sentence identical across all examples. The loss was computed over the tokens in the entire assistant turn using the standard next-token cross-entropy objective.

Training Data Sampling. One key consideration in training on a pairwise task is the number of training samples. Because the label is defined at the pair level, the number of possible training pairs grows quadratically, reaching as high as $\binom{n}{2}$ pairs for n items. With 975 training items, the size

of training samples can amount to 474,825 pairs. Training on these many samples is not only unnecessary but also inefficient as each item recurs across $n - 1$ pairs and incurs highly redundant information. To address this, we adopted a structured sampling approach that curates a subset.

Our structured sampling was conducted in three steps: (i) *margin filtering*, (ii) *stratification by difficulty gap* and (iii) *stub matching sampling*. The margin filtering considers the idea that a pair whose difficulty difference falls within the estimation error of the banked parameters provides noise rather than signal. For each candidate pair (i, j) , we thus computed a standardized gap $d_{ij} = |b_i - b_j| / \sqrt{\text{se}_i^2 + \text{se}_j^2}$, and retained only pairs with $d_{ij} \geq 1$. Next, we stratified the remaining pairs by their difficulty gap $|b_i - b_j|$ into ten strata using equal-percentile boundaries. Finally, within this stratified structure, we sampled pairs such that each item appeared approximately $K = 80$ times in the training set. This sampling was done through a stub-matching procedure (Newman et al., 2001), where each item is assigned K stubs distributed across the strata and stubs are randomly paired with other stubs within the stratum.

Training Configuration. To fine-tune the Gemma model, we used Low-Rank Adaptation (LoRA; Hu et al., 2021) with a rank of 16 and a scaling factor of 32, applying adapters to the attention and projection matrices. We used a batch size of 1 with gradients accumulated over 16 steps. To guard against overfitting while selecting the best-performing model, we examined pairwise prediction accuracy—specifically accuracy on the verdict token—on the validation set of 133 items every 100 optimization steps. Training was terminated early if validation accuracy did not improve over 10 consecutive evaluations, and the final model was selected from the checkpoint with the highest accuracy as long as it shows decent diagnostics (see Figure 1 for details). In addition to early stopping, training was capped at 1 epoch as our preliminary analysis showed that completing a full epoch was unnecessary and led to degraded performance. Given that individual items reappeared many times across the training set in pairwise comparison, it is reasonable that optimal performance was achieved before a full epoch.

Learning Rate Optimization Using Agents. All the hyperparameters above were fixed a pri-

ori, while the *learning rate*, which arguably is one of the most important hyperparameter, was selected through a dedicated search procedure using an agent, which will be detailed in Section 4.2.

4 Workflows and Agents in the Research Pipeline

We now characterize the current study through a lens of agentic system, following the framework of Anthropic (2024). The pipeline implemented here requires a continuous stream of LLM calls. When a task is complex enough to warrant multiple LLM calls and yet its steps can be specified deterministically, a *workflow* is appropriate. An *agent* is reserved for open-ended tasks whose steps depend on decisions that cannot be specified in advance and the LLM must dynamically determine how to proceed based on information obtained during execution. Figure 1 depicts how both workflows and an agent are incorporated into the current research pipeline.

4.1 Workflows in the Pipeline

As shown in Figure 1, most of the LLM components in the current pipeline constitute a workflow: the pipeline proceeds as a linear chain where each LLM call processes the output of the preceding step along a deterministic path. For example, the pairwise judgment operates on the difficulty breakdowns produced from the preceding LLM call while the tie-breaker gets invoked conditioning on the preceding swap test results. Importantly, however, this branching decision is entirely governed by the prescribed script rather than by the LLM itself. That is, although the LLM outputs provide information required by subsequent steps, the model itself never determines which action is executed next.

4.2 An Agent for Learning Rate Optimization

An *agent*, in contrast, dynamically determines its course of action. Learning rate optimization provides a natural setting for such an agentic approach. Selecting an appropriate learning rate is a critical component of LLM fine-tuning and yet such hyperparameter tuning has traditionally relied on manual experimentation (Bischi et al., 2023; Chaudhry and Mizuno, 2024). Although grid search offers a more systematic alternative, it can become computationally prohibitive for large generative models and may expend unnecessary resources for evaluating

candidates that are unlikely to perform well. An agent can offer an adaptive solution. Rather than exhaustively sweeping a predefined set of candidates or relying on repeated human intervention, the agent can propose the next learning rate by intelligently interpreting evidence from previous runs, such as validation performance and training diagnostics.

Figure 1 depicts the learning rate search procedure in which the agent dynamically directs each successive learning rate. Operating on a specification written in a natural language (e.g., a markdown file) that defines its goal, constraints, and guidance on what evidence to weigh, the agent proceeds through a dynamic loop until it determines that sufficient experiments have been conducted to identify an optimal learning rate. Specifically, at each iteration, the agent invokes a *sub-workflow* that (i) conducts a smoke test to verify that the verdict token generation is done properly and (ii) performs the Bradley-Terry estimation on the development set to provide information about predictive performance. The agent then evaluates the outputs of this sub-workflow together with training diagnostics, such as gradient norms, to assess training stability and decides the next learning rate to test. For example, if substantial spikes in gradient norms are observed, the agent may suggest a lower learning rate.

5 Results

We next present the model performance results. Model performance was evaluated by examining how closely the latent scale estimated from the model’s pairwise outcomes through the Bradley-Terry estimation reproduced the banked difficulty parameters. The Spearman and Pearson correlations and the root mean squared error (RMSE) were computed to gauge this. In addition to prediction accuracy, we report swap-test pass rates (i.e., the proportion of item pairs for which the model consistently selected the same item despite the reversed presentation order) as a metric to assess the model’s robustness to positional bias. All results reported below were computed on the development set (except for verdict-token accuracy computed on the inner validation set).

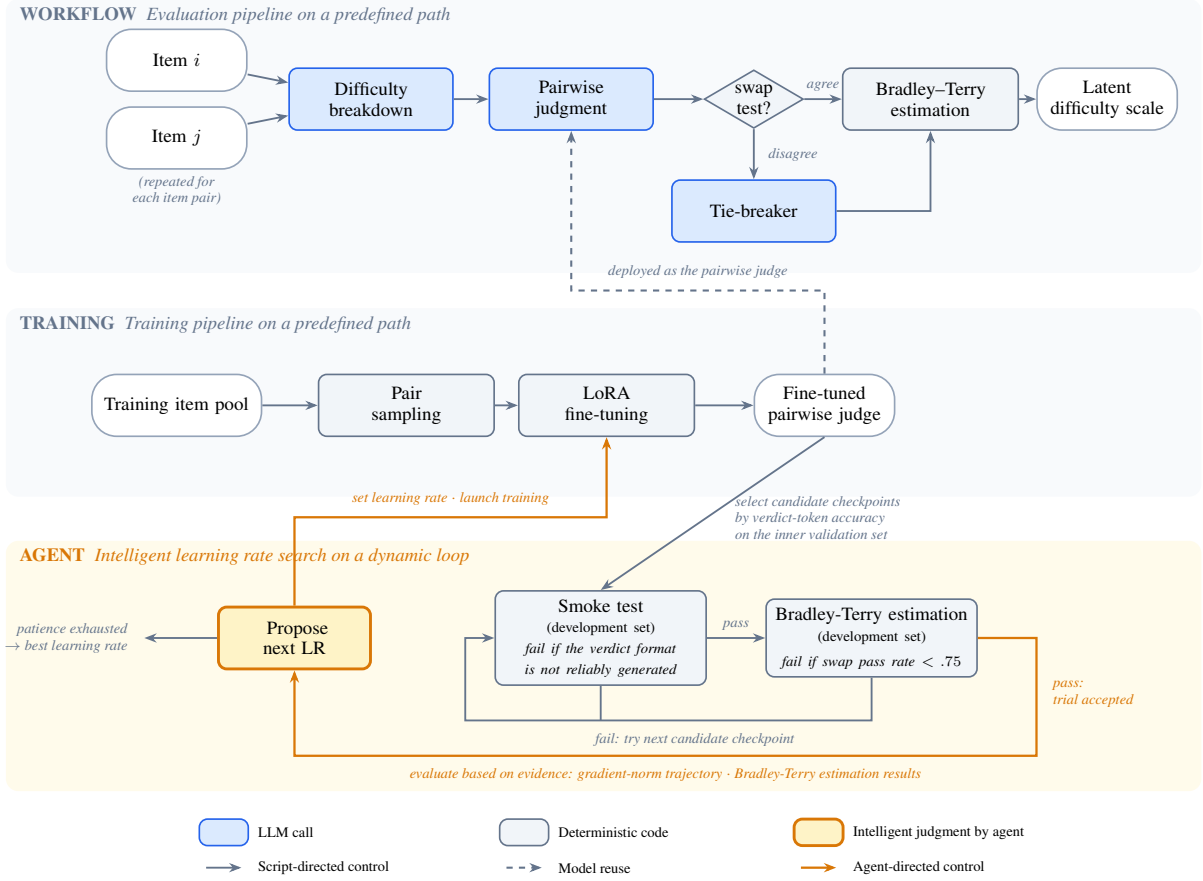


Figure 1: The research pipeline described as an agentic system

5.1 Difficulty Prediction Across the Learning Rate Search by Agent

Table 1 summarizes the difficulty prediction results across the learning rates proposed by the agent. The results show that the agent successively halved the learning rate, despite an apparently acceptable prediction outcome at Trial 1 (i.e., $\rho = .702$). This decision was driven by the training diagnostics: extreme spikes in the gradient norm and evaluation loss together with the failure of two candidate checkpoints on the smoke test led the agent to conclude that training was unstable at that learning rate (see Appendix G for details). As the learning rate was halved, the correlations improved monotonically, from .702 to .766 in Spearman correlation and from .661 to .730 in Pearson correlation. Verdict-token accuracy increased monotonically as well although the gain from Trial 2 to 3 was marginal. Swap-test pass rates were uniformly high across all trials, indicating that the fine-tuned models were robust to positional bias.

Trial	Learning rate	ρ	r	Swap pass	Verdict-token accuracy
1	1×10^{-4}	.702	.661	.922	.813
2	5×10^{-5}	.747	.709	.914	.824
3	2.5×10^{-5}	.766	.730	.925	.826

Table 1: Difficulty prediction results across the learning rate search by the agent

5.2 Difficulty Prediction by Model

Table 2 presents difficulty prediction results for three models. In addition to the models based on the pairwise comparison method, results based on a fine-tuned DeBERTa model following the method of Han et al. (2025) are presented for comparison. The fine-tuned pairwise model predicted the banked difficulty parameters most closely ($\rho = .766$, $r = .730$, RMSE = 1.057), with the encoder fine-tuning method slightly behind on all reported metrics. By contrast, the effect of task-specific fine-tuning within the pairwise comparison framework was pronounced. Applying the identical evaluation pipeline with an off-the-shelf model achieved only $\rho = .448$ and $r = .401$. The higher swap-test pass rate of the fine-tuned model (.925 vs. .883)

further suggests that fine-tuning also improved the positional consistency of the generative model.

Model	ρ	r	RMSE	Swap pass
Pairwise, fine-tuned	.766	.730	1.057	.925
Pairwise, off-the-shelf	.448	.401	1.573	.883
Encoder fine-tuning	.733	.693	1.127	—

Table 2: Difficulty prediction results by model

6 Discussion and Limitations

This study examined the feasibility of using LLMs as judges for pairwise comparison of item difficulty and the extent to which the resulting outcomes can predict banked difficulty parameters when aggregated through the Bradley-Terry model. Using operational Grade 6 mathematics items from a large-scale K–12 summative assessment, the results showed that banked difficulty parameters can be recovered with adequate accuracy when the LLM was appropriately fine-tuned for the pairwise comparison task. In addition, the study characterized the research pipeline through the lens of an agentic system and demonstrated how the learning-rate selection task can be delegated to an agent. Guided by a specification document written in a natural language, the learning-rate-search agent autonomously selected successive candidate values by interpreting training dynamics and downstream prediction performance. This approach provides an adaptive alternative to manual tuning or exhaustive grid search, with the potential to improve both the efficiency of hyperparameter optimization and the use of computational resources in LLM fine-tuning.

This study is not without limitations. First, the current study remains in progress, and the fine-tuned model has not yet been evaluated on a completely held-out test set. Evaluation on the held-out set will provide a more rigorous assessment of generalization and help determine whether the learning-rate exploration by agent translates to improved performance beyond the development data. Second, the procedure evaluated in this study compares all possible item pairs within a pool, making it most directly applicable to placing an existing set of items on a common scale. For the more common scenario of placing a smaller number of new items relative to an existing calibrated pool, an anchoring design could instead be used where new items are compared against items with known difficulty values and the anchor-item parameters are fixed at

their known values during the model estimation.

Disclaimer

Generative AI tools, including Claude and Gemini, were used to assist with writing and coding during the development of this study. All AI-assisted content and code were reviewed and validated by the authors, who retain full responsibility for the final work.

References

- Samah AlKhazaey, Floriana Grasso, Terry R Payne, and Valentina Tamma. 2024. Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, 34(3):862–914.
- Anthropic. 2024. Building effective agents. <https://www.anthropic.com/engineering/building-effective-agents>. Anthropic Engineering Blog. Accessed: 2026-09-04.
- Australian Council for Educational Research. 2022. [Pairwise comparison method: Concept note](#). Technical report, Australian Council for Educational Research.
- Allan Birnbaum. 1968. Some latent trait models. *Statistical theories of mental test scores*.
- Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, and 1 others. 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2):e1484.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Zan Chaudhry and Naoko Mizuno. 2024. Exptest: Automating learning rate searching and tuning with insights from linearized neural networks. *arXiv preprint arXiv:2411.16975*.
- Gemma Team. 2026. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*.
- Suhwa Han and Frank Rijmen. 2026. Scaling item difficulty using llm-based pairwise judgments: An application of the bradley-terry-luce model. Paper presented at the annual meeting of the National Council on Measurement in Education April 2026, Los Angeles, CA.
- Suhwa Han, Frank Rijmen, Allison Ames Boykin, and Susan Lottridge. 2025. Leveraging fine-tuned large

- language models in item parameter prediction. In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Full Papers*, pages 250–264.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Lanrong Li and Matthew Gushta. 2026. Using ai judges’ paired comparisons to estimate item difficulty. Paper presented at the annual meeting of the National Council on Measurement in Education April 2026, Los Angeles, CA.
- Hauke Licht, Rupak Sarkar, Patrick Y Wu, Pranav Goel, Niklas Stoehr, Elliott Ash, and Alexander Miserlis Hoyle. 2025. Measuring scalar constructs in social science with llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32132–32159.
- Hotaka Maeda and Yikai “EK” Lu. 2026. [Multimodal test item parameter prediction from text, images, and metadata: Fusing together ai vision and language models](#). *Educational and Psychological Measurement*, 0(0):00131644261460779.
- Eiji Muraki. 1992. A generalized partial credit model: Application of an em algorithm. *Applied psychological measurement*, 16(2):159–176.
- Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. 2001. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118.
- OpenAI. 2023. [GPT-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Pooya Razavi and Sonya J Powers. 2025. Estimating item difficulty using large language models and tree-based machine learning algorithms. *arXiv preprint arXiv:2504.08804*.
- Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314.
- Nate Smith and Kelly Rewley. 2026. Predicting item statistics: Thurstone scaling with large language models. Paper presented at the annual meeting of the National Council on Measurement in Education April 2026, Los Angeles, CA.
- Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*.
- Louis L Thurstone. 2017. A law of comparative judgment. In *Scaling*, pages 81–92. Routledge.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and 1 others. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 9440–9450.
- Victoria Yaneva, Kai North, Peter Baldwin, Le An Ha, Saed Rezayi, Yiyun Zhou, Sagnik Ray Choudhury, Polina Harik, and Brian Clauser. 2024. [Findings from the first shared task on automated prediction of difficulty and response time for multiple-choice questions](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 470–482, Mexico City, Mexico. Association for Computational Linguistics.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Descriptive Statistics of the Banked Difficulty Parameters

	Train ($n = 975$)	Inner validation ($n = 133$)	Development ($n = 139$)
Min	-3.934	-2.396	-1.900
1st Qu.	0.105	-0.125	-0.065
Median	0.921	0.853	0.678
Mean	0.933	0.944	0.856
3rd Qu.	1.720	1.772	1.637
Max	7.522	9.158	6.778
SD	1.400	1.593	1.442

Table 3: Descriptive statistics for the banked item difficulty parameter (b_i) across the subsets

B Example of the Augmented Item Text Input (Synthetic Item)

Augmented item text input

[Item Text: the question or prompt as presented to the student]

A bag contains red and blue marbles in a ratio of 3 to 5. If there are 25 blue marbles, how many red marbles are in the bag?

[Accessibility Context: spoken and Braille descriptions of images, graphs, and interactive elements not visible in the item text]

A double number line is shown. The top line is labeled “Red marbles” with tick marks at 0, 3, 6, 9, 12, and 15. The bottom line is labeled “Blue marbles” with tick marks at 0, 5, 10, 15, 20, and 25. Corresponding values are vertically aligned.

[Scoring Rubric: the answer key for simple items, or a full scoring rubric describing acceptable responses for complex item types]

Correct answer: 15 . 1 point for the correct value; 0 points otherwise. Equivalent unsimplified forms are not accepted.

Figure 2: Illustrative example of the augmented text input on a synthetic item

C Example of the Difficulty Breakdown Prompt Chain (Synthetic Item)

Difficulty breakdown prompt chain

Stage 1 – Expert perspective

User: Analyze the following math item:
{Item Text}

You are a math content expert. Analyze this item from an expert perspective:
(1) **Content Domain:** what concept(s) does this assess, and is it easy or hard for the intended students? (2) **Cognitive/Competency Demand:** how much structured reasoning, coordination of steps, or abstraction is required? (3) **Representational/Processing Demand:** how much translation between representations (notation, graphs, tables) is required? Be concise; use bullet points; max 120 words; do NOT solve the problem.

Assistant: - Content Domain: Ratios and proportional reasoning (Grade 6); on-grade, moderately familiar. - Cognitive Demand: Set up a part-to-part ratio and scale it to a given quantity; a few coordinated steps. - Representational Demand: Must read a double number line and relate the visual to the numeric ratio.

Stage 2 – Student perspective

User: Based on your previous analysis:
{previous_answers}

Now shift perspective. Imagine a typical student encountering this item:
(1) **Prior/Prerequisite Knowledge:** is the required knowledge common, emerging, or fragile for a typical student? (2) **Response Format:** does the format (multiple choice vs. open-ended) add or remove difficulty? Be concise; max 100 words.

Assistant: - Prior Knowledge: Equivalent-ratio reasoning is emerging for many Grade 6 students; scaling a ratio can be fragile. - Response Format: Open-ended numeric entry removes answer-elimination cues, adding difficulty relative to multiple choice.

Stage 3 – Synthesis

User: Based on all previous analysis:
{previous_answers}

Synthesize the overall difficulty for a typical student. State: (1) difficulty level (LOW / MODERATE / HIGH), (2) the 2 most significant difficulty drivers. Max 80 words.

Assistant: Difficulty level: MODERATE. Most significant drivers: (1) scaling a part-to-part ratio to a target quantity; (2) integrating the double-number-line representation with the numeric ratio.

Figure 3: Illustrative example of the three-stage difficulty breakdown prompt chain on a synthetic item

D Example of the Final Difficulty Breakdown (Synthetic Item)

Final difficulty breakdown

- Content Domain: Ratios and proportional reasoning (Grade 6); on-grade, moderately familiar.
- Cognitive Demand: Set up a part-to-part ratio and scale it to a given quantity; a few coordinated steps.
- Representational Demand: Must read a double number line and relate the visual to the numeric ratio.
- Prior Knowledge: Equivalent-ratio reasoning is emerging for many Grade 6 students; scaling a ratio can be fragile.
- Response Format: Open-ended numeric entry removes answer-elimination cues, adding difficulty relative to multiple choice.

Difficulty level: MODERATE.

Most significant drivers: (1) scaling a part-to-part ratio to a target quantity;
(2) integrating the double-number-line representation with the numeric ratio.

Figure 4: The final difficulty breakdown on a synthetic item

E Example of the Pairwise Comparison Prompt

Pairwise comparison prompt

System: You are a precise and detail-oriented educational assessment expert.

User: You are comparing the difficulty of two math items based on detailed analyses.

ANALYSIS A: {difficulty breakdown of Item A}

ANALYSIS B: {difficulty breakdown of Item B}

INSTRUCTIONS:

- Based ONLY on the analyses above, determine which item (A or B) is MORE DIFFICULT for typical students
- Consider ALL difficulty factors mentioned in the analyses
- The order of presentation (A vs B) is completely arbitrary and should NOT influence your decision
- You MUST choose either A or B

RESPONSE FORMAT:

FINAL ANSWER: <Your choice of "A" or "B">

JUSTIFICATION: <1-2 sentences explaining the key difficulty factors>

Assistant: FINAL ANSWER: B

JUSTIFICATION: Both items are moderate, but Item B additionally requires multiplying two fractions and interpreting the product as an area, which is more error-prone than scaling a single part-to-part ratio.

Figure 5: Illustrative example of the pairwise comparison prompt

F Example of the Tie-Breaking Prompt

Tie-breaking prompt

System: You are a precise and detail-oriented educational assessment expert.

User: You are resolving a disagreement about which of two math items is more difficult. Two previous comparisons of the same pair produced OPPOSITE conclusions:

- Comparison 1 conclusion: Item A is more difficult, owing to integrating the double-number-line representation with the numeric ratio.
- Comparison 2 conclusion: Item B is more difficult, because multiplying two fractions is more error-prone.

A: {difficulty breakdown of Item A}

B: {difficulty breakdown of Item B}

INSTRUCTIONS:

- The two previous comparisons disagreed. Look MORE CAREFULLY to break the tie.
- Focus on the factors most predictive of student difficulty: response format (multiple choice vs open-ended), prior-knowledge fragility, and cognitive demand.
- You MUST choose either A or B. Do NOT say they are equal.

RESPONSE FORMAT:

FINAL ANSWER: <Your choice of “A” or “B”>

JUSTIFICATION: <1-2 sentences explaining what tipped the balance>

Assistant: FINAL ANSWER: B

JUSTIFICATION: Both items share open-ended formats and emerging prerequisites, but Item B’s two-step fraction multiplication imposes a higher cognitive demand than scaling a single ratio.

Figure 6: Illustrative example of the tie-breaking prompt

G Training Diagnostics from the First Learning Rate Trial (1×10^{-4})

Step	Training loss	Maximum gradient norm	Evaluation loss	Verdict-token accuracy
100	2.308	114.52	.022	.721
200	0.024	0.82	.019	.769
300	0.021	0.99	.018	.819 [†]
400	0.095	14.59	.020	.733
500	0.018	4.94	.020	.770
600	0.017	1.56	.019	.804
700	0.069	66.69	.021	.758
800	0.128	408.75	.823	.468
900	0.045	226.98	.017	.806
1000	0.014	3.88	.021	.786
1100	0.013	0.56	.021	.813 [†]
1200	0.011	0.41	.019	.818 [†]
1300	0.009	0.69	.019	.813

Table 4: Diagnostics from the first learning-rate trial (1×10^{-4}) as available to the agent. Training loss was the mean and the gradient norm the maximum over the 100 optimization steps preceding each evaluation. Early stopping terminated the run at step 1300 of a 2071-step ceiling. [†] marks the three checkpoints with the highest verdict-token accuracy, which the agent screened in descending order. Those at steps 300 and 1200 failed the smoke test, and the checkpoint at step 1100 was therefore selected as the final model for this trial.