

Predicting Multiple-Choice Question Difficulty and Distractor Selection with Large Language Models

Jonathan Koff¹, Lisa Zhang², Juho Leinonen^{3,*} and Paul Denny⁴

¹*Infinite Pod*

²*University of Toronto*

³*Aalto University*

⁴*The University of Auckland*

Abstract

Estimating the difficulty of questions before administering assessments can be useful for computing instructors, both to calibrate questions and improve them. However, this problem is challenging. We investigate whether large language models (LLMs) can serve as “synthetic students” to address two aspects of this problem, evaluating LLMs’ ability to (1) rank CS1 multiple-choice questions by difficulty and (2) predict which distractors students select. Crucially, we systematically vary contextual information (e.g., course descriptions, prior test distributions) and output format (e.g., rank orderings, probability distributions) across multiple model families. Our results reveal a mixed but informative picture. On (1) difficulty ranking, the best LLM configuration outperforms an ensemble of human raters ($\rho = 0.665$ vs. $\rho = 0.577$ Spearman correlation with ground truth). Surprisingly, output format matters more than the amount of context provided. Performance also varies significantly across LLMs. However, on (2) distractor prediction, humans maintain an advantage (52% vs. 36% accuracy). Matching fine-grained psychometric properties—such as sensitivity to answer-option ordering—remains particularly elusive. These findings suggest that simulating student cognition on multiple-choice questions is genuinely difficult, yet the strong performance of some configurations indicates the problem is tractable.

Keywords

synthetic data generation, multiple-choice questions, estimating difficulty, generative AI, large language models

1. Introduction

Creating effective multiple choice assessments is often routine for computing instructors, particularly in large courses. However, it is surprisingly challenging to estimate question difficulty [1, 2]. In Simon et al. [3], while the examiner’s estimated difficulty of a CS1 test correlated moderately with student performance, the estimates of other educators correlated more strongly with actual performance; it is easy to overlook issues in wording and distractor choice. Good distractors are important to question quality [4], but are difficult to write [5]; there are many guidelines [6, 7] on their development and assessment. Identifying such issues *before* administering an assessment is particularly valuable: in large CS1 courses serving hundreds of students, a flawed question is difficult to revise after the fact, and a careful expert pre-review is often costly or unrealistic.

Large language models (LLMs) offer a potential new approach: they can potentially be used to simulate how real students might respond to multiple choice questions in tests. Prior work has shown that LLMs can effectively adopt various roles [8], and they have been used to generate synthetic buggy code with patterns seen in students [9]. Furthermore, LLMs are trained on large corpora that include student-generated content, such as programming forums, Q&A sites, and online coursework. Thus, they may have internalized common beginner misconceptions, making them plausible as proxies for novice student reasoning. While there is work on evaluating LLMs’ ability to answer multiple choice

CSEDM’26: 10th Educational Data Mining in Computer Science Education (CSEDM) Workshop, June 28, 2026, Seoul, South Korea

*Corresponding author.

✉ jonathan@jkoff.ca (J. Koff); lc Zhang@cs.toronto.edu (L. Zhang); juho.2.leinonen@aalto.fi (J. Leinonen);

paul@cs.auckland.ac.nz (P. Denny)

0009-0000-9168-592X (J. Koff); 0000-0002-7302-6530 (L. Zhang); 0000-0001-6829-9449 (J. Leinonen); 0000-0002-5150-9806 (P. Denny)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

questions (e.g., for high-profile exams like medical exams [10]), research has generally focused outside of computing.

In this paper, we investigate whether LLMs can be used in place of real students or instructors to (1) reliably estimate question difficulty by ranking questions from easiest to hardest, and (2) predict which distractors students are most likely to select. We focus on out-of-the-box LLM tools, using contextual information that instructors will realistically have. We compare LLM performance against an ensemble of human raters with CS backgrounds, providing a meaningful point of comparison for what expert judgment can achieve without AI assistance.

A key contribution in this paper is that we systematically vary the contextual information provided to the models (e.g., course descriptions, last year's test distributions) and the output format requested (e.g., rank orderings, probability distributions, answers for each simulated student), evaluating performance across multiple LLM models. We prioritize input contexts that are realistic for an instructor to have, and output formats that can be used downstream to compute key metrics, e.g., distractor usage and other measures like the discrimination index, although we leave the latter for future work.

2. Related Work

Synthetic Data Generation in Education. Beyond its usefulness, synthetic data generation addresses privacy, data scarcity, and reproducibility challenges in learning analytics [11]. Zhan et al. [12] compared synthetic data generators on tabular student data, finding that top performers produce realistic data for downstream predictive tasks. However, Flanagan et al. [13] showed that models trained on fine-grained synthetic learner data may not transfer reliably to real-world settings. Recent work explores LLMs for richer synthetic educational data. Khalil et al. [14] showed LLM-based generators can rival traditional tabular synthesis methods. For programming submissions, MacNeil et al. [15] found that carefully engineered prompts align synthetic buggy code with authentic student patterns, and follow-up work confirmed LLM-generated incorrect submissions can match real data when evaluated via test case outcomes [9]. LLMs have also been used to generate MCQs directly; in medical education, AI-generated questions can be comparable to human-authored ones [10].

Predicting Question Difficulty. Question difficulty is traditionally estimated via classical test theory (CTT) or item response theory (IRT), which require sufficiently large samples of examinees [16, 17, 18]. This motivates predicting item characteristics from content to reduce field testing costs. Pre-LLM work used linguistic features and learned representations to predict difficulty from item text. Hsu et al. [19] modeled MCQ components using word embeddings; Ha et al. [20] predicted medical MCQ difficulty controlling for reading level; Xue et al. [21] showed transfer learning improves prediction for medical MCQs. Recent surveys synthesize these approaches and their challenges [22, 23], and the BEA 2024 APIDIRT shared task provides standardized benchmarks [24].

LLMs enable a new paradigm where models act as respondents to approximate field-testing. Liu et al. [25] showed that LLM ensembles can approximate human ability distributions and yield item parameters aligning with human-calibrated data. Maeda [26] fine-tuned transformers under IRT to predict option-selection probabilities, achieving strong correlation with human data ($r = .82$) though below human-response baselines. Park et al. [27] proposed LLaSA, which uses multiple LLMs and prompting methods to represent students at different ability levels and estimates question difficulty by applying IRT to LLM-generated responses. Benedetto et al. [28] prompted LLMs to answer MCQs while simulating learners of different skill levels, finding that the approach can produce increasing accuracy across simulated levels, but that prompts may not generalize reliably across LLMs. Uto et al. [29] used QA systems as virtual test-takers with IRT linking, outperforming text-based predictors. LLMs have also appeared in shared-task systems via data augmentation [24, 30]. Related work on distractor quality shows that generating effective distractors requires pedagogical expertise to capture authentic student misconceptions [6], and while LLMs can generate plausible distractors, they often struggle to reflect students' typical reasoning errors [31, 32].

Student Modeling and Performance Prediction. A related but distinct line of research models

student performance from observed learner interaction histories, often to predict correctness on future items using latent ability models or sequence-based approaches that capture learning over time [33, 34, 35]. In contrast to difficulty prediction from item text alone, these methods leverage response histories to infer student knowledge states and item characteristics [33, 35]. Recent work also explored deep learning approaches to knowledge tracing, with mixed evidence on when they provide consistent advantages over strong classical baselines [34, 36, 37].

More recently, LLM-based work has simulated student profiles to support item evaluation without requiring observed response histories. Nguyen et al. [38] highlight that while LLMs can reliably assess surface-level properties like topic coverage, they struggle to predict psychometric properties such as item difficulty, discrimination, and distractor efficiency without explicit student modeling. Lu and Wang [39] model students using the Knowledge-Learning-Instruction framework, defining student profiles based on the knowledge components that a student has mastered, is confused about, or has no evidence of knowing, and use these profiles to generate student answers. Our approach differs in the context provided to the LLM, focusing on information to which an instructor has ready access. Our contribution is also in exploring this context, along with the output format—characteristics that are under-explored.

3. Methods

We use data from an introductory programming course (CS1) in Matlab and C at The University of Auckland. In particular, we use a multiple choice test completed by 1013 students in the year 2025. The test contains 25 multiple choice questions, each with 5 possible answers; a small proportion of responses (0.37% overall) were left blank. Four versions of the test were used: the versions had identical questions and answers, but had different orders for answer options.

Given various information about the course and students, we wish to evaluate whether LLMs are able to estimate student performance (and thus question difficulty), predict student answers, and evaluate the quality of the 25 multiple choice questions. We explore several variations in the following:

Context. In considering the context provided to the LLM, we chose information that is likely useful for the task, and that instructors may be able to realistically obtain. We always provide the test questions and answer choices, including for the context ϕ in Table 1. We use an XML-encoded version of the full test, including questions and an ordered set of multiple choice answers. We then add:

- **CD:** The calendar description of the course; this information is readily available, and can provide the LLM with a high-level overview of the course, along with the likely student knowledge level.
- **CA:** The correct answer to each question. While the LLM can rederive the correct answer from the question and answer choices, doing so is a potential source of error and requires significant reasoning compute.
- **SS:** Survey answers from students in this year’s course about their perceptions of programming, including free-form responses. Students were asked to rate their agreement with the following statements on a five-point Likert scale: (1) I find programming difficult, (2) I am confident that I can successfully complete the programming tasks in this course. Additionally, they were asked to describe what they enjoy the most about programming, and what they find most frustrating.
- **LYG:** The questions from the previous year’s test, as well as the gradebook data for the students who took the test. This provides a sense of the expected distribution for the questions.

These are ordered by how likely it is that instructors will have access to the information. As such, we iteratively *add* to the context, rather than explore all possible combinations.

Output Information and Format. We experiment with asking for different output formats corresponding to the degree of specificity about the student answer distribution. We use the following output formats, sorted by increasing complexity:

- **Rank Order:** The output is a list of question numbers, from the least to most difficult.
- **Binomial:** The output is an estimate of the probability of a student correctly answering each question.
- **Marginal:** The output is the estimated probability distribution of student answers for each question: i.e., the proportion of students answering ‘A’, ‘B’, ‘C’, etc.
- **Gradebook:** The output is a gradebook of synthetic student answers to each multiple choice question, for any number of students. That is, for each (synthetic) student, output that student’s answer choice for each question. To manage the size of the output tokens, $n = 25$ students were used.

Our rationale for including these more complex formats is that additional metrics may be estimated. For the **Gradebook** output, any metric that can be computed from a real student gradebook could be (in theory) estimated from the LLM output. We note that prior work on both human [3] and LLM [28] estimates of question difficulty uses Likert scale rating as an output format; we explicitly avoided this format for the same reason.

Generative AI Model. To explore the state of the art while avoiding bias toward any particular model size or vendor, we evaluated OpenAI’s gpt-5-nano, gpt-5-mini, gpt-5.2, gpt-5.2-codex, and gpt-5-pro models, Anthropic’s sonnet-4.5 and opus-4.5 models, and Google’s gemini-2.5-flash and gemini-2.5-pro models. The OpenAI Python library was used to interface with OpenAI models on OpenAI using the Responses API, and non-OpenAI models on OpenRouter using the Chat Completions API. “medium” reasoning effort (as defined by OpenAI and OpenRouter) was used for all models, with the exception of gpt-5-pro which produces an error if “high” reasoning effort is not requested. In all cases, the *System Prompt* contained a fixed string providing instructions for the LLM to follow, based on the prompting approach. The *user prompt* contained an XML-encoded concatenation of context information described above. Finally, a *schema for structured outputs* was provided since structured outputs were used for all models; the schema included JSON structure, types, and field descriptions.

3.1. Evaluation Process

We measure the quality of the LLM-generated answer distributions by comparing them to the empirical distribution of the 1013 student answer choices for the corresponding question. We compute two metrics to evaluate the LLM-generated answer distributions.

Estimating Question Difficulty. In order to evaluate how well LLM-generated answers correctly estimate the difficulty of a question, we use the **Spearman Correlation** coefficient, a rank-based correlation measure. Specifically, we compute the predicted order (least to most difficult) of the 25 questions, and compare this order to the ground-truth. A rank-based measure is used because we do not expect the question difficulty measures to be normally distributed.

Best Distractor Prediction. The Spearman Correlation does not provide insight as to whether the LLM-estimated distribution of answer choices matches the real student distribution. If an LLM’s estimated distribution across answer choices is reasonably accurate, then instructors can gather further insight as to which distractors are most meaningful. To evaluate the extent to which the answer distributions match, we use **best distractor prediction accuracy**, defined as the extent to which the predicted “best” distractor (the incorrect response with the highest proportion of students choosing it) matches the ground-truth.

We evaluate each (context, prompting approach, model) combination separately. To do so, we compute the two evaluation metrics, comparing the distributions against the empirical distributions from 1013 real students completing the multiple choice questions. However, there were 4 versions of the test with the same multiple choice questions, but with different answer orders. Roughly $\frac{1}{4}$ of students were assigned to each test version. During our exploratory analysis, we found that the answer order affects the distribution of student answers. Thus, each (context, prompting approach, model) is evaluated four times, corresponding to the four answer shuffle orders in the test versions. Each time, a comparison

is made between the LLM-generated distribution and the empirical distribution of the $\frac{1}{4}$ of students assigned to the corresponding test. The statistics evaluated are then averaged for the main analysis. The question ordering effect is analyzed separately for the best approaches.

3.2. Human Evaluation Baseline

To produce a reasonable baseline, we asked 5 human raters with at least a master’s degree in Computer Science or equivalent industry experience to evaluate the questions; one rater was a computing educator with CS1 teaching experience. The original instructor who served as the examiner was not asked to rate the question difficulty, since the examiner already knew the actual grade distribution of the questions. Prior work has shown that programming teachers are fairly accurate in rating the difficulty of questions for a course they did not teach [3].

Each human rater was asked to estimate the proportion of students choosing each multiple choice answer, akin to the “Marginal” prompting strategy. A course description was provided to the human raters. Due to the time constraints on the part of the human raters, not all raters rated all questions: one rater missed 1 question, and another rater missed 3.

To form a competitive comparison, we use an **ensemble** of human ratings. That is, our ensemble estimate of the proportion of students choosing each multiple choice answer choice is the *average* of the (up to 5) individual human rater estimates. Crucially, **only one test variation is rated: V1**. The other variations have identical questions and differ only in the ordering of the answer choices, so it was not reasonable to ask the raters to rate multiple versions.

4. Results

Table 1 shows, for each context, model, and output format combination, the Spearman rank correlation and the top-distractor accuracy. The latter metric was only computed for “marginal” and “gradebook” outputs, since “rank order” and “binomial” outputs lack the necessary information to determine the probability that a given distractor is chosen.

4.1. Estimating Multiple Choice Question Difficulty

From Table 1, we observed the highest Spearman rank correlation using Claude Sonnet 4.5 with the Rank Order prompting approach and the Course Description as context. The Rank Order prompting approach outperformed other approaches in all models and contexts. In fact, the Gradebook approach sometimes produced negative correlations. Moreover, larger models did not always outperform smaller ones: e.g., Claude Sonnet generally outperforms Opus. Anthropic models generally outperform the others, even with little information about the course. This model-dependent behavior is consistent with prior findings by Benedetto et al. [28], who found that prompts for simulated student responses did not generalize reliably across LLMs. We also did not observe any improvements when adding additional context to the prompt: using only the Course Description generally produced the best performance.

Table 2 (left) compares the difficulty ranking between GT, human, and this “Best LLM”. Since the human ranking is for V1 of the test only, we compare performance on V1: the Best LLM achieves $\rho = 0.665$ compared to human raters at $\rho = 0.577$. We find that the Best LLM actually outperforms the human ensemble. However, Table 1 also shows, even for the ground truth, that while the 4 test versions contain the same questions, there is slight variation between the 4 test versions. For the ground-truth student data, the Spearman correlations between question difficulty ranks across versions range from 0.932-0.963 (mean=0.957), whereas for the LLM, the ranges are 0.959-0.969 (mean=0.963). While the differences appear small, for each of the 6 pairwise comparisons between test versions, the LLM’s rank similarity is higher than that of the ground-truth student distribution. Indeed, the LLM prediction of the 3 easiest and 1 hardest question is consistent, regardless of answer ordering.

To further explore the answer ordering effect, we see from Table 2 (right) that the proportions of correct responses across the 4 test variations differ. For Q5 and Q19, the difference in average

Table 1

Comparison of generated distributions to empirical data across different context combinations, models, and output formats. Cells marked ‘err’ indicate that the model did not return a valid, instruction-following response.

Context	Model	Spearman rank correlation				Top-distractor accuracy	
		Rank Order	Binomial	Marginal	Gradebook	Marginal	Gradebook
ϕ	gpt-nano	0.159	-0.048	0.095	-0.128	0.196	0.201
ϕ	gpt-mini	0.159	0.009	0.167	0.170	0.207	0.216
ϕ	gpt-5.2	0.160	0.083	0.120	0.065	0.203	0.213
ϕ	gpt-5.2-codex	0.242	0.012	-0.031	-0.233	0.205	0.204
ϕ	gpt-5.2-pro	0.238	0.062	0.078	-0.001	0.210	0.210
ϕ	sonnet-4.5	0.584	0.151	0.079	0.196	0.212	0.203
ϕ	opus-4.5	0.572	0.204	0.040	0.118	0.253	0.237
ϕ	gemini-2.5-flash	0.422	err	0.043	0.253	0.259	0.233
ϕ	gemini-2.5-pro	0.414	0.140	0.055	0.185	0.238	0.183
CD	gpt-nano	0.263	0.042	0.041	-0.099	0.214	0.312
CD	gpt-mini	0.337	0.099	0.078	0.017	0.221	0.236
CD	gpt-5.2	0.182	0.090	0.101	0.025	0.220	0.200
CD	gpt-5.2-codex	0.297	0.088	0.034	-0.341	0.212	0.195
CD	gpt-5.2-pro	0.186	0.093	0.079	0.089	0.210	0.210
CD	sonnet-4.5	0.640	0.113	0.140	0.235	0.243	0.177
CD	opus-4.5	0.591	0.233	0.087	0.146	0.273	0.195
CD	gemini-2.5-flash	0.436	0.231	err	0.280	err	0.248
CD	gemini-2.5-pro	0.435	0.237	0.259	0.061	0.206	0.195
CD+CA+SS	gpt-nano	0.417	-0.047	-0.060	-0.010	0.208	0.244
CD+CA+SS	gpt-mini	0.228	0.032	0.023	0.029	0.185	0.282
CD+CA+SS	gpt-5.2	0.281	0.086	0.045	0.217	0.198	0.212
CD+CA+SS	gpt-5.2-codex	0.311	0.136	0.029	-0.159	0.225	0.243
CD+CA+SS	gpt-5.2-pro	0.265	0.046	0.071	0.115	0.210	0.210
CD+CA+SS	sonnet-4.5	0.540	0.191	0.013	err	0.257	err
CD+CA+SS	opus-4.5	0.570	0.291	-0.130	0.200	0.255	0.250
CD+CA+SS	gemini-2.5-flash	0.440	0.336	-0.031	err	0.198	err
CD+CA+SS	gemini-2.5-pro	0.305	0.283	0.200	0.105	0.226	0.225
CD+CA+SS+LYG	gpt-nano	0.220	-0.392	0.144	0.178	0.241	0.229
CD+CA+SS+LYG	gpt-mini	0.282	0.033	-0.039	0.071	0.198	0.258
CD+CA+SS+LYG	gpt-5.2	0.250	0.071	0.176	0.001	0.205	0.235
CD+CA+SS+LYG	gpt-5.2-codex	0.522	0.168	0.053	-0.038	0.190	0.218
CD+CA+SS+LYG	gpt-5.2-pro	0.225	0.103	0.121	0.071	0.210	0.215
CD+CA+SS+LYG	sonnet-4.5	0.583	0.287	0.047	0.198	0.222	0.183
CD+CA+SS+LYG	opus-4.5	0.572	0.228	0.050	-0.056	0.247	0.215
CD+CA+SS+LYG	gemini-2.5-flash	0.497	0.340	0.037	0.014	0.208	0.192
CD+CA+SS+LYG	gemini-2.5-pro	0.416	0.288	0.214	0.177	0.228	0.199

correctness between the “easiest” and “hardest” versions is more than 10%; in fact, 15 of the 25 questions have correctness rate variations of greater than 5%, showing that distractor placement significantly impacts item difficulty [40, 41, 42, 43]. The LLM result, shown in Table 2, does not show psychometric effects similar to those observed in real students. However, some order variations show degradation in performance (e.g., gpt-nano/CD/Gradebook/V4).

To explore potential failure modes, we analyze, for each context/model/output combination (where the correct answer (CA) is not provided and where the output allows), the percentage of cases in which the model’s top predicted student answer matches the correct answer. We find that the gpt-5.2 models tended towards 100% correct answers for both the marginal and gradebook output formats, and for both ϕ and CD contexts. Conversely, the higher-performing models like CD/gpt-nano/Gradebook (see Table 2 right) and sonnet/opus tended to have a lower percentage of correct answers, more consistent with student answers. We hypothesize, based on results from Benedetto et al. [28], that it is difficult to get LLMs to role-play as a novice student, rather than predict the correct solution, in the MCQ answer prediction setting.

Table 2

Comparison of question difficulty ranking and percent correct by question across four test versions (V1-V4). Best LLM for ranking is CD-sonnet-4.5-rankorder; best LLM for percent correct is gpt-nano/CD Gradebook.

Q	Question Difficulty Ranking										Percent Correct Answer									
	Ground Truth				Human	Best LLM					Ground Truth				Human	Best LLM				
	V1	V2	V3	V4	V1	V1	V2	V3	V4		V1	V2	V3	V4	V1	V1	V2	V3	V4	
1	5	8	7	9	12	1	1	1	1		82	79	82	76	51	88	80	4	16	
2	13	13	13	12	22	3	3	3	3		69	72	73	74	34	76	76	8	20	
3	8	10	8	10	18	9	7	4	7		80	75	81	76	44	84	88	12	28	
4	11	14	9	11	23	5	5	6	5		73	72	77	75	34	12	84	16	12	
5	19	23	16	21	21	6	6	7	6		59	54	66	58	35	68	88	20	24	
6	7	7	4	7	3	8	8	8	9		80	80	84	79	68	88	88	24	20	
7	6	5	2	2	11	4	4	5	4		81	83	87	87	54	88	92	28	12	
8	1	1	1	1	1	13	14	12	13		87	88	90	91	76	76	88	32	28	
9	14	15	15	15	5	7	9	9	8		69	70	71	69	64	96	76	32	16	
10	9	6	12	6	10	16	18	17	16		78	81	75	80	55	88	88	40	16	
11	4	3	5	3	2	15	12	14	14		83	84	83	85	70	84	96	44	12	
12	22	22	23	24	16	22	22	23	17		53	54	50	49	49	76	88	44	16	
13	10	9	11	8	20	14	13	15	15		77	78	75	78	36	84	100	40	20	
14	3	2	3	4	4	2	2	2	2		84	87	87	84	66	68	80	48	24	
15	24	24	25	25	25	20	24	24	24		51	53	46	48	32	88	92	96	12	
16	25	25	24	23	14	24	20	21	20		50	42	47	50	50	88	76	68	24	
17	2	4	6	5	8	10	11	10	10		86	84	83	82	57	84	0	92	12	
18	15	11	14	13	13	12	17	13	12		68	75	73	72	50	76	4	0	24	
19	21	16	19	18	19	18	19	18	21		57	70	62	62	44	84	100	4	20	
20	17	18	20	16	17	21	21	19	22		67	65	62	68	47	84	88	92	20	
21	18	19	18	19	6	23	23	20	23		61	63	63	61	60	76	100	92	24	
22	16	17	17	17	15	19	15	22	18		67	67	63	65	50	0	100	92	16	
23	12	12	10	14	7	11	10	11	11		70	74	77	72	59	76	92	96	20	
24	20	20	21	20	9	17	16	16	19		57	61	61	58	57	12	76	92	24	
25	23	21	22	22	24	25	25	25	25		51	57	56	53	33	84	4	96	20	
Avg											70	71	71	70	51	73	78	48	19	

4.2. Best Distractor

We compared the best distractor predictions from humans and LLMs against the ground truth student data. For the LLM, we report results using gpt-nano with Course Description context and Gradebook prompting, which achieved the highest top-distractor accuracy (0.312) in Table 1.

On test version V1, human raters achieved 52% accuracy in predicting the most frequently selected distractor, compared to 36% for the LLM. Thus, humans maintain an advantage in identifying attractive distractors. This suggests that predicting which specific wrong answer students will choose requires different reasoning than ordering questions by overall difficulty [32].

The LLM’s accuracy also varies across test versions with no consistent pattern. For the other three versions of the test (V2, V3, and V4), we did not have human data. For these, the LLM was able to predict the most selected distractor 43%, 20%, and 13% of the time for V2, V3, and V4, respectively. This suggests that the ordering of answer options seems to correlate with the performance of the LLM in this task. Furthermore, the LLM frequently predicts ties or puts all its weight on the correct answer, suggesting that current models struggle to simulate how a novice student would realistically approach a distractor. However, even for the ground-truth student data, the actual best distractor answer differs depending on the order in which the answers are shown.

5. Discussion

Estimating Multiple Choice Question Difficulty: Output Format Matters. The best LLM configuration (Claude Sonnet with Rank Order prompting) achieved $\rho = 0.665$ Spearman correlation with ground-truth difficulty, actually outperforming our ensemble of human raters ($\rho = 0.577$). The Rank

Order approach significantly outperformed other prompting strategies. We attribute this to several factors: the simplicity of the output reduces opportunities for error; the format naturally encodes relationships between questions rather than treating each in isolation; and it avoids degenerate distributions (e.g., predicting 100% probability of selecting the right answer). Surprisingly, adding more context (e.g., student survey answers) did not consistently improve performance, and larger models did not always outperform smaller ones. These results suggest that when designing LLM-based evaluation tasks, the choice of output format deserves as much attention as the information provided in the prompt. We note again that prior work sometimes uses Likert scale predictions of output difficulty [28, 3], or information (like the knowledge component profiles of simulated students) that is not easily accessible [25].

Distractor Prediction is Harder. While LLMs can reasonably estimate overall question difficulty, predicting which specific distractors students will select remains challenging. On this task, the human ensemble maintained an advantage (52% accuracy vs. 36% for the best LLM). This likely requires modeling student misconceptions rather than just question content—a task that benefits from pedagogical expertise [6, 31, 32]. Moreover, models were not universally capable of solving the test questions themselves; smaller models sometimes erred even with reasoning enabled. If a model cannot reliably identify the correct answer, it is unlikely to predict which wrong answers students find attractive.

Answer Ordering Effects. Answer order affects human test-takers, but LLMs appear less sensitive to option ordering. This consistency is a double-edged sword. On the one hand, it suggests LLMs may capture question-intrinsic difficulty. On the other hand, real students *are* affected by option ordering—for some questions, correctness rates differed by more than 10% across versions—so an LLM that ignores these effects may miss psychometric properties. This highlights a key challenge: should synthetic students mimic human biases, or should they estimate some “true” underlying difficulty?

5.1. Limitations

This study used a single 25-question exam from one introductory programming course at one institution, limiting generalizability to other subjects, course levels, or student populations. While the ground truth is based on responses from a single cohort, our sample size was large ($n=1013$). Our human baseline was an ensemble of raters, not all of whom were instructors, and is intended to provide a point of comparison rather than being a rigorous measure.

On the technical side, we used one run per test variation, and thus 4 runs per value shown in Table 1. While we experimented extensively with LLM parameters (e.g., temperature), we were limited by the LLM’s context window. Thus, the Gradebook approach generated only 25 synthetic students per run, which may limit the fidelity of distribution estimates. We observed substantial variation across runs. This variability likely stems from multiple sources: stochastic sampling during generation, different reasoning paths taken by the model, and sensitivity to subtle differences in input formatting across test versions. Understanding and reducing this variance is an important direction for future work—e.g., through ensemble methods that aggregate predictions across multiple runs or prompting strategies that encourage more deterministic reasoning.

5.2. Negative Results

We initially attempted to evaluate synthetic responses using absolute error and KL divergence between predicted and actual answer distributions. However, these metrics were dominated by calibration differences rather than question-specific accuracy: if an LLM predicts 70% correct when the true rate is 30%, it scores poorly even if it perfectly captures relative difficulty. Human raters exhibited similar calibration challenges, struggling to predict absolute percentages even when their rankings were reasonable. This motivated our focus on rank-based evaluation.

We also explored automated prompt optimization using OPRO [44], but our attempts yielded only marginal improvements even with highly prescriptive prompts. Given the added complexity and overfitting risk, we did not report results for optimized prompts.

6. Conclusion

This work investigated whether LLMs can serve as “synthetic students” to help instructors evaluate multiple choice question difficulty. We show that the best LLM configuration outperformed human raters on the question difficulty ranking task, but predicting which specific distractors students would select proved more challenging. Generating response distributions that capture fine-grained psychometric properties, such as sensitivity to answer-option ordering, remains an open problem.

Our results suggest that simulating student cognition on MCQs is genuinely difficult, likely because it requires modeling not just knowledge but the specific misconceptions that lead students to particular wrong answers. At the same time, the fact that some configurations perform well indicates that the problem may be tractable and merits further investigation. Future work could explore more complex reasoning approaches, so that educators could use LLMs to estimate question difficulty, and ultimately create better assessments for students.

Acknowledgments

This work was supported by Research Council of Finland grant #356114.

Declaration on Generative AI

During the preparation of this work, the authors used Claude, ChatGPT, and Cursor to assist with the following: drafting content, identifying relevant literature, paraphrasing and rewording, improving writing style, checking grammar and spelling, formatting, and enhancing content. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] L. Zotos, I. P. de Jong, M. Valdenegro-Toro, A. I. Sburlea, M. Nissim, H. van Rijn, NLP methods may actually be better than professors at estimating question difficulty, in: Proceedings of the 2nd International Workshop on AI in Society, Education and Educational Research (AISEER), 2025. ArXiv:2508.03294.
- [2] G. van de Watering, J. van der Rijt, Teachers’ and students’ perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items, *Educational Research Review* 1 (2006) 133–147. doi:10.1016/j.edurev.2006.05.001.
- [3] Simon, D. D’Souza, J. Sheard, J. Harland, A. Carbone, M.-J. Laakso, Can computing academics assess the difficulty of programming examination questions?, in: Proceedings of the 12th Koli Calling International Conference on Computing Education Research, Koli Calling ’12, Association for Computing Machinery, New York, NY, USA, 2012, p. 160–163. URL: <https://doi.org/10.1145/2401796.2401822>. doi:10.1145/2401796.2401822.
- [4] M. J. Gierl, O. Bulut, Q. Guo, X. Zhang, Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review, *Review of Educational Research* 87 (2017) 1082–1116. URL: <https://doi.org/10.3102/0034654317726529>. doi:10.3102/0034654317726529. arXiv:<https://doi.org/10.3102/0034654317726529>.
- [5] M. Tarrant, J. Ware, A. M. Mohammed, An assessment of functioning and non-functioning distractors in multiple-choice questions: a descriptive analysis, *BMC Medical Education* 9 (2009) 40. doi:10.1186/1472-6920-9-40.
- [6] T. M. Haladyna, M. C. Rodriguez, *Developing and Validating Test Items*, Routledge, 2013.
- [7] B. B. Frey, S. Petersen, L. M. Edwards, J. T. Pedrotti, V. Peyton, Item-writing rules: Collective wisdom, *Teaching and Teacher Education* 21 (2005) 357–364. URL: <https://www.sciencedirect.com/science/article/pii/S0742051X0500020X>. doi:<https://doi.org/10.1016/j.tate.2005.01.008>.

- [8] Y.-M. Tseng, Y.-C. Huang, T.-Y. Hsiao, W.-L. Chen, C.-W. Huang, Y. Meng, Y.-N. Chen, Two tales of persona in llms: A survey of role-playing and personalization, 2024. URL: <https://arxiv.org/abs/2406.01171>. arXiv:2406.01171.
- [9] J. Leinonen, P. Denny, O. Kiljunen, S. MacNeil, S. Sarsa, A. Hellas, LLM-itation is the sincerest form of data: Generating synthetic buggy code submissions for computing education, in: Proceedings of the 27th Australasian Computing Education Conference, 2025, pp. 56–63.
- [10] H. Wu, D. Lee, T. Zerner, S. Court-Kowalski, P. Devitt, E. Palmer, A comparison of the psychometric properties of GPT-4 versus human novice and expert authors of clinically complex MCQs in a mock examination of Australian medical students, *Medical Teacher* 48 (2026) 74–84.
- [11] M. Dorodchi, E. Al-Hossami, A. Benedict, E. Demeter, Using synthetic data generators to promote open science in higher education learning analytics, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 4672–4675.
- [12] C. Zhan, O. B. Deho, X. Zhang, S. Joksimovic, M. de Laat, Synthetic data generator for student data serving learning analytics: A comparative study, *Learning Letters* 1 (2023) 5–5.
- [13] B. Flanagan, R. Majumdar, H. Ogata, Fine grain synthetic educational data: challenges and limitations of collaborative learning analytics, *IEEE Access* 10 (2022) 26230–26241.
- [14] M. Khalil, F. Vadiiee, R. Shakya, Q. Liu, Creating artificial students that never existed: Leveraging large language models and CTGANs for synthetic data generation, in: Proceedings of the 15th International Learning Analytics and Knowledge Conference, 2025, pp. 439–450.
- [15] S. MacNeil, M. Rogalska, J. Leinonen, P. Denny, A. Hellas, X. Crosland, Synthetic students: A comparative study of bug distribution between large language models and computing students, in: Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1, 2024, pp. 137–143.
- [16] G. Rasch, Probabilistic models for some intelligence and attainment tests, *The SAGE Encyclopedia of Research Design* (1981). URL: <https://api.semanticscholar.org/CorpusID:61203382>.
- [17] F. M. Lord, Applications of item response theory to practical testing problems, Routledge, 2012.
- [18] S. E. Embretson, S. P. Reise, Item response theory for psychologists, Psychology Press, 2013.
- [19] F. Y. Hsu, H. M. Lee, T. H. Chang, Y. T. Sung, Automated estimation of item difficulty for multiple-choice tests: An application of word embedding techniques, *Information Processing & Management* 54 (2018) 969–984.
- [20] L. A. Ha, V. Yaneva, P. Baldwin, J. Mee, Predicting the difficulty of multiple choice questions in a high-stakes medical exam, in: H. Yannakoudakis, E. Kochmar, C. Leacock, N. Madnani, I. Pilán, T. Zesch (Eds.), Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, Florence, Italy, 2019, pp. 11–20. URL: <https://aclanthology.org/W19-4402/>. doi:10.18653/v1/W19-4402.
- [21] K. Xue, V. Yaneva, C. Runyon, P. Baldwin, Predicting the difficulty and response time of multiple choice questions using transfer learning, in: Proceedings of the fifteenth workshop on innovative use of NLP for building educational applications, 2020, pp. 193–197.
- [22] L. Benedetto, P. Cremonesi, A. Caines, P. Buttery, A. Cappelli, A. Giussani, R. Turrin, A survey on recent approaches to question difficulty estimation from text, *ACM Computing Surveys* 55 (2023) 1–37.
- [23] S. AlKhuyaey, F. Grasso, T. R. Payne, V. Tamma, Text-based question difficulty prediction: A systematic review of automatic approaches, *International Journal of Artificial Intelligence in Education* 34 (2024) 862–914.
- [24] V. Yaneva, K. North, P. Baldwin, L. A. Ha, S. Rezayi, Y. Zhou, S. R. Choudhury, P. Harik, B. Clauser, 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), 2024, pp. 470–482.
- [25] Y. Liu, S. Bhandari, Z. A. Pardos, Leveraging LLM respondents for item evaluation: A psychometric analysis, *British Journal of Educational Technology* 56 (2025) 1028–1052.
- [26] H. Maeda, Field-testing multiple-choice questions with AI examinees: English grammar items, *Educational and Psychological Measurement* 85 (2025) 221–244.

- [27] J.-W. Park, S.-J. Park, H.-S. Won, K.-M. Kim, Large language models are students at various levels: Zero-shot question difficulty estimation, in: *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 8157–8177.
- [28] L. Benedetto, G. Aradelli, A. Donvito, A. Lucchetti, A. Cappelli, P. Buttery, Using LLMs to simulate students' responses to exam questions, in: *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 11351–11368.
- [29] M. Uto, Y. Tomikawa, A. Suzuki, Question difficulty prediction based on virtual test-takers and item response theory, in: *Workshop on Automated Evaluation of Learning and Assessment Content: International Conference on Artificial Intelligence in Education*, 2024.
- [30] A.-C. Rogoz, R. T. Ionescu, UnibucLLM: Harnessing LLMs for automated prediction of item difficulty and response time for multiple-choice questions, in: *The 19th Workshop on Innovative Use of NLP for Building Educational Applications*, 2024, p. 493.
- [31] E. Alhazmi, Q. Z. Sheng, W. E. Zhang, M. Zaib, A. Alhazmi, Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation, in: *Proceedings of the 2024 conference on empirical methods in natural language processing*, 2024, pp. 14437–14458.
- [32] Z. Liu, H. Li, J. Chao, W. Xing, Brains vs. algorithms? How experts and students see AI-generated distractors, in: *Proceedings of the Symposium on Educational Advances in Artificial Intelligence, AAAI*, 2026.
- [33] A. T. Corbett, J. R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, *User modeling and user-adapted interaction* 4 (1994) 253–278.
- [34] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, J. Sohl-Dickstein, Deep knowledge tracing, *Advances in neural information processing systems* 28 (2015).
- [35] P. I. Pavlik Jr, H. Cen, K. R. Koedinger, Performance factors analysis—a new alternative to knowledge tracing., *Online submission* (2009).
- [36] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, When is deep learning the best approach to knowledge tracing?, *Journal of Educational Data Mining* 12 (2020) 31–54.
- [37] S. Sarsa, J. Leinonen, A. Hellas, Empirical evaluation of deep learning models for knowledge tracing: Of hyperparameters and metrics on performance and replicability, *Journal of Educational Data Mining* 14 (2022).
- [38] B. Nguyen, T. Du, M. Yu, L. Angrave, M. Jiang, Qg-sms: enhancing test item analysis via student modeling and simulation, in: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 26152–26168.
- [39] X. Lu, X. Wang, Generative students: Using llm-simulated student profiles to support question item evaluation, in: *Proceedings of the Eleventh ACM conference on learning@ Scale*, 2024, pp. 16–27.
- [40] S. Lions, P. Dartnell, G. Toledo, M. I. Godoy, N. Córdova, D. Jiménez, J. Lemarié, Position of correct option and distractors impacts responses to multiple-choice items: Evidence from a national test, *Educational and Psychological Measurement* 83 (2023) 861–884. doi:10.1177/00131644221132335, PMID: 37663536.
- [41] E. L. Clark, General response patterns to five-choice items., *Journal of Educational Psychology* 47 (1956) 110.
- [42] J. Schroeder, K. L. Murphy, T. A. Holme, Investigating factors that influence item performance on ACS exams, *Journal of Chemical Education* 89 (2012) 346–350.
- [43] J. Tellinghuisen, M. M. Sulikowski, Does the answer order matter on multiple-choice exams?, *Journal of Chemical Education* 85 (2008) 572–575. doi:10.1021/ed085p572.
- [44] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, X. Chen, Large language models as optimizers, in: *The Twelfth International Conference on Learning Representations*, 2024. URL: <https://openreview.net/forum?id=Bb4VGOWELL>.