It's Hard to Know How Hard It Is: Mapping the Design Space of LLM Item Difficulty Estimation

Anonymous

Anonymous Institution

Abstract. Can LLMs estimate how hard a test question is? Published results disagree wildly—correlations range from near-zero to $r = 0.87$—but each study typically tests one prompt on one model on one dataset. We map the design space systematically: 15 prompts, 6 models, 3 datasets (~120 configurations). Prompts that direct the model to enumerate structural item features—prerequisites, cognitive load, error paths—achieve $\rho = 0.65$–0.69 on open-ended items. This advantage attenuates on MCQs ($\rho \approx 0.45$–0.58), where a simple teacher-role prompt nearly matches. Prompt design is the primary lever; model size, temperature, and deliberation each contributes minimally ($\eta^2 < 0.05$). Methodologically, correlations from small item sets ($n < 50$) are unreliable, and unbalanced hyperparameter sweeps produce Simpson's paradox artifacts. At ~$0.01 per item, LLM estimates can triage items into difficulty bands, but the signal is strongest for open-ended items spanning knowledge domains.

Keywords: item difficulty estimation · LLM evaluation · design of experiments · psychometrics · prompt engineering

1 Introduction

A well-designed test gives each student questions at the right level of challenge—hard enough to reveal what they know, easy enough to avoid frustration. Getting this right depends on knowing how difficult each question is before it reaches a student. In practice, this means field-testing items with hundreds or thousands of real students and computing how many answer correctly. In classical test theory, this proportion is called p-correct; item response theory refines it with a latent difficulty parameter b that accounts for examinee ability [8]. Either way, calibrating each item requires collecting responses from 200 to 1,000+ students depending on the IRT model [5], making difficulty estimation one of the most resource-intensive steps in assessment development.

That cost is becoming a bottleneck. LLMs can now generate assessment items at scale [15], but generated items arrive without calibrated difficulty estimates. Without such estimates, items cannot be assembled into well-targeted tests, used in adaptive systems, or compared to existing item banks. The question is whether LLMs can also estimate difficulty—replacing or supplementing the expensive field-testing step.

The task is harder than it appears. LLMs are trained on expert-written text and tend to find most items trivial; Li, Chen et al. [13] find that high model performance paradoxically impedes difficulty estimation—models that solve items easily cannot perceive what makes them hard for humans. This "curse of knowledge" creates a systematic bias: models overestimate student ability and compress difficulty estimates toward the easy end of the scale. Expert teacher judgment, by comparison, achieves correlations of $r = 0.63$–0.66 with actual student performance [11,19]—a useful reference point, though these studies measure judgment of individual students rather than aggregate item difficulty.

Recent work shows wide variation: some studies report strong results (up to $r = 0.87$ with feature extraction and gradient boosting [17]), while others find near-zero correlations for direct estimation [1]. A systematic review of 37 papers found no consensus on methods [16], and the first shared task on difficulty prediction (BEA 2024) found best results only marginally beat baselines [22]. Li, Chen et al. [13] evaluated 20+ models and found they converge on a "machine consensus" that diverges from human perception.

A key limitation is that each study typically tests one model with one prompt on one dataset. When results disagree, it is unclear whether the discrepancy reflects model choice, prompt design, item properties, or student population. We address this by mapping the design space rather than testing any single configuration.

Our optimization target is **rank-order agreement** between LLM-predicted difficulty and empirical p-correct, measured by Spearman's ρ. We choose rank correlation because practical applications—item sequencing, adaptive test assembly, item bank stratification—depend on difficulty *ordering* rather than exact calibration. Each experimental condition prompts an LLM to estimate the proportion of students who would answer each item correctly; we then correlate these estimates against observed p-correct from real student responses.

Contributions. (1) The first systematic comparison of 15 theory-grounded prompts for LLM difficulty estimation, tested across 6 models and 3 datasets (~120 configurations). (2) An *item-centric vs. population-centric* distinction explaining why hybrid prompts like buggy_rules ($\rho = 0.66$) succeed while pure simulation (synthetic_students, $\rho = 0.19$) fails. (3) Methodological guidance: Simpson's paradox artifacts in unbalanced hyperparameter sweeps, $\geq$100 items required for reliable validation, and two-stage sequential DOE as the recommended design.

2 Related Work

We organize prior work by approach and extract testable claims from each.

Direct LLM Estimation. Razavi & Powers [17] report $r = 0.83$ (math) and $r = 0.81$ (reading) on K–5 items using GPT-4o. However, Acquaye et al. [1] find $r \approx 0$ for direct estimation on NAEP items, and Li, Chen et al. [13] find average $\rho = 0.28$ across 20+ models on four assessment domains. *Testable claim: direct* *estimation produces meaningful correlations ($\rho > 0.30$) on curriculum-aligned items.*

Feature Extraction + ML. Razavi & Powers [17] achieve their strongest results ($r = 0.87$) by extracting features from LLMs and training gradient-boosted models. *Testable claim: LLM-extracted features combined with ML outperform direct estimation.*

Simulated Classrooms. Acquaye et al. [1] simulate classrooms of LLM students at multiple ability levels, achieving $r = 0.75$–0.82 on NAEP items. Counter-intuitively, weaker math models predicted difficulty better than stronger ones. However, Li, Chen et al. [13] find significant misalignment between AI and human difficulty perception. *Testable claim: simulation outperforms direct estimation.*

Model Uncertainty. Zotos et al. [23] use LLM uncertainty features (first-token probability, choice-order sensitivity) to predict difficulty. *Testable claim: items the model finds uncertain are items students find difficult.*

Reasoning Augmentation. Feng et al. [9] report up to 28% MSE reduction when generating reasoning before predicting difficulty. However, Li, Jiao et al. [14] find minimal improvements from chain-of-thought prompting for fine-tuned models. *Testable claim: structured deliberation improves difficulty estimation.*

Cognitive Item Models. A separate tradition predicts difficulty from structural item features. Embretson's [7] cognitive design system counts processing steps and working memory demands. Gorin [10] manipulates item features to generate items at target difficulties. The KLI framework [12] posits that the number and nature of knowledge components drive learning difficulty. Recent work has used LLMs for KC tagging [15], but not to operationalize cognitive item models as difficulty estimation prompts. *Testable claim: prompts grounded in cognitive item modeling outperform atheoretical prompts.*

3 Hypotheses

We derive hypotheses from three sources: replication of prior claims, predictions from learning science theory, and exploratory questions about the design space. Table 1 lists all 12 hypotheses with their sources and predictions.

4 Method

4.1 Datasets

SmartPaper (India, Open-ended)—Primary. 140 open-ended questions from Indian state assessments across four subjects (English, Mathematics, Science, Social Science), Grades 6–8, with 728,000+ responses. Ground truth: classical p-correct (proportion scoring full marks). Difficulty range: 0.04–0.83, mean 0.29.

Table 1: Hypotheses tested in this study. Verdicts: ✓ = supported, × = rejected, ∼ = partial/mixed.

#	Hypothesis	Source/Theory	Prediction	Verdict
From Prior Work (Replication)				
H1	Direct estimation yields $\rho > 0.30$	Razavi & Powers	$\rho > 0.30$	✓ .56
H2	Simulation beats direct	Acquaye et al.	$\rho_{\text{sim}} > \rho_{\text{direct}}$	× .19<.56
H3	Larger models do better	Scaling expectation	Larger → higher ρ	∼ mixed
From Learning Science Theory				
H4	Item analysis beats direct	Embretson (1998)	$\rho_{\text{analysis}} > \rho_{\text{direct}}$	✓ .69>.56
H5	Prerequisite counting works	KLI (Koedinger)	More prereqs → harder	✓ .69
H6	Cognitive load counting works	CLT (Sweller)	More elements → harder	✓ .67
H7	Buggy rules analysis works	Brown & Burton	More error paths → harder	✓ .66
Exploratory (Design Space)				
H8	Temperature enhances predictions	Stochastic diversity	Higher temp → higher ρ	× η^2=.001
H9	Multi-sample averaging helps	Wisdom of crowds	$\rho_{\text{3-rep}} > \rho_{\text{1-rep}}$	∼ +.03
H10	Prompt is primary lever	Design space	Prompt > model + temp	✓ on open; ∼ on MCQ
H11	Analysis needs capable models	Capacity limits	Analysis × capability	✓ interaction
H12	Signal transfers across datasets	Generalizability	Signal on D1 → D2	✓ attenuates

Table 2: Experimental stages.

Stage	Purpose	Dataset	Design
Screening	Map prompt space	SmartPaper (140)	15 prompts × 2–5 temps × 3 reps
Model survey	Model generalization	SmartPaper (140)	6 models × 3 prompts × 3 reps
Confirmation	Cross-dataset transfer	DBE-KT22 (168)	3 prompts × 2 temps × 3 reps
Validation	Published benchmark	BEA 2024 (595)	3 prompts × 1 temp × 3 reps

DBE-KT22 (South Africa, MCQ)—Confirmation. 168 undergraduate computer science MCQs spanning 27 knowledge components, administered to 1,300+ students. Ground truth: classical p-correct. Differs from SmartPaper in format, domain, and population.

BEA 2024 Shared Task (USA, USMLE MCQ)—Validation. 595 text-only items from the BEA 2024 automated difficulty prediction shared task [22], comprising USMLE Steps 1–3 questions (667 total items minus 72 requiring images). Ground truth: IRT difficulty parameters from operational administrations. This benchmark allows direct comparison with other difficulty estimation approaches.

4.2 Design Space

4.3 Prompts

All 16 prompts share a common output: the LLM predicts what proportion of students would answer each item correctly. They differ along two binary di-

Table 3: Prompt taxonomy with factor coding. Item = analyzes item structure; Pop. = models student population. All 15 prompts except synthetic_students were tested in screening.

Prompt	Item	Pop.	Theory/Prior Work
teacher			PCK (Shulman, 1986)
verbalized_sampling			Wisdom of crowds
familiarity_gradient	✓		Transfer distance
contrastive	✓		Comparative judgment
prerequisite_chain	✓		KC theory (Koedinger et al.)
cognitive_load	✓		CLT (Sweller, 1988)
cognitive_profile	✓		CLT + profiling
devil_advocate		✓	Debiasing heuristics
teacher_decomposed		✓	IRT stratification
classroom_sim		✓	Acquaye et al. (2025)
imagine_classroom		✓	Imagery-based reasoning
synthetic_students*		✓	Student simulation
error_analysis	✓	✓	Error analysis tradition
error_affordance	✓	✓	Brown & Burton (1978)
buggy_rules	✓	✓	Brown & Burton (1978)
misconception_holistic	✓	✓	Chi et al. (1994)

*Tested in model survey only ($\rho = 0.19$, worst performer).

mensions: whether the prompt directs the model to *analyze structural item features* (prerequisites, steps, cognitive demands) before estimating, and whether it directs the model to *model the student population* (reason about proficiency levels, struggles). Table 3 lists all 16 prompts; 15 were tested in screening (synthetic_students was added in the model survey). All prompts are available in the project repository.

4.4 Models

Six models spanning 8B to frontier-scale: Gemini 3 Flash (screening model), GPT-4o, Llama-3.3-70B, Gemma-3-27B, Llama-4-Scout (17B active/109B MoE), and Llama-3.1-8B. Qwen3-32B was tested but excluded due to low parse rates (<50% valid responses).

4.5 Statistical Approach

Our primary metric is Spearman's ρ between LLM-predicted and observed p-correct. We report 95% bootstrap confidence intervals (10,000 resamples) for primary results and MAE as a calibration measure. Each condition uses 3 replications; reported ρ is the averaged-prediction correlation (mean prediction across reps correlated with ground truth).

Table 4: Prompt screening results (SmartPaper, $n = 140$): top 10 of 15 screened prompts, ranked by best ρ. Five additional prompts (verbalized_sampling, familiarity_gradient, imagine_classroom, teacher_decomposed, error_affordance) achieved $\rho < 0.55$. CIs shown for prompts advanced to model survey.

Prompt	Item	Pop.	Best ρ	Temp	95% CI	MAE
prerequisite_chain	✓		**0.686**	0.5	[.576,.775]	.155
cognitive_load	✓		**0.673**	2.0	[.550,.766]	.190
buggy_rules	✓	✓	**0.655**	1.0	[.532,.752]	.117
misconception_holistic	✓	✓	0.636	2.0	—	.204
error_analysis	✓	✓	0.596	2.0	—	.121
devil_advocate		✓	0.596	1.0	—	.098
cognitive_profile	✓		0.586	1.0	[.456,.693]	.214
contrastive	✓		0.584	1.0	—	.123
classroom_sim		✓	0.562	2.0	—	.240
teacher			0.555	1.0	[.422,.664]	.439

5 Results

We present results in five parts: the headline finding on item analysis prompts, the mechanism behind it, factors that matter less than expected, generalization across three datasets, and practical implications.

5.1 Item Analysis Prompts Achieve $\rho = 0.65$–0.69

Fifteen prompts were screened on SmartPaper (140 open-ended items) using Gemini 3 Flash at 2–5 temperature settings with 3 replications per condition. The top three prompts—all directing the model to analyze structural item features before estimating difficulty—achieved $\rho = 0.65$–0.69 (Table 4, Figure 1): **prerequisite_chain** ($\rho = 0.686$) enumerates knowledge prerequisites before estimating; **cognitive_load** ($\rho = 0.673$) counts interacting memory elements; and **buggy_rules** ($\rho = 0.655$) identifies procedural error paths. For context, meta-analyses report that experienced teachers can predict individual student performance with $r = 0.63$–0.66 [11,19]. The constructs differ: teachers judge individual students while we predict aggregate item difficulty. Direct comparison is inappropriate, but the magnitude suggests LLM estimates capture meaningful difficulty information.

5.2 The Mechanism: Item Analysis, Not Population Modeling

Grouping prompts by their 2×2 factor membership (Table 3) reveals a clear pattern: item analysis is the primary driver. Prompts that model student populations without analyzing item structure perform *worse* than baseline. Mean ρ by group: item analysis only = 0.61 (5 prompts); item + population = 0.59 (4 prompts); neither factor = 0.52 (2 prompts); population only = 0.47 (4 prompts).

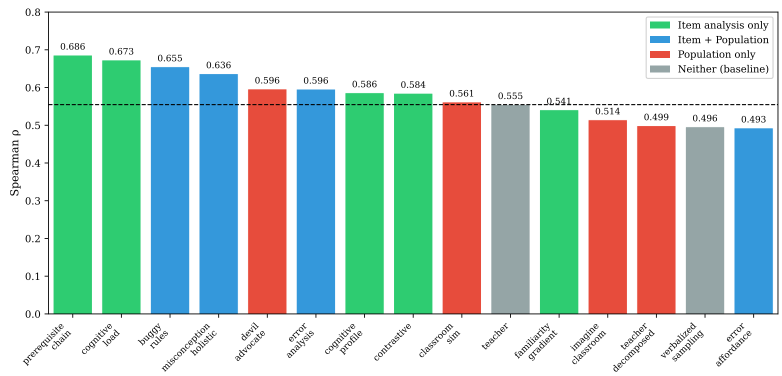


Fig. 1: Prompt screening results on SmartPaper ($n = 140$). Color indicates factor membership: green = item analysis only, blue = item + population, red = population only, grey = neither. Dashed line = teacher-role prompt baseline. Error bars show 95% bootstrap CIs.

This parallels Embretson's [7] cognitive design system, where item difficulty is predicted from counts of processing steps and working memory demands. The LLM operationalizes similar constructs when directed to enumerate prerequisites or interacting elements.

The critical distinction is *item-centric* vs. *population-centric* framing. Hybrid prompts like buggy_rules ($\rho = 0.66$) succeed because "identify procedural errors students might make" is fundamentally item analysis—it asks what error paths the item affords, not how a simulated population would respond. Population modeling *alone* is unreliable: explicit student simulation (synthetic_students: $\rho = 0.19$) performs worst, consistent with Li, Chen et al.'s [13] finding that proficiency simulation does not reliably improve difficulty estimation. Framing analysis through student behavior works; simulating student populations does not.

5.3 What Doesn't Matter

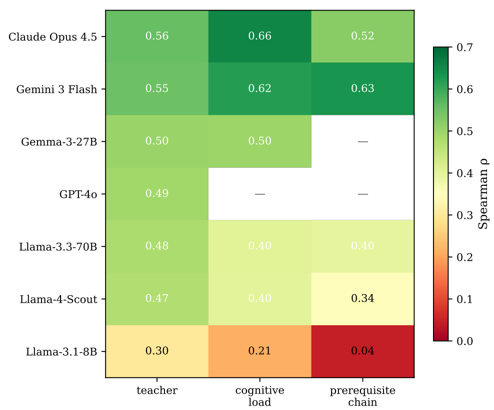
Three factors had minimal effects (each unique $\eta^2 < 0.05$):

Temperature. Most prompts show $\Delta\rho < 0.05$ across temperatures 0.5–2.0. The exception: prerequisite_chain peaks at temperature 0.5 while cognitive_load peaks at 2.0.

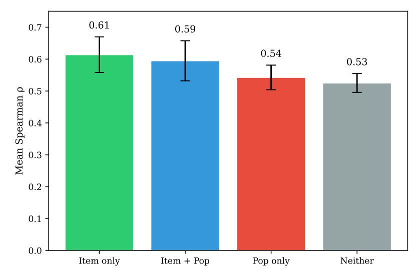
Model size. Among frontier models, size does not predict success. GPT-4o ($\rho = 0.49$) underperforms Gemini 3 Flash ($\rho = 0.55$) and matches the smaller Gemma-3-27B. A capacity floor exists—Llama-3.1-8B struggles ($\rho = 0.30$)—but above that threshold, other factors dominate (Table 5).

Table 5: Model survey (ρ, SmartPaper, $t = 1.0$). "—" indicates condition not run or <50 valid responses.

Model	Params	teacher	cog_load	prereq
Gemini 3 Flash	—	0.55	0.62	**0.66**
Gemma-3-27B	27B	0.50	0.50	—
GPT-4o	—	0.49	—	—
Llama-3.3-70B	70B	0.48	0.40	0.40
Llama-4-Scout	17B/109B	0.47	0.40	0.35
Llama-3.1-8B	8B	0.30	0.21	0.04



(a) Model survey. Frontier models achieve $\rho > 0.50$; Llama-8B fails on structured prompts.



(b) Post-hoc grouping by factor membership. Item analysis is the primary driver; population modeling alone underperforms baseline.

Fig. 2: (a) Model $\times$ prompt heatmap on SmartPaper. (b) Mean ρ by 2$\times$2 factor membership with 95% CIs.

Deliberation. Undirected chain-of-thought reasoning degrades performance. The structured prompts that work use *directed* analysis—they specify exactly what to enumerate (prerequisites, interacting elements, error paths). Undirected "think step by step" produces unfocused reasoning that drifts toward solving the item.

Surface features. Text length correlates $\rho = -0.44$ with difficulty on SmartPaper (shorter questions are harder). Controlling for text length, partial $\rho = 0.59$–0.66 for item analysis prompts—the correlation changes minimally ($\Delta\rho \leq 0.06$), confirming that LLM predictions are not merely proxies for question length.

Model $\times$ prompt interaction. Item analysis prompts help capable models but hurt weak ones. On Gemini, prerequisite_chain gains 0.08 over teacher; on Llama-8B, it loses 0.26.

Table 6: Cross-dataset generalization (Gemini 3 Flash, best temperature per dataset).

Dataset	n	teacher	prereq	buggy	Best 95% CI
SmartPaper (open-ended)	140	0.56	**0.69**	0.66	[.58,.78]
DBE-KT22 (CS MCQ)	168	0.52	0.53	**0.58**	[.46,.68]
BEA 2024 (USMLE MCQ)	595	0.45	0.44	**0.45**	[.39,.52]

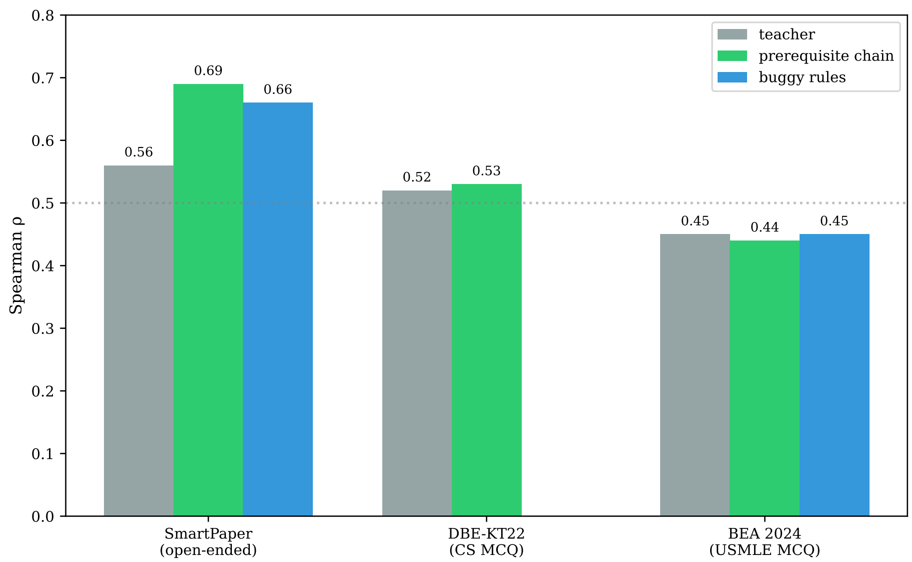


Fig. 3: Cross-dataset generalization. The item-analysis advantage (prerequisite_chain, green) is strongest on open-ended items and attenuates on MCQs, where the teacher-role prompt (grey) nearly matches structured prompts.

5.4 Generalization Across Datasets

We validated on two additional datasets: DBE-KT22 (168 undergraduate CS MCQs) and the BEA 2024 shared task (595 USMLE medical MCQs with IRT difficulty). Table 6 shows results.

The signal transfers: all prompts achieve significant correlations on all three datasets. However, the item analysis advantage attenuates on MCQs. On SmartPaper, prerequisite_chain beats teacher by 0.13; on DBE-KT22, buggy_rules achieves the highest $\rho = 0.58$ but the gap over teacher is only 0.06; on BEA, all three prompts are equivalent ($\rho \approx 0.45$).

This suggests a **structural predictability hypothesis**: LLMs estimate difficulty well when it varies across knowledge domains (open-ended items spanning topics), but the advantage of counting prompts diminishes when difficulty is driven by distractor design within a domain. The teacher-role prompt is equally effective for MCQs.

BEA 2024 Benchmark Comparison. The BEA 2024 shared task [22] evaluated difficulty prediction on 667 USMLE items using RMSE on IRT difficulty as the primary metric. The winning system (EduTec) achieved RMSE=0.299, barely improving over a mean-prediction baseline of 0.311; the best post-competition result (UnibucLLM, SVR+BERT features [18]) achieved RMSE=0.281 with Kendall $\tau = 0.28$. Our zero-shot predictions show systematic bias: models overestimate student ability. However, when calibrated via linear scaling on the BEA training set, our three-prompt ensemble (averaging teacher, prerequisite_chain, buggy_rules) achieves RMSE=0.280 and $\tau = 0.31$. This matches or slightly exceeds the best published result—despite using prompts designed for K–8 education rather than medical licensing exams. This suggests that zero-shot rank-order signal, combined with simple calibration, can match supervised feature-extraction approaches.

5.5 Practical Implications

At approximately $0.01 per item (3 replications × Gemini 3 Flash at $0.50/$3.00 per 1M tokens input/output), LLM difficulty estimation is orders of magnitude cheaper than field testing. Rankings transfer across datasets; absolute calibration does not—models overestimate student ability on SmartPaper. Notably, item analysis prompts achieve lower MAE (0.12–0.19) than the teacher-role prompt (MAE=0.44), but the rank-order is preserved regardless of calibration bias.

For practitioners: use a capable model (Gemini, GPT-4o), prompt it to analyze structural item features, and average 3+ replications. For standard MCQs, the teacher-role prompt suffices.

6 Discussion

Why Directed Analysis Works. The success of item analysis prompts suggests LLMs possess implicit pedagogical content knowledge—understanding of what makes content difficult—but this knowledge must be *elicited* through directed enumeration. Undirected chain-of-thought drifts toward solving the item; directed analysis (“list prerequisites, then estimate”) focuses that knowledge on the estimation task. This parallels findings in expertise research: experts often cannot articulate tacit knowledge until prompted with specific frameworks.

Boundary Conditions. The attenuation on MCQs reveals when LLM estimation fails: difficulty driven by *structural* features visible in the item text (prerequisites, cognitive load, error paths) is predictable; difficulty driven by *population* features absent from the text (prior exposure, distractor plausibility for specific misconceptions) is not. Practical heuristic: use structured prompts for open-ended items spanning knowledge domains; use the teacher-role prompt for MCQs within a domain.

Methodological Recommendations. Our design-space exploration surfaced three lessons. (1) *Beware Simpson's paradox in unbalanced designs.* We initially swept temperature only for promising prompts; a marginal ANOVA showed temperature as dominant ($\eta^2 = 0.74$), but this was an artifact—low temperatures were confounded with high-performing prompts. When restricted to balanced conditions, temperature explained $< 1\%$ of variance. (2) *Use two-stage sequential DOE.* Screen all prompts cheaply (1 rep, single temp) to eliminate the bottom half, then run a balanced factorial on survivors. (3) *Validate on ≥100 items.* Our 20-item probe yielded $\rho = 0.46$; the full 140-item set achieved $\rho = 0.55$. Bootstrap analysis showed only 2.5% of 20-item subsets would produce $\rho \geq 0.50$. Many published results use <50 items—treat such samples as exploratory only.

Limitations. Four limitations warrant mention. First, all 15 prompts were screened on a single model (Gemini 3 Flash); the model survey shows effects generalize but with diminished magnitude. Second, calibration does not transfer—models overestimate ability on SmartPaper but rankings are preserved, so population-specific adjustment would be needed for absolute predictions. Third, we do not compare to fine-tuned models; fine-tuned BERT/RoBERTa can outperform zero-shot LLMs when training data is available [14], but our approach targets the zero-shot case where items arrive without calibration data. Fourth, and most fundamentally: LLM estimation tells you how hard an item *looks*, not how hard it *is* for real students. The model has never been a confused 12-year-old parsing an English word problem. Field testing yields individual-level diagnostics, validity evidence, and distractor analysis that no proxy can replace.

Future Work. Three directions warrant investigation. First, batching multiple items per prompt showed preliminary improvements (+0.03–0.21 ρ) but requires systematic validation; if robust, this would further reduce per-item costs. Second, extended reasoning models (o1, DeepSeek-R1) may behave differently than the standard models tested here—our preliminary finding that thinking tokens *hurt* performance deserves replication on reasoning-optimized architectures. Third, the mechanism behind enumeration prompts remains unclear; ablation studies removing the "count the X" instruction could isolate the active ingredient and inform prompt design for other estimation tasks.

7 Conclusion

We mapped the design space for LLM item difficulty estimation across 15 prompts, 6 models, and 3 datasets (~120 configurations, ~$100 in API calls). Item analysis prompts that enumerate structural features (prerequisites, cognitive load, error paths) achieve $\rho = 0.65$–0.69 on open-ended items, but this advantage attenuates on MCQs ($\rho \approx 0.45$–0.58), where the teacher-role prompt nearly matches. Prompt design is the primary lever; model size, temperature, and deliberation each contribute ≤ 0.05 ρ. Rankings transfer across datasets but calibration does not—at ~$0.01 per item, LLM estimates can triage items into difficulty bands, but absolute predictions require population-specific adjustment. Methodologically, correlations from <50 items are unreliable, and unbalanced hyperparameter sweeps produce Simpson's paradox artifacts; we recommend two-stage sequential DOE.

Ethics Statement. This study uses anonymized, aggregated student response data. SmartPaper data consists of item-level statistics without individual student identifiers. DBE-KT22 and BEA 2024 are publicly available research datasets. No personally identifiable information was accessed or processed.

Data and Code Availability. Prompts and analysis code are available in the project repository. SmartPaper data are available upon request from the originating organization. DBE-KT22 is publicly available [6]. BEA 2024 data are available through the shared task organizers.

References

1. Acquaye, C., Huang, Y.T., Carpuat, M., Rudinger, R.: Take out your calculators: Estimating the real difficulty of question items with LLM student simulations. arXiv:2601.09953 (2025)
2. Benedetto, L., et al.: A survey on recent approaches to question difficulty estimation from text. ACM Computing Surveys **56**(5), 1–37 (2023)
3. Brown, J.S., Burton, R.R.: Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science **2**(2), 155–192 (1978)
4. Chi, M.T.H., Slotta, J.D., de Leeuw, N.: From things to processes: A theory of conceptual change. Learning and Instruction **4**(1), 27–43 (1994)
5. de Ayala, R.J.: The Theory and Practice of Item Response Theory. Guilford Press (2009)
6. Du Plessis, C., van Vuuren, J.: DBE-KT22: A Knowledge Tracing Dataset from South African Secondary Schools. Zenodo (2022). `https://doi.org/10.5281/zenodo.7096666`
7. Embretson, S.E.: A cognitive design system approach to generating valid tests. Psychological Methods **3**(3), 380–396 (1998)
8. Embretson, S.E., Reise, S.P.: Item Response Theory for Psychologists. Lawrence Erlbaum (2000)
9. Feng, W., Tran, P., Sireci, S.G., Lan, A.: Reasoning and sampling-augmented MCQ difficulty prediction via LLMs. arXiv:2503.08551 (2025)
10. Gorin, J.S.: Manipulating processing difficulty of reading comprehension questions. J. Educational Measurement **42**(4), 351–373 (2005)
11. Hoge, R.D., Coladarci, T.: Teacher-based judgments of academic achievement: A review. Review of Educational Research **59**(3), 297–313 (1989)
12. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The Knowledge-Learning-Instruction framework. Cognitive Science **36**(5), 757–798 (2012)
13. Li, M., Chen, H., Xiao, Y., et al.: Can LLMs estimate student struggles? arXiv:2512.18880 (2025)
14. Li, M., Jiao, H., Zhou, T., et al.: Item difficulty modeling using fine-tuned small and large language models. Educ. and Psych. Measurement **85**(6), 1065–1090 (2025)

15. Moore, S., Schmucker, R., Mitchell, T., Stamper, J.: Automated generation and tagging of knowledge components from multiple-choice questions. In: L@S 2024, pp. 122–133 (2024)
16. Peters, S., et al.: Text-based approaches to item difficulty modeling: A systematic review. arXiv:2509.23486 (2025)
17. Razavi, P., Powers, S.J.: Estimating item difficulty using large language models and tree-based ML. arXiv:2504.08804 (2025)
18. Rogoz, A., Ionescu, R.T.: UnibucLLM: Harnessing LLMs for automated prediction of item difficulty and response time. In: BEA 2024, pp. 502–508 (2024)
19. Südkamp, A., Kaiser, J., Möller, J.: Accuracy of teachers' judgments: A meta-analysis. J. Educational Psychology **104**(3), 743–763 (2012)
20. Sweller, J.: Cognitive load during problem solving. Cognitive Science **12**(2), 257–285 (1988)
21. Yaneva, V., et al.: Findings from the first shared task on automated prediction of difficulty. In: BEA 2024, pp. 470–482 (2024)
22. Zotos, L., van Rijn, H., Nissim, M.: Are you doubtful? Exploring model uncertainty for difficulty estimation. In: EDM 2025, pp. 77–89 (2025)

A Example Prompts

Teacher (baseline).

```
You are an experienced teacher in [subject] for Grade [N] students in
India.
For this open-ended question, estimate what proportion of students would
score full marks.
Question: [question_text]
Rubric: [rubric]
Maximum score: [max_score]
Think about:
- What specific errors or misunderstandings would cause students to
lose marks?
- How clearly does the question communicate what's expected?
- What prerequisite knowledge is needed?
- How likely are students at this grade level to have that knowledge?
Respond with ONLY a number between 0 and 1 representing the proportion
of students who would get full marks. For example: 0.45
```

Prerequisite Chain (best performer, $\rho = 0.69$).

```
[Population context: economically weaker sections, Hindi-medium backgrounds]
For this question, identify the prerequisite knowledge and skills a
student needs. Count how many independent things must ALL go right for
a correct answer. Each prerequisite is a potential failure point.
Examples of prerequisites: reading comprehension, specific vocabulary,
a math operation, a concept definition, multi-step reasoning, writing
ability.
[Question, rubric, max score]
List the prerequisites, then estimate what proportion would get full
marks.
```

```
PREREQUISITES: [list them]
COUNT: [N]
Respond with ONLY a number between 0 and 1 on the last line.
```

Buggy Rules (best hybrid, $\rho = 0.66$).

```
You are an expert in mathematical cognition and systematic student errors
(Brown & Burton, 1978).
[Population context]
For the following test item, analyze the cognitive demands:
[Grade, subject, question, rubric, max score]
Step 1: List the specific procedural steps a student must execute correctly.
Step 2: For each step, identify any known "buggy rules" -- systematic
procedural errors students commonly make (e.g., subtracting smaller
from larger regardless of position, forgetting to carry).
Step 3: Consider the target student population.
Step 4: Taking into account ALL of the above analysis holistically,
estimate what proportion of students would produce the fully correct
answer.
Respond with ONLY a number between 0 and 1 on the last line.
```

Synthetic Students (worst performer, $\rho = 0.19$). This two-stage approach first generates 10 student personas, then simulates each attempting each item.

Stage 1 — Persona generation:

```
Generate 10 diverse student profiles for a Class [N] government school
in India. [Population context]
The class distribution should reflect a typical government school:
- 4 students: Below Basic (barely literate, struggle with basic concepts)
- 3 students: Basic (can read Hindi well, weak English)
- 2 students: Competent (understand most concepts, some errors)
- 1 student: Advanced (strong understanding, rarely makes mistakes)
STUDENT 1: [Name] | Level: [level] | [2-3 specific traits]
...
```

Stage 2 — Student simulation (per item, per persona):

```
You are role-playing as this student: [persona]
[Grade, question, rubric, max score]
Role-play this specific student attempting this question. Consider their
reading ability, knowledge level, attention, and typical behaviors.
Write their actual response as they would write it, then score it.
RESPONSE: [what this student would actually write]
SCORE: [0 to max_score]
```