Quick, Create a Distractor!
Evaluating LLM Distractors for Multiple-Choice Benchmarks

Anonymous ACL submission

Abstract

Researchers use LLMs to create wrong answers (i.e., distractors) for multiple-choice question answering (MCQA) datasets, but whether LLM distractors can replace human-guided ones is unclear. We test this in three strong LLMs and three MCQA benchmarks under two tasks: *distractor generation* from question–answer pairs and *distractor extension* from MCQs. We score MCQs via metrics in education research, showing: 1) LLM distractors have trade-offs in how well they challenge test-taker models, discern model ranks, and adhere to writing quality; 2) LLM distractors rarely match the original MCQA benchmark items, so benefits are unlikely from contamination alone; and 3) LLMs struggle to balance plausibility, option length, and simulated mistakes, which humans can help monitor. From these analyses, we outline steps for rigorously using LLMs to scale MCQA benchmarks.

1 Introduction

NLP evaluation relies on multiple-choice questions (Clark et al., 2020, MCQs): models select the best answer to a question while avoiding plausible but wrong distractor choices. Writing MCQs is tedious, so researchers now prompt LLMs for support: creating distractors for MCQs from an input question/answer (Akter et al., 2026) or initial MCQ (Wang et al., 2024). Despite this popular practice, work has not tested trade-offs of using LLM distractors in MCQA without human guidance (Alhazmi et al., 2024, §4).

Our short paper proposes a focused evaluation of LLM distractors versus human-guided items with three LLMs and three MCQA datasets in two tasks: 1) *distractor generation* from an input question/answer pair; and 2) *distractor extension* from an existing MCQ (Figure 1, center). Drawing on education research (Haladyna et al., 2002), we use three metrics to test distractors' value for MCQA benchmarks (Figure 1, right): difficulty—how well the resulting MCQ challenges test-takers; discriminability—how well the MCQ separates weak and strong test-takers; and quality—how well the MCQ follows best writing practices. We instantiate (1) and (2) by fitting Item Response Theory (Lord and Novick, 2008)—a psychometrics tool—with responses from 14 test-taker LLMs, and (3) via LLM judges equipped with a 19-rule rubric from education (Tarrant et al., 2006).

In distractor generation and extension, LLM distractors often exceed human-guided items in difficulty, but can jointly sacrifice writing quality and discriminability (§3.1). We detect few overlaps between human-guided and LLM distractors, suggesting LLMs create unique distractors and are not just recalling existing benchmark items (§3.2). Qualitative analyses of such differences show weaknesses in LLM distractors: relative to humans, LLMs tend to generate distractors that are plausibly correct or obviously wrong due to distinct length (§3.3) and struggle to target high-level misconceptions (§3.4).

We validate LLMs as a promising tool for distractor generation, but find areas benchmark creators must supervise. Thus, we end with a protocol for using LLMs in MCQA design (§5). We contribute: **1)** A comparison of LLM-only and human-guided MCQA items across difficulty, discriminability, and validity—metrics grounded in education research. **2)** Evidence that LLM-written MCQs match human analogues for NLP evaluation, but qualitatively differ across writing flaw types and distractor design. **3)** A discussion of best practices for using LLMs to support the generation of new MCQA benchmarks.

2 Experimental Setup

A multiple-choice question (MCQ) is a stem q and n choices $\mathcal{C}$, with one right answer $a \in \mathcal{C}$; the rest are plausible but incorrect distractors $\mathcal{D} = \mathcal{C} \setminus \{a\}$. Distractor creation used to rely on humans, but now work uses LLMs to replace this (Qu et al., 2024).

Our evaluation studies the value of LLM distrac-

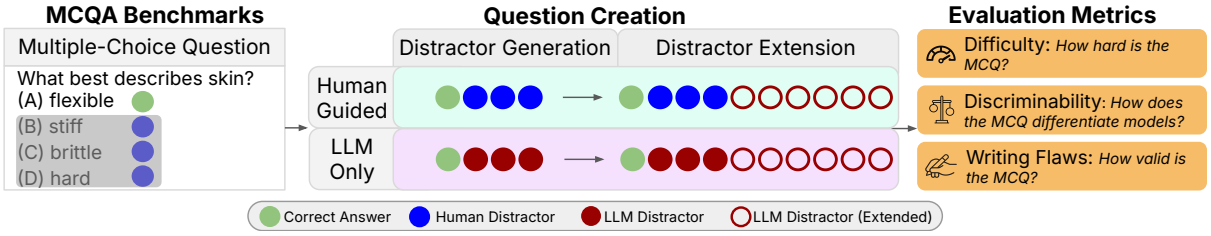

Figure 1: **Experiment Overview.** We compare MCQA distractors generated with human guidance versus LLMs alone in two tasks: 1) *distractor generation* from a question-answer pair and 2) *distractor extension* from an existing MCQ. We assess ensuing MCQs via metrics in education research: difficulty for test-taker models, discriminability of model ranks, and writing quality.

tors for NLP evaluation (Figure 1). We form tasks (§2.1) where strong generator LLMs make distractors (§2.2) to test weaker test-taker models (§2.3), scored across datasets and metrics (§2.4, §2.5).

2.1 Distractor Generation and Extension

Surveying Long et al. (2024), we test two common tasks for constructing distractors (Figure 1, center): **1) Distractor Generation (D-GEN):** For a question/answer pair (q, a), return $n-1$ distractors $\mathcal{D}$ to form an MCQ $\mathcal{Q} = (q, \mathcal{D} \cup \{a\}, a)$. D-GEN is used to format open QA tasks as MCQA or make MCQs via factual text (Su et al., 2025; Akter et al., 2026). **2) Distractor Extension (D-EXT):** For an MCQ $\mathcal{Q} = (q, \mathcal{C}, a)$, create m extra distractors $\mathcal{D}'$ to form a new MCQ $\mathcal{Q}' = (q, \mathcal{C} \cup \mathcal{D}', a)$—heightening MCQ difficulty (Wang et al., 2024; Du et al., 2025).

We aim to compare distractors generated via D-GEN and D-EXT with **LLMs-only** versus **human-guided** items. We collect MCQs $\mathcal{Q}_{\text{human}} = (q, \mathcal{C}, a)$ in MCQA datasets with $n = 4$ choices (Fig 1, left, datasets in §2.4) as the human-guided baseline for D-GEN. As the LLM-only version, a generator LLM (§2.2) executes D-GEN with $(q, a) \in \mathcal{Q}_{\text{human}}$ as input, yielding an MCQ $\mathcal{Q}_{\text{model}}$ with LLM distractors.

We study D-EXT under MMLU-Pro's protocol (Wang et al., 2024): LLMs make $m = 6$ extra distractors for an input MCQ. To collect LLM-only and human-guided versions, we use $\mathcal{Q}_{\text{model}}$ and $\mathcal{Q}_{\text{human}}$ as task inputs for D-EXT, respectively; the latter is human-guided, as humans write initial distractors.

2.2 Generator Models

Benchmark creation is a one-time operation, so researchers use stronger LLMs for help (Wang et al., 2024). We thus use three frontier LLMs as generator models for §2.1: GPT-5.2 (OpenAI, 2025), Gemini-3.1 Pro (Google DeepMind, 2026), and Qwen-3.5 397B (Qwen Team, 2026). We prompt models to return three and six distractors in D-GEN and D-EXT, and format outputs as JSON for parsing. In D-EXT's LLM-only setting, generator LLMs only extend MCQs they generated in D-GEN. We run LLMs in InspectAI (UK AISI, 2024) with default temperature, 10k max. tokens, medium reasoning effort, and two retries on errors (Appendix A.2).

2.3 Test-Taker Models

We have 14 LLMs answer MCQs from §2.2 as test-taker models to test NLP evaluation impact (§2.5). We use GPT-5.4-mini[1] and 13 open-weight models like Gemma-3 (Team et al., 2025, 4B) and Qwen-3 (Yang et al., 2025, 4B). LLMs use up to 32,768 max output tokens. GPT uses high reasoning. Before evaluation, we shuffle MCQ choices for positional bias (Wei et al., 2024). We evaluate models with InspectAI's MCQA prompt (UK AISI, 2024). Appendix A.3 has all models/parameters/prompts.

2.4 Datasets

We use MCQs in three common benchmarks of varied difficulty: **1) ARC** (Clark et al., 2018), grade-school science; **2) MMLU** (Hendrycks et al., 2021), 57 college topics; and **3) GPQA** (Rein et al., 2024), graduate science. We sample 1000 MCQs in (1-2) and all 448 in (3); we omit items where any generator cannot output JSON in D-GEN or D-EXT, yielding 995/954/415 items in ARC/MMLU/GPQA.[2]

2.5 Multiple-Choice Quality Metrics

To score MCQs, we review work in education and NLP (Haladyna et al., 2002; Balepur et al., 2025)—what educators deem useful and what makes it effective for NLP evaluation—yielding three metrics: **1) Difficulty:** how well the MCQ challenges test-takers (Akhtar et al., 2026); **2) Discriminability:** how well it separates low/high-skill test-takers to informatively discern their abilities (Polo et al., 2024); and **3) Quality:** how well it tests intended skills versus confusing test-takers, based on writing flaw count (Moore et al., 2022, e.g., ambiguity).

[1] https://tinyurl.com/gpt54-mini-api

[2] Omitted questions often require lengthy distractors infeasible to use an LLM for (Appendix A.1, e.g., DNA sequences).

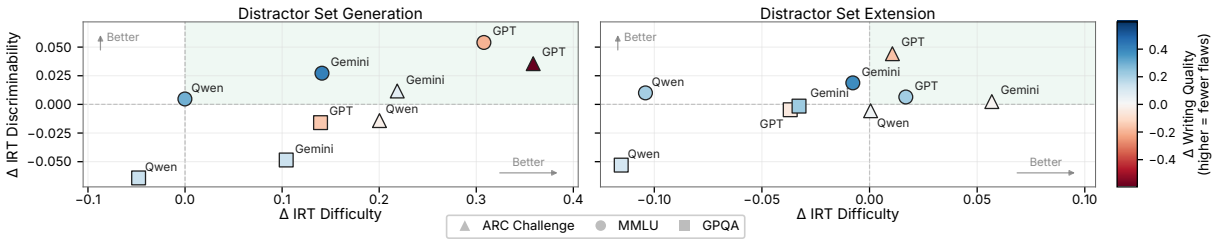

Figure 2: Differences in difficulty, discriminability, and writing quality of MCQs with generated or extended distractors using LLMs alone compared to with human guidance. Positive differences mean the LLM distractors surpass human-guided ones in that metric. Three LLM-generated distractor sets beat their human-guided analogues in all three metrics but only one is worse in all metrics. Thus, deciding whether to use LLM distractors largely depends on which metrics the benchmark developer prioritizes.

We implement (1) and (2) by fitting an Item Response Theory (Lord and Novick, 2008, IRT, details in §A.4) model on test-taker responses in §2.3. IRT models each test-taker's skill from their overall accuracy, then combines that with the correctness of model responses on each MCQ to learn latent per-item difficulty/discriminability scores. We measure (3) with BenchMarker (Balepur et al., 2026), a tool that uses LLM judges to flag 19 writing flaws from education research that mislead test-takers (Tarrant et al., 2006)—like ambiguity, implausible distractors, and many right answers. We report the number of writing flaws, but study specific flaws in §3.3.

3 Results: Do LLMs Write Useful MCQs?

We now answer our research question: can LLM distractors usurp human-guided ones in benchmarks? We show LLM distractors have difficulty, discriminability, and writing quality trade-offs (§3.1), so this answer largely depends on metrics researchers value. After finding LLMs do not copy benchmarks (§3.2), we qualitatively reveal LLM issues in flaw types and error simulation (§3.3, §3.4)—informing our protocol for LLM use in MCQA creation (§5).

3.1 LLM-Created Distractors Have Trade-Offs

We test how LLM distractors impact MCQA evaluation, studying differences in our metrics (§2.5) over LLM-only and human-guided distractors (Figure 2). In nine generator-benchmark combinations, two LLM distractor sets (Gemini on MMLU/ARC) beat humans over all metrics in D-GEN and one for D-EXT (GPT on MMLU); only one distractor set is worse in all metrics (GPT on GPQA for D-EXT).

LLM distractors often have trade-offs. In D-GEN, 8 LLM distractor sets exceed humans in difficulty, but 4 lower discriminability and writing quality; on ARC, GPT heightens difficulty by 0.4 but adds 0.6 writing flaws per item, so difficulty here may stem from confusing quality issues (Moore et al., 2022). In D-EXT, metric differences are model/data specific, but we find trends: GPT tends to add writing flaws; LLM distractors on GPQA are easier.

	GPT-5.2	Gemini-3.1 Pro	Qwen-3.5 397B
ARC	14.7%	22.2%	19.8%
MMLU	8.3%	15.8%	12.3%
GPQA	19.4%	24.8%	21.4%

Table 1: LLM-only distractors in D-GEN that exactly match an original benchmark distractor. LLMs create distinct MCQs.

Deciding to use LLM distractors depends on the metric researchers value—test-taker scores (difficulty), rankings (discriminability), or item validity (writing quality). We later outline a decision protocol (§5) researchers can apply to their own datasets.

3.2 LLM Questions Are Not Just Recycled

If human-guided and LLM-only distractors are identical (e.g., contamination), studying them qualitatively or debating which to use would be moot. To test this, we follow Du et al. (2025) and find few LLM distractors in D-GEN that exactly match any distractor from the original benchmark items (Table 1)—based on a 0.90 cosine similarity threshold via MiniLM-L6-v2 (Reimers and Gurevych, 2019) for text-based distractors, and exact match for numeric distractors. While this does not flag all overlap types (e.g., knowledge), it detects the simplest ones and motivates qualitative analyses on differences in human-guided and LLM items (§3.3, §3.4).

3.3 LLM Distractors Contain Writing Flaws

As LLM distractors differ from human-guided ones (§3.2), we now compare them qualitatively. We first analyze the largest differences in writing flaw prevalence between LLM and human-guided distractors, based on the 19 writing flaw types we detect (§2.5).

In D-GEN, LLM distractors have mixed effects on writing quality over human-guided items (Table 2). Reduced convergence cues limit shortcuts test-takers can use to bypass the item (Yuan et al., 2024), but LLMs negate this by adding long, obvious distractors. LLMs lower implausible distractors,

Writing Flaw Type	D-GEN Human	D-GEN LLM	D-EXT Human	D-EXT LLM
Unequal Option Length	20.4	28.0	37.5	37.0
Implausible Distractors	11.2	7.8	22.4	19.3
Has Absolute Terms	9.6	14.4	23.2	23.7
Multiple Best Answers	12.2	15.3	29.3	28.6
Has Convergence Cues	21.6	19.1	20.6	19.3

Table 2: Percentage of five writing flaws in human-guided and LLM-only distractors in D-GEN and D-EXT. LLMs struggle to maintain option length and limit plausible answers. D-EXT tends to increase writing flaws. Appendix A.6 has all 19 flaws.

Model	SOLVE	SIMULATE	RECALL	SHALLOW	REFINE
GPT	15%	82%	40%	13%	7%
Gemini	57%	83%	48%	12%	82%
Qwen	60%	88%	53%	12%	100%

Table 3: How often LLMs use varied reasoning trace strategies when making 60 distractors in D-GEN on GPQA. LLMs mirror educators: they re-solve MCQs, simulate mistakes, and recall facts. Appendix A.8 has examples and MMLU/ARC results.

but more "many best answers" show they overcompensate, struggling to weigh plausibility (Lee et al., 2025). By applying NLP tools such as LLM judges (§2.5) to LLM distractors, researchers can find areas to intervene (e.g., shortening lengthy distractors).

In D-EXT, differences are tempered, as writing flaws increase over all distractor sets. Researchers using LLMs for distractor extension (Lee and Son, 2026) should report their increased writing flaws, and evaluate how they influence test-taker accuracy.

3.4 LLMs Construct Distractors like Humans

We lastly see *how* LLMs create distractors via their reasoning traces (Deng et al., 2025). We review 60 random traces for all generator LLMs on GPQA in D-GEN and find five strategies LLMs often use: 1) **SOLVE** the item, 2) **SIMULATE** test-taker mistakes, 3) **RECALL** relevant facts; 4) add a **SHALLOW** perturbation of the gold answer (e.g., negative sign); and 5) **REFINE** distractors (Table 3). Except SHALLOW, educators use similar strategies (Gierl et al., 2017); future work can test whether education can further steer LLM reasoning for distractor design.

While this is promising, generator LLMs do not always aptly target test-taker errors in SIMULATE; they often perturb steps in their own solution versus focusing on high-level mistakes real test-takers would make. To illustrate, for one MCQ where test-takers need to solve a physics equation with the right answer 3pc, Gemini's distractors drop a correction or temperature term, or invert the former (Appendix Figure A.1). While still valid, PhD-level test-takers—the expected skill level of GPQA—are unlikely to make such simple errors. Thus, we advise researchers to study generator reasoning traces for LLM distractors—ensuring they target failures test-takers at their desired skill level would make.

4 Related Work

Distractor Generation: Distractor generators curb MCQ writing effort. Early work used heuristics/neural models (Mitkov et al., 2003; Liang et al., 2018) before LLMs (Bitew et al., 2023; Mucciaccia et al., 2025). Most work uses such models for education (Alhazmi et al., 2024; Feng et al., 2024; Lee et al., 2025, e.g., student exams), but MCQA's use in NLP has motivated applying such methods to benchmark items (Zellers et al., 2018; Liu et al., 2022; Wang et al., 2024). We bridge these fields: drawing on education to score LLM distractor quality for NLP evaluation (Haladyna et al., 2002). Closest to us, Byun and Choi (2025) train models on MMLU to create distractors, but do not prompt stronger LLMs.
Synthetic Data: Synthetic data is rife (Long et al., 2024), used to train models (Li et al., 2025; Nadăș et al., 2025), extend evaluations (Wang et al., 2024; Zhang et al., 2025), or imbue model traits (Thrush et al., 2026). Work increasingly tests the benefits and risks of such data, including for AI capabilities (Shumailov et al., 2024; Liu et al., 2024), biases (Breugel et al., 2023; Rahmani et al., 2025), and privacy (Liu et al., 2025; Xu et al., 2026). Complementing this, we audit LLM-generated distractors.

5 Protocol: When to Use LLM Distractors

Our evaluation of generator models, MCQA benchmarks, and education-grounded metrics shows the potential and weaknesses of LLMs for constructing distractors. Given our mixed results, we outline the following protocol to help benchmark creators decide when to use LLMs over humans for distractors:

1. Use humans to author a subset of distractors.
2. Adapt our evaluation protocol to score human-written and LLM-generated distractors in difficulty, discriminability, and writing flaws (§2); evaluate distractor extension tasks if needed.
3. Compare metric trade-offs and locate which model optimizes the metrics you value (§3.1).
4. Qualitatively analyze items, including writing flaw breakdowns (§3.3) and reasoning trace analyses (§3.4), to identify areas for improvement or where human intervention is needed.

Creating MCQA benchmarks is tedious, but necessary to make progress in NLP (Clark et al., 2020). Our paper shows LLMs can facilitate these efforts with distractor generation—if validated rigorously.

6 Limitations

LLMs are sensitive to prompting (Alzahrani et al., 2024), so modifying our prompts for distractor generation and extension (§2.2) and MCQA evaluation (§2.3) could impact our results. However, we mostly argue that LLMs may already match the quality of human-written MCQs for NLP evaluation, so our results form a lower bound; improvements to prompt design would only strengthen the trend. Appendix A.3 has our prompts for reproducibility.

Further, while we identify no instances of contamination (§3.2), we cannot eliminate the possibility of generation models having already seen the human MCQs without access to their training data (Soldaini et al., 2024). Thus, as mentioned in §5, we encourage future work to adopt our evaluation setup on their own—and presumably unseen–MCQA benchmarks. Such evaluations would also strengthen the generalization of our analyses. We open-source all code/data to facilitate these efforts.

Finally, we acknowledge many auxiliary dimensions that we could focus our evaluation on—such as generation model costs, the number of distractors, and studying inter-model versus intra-model extension—that we did not consider due to resource constraints. For the scope of a short paper, we specifically target the question of the role of human guidance in MCQA benchmark design with the strongest possible generator models—a useful prerequisite before exploring these follow-up analyses. We release this paper to inspire such future work.

7 Ethical Considerations

While we show the benefits of using LLMs to support MCQA benchmark design, we never argue that these items are sufficient for real student testing. In fact, §3.4 reveals issues in LLM-generated MCQs that may degrade usefulness in education settings, so we urge researchers to more rigorously evaluate MCQs before using them with real students. Further, there exist other instantiations of MCQ quality for NLP evaluation that our paper does not consider—including shortcuts (Lai et al., 2021), biases (Wei et al., 2024), and self-preference (Panickssery et al., 2024)—that may be worth measuring before deciding to release LLM-generated MCQA benchmarks.

Generative AI (GenAI) was used in this project. We used Codex[3] to design plots and refactor code, and GPT-5 to refine paper writing for brevity. We never use GenAI for writing text from scratch in this paper. We take complete responsibility for any GenAI errors. By discussing GenAI usage here, we aim to encourage other researchers to do the same.

[3] https://openai.com/codex/

References

Mubashara Akhtar, Anka Reuel, Prajna Soni, Sanchit Ahuja, Pawan Sasanka Ammanamanchi, Ruchit Rawal, Vilém Zouhar, Srishti Yadav, Chenxi Whitehouse, Dayeon Ki, Jennifer Mickel, Leshem Choshen, Marek Šuppa, Jan Batzner, Jenny Chim, Jeba Sania, Yanan Long, Hossein A. Rahmani, Christina Knight, and 18 others. 2026. When ai benchmarks plateau: A systematic study of benchmark saturation. *Preprint*, arXiv:2602.16763.

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, and 1 others. 2026. Nemotron-crossthink: Scaling self-learning beyond math reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 984–1002.

Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Munazza Zaib, and Ahoud Alhazmi. 2024. 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*, pages 14437–14458, Miami, Florida, USA. Association for Computational Linguistics.

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Balepur, Bhavya Rajasekaran, Jane Oh, Michael Xie, Atrey Desai, Vipul Gupta, Steven James Moore, Eunsol Choi, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2026. Benchmarker: An education-inspired toolkit for highlighting flaws in multiple-choice benchmarks. *arXiv preprint arXiv:2602.06221*.

Nishant Balepur, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2025. Which of these best describes multiple choice evaluation with LLMs? a) forced B) flawed C) fixable D) all of the above. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3394–3418, Vienna, Austria. Association for Computational Linguistics.

Semere Kiros Bitew, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Distractor generation for multiple-choice questions with predictive prompting and large language models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 48–63. Springer.

B. V. Breugel, Nabeel Seedat, F. Imrie, and M. Schaar. 2023. Can you rely on your model evaluation? improving model evaluation with synthetic test data. *ArXiv*, abs/2310.16524.

Grace Byun and Jinho D. Choi. 2025. *D-GEN*: Automatic distractor generation and evaluation for reliable assessment of generative models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3316–3349, Vienna, Austria. Association for Computational Linguistics.

Gregory Camilli. 1994. Teacher's corner: origin of the scaling constant d= 1.7 in item response theory. *Journal of educational and behavioral statistics*, 19(3):293–295.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, and 1 others. 2020. From 'f'to 'a'on the ny regents science exams: An overview of the aristo project. *Ai Magazine*, 41(4):39–53.

Amy Deng, Sydney Von Arx, Ben Snodin, Sudarsh Kunnavakkam, and Tamera Lanham. 2025. CoT may be highly informative despite "unfaithfulness".

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, and 76 others. 2025. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wanyong Feng, Jaewook Lee, Hunter McNichols, Alexander Scarlatos, Digory Smith, Simon Woodhead, Nancy Otero Ornelas, and Andrew Lan. 2024. Exploring automated distractor generation for math multiple-choice questions via large language models. *arXiv preprint arXiv:2404.02124*.

Mark J Gierl, Okan Bulut, Qi Guo, and Xinxin Zhang. 2017. Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of educational research*, 87(6):1082–1116.

Google DeepMind. 2026. Gemini 3.1 Pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`. Accessed: 2026-05-17.

Thomas M Haladyna, Steven M Downing, and Michael C Rodriguez. 2002. A review of multiple-choice item-writing guidelines for classroom assessment. *Applied measurement in education*, 15(3):309–333.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? *arXiv preprint arXiv:2106.01024*.

Nahyun Lee and Guijin Son. 2026. Pushing the boundaries of multiple choice evaluation to one hundred options. *arXiv preprint arXiv:2604.14634*.

Yooseop Lee, Suin Kim, and Yohan Jo. 2025. Generating plausible distractors for multiple-choice questions via student choice prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23669–23692, Vienna, Austria. Association for Computational Linguistics.

Ming Li, Pei Chen, Chenguang Wang, Hongyu Zhao, Yijun Liang, YuPeng Hou, Fuxiao Liu, and Tianyi Zhou. 2025. Mosaic-IT: Cost-free compositional data synthesis for instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25287–25318, Vienna, Austria. Association for Computational Linguistics.

Chen Liang, Xiao Yang, Neisarg Dave, Drew Wham, Bart Pursel, and C Lee Giles. 2018. Distractor generation for multiple choice questions using learning to rank. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 284–290.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*.

Qinyi Liu, Oscar Blessed Deho, Farhad Vadiee, Mohammad Khalil, Srecko Joksimovic, and George Siemens. 2025. Can synthetic data be fair and private? A comparative study of synthetic data generation and fairness algorithms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK 2025, Dublin, Ireland, March 3-7, 2025*, pages 591–600. ACM.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024. Best practices and lessons learned on synthetic data. In *First Conference on Language Modeling*.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

Frederic M Lord and Melvin R Novick. 2008. *Statistical theories of mental test scores*. IAP.

Ruslan Mitkov and 1 others. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*, pages 17–22.

Steven Moore, Huy A Nguyen, Norman Bier, Tanvi Domadia, and John Stamper. 2022. Assessing the quality of student-generated short answer questions using gpt-3. In *European conference on technology enhanced learning*, pages 243–257. Springer.

Sérgio Silva Mucciaccia, Thiago Meireles Paixão, Filipe Wall Mutz, Claudine Santos Badue, Alberto Ferreira de Souza, and Thiago Oliveira-Santos. 2025. Automatic multiple-choice question generation and evaluation systems based on llm: A study case with university resolutions. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2246–2260.

Mihai Nadăș, Laura Dioșan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*.

OpenAI. 2025. Introducing GPT-5.2. https://openai.com/index/introducing-gpt-5-2/. Accessed: 2026-05-17.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinybenchmarks: evaluating llms with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Fanyi Qu, Hao Sun, and Yunfang Wu. 2024. Unsupervised distractor generation via large language model distilling and counterfactual contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 827–838.

Qwen Team. 2026. Qwen3.5-397B-A17B. https://huggingface.co/Qwen/Qwen3.5-397B-A17B. Accessed: 2026-05-17.

Hossein A Rahmani, Varsha Ramineni, Emine Yilmaz, Nick Craswell, and Bhaskar Mitra. 2025. Towards understanding bias in synthetic data for evaluation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5166–5170.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2459–2475.

Marie Tarrant, Aimee Knierim, Sasha K Hayes, and James Ware. 2006. The frequency of item writing flaws in multiple-choice questions used in high stakes nursing assessments. *Nurse Education Today*, 26(8):662–671.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Tristan Thrush, Sung Min Park, Herman Brunborg, Luke Bailey, Marcel Roed, Neil Band, Christopher Potts, and Tatsunori Hashimoto. 2026. Synthetic data for any differentiable target. *arXiv preprint arXiv:2604.08423*.

UK AISI. 2024. Inspect AI: Framework for Large Language Model Evaluations.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sheng-Lun Wei, Cheng-Kuang Wu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. Unveiling selection biases: Exploring order and token sensitivity in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5598–5621, Bangkok, Thailand. Association for Computational Linguistics.

Fangyuan Xu, Sihao Chen, Zinan Lin, Taiwei Shi, Sydney Graham, Pei Zhou, Mengting Wan, Alex Stein, Virginia Estellers, Charles Chen, Morris Sharp, Richard Speyer, Tadas Baltrusaitis, Jennifer Neville, Eunsol Choi, and Longqi Yang. 2026. Dp-rft: Learning to generate synthetic text via differentially private reinforcement fine-tuning. *Preprint*, arXiv:2602.18633.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. 2024. Do LLMs overcome shortcut learning? an evaluation of shortcut challenges in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12188–12200, Miami, Florida, USA. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah Aklilu, Alejandro Lozano, Anjiang Wei, and 1 others. 2025. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29580–29590.

A Appendix

A.1 Data

We initially sample 1000 random examples from the test set of ARC-Challenge (Clark et al., 2018) and MMLU (Hendrycks et al., 2021). MMLU questions were drawn from a pool of question seeds used in MMLU-Pro (Wang et al., 2024). We use 448 examples of GPQA (Rein et al., 2024), the full question set. All datasets are publicly available, so our experiments are within their intended use. ARC-Challenge and GPQA are licensed under CC BY 4.0 and MMLU under MIT. We did not collect any datasets, so we did not check for PII. To our knowledge, all questions are in English.

We check for bad questions using automated methods—check for missing question stems and answer choices—and manual inspection. We filter out one MMLU question (QA.1) and four GPQA questions (QA.2, QA.3, QA.4, QA.5).

Additionally, we filter questions if a generator model creating MCQs fails to produce a valid output across any setting for a given human MCQ seed, yielding **995** total items in ARC-Challenge, **954** in MMLU, and **415** in GPQA.

We use datasets consistent with their intended use in NLP research and evaluating model knowledge and reasoning capabilities. We use this data as seeds for downstream question generation. Any generated data released with this paper should not be used unsupervised or replacements for educational tests without additional review as mentioned in recommended guidelines in Section 5.

We do not collect new human-subject data. We manually inspect a portion of the generated artifacts—stems, answer options, generated distractors, model traces—and to our best knowledge, we do not find PII or offensive content.

A.2 InspectAI Implementation

InspectAI[4] is a recent effort from the United Kingdom's AI Security Institute to standardize NLP evaluations (UK AISI, 2024). Additionally, we also use InspectAI for initial creation of LLM-generated questions, as described below. Any InspectAI framework contains three parts:

1. **Task:** The data for the task. We have separate tasks for (1) question generation and (2) test-taker evaluation. For (1), each task is made up of a question stem, gold answer, and human distractor in the distractor set generation setting and required distractors from the prior setting for the distractor set extension setting. For (2), each task contains the question stem, randomized answer choices, the index of the gold answer, and associated metadata for data classification.

2. **Solver:** The NLP system that solves the task. For (1), given an QA pair or complete MCQ, the solver prompts a generator LLM to create three distractors or six distractors, respectively. Following generation, it checks JSON output for invalid or duplicate distractors, shuffles answer choices, and logs the question with associated metadata including model reasoning traces. For (2), the solver prompts a test-taker LLM with either a full question or choices-only prompt and parses for an answer letter in the generated JSON.

3. **Scorer:** How task success/failure is evaluated. For (1) and (2), the scorer records if an item successfully generated; additionally, the scorer for (2) computes item accuracy and records other information such as the gold answer letter.

A.3 LLM Details

We access all models through InspectAI; models are passed through using provider-prefixed model identifiers. Hosted models use InspectAI provider strings such as `openai/...` for GPT models, `google/...` for Gemini models, and `together/...` for Qwen models. Open-weight models are locally run using `vllm/...` identifiers.

For generation, we use three generator LLMs:

- openai/gpt-5.2-2025-12-11
- google/gemini-3.1-pro-preview
- together/Qwen/Qwen3.5-397B-A17B

For generation, we specify the following parameters: maximum generation length of 10,000 tokens, default temperature, two model-call retries in case of initial generation failure, and a maximum for three parsing attempts for each item. When supported by models, we use medium reasoning effort. This corresponds to medium in GPT-5.2 and Gemini-3.1-Preview, and goes unused for Qwen3.5-397B-A17B (not supported). We instead enable thinking mode through the binary toggle.

GPT-5.2 and Gemini-3.1 Pro are proprietary models, so parameter counts are not publicly dis-

[4] https://inspect.aisi.org.uk/

closed. Qwen3.5-397B-A17B is a mixture of experts (MoE) model with 397B total parameters and 17B active parameters.

For model evaluation, we use the following 14 test-taker models:

- openai/gpt-5.4-mini
- nvidia/NVIDIA-Nemotron-Nano-9B-v2
- google/gemma-3-4b-it
- google/gemma-3-12b-it
- Qwen/Qwen3-4B-Instruct-2507
- allenai/Olmo-3-7B-Instruct
- cyankiwi/Olmo-3.1-32B-Instruct-AWQ-4bit
- meta-llama/Llama-3.2-3B-Instruct
- meta-llama/Llama-3.1-8B-Instruct
- LiquidAI/LFM2-2.6B
- LiquidAI/LFM2-1.2B
- LGAI-EXAONE/EXAONE-4.0-1.2B
- ibm-granite/granite-4.0-1b
- ibm-granite/granite-4.0-micro

For evaluation, we specify the following parameters: maximum generation length of 32,768 tokens, no temperature override, two model-call retries, and no stop sequence override. Due to architectural constraints of some models, we modify the max model length for the following: Gemma (124k) and certain LFM & older Qwen3 models (32k). GPT-5.4-mini is run on high reasoning effort and the remaining vLLM-based test-taker models are run on medium reasoning. No used vLLM models support reasoning effort, but we enable thinking mode if available.

Test-taker model sizes range from 1B to 9B, with Olmo-3.1-32B-Instruct as the only exception. Additionally, GPT-5.4-mini is proprietary and parameter count is not publicly disclosed.

We use CPUs only when running LLMs via APIs and one NVIDIA rtxa6000 GPU for open-weight LLMs below 8B parameters called via vLLM. We allocate 12 hours for each run, though models jobs are split into separate SLURM jobs by each setting and dataset for parallel speedup. All results are reported from a single run.

Across all generation and evaluation runs, experiments use 657.1M tokens across 377.8 job-hours. Generation runs used a collective 94.9M tokens and 40.5 job-hours. Evaluation used 562.3M tokens across 337.3 job-hours. Token counts include input, reasoning, and output tokens.

We do not perform a hyperparameter search over sampling settings to keep settings consistent between different models in InspectAI. Generation and evaluation settings, as described above, are fixed for final runs.

A.3.1 Prompts

Prompts are based on prompts used by (Wang et al., 2024) and general recommendations by InspectAI. After testing prompts, we mainly change the output format from looser labeled distractors to a formal JSON format with a "distractor" list, containing a list of exact choices depending on the input distractor count. We also move away from formal forbidden structures through language such as "Do not use any forbidden value listed below" and use semantic guidance such as generating plausible incorrect answers and avoiding synonyms of distractors in the returned set.

For final prompts used in all reported runs, see Prompts A.2 and A.3 for generator model generation prompts and Prompt A.1 for the test-taker model evaluation prompt.

A.3.2 Licenses

We follow original licenses and terms of use of all models and other software used. We use InspectAI under its MIT license and vLLM and sentence-transformers/all-MiniLM-L6-v2 under Apache 2.0. Hosted models APIs each have relevant provider terms: OpenAI (GPT5.2, GPT-5.4-mini), Google (Gemini-3.1-Pro-Preview), and Together AI (Qwen3.5-397B-A17B generation through Together). For open-weight model, we mainly obtain license details from the corresponding HuggingFace model cards. These include licenses such as Apache 2.0 (Qwen, OLMo, IBM Granite), Gemma terms, Llama 3.x terms, and other model-specific licenses for Nemotron-Nano, LFM2, and EXAONE.

A.4 IRT Details

We use a 3PL model fit over full-question model-evaluation responses only. Given test-taker (j) and instantiated item (i):

$$P(Y_{ij} = 1) = c_i + (1 - c_i)\sigma\left(1.702 a_i(\theta_j - b_i)\right),$$

such that (θ_j) is test-taker ability, (b_i) is item difficulty, (a_i) is item discrimination, and (c_i) is the guessing parameter (either 0.25 for 4-choice questions or 0.10 for 10-choice questions). The 1.702 scaling factor is a standard constant to minimize the maximum difference between the normal and logistic distribution functions (Camilli, 1994).

Rank	Test-taker model	Relative ability
1	GPT-5.4-mini	2.591
2	Nemotron-Nano-9B-v2	1.349
3	Gemma-3-12B-it	0.543
4	Qwen3-4B-Instruct	0.487
5	Olmo-3.1-32B-Instruct-AWQ	0.361
6	Llama-3.1-8B-Instruct	−0.002
7	LFM2-2.6B	−0.219
8	EXAONE-4.0-1.2B	−0.358
9	Gemma-3-4B-it	−0.447
10	Olmo-3-7B-Instruct	−0.487
11	Granite-4.0-micro	−0.557
12	Llama-3.2-3B-Instruct	−0.622
13	Granite-4.0-1B	−1.050
14	LFM2-1.2B	−1.589

Table 4: Relative abilities of each test-taker model based on θ_j parameter. Ability estimates are centered by subtracting the mean ability across the 14 test-taker models.

Dataset	Source stems	Difficulty effect
GPQA	415	−0.046
MMLU	954	−0.723
ARC-Challenge	995	−2.493

Table 5: Relative abilities of each dataset based on $\beta^{\text{dataset}}_{d(i)}$ sub-parameter. Larger estimates correspond to harder datasets.

Difficulty is further broken down into components as:

$$b_i = \beta^{\text{dataset}}_{d(i)} + \beta^{\text{stem}}_{s(i)} + \beta^{\text{generator}}_{g(i)} + \beta^{\text{setting}}_{r(i)} + \epsilon_i.$$

Each instantiated item is based on the dataset, question stem, generator model, and setting.

We also include fit details:

- Observations: 496,440
- Test-taker models: 14
- Unique stems: 2,364
- Instantiated items: 35,460
- Parameters: 108,764
- Optimizer: L-BFGS-B
- Converged: YES, after 1,870 iterations
- Log likelihood: -169915.14
- AIC: 557358.27
- BIC: 1766293.83

Additionally, we provide per-parameters results: Tables 4 (test-taker), 5 (dataset), and 6 (setting).

A.5 Full Quality Results

We report the full details of the delta differences shown in Figure 2 in Figure 3. Difficulty and discriminability are estimated with the IRT model described in Appendix A.4; writing quality is the average number of BenchMarker writing flaws per item.

We also attempt to generate the complete extended distractor set all at once but found it was worse than the two-step process (Figure 4).

Setting	Difficulty effect
Extend Human	0.611
Extend Model	0.597
Extend Ablation	0.430
Generate Model	0.139
Generate Human	0.000

Table 6: Setting-level IRT difficulty effects based on $\beta^{\text{setting}}_{r(i)}$ sub-parameter. Using Generate Human as the reference setting, larger estimates indicate that a setting increases item difficulty relative to the reference.

Statistic	Value
Mean flaws/item	2.641 ± 0.009
SD	1.690
Median	2
IQR	1–4
Range	0–12
Zero flaws	7.7%
≥ 2 flaws	73.0%

Table 7: Overall flaw-count distribution.

A.6 Full Writing-Flaw Results

Table 7 and 8 have general flaw-count information and a full BenchMarker writing-flaw breakdown.

A.7 Memorization Details

We test if LLM-generated distractors produce identical distractors to the original human distractors and break down matches in Table 9.

A.8 Qualitative Analysis Details

We define the five annotated metrics as:

SOLVE The model re-derives the solution for the gold answer from the question step by step.

SIMULATE The model attempts to simulate test-taker mistakes such as wrong calculation, formula misuse, etc. Then, the model assigns each distractor to a specific error.

RECALL The model creates distractor based on related-but-distinct subject information, such as alternative reaction products, adjacent elements/compounds, parallel mechanisms, etc.

SHALLOW The model makes simple perturbation of the correct answer, such as swapping signs, adjacent integers, rearrange, etc.

REFINE The model reflects over the entire set and critics distractors for aspects like, distinctness, plausibility, and format. Then, it revises the distractors.

Data in Table 3 is broken down into component datasets in Table 10. We also provide the full trace of the Gemini question referred to in 3.3: A.1.

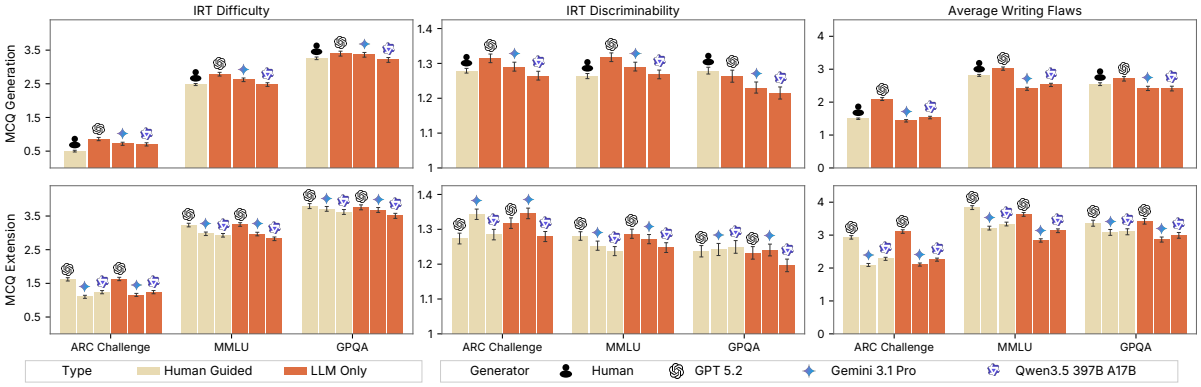

Figure 3: Per-model breakdown of MCQ question quality across three metrics: IRT difficulty (how hard is the question), IRT Discriminability (how well does the question tell test-takers apart), and Average Writing Flaws (a proxy for question validity).

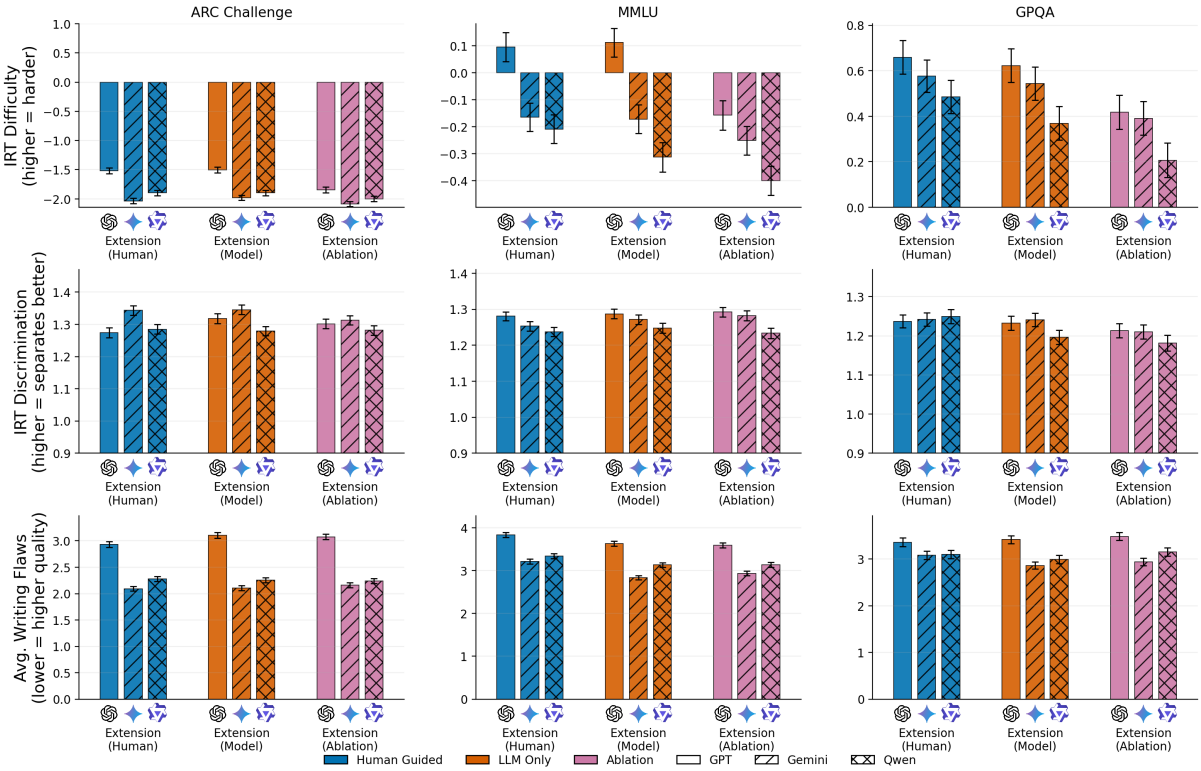

Figure 4: Per model measurements of difficulty, discriminability, and writing quality for extension setting. We find that the two-step methods in 2.1 outperform the one-step ablation in generating the extended distractor set across most metrics, particularly in overall MCQ difficulty.

Writing Flaw Type	D-GEN		D-EXT	
	Human	Model	Human	Model
Unequal Option Length	$\mathbf{20.4 \pm 0.48}$	28.0 ± 0.53	37.5 ± 0.57	$\mathbf{37.0 \pm 0.57}$
Implausible Distractors	11.2 ± 0.37	$\mathbf{7.8 \pm 0.32}$	22.4 ± 0.50	$\mathbf{19.3 \pm 0.47}$
Has Absolute Terms	$\mathbf{9.6 \pm 0.35}$	14.4 ± 0.42	$\mathbf{23.2 \pm 0.50}$	23.7 ± 0.50
Unclear Language	8.7 ± 0.33	$\mathbf{6.9 \pm 0.30}$	11.2 ± 0.37	$\mathbf{9.2 \pm 0.34}$
Multiple Best Answers	$\mathbf{12.2 \pm 0.39}$	15.3 ± 0.43	29.3 ± 0.54	$\mathbf{28.6 \pm 0.54}$
Has Convergence Cues	21.6 ± 0.49	$\mathbf{19.1 \pm 0.47}$	20.6 ± 0.48	$\mathbf{19.3 \pm 0.47}$
Has Extraneous Information	52.7 ± 0.59	$\mathbf{51.4 \pm 0.59}$	$\mathbf{50.0 \pm 0.59}$	50.4 ± 0.59
Unfocused Stem	$\mathbf{28.4 \pm 0.54}$	28.5 ± 0.54	28.6 ± 0.54	28.6 ± 0.54
Problem Not in Stem	12.2 ± 0.39	$\mathbf{11.4 \pm 0.38}$	11.2 ± 0.37	11.2 ± 0.38
Unordered Options	11.3 ± 0.38	$\mathbf{11.2 \pm 0.37}$	$\mathbf{11.3 \pm 0.38}$	11.5 ± 0.38
Has Vague Terms	5.6 ± 0.27	5.6 ± 0.27	$\mathbf{12.2 \pm 0.39}$	13.0 ± 0.40
Has Logical Cues	6.1 ± 0.28	$\mathbf{4.4 \pm 0.24}$	11.7 ± 0.38	$\mathbf{10.0 \pm 0.36}$
Grammatical Inconsistency	5.1 ± 0.26	$\mathbf{4.4 \pm 0.24}$	8.6 ± 0.33	$\mathbf{7.3 \pm 0.31}$
Has K-Type Options	3.0 ± 0.20	$\mathbf{2.4 \pm 0.18}$	9.5 ± 0.35	$\mathbf{8.1 \pm 0.32}$
Has Repetition	5.9 ± 0.28	$\mathbf{4.0 \pm 0.23}$	$\mathbf{2.9 \pm 0.20}$	3.4 ± 0.21
Has Negatives	3.8 ± 0.23	3.8 ± 0.23	$\mathbf{3.5 \pm 0.22}$	3.6 ± 0.22
Has Fill-in-the-Blank	3.2 ± 0.21	$\mathbf{3.1 \pm 0.21}$	3.3 ± 0.21	3.3 ± 0.21
Has None of the Above	$\mathbf{0.1 \pm 0.04}$	0.2 ± 0.05	0.7 ± 0.10	$\mathbf{0.6 \pm 0.09}$
Has All of the Above	0.1 ± 0.04	0.1 ± 0.03	0.3 ± 0.06	$\mathbf{0.2 \pm 0.05}$

Table 8: Prevalence (%) of all 19 writing flaws in human-guided and LLM-only distractors across distractor generation and extension tasks. Values are BenchMarker rule failure rates, so lower is better.

Dataset	0 matches	1 match	2 matches	3 matches
ARC	63.4%	21.1%	11.1%	4.5%
MMLU	74.8%	16.1%	6.8%	2.2%
GPQA	62.5%	18.7%	9.6%	9.2%

Table 9: Distribution of item-level distractor overlap, averaged over generator models. Most generated items share no distractors with the original human-written item.

Dataset	Model	SOLVE	SIMULATE	RECALL	SHALLOW	REFINE
GPQA	GPT	25%	70%	25%	35%	15%
	Gemini	85%	75%	40%	25%	65%
	Qwen	90%	90%	50%	25%	100%
ARC	GPT	10%	80%	25%	0%	5%
	Gemini	10%	80%	25%	5%	90%
	Qwen	15%	80%	30%	5%	100%
MMLU	GPT	10%	95%	70%	5%	0%
	Gemini	75%	95%	80%	5%	90%
	Qwen	75%	95%	80%	5%	100%

Table 10: Reasoning-trace strategy use in D-GEN (% of 20 MCQs per dataset–LLM), split by dataset.

Question QA.1: MMLU-Pro:996 (Text Corruption)

Question: A homeowner resides downhill from a metal fabrication facility. She has sued both the owner of the facility and the supplier of a solvent used at the facility. She contends that contaminants, consisting mostly of the solvent, were released into the ground at the facility and have migrated and continue to migrate to her property, contaminating the soil, the groundwater, and her well. She alleges various acts of negligence on the part of the facility owner in causing the release of the contaminants into the ground. She also alleges that employees of the solvent supplier were negligent in frequently the solvent supplier were negligent in frequently spilling some of the solvent onto the ground while filling a rooftop tank at the facility. The solvent supplier has moved for summary judgment, arguing that if there was any contamination, the facility owner and the supplier independently contributed indeterminate amounts to the contamination and that therefore the homeowner cannot show how much damage each has inflicted on her. There is no evidence that the facility owner and the solvent supplier acted in concert. Should the court grant the summary judgment motion?

Choices:
A. No, because concurrent tortfeasors are jointly and severally liable for an indivisible injury.
B. No, because the solvent supplier is vicariously liable for damage inflicted by the facility owner.
C. Yes, because there is no basis for allocating damages against the solvent supplier. damages against the solvent supplier.
D. Yes, because there is no evidence that the facility owner and the solvent supplier acted in concert.

Correct Answer: A

Question QA.2: GPQA:recDjE01bu72pPUU2 (Duplicate correct answer among distractors)

Question:
When 500 mL of PH3 is decomposed the total volume of the reaction mixture becomes 600 mL only. The H2 obtained in the above reaction is used to create electricity in a fuel cell. Calculate the volume of unreacted H2 in the fuel cell when only 50 mL of O2 is used.
Choices:
A. 200 mL
B. 100 mL
C. 180 mL
D. 200 mL
Correct Answer: A

Question QA.3: GPQA:recZSGUkn56v9kEp1 (Duplicate distractor option)

Question:
Determine the product obtained when 3,4-dimethylhexanedial is reacted with the following set of reagents:
1. KOH, H2O, THF, Heat
2. CH3CH2MgBr, H3O+
3. PCC, CH2Cl2
4. O3, H2O
Choices:
A. 4,5-dimethylnonane-2,6,7-trione
B. 4,5-dimethylnonane-2,6,7-trione
C. 3,4-dimethyl-5,6-dioxooctanal
D. 3,4-dimethyl-5,6-dioxooctanoic acid
Correct Answer: D

Question QA.4: GPQA:recnGEpF1srQpaqWq (Duplicate correct answer among distractors)

Question:
Which of the following molecules is formed when 5-butylnona-2,6-diene is heated?
Choices:
A. 5-ethylundeca-2,6-diene
B. 5-ethyl-4-methyldeca-2,6-diene
C. 4-ethyl-3-methyldeca-1,5-diene
D. 5-ethyl-4-methyldeca-2,6-diene
Correct Answer: C

Question QA.5: GPQA:recnGEpF1srQpaqWq (Impractically long DNA sequence)

Question:
Congratulations, you just landed your dream job at one of the top structural biology labs at Stanford University in California. Without further ado, the Principal Investigator explains that the priority of the moment is to produce in E. coli BL21 then purify as quickly as possible the Human P53 protein.
The amino acid sequence of this protein is as follows:
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLPSQAMDDLMLSPDDIEQWFTEDPGPDEA
PRMPEAAPPVAPAPAAPTPAAPAPAPSWPLSSSVPSQKTYQGSYGFRLGFLHSGTAKSVTCTY
SPALNKMFCQLAKTCPVQLWVDSTPPPGTRVRAMAIYKQSQHMTEVVRRCPHHERCSDSDGLA
PPQHLIRVEGNLRVEYLDDRNTFRHSVVVPYEPPEVGSDCTTIHYNYMCNSSCMGGMNRRPIL
TIITLEDSSGNLLGRNSFEVRVCACPGRDRRTEEENLRKKGEPHHELPPGSTKRALPNNTSSS
PQPKKKPLDGEYFTLQIRGRERFEMFRELNEALELKDAQAGKEPGGSRAHSSHLKSKKGQSTS
RHKKLMFKTEGPDSD

You are given 4 plasmids containing 4 different sequences, which one are you going to use?
Choices: (cut off for brevity)
A. ATGGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCGCCTCTGAGTCAGGA…
B. ATGGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCGCCTCTGAGTCAGGA…
C. ATGGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCCCCTCTGAGTCAGGA…
D. ATGGAGGAGCCGCAGTCAGATCCTAGCGTCGAGCCGCCTCTGAGTCAGGA…
Correct Answer: C

Prompt A.1: Evaluation

Answer the following multiple choice question.

<question>
question
</question>

<choices>
choices
</choices>

<format>
Return your output as valid JSON with the key “answer” which is one of letters.
json_example
Do not include anything else.
</format>

Prompt A.2: Generate given QA pair (Generation Setting)

You are an expert writer of plausible but incorrect distractors for multiple-choice questions (MCQs).

Given a question stem and an answer to this question, your job is to generate exactly count plausible but incorrect answer choices that could be used as distractors to form a full multiple-choice question.

Here is the question stem and the correct answer:
<multiple-choice question>
Question: question
Answer: gold_answer
</multiple-choice question>

<distractor instructions>
- Each distractor option must be a plausible answer to this question that tests common misconceptions, but ultimately should not be more correct than the provided answer
- Each of the distractor options must be distinct; they cannot be synonyms of each other
</distractor instructions>

<format>
Return your output as valid JSON with exactly one key: "distractors" - a list of exactly count incorrect answer choices
json_example
Do not include anything else.
</format>

Prompt A.3: Generate given MCQA (Extension Setting)

You are an expert writer of plausible but incorrect distractors for multiple-choice questions (MCQs).

Given a multiple-choice question stem, old_count existing choices for this question, and an answer to this question, your job is to generate exactly count additional plausible but incorrect answer choices to form an expanded multiple-choice question.

Here is the question stem, the choices, and the correct answer:
<multiple-choice question>
Question: question
Choices:
choices
Answer: gold_answer
</multiple-choice question>

<distractor instructions>
- Each distractor option must be a plausible answer to this question that tests common misconceptions, but ultimately should not be more correct than the provided answer
- Each of the distractor options must be distinct; they cannot be synonyms of each other
- None of the generated distractor choices should be synonyms with the provided answer choices above.
</distractor instructions>

<format>
Return your output as valid JSON with exactly one key: "distractors" - a list of exactly count incorrect answer choices
json_example
Do not include anything else.
</format>

Trace A.1: Question (gpqa:rec2xN0SZSiyfKb4E)

Analyzing AGN Torus Radius Calculations and Potential Student Errors
Okay, so we're looking at calculating the inner radius of an AGN torus using the Barvainis model, given a dust sublimation temperature of 900 K, a bolometric luminosity of 1.5×10^{46} erg/s, and a UV bolometric correction of 5. The correct answer, we're told, is 3 pc. Let's break this down systematically.
The Barvainis model is the key here; it relates the dust sublimation radius to the UV luminosity. The formula is $R \approx 1.3 \left(\frac{L_{UV}}{10^{46}\ \text{erg/s}}\right)^{0.5} \left(\frac{T}{1500\ \text{K}}\right)^{-2.8}$ pc. First, we need to calculate the UV luminosity from the bolometric luminosity: $L_{UV} = L_{\text{bol}}/BC = (1.5 \times 10^{46})/5 = 0.3 \times 10^{46} = 3 \times 10^{45}$ erg/s. Let's make sure we are clear about the parameters. Then we plug that into the formula.
Substituting the values into the Barvainis formula, we have $L_{UV,46} = 0.3$, $T = 900$ K, so $T_{1500} = 900/1500 = 0.6$. The radius becomes $R \approx 1.3 \times \sqrt{0.3} \times (0.6)^{-2.8}$. Let's calculate this step by step. $\sqrt{0.3} \approx 0.5477$, and $0.6^{-2.8} \approx 4.18$. Multiplying everything together, we get $R \approx 1.3 \times 0.5477 \times 4.18 \approx 2.97$ pc. That is almost exactly 3 pc.
Now, let's play "what if" and consider what common errors a student might make. If they forget to divide by the bolometric correction (a likely scenario), they'd use the bolometric luminosity directly. In that case, $L = 1.5 \times 10^{46}$ erg/s, and $R = 1.3 \times \sqrt{1.5} \times (0.6)^{-2.8} \approx 6.66$, or around 6.7 pc.
If the student uses the bolometric correction backwards and multiplies, we get $L = 1.5 \times 10^{46} \times 5 = 7.5 \times 10^{46}$. The radius would become $R = 1.3 \times \sqrt{7.5} \times (0.6)^{-2.8} \approx 14.89$ pc, or about 15 pc.
If they forget the temperature dependence entirely, or assume a standard 1500K temperature, the calculation simplifies to $R = 1.3 \times \sqrt{0.3} \approx 0.71$, which is about 0.7 pc.
We can consider the Stefan-Boltzmann law and its T^{-2} relation, instead of the Barvainis $T^{-2.8}$. If this happens, we get $R = 1.3 \times \sqrt{0.3} \times (0.6)^{-2} = 1.3 \times \sqrt{0.3} \times 2.77$. Thus $R \approx 1.97$, which is about 2 pc.
Finally, what are plausible wrong answers (distractors)? We have 6.7 pc (no bolometric correction), 15 pc (multiplying by the BC), 0.7 pc (no temperature dependence), and we've already done 1.6 pc (forgetting the square root) and 2 pc. The distractors should also be formatted the same. I will go with "6.7 pc", "15 pc", and "0.7 pc".