Check The Scoreboard: An Analysis of Scoring Schemes on Multiple-Choice Evaluation

Nishant Balepur[1,2] **Paiheng Xu**[1]
Wei Ai[1] **Eunsol Choi**[2] **Rachel Rudinger**[1] **Jordan Boyd-Graber**[3]*

[1]University of Maryland [2]New York University [3]Nanyang Technological University

nbalepur@umd.edu jordan.ying@ntu.edu.sg

Abstract

Multiple-choice question answering (MCQA) benchmarks in NLP use number-right scoring (accuracy), but in educational testing, the scoring scheme—the combination of the response mode models follow and the rule for grading responses—is a key design choice that dictates which abilities to reward. We examine how alternatives to number right change what MCQA measures with six education-inspired schemes that assess abilities beyond accuracy: distractor elimination, abstention, confidence calibration, and self-correction. On LLM benchmarks, these schemes: 1) shift rankings of 31 LLMs beyond rephrased number right prompts; 2) better predict the LLMs users prefer in LLM Arena; and 3) reveal distinct model capabilities, like that GPT-5 rarely abstains and readily self-corrects, while weaker open-weight models often abstain and hesitate to eliminate choices. Given the benefits of alternative scoring schemes, we discuss ways to extend them to tasks beyond MCQA.[1]

1 Introduction: Keeping Score in MCQA

In multiple-choice question answering (Clark et al., 2020, MCQA) benchmarks, NLP researchers carefully study question and inference design (Gu et al., 2025), but stick to the simplest **scoring scheme** of *number right scoring*: models receive one point for each correct answer (i.e., accuracy). Despite long-standing concerns that this rewards guessing (Du et al., 2023) and poorly reflects abilities users value (Saxon et al., 2024), it persists due to its simplicity.

Conversely, education research offers alternative schemes that punish guessing, elicit confidence, or add re-attempts (Lau et al., 2011). These schemes surface student abilities that number right scoring cannot distinguish (Kanzow et al., 2023), but add costs like test anxiety that limit adoption (Ndu et al., 2016). However, these schemes test abilities LLMs lack—confidence, abstention, and self-correction—and LLMs do not experience test anxiety (Shanahan, 2024), making NLP well-positioned to adapt them.

Our paper presents six MCQA scoring schemes for LLM evaluation that use alternative prompt instructions and metrics (Table 1). Grounded in education research, our schemes assess how well LLMs *abstain* when uncertain to avoid penalties, use *partial knowledge* to eliminate distractors, express calibrated *confidence* in predictions, and *self-correct* after being informed they are wrong—distinct abilities that MCQA neglects with number right scoring.

Across 31 LLMs and three MCQA datasets (§2), many schemes shift LLM ranks more than rephrasing instructions in our number right prompt, testing unique abilities (§3.1). Answer until correct scoring best predicts the LLMs that users prefer in LLM Arena (Zheng et al., 2024), helping MCQA better reflect what users value (§3.2). Our schemes also reveal LLM behaviors: larger LLMs with test-time reasoning are more consistent over schemes (§3.3), strong GPT-5 models rarely abstain and are quick to change answers, Command-R+ often abstains, and LLaMA-3.1 hesitates to eliminate incorrect choices (§3.4)—quirks that number right fails to surface.

These scoring schemes require only prompt and metric changes, making them simple to incorporate into existing MCQA benchmarks. We more broadly argue for the importance of scoring scheme design in benchmarks and conclude with ways to extend them beyond MCQA (§5). Our contributions are:
1) Six education-grounded scoring schemes, evaluated across three MCQA benchmarks and 31 LLMs.
2) Evidence that scoring schemes test diverse abilities and better predict model ranks in LLM Arena.
3) Cross-scheme analyses of consistency, measurements, confidence, and self-correction behaviors.

2 Experimental Setup: Scoring Schemes

Drawing on Koele et al. (1987), a **scoring scheme** combines a *response mode* (how test-takers must

[1]Our code: https://github.com/nbalepur/mcqa-scoring/
[1]Work done at University of Maryland

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 246–276
October 24-29, 2026 ©2026 Association for Computational Linguistics

Name	Abilities Tested	Description
Number Right (Kelly, 1916, NR)	Answer Accuracy	The standard method where the model picks a best answer. +1 point if correct and 0 if incorrect.
Control Variation (NR-VAR)	Prompt Sensitivity	To set a baseline prompt sensitivity when testing different schemes, we rephrase the prompt of NR.
Negative Marking (Holt, 2006, NM)	Uncertainty, Abstention	The model picks one best answer or abstains. To penalize guessing, scoring awards +1 point for right answers, $-1/(n-1)$ for wrong ones, and 0 for abstaining, so the expected value of guessing is 0.
Elimination Scoring (Coombs et al., 1956, ELIM)	Partial Knowledge, Eliminative Reasoning	The model picks as many answers it thinks are wrong. For granular partial knowledge, the score is the proportion of all distractors eliminated, or 0 points if the correct answer is eliminated.
Elimination + NM (Coombs et al., 1956, ELIM-NM)	Partial Know., Elim. Reasoning, Abstention	A scoring scheme that combines the benefits and abilities tested in ELIM and NM: models receive $1/(n-1)$ points for eliminating wrong answers and -1 for correct ones.
Confidence-Based Marking (Gardner-Medwin, 2006, CBM)	Discrete Confidence Calibration	The model predicts one answer choice and a confidence level ($C = 1$ for low, $C = 2$ for medium, or $C = 3$ for high). To reward calibration, predicting $C = 1$ yields $+1$ if correct and 0 if wrong, $C = 2$ yields $+2$ if correct and -2 if wrong, and $C = 3$ yields $+3$ if correct and -6 if wrong.
Personal Point Allocation (MacNeil et al., 2023, PPA)	Probabilistic Conf. Calibration	The model outputs a probability distribution for each choice being correct. To reward confidently right answers, we use Brier score between the model's distribution and the true distribution (one-hot).
Answer Until Correct (Wilcox, 1982, AUC)	Self-Correction, Adaptability	Over a fixed number of tries, the model answers until correct—told when incorrect. To reward fewer attempts, the score is $(r-1)/(n-1)$, where r is # choices left when the correct answer is picked.

Table 1: The MCQA scoring schemes from education we adopt and abilities they test. Appendix A.7 has examples.

respond to inputs) and a *scoring rule* (how to score responses). Practically in LLM evaluation, prompts define response modes (e.g., "*Select the correct answer*") and metrics (e.g., accuracy) dictate scoring.

MCQA's default scoring scheme is number right (Kelly, 1916, NR): given a question q and n choices $\mathcal{C}$, test-takers predict one answer $\hat{a} \in \mathcal{C}$—the *response mode*. The *scoring rule* assigns one point if $\hat{a}$ matches the gold answer ($\mathbb{1}(a = \hat{a})$). Despite its popularity, NR cannot distinguish lucky guesses from true understanding and provides no credit for test-takers' partial knowledge (Lau et al., 2011), which also applies to NLP models (Du et al., 2023).

To see whether alternative scoring schemes can bolster LLM evaluation, we study MCQA schemes inspired by education (Table 1): negative marking (NM) to penalize guessing (Frary, 1988), answer elimination (ELIM, ELIM-NM) for partial credit based on the number of ruled-out choices (Coombs et al., 1956), confidence elicitation (CBM, PPA) to reward calibrated certainty (de Finetti, 1965), and answering until correct (AUC) to evaluate how test-takers adapt to feedback (Wilcox, 1982). These test desired abilities of LLMs that NR cannot capture (Liang et al., 2023)—abstention (Elhady et al., 2025), eliminative reasoning (Ma and Du, 2023), calibration (Guo et al., 2017), and self-correction (Huang et al., 2024)—valuable additions to MCQA.

We implement each scheme via custom prompts outlining response modes, and metrics as scoring rules. To illustrate, for NM, we adapt NR's prompt to let LLMs abstain as part of the response mode, and the scoring rule to add penalties for wrong answers (Table 1, row 3). We repeat this process for each row in Table 1—prompting LLMs to return wrong choices in a list for ELIM/ELIM-NM, predictions with confidence scores in CBM/PPA, and add past predictions as inputs for AUC—specified in Appendix A.7. All prompts use MCQA templates drawn from the InspectAI library (UK AISI, 2024).

2.1 Datasets

We sample 1000 MCQs from three popular benchmarks of varied difficulty: **1) ARC**–grade-school scientific knowledge/commonsense (Clark et al., 2018); **2) MMLU**—knowledge of 57 college topics (Hendrycks et al., 2021); and **3) SuperGPQA**—knowledge of 285 graduate topics (Du et al., 2025).

2.2 Models

We assess 31 frontier LLMs: 1) *Gemini* (Comanici et al., 2025, 2/3 Flash, 2.5 Lite, 2.5/3 Pro); 2) *GPT* (Singh et al., 2025, 4.1, 5 Nano, 5 Mini, 5, 5.2); 3) *Claude* (Anthropic, 2025, 3.7/4/4.5 Sonnet, 4.5 Haiku); 4) *Cohere* Command-R (Gomez, 2024, R, R+, R7B); 5) *DeepSeek* (Guo et al., 2025, V3.1, R1); 6) *Qwen*-3 (Yang et al., 2025, 80B Inst/Think, 235B Inst/Think); 7) *LLaMA*-3.1 (Dubey et al., 2024, 8B, 70B, 405B); 8) *GPT OSS* (Agarwal et al., 2025, 20B, 120B); 9) *GLM* (Zeng et al., 2025, 4.5, 4.6); and 10) *Kimi*-K2 (Team et al., 2025). GPT-OSS and Cmd-R+ take a week on SuperGPQA, so we omit them. We use default model parameters.

3 Results: Analyzing Scoring Schemes

Having designed education-based MCQA schemes (§2), we now reveal their benefits for NLP: they reward new abilities (§3.1), predict user preferences (§3.2), and surface model behaviors (§3.3, §3.4).

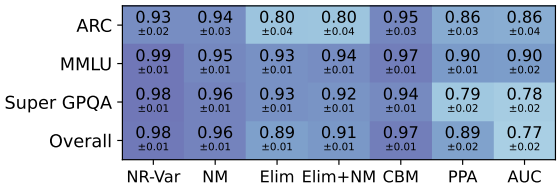

Figure 1: Spearman correlations between number right (NR) scoring with bootstrapped standard error ($n = 1000$). "Overall" micro-averages scores across all datasets. All shift ranks beyond the prompt sensitivity control (NR-VAR)—especially ELIM ELIM-NM, PPA and AUC—so they test diverse model abilities. Appendix A.3 has correlations across all schemes.

3.1 Different Schemes, Different Ranks

While our alternative scoring schemes differ from NR by design, we do not need them if they measure the same model abilities as NR, as their adoption would never impact downstream evaluation (Cronbach and Meehl, 1955). To probe this, we measure the Spearman's correlation between LLM rankings under each scheme versus NR. Low correlation indicates discriminant validity (Campbell and Fiske, 1959): the scheme evaluates abilities NR does not.

NM and CBM have high rank correlation with NR (Fig 1), but ELIM, ELIM-NM, PPA, and AUC consistently show lower correlation than the NR-VAR control, testing skills NR does not capture beyond rephrased instructions. This trend also holds on ARC when shuffling choices and replacing letters with numbers (e.g., (A) → (1)) as sensitivity baselines in Appendix Table 2. There is not a best scheme, as this depends on what researchers value: ELIM for eliminative reasoning, PPA for calibration, and AUC for self-correction (Table 1). NR cannot target such skills, which have downstream use: medical chatbots apply ELIM in diagnosis by exclusion (Fred, 2013), while AI tutors apply AUC when adapting to student errors (Chen et al., 2024).

For researchers who prefer the traditional NR scheme, applying many schemes can still break ties: ARC is a simpler dataset where LLMs score closely with NR (Appendix A.3), but ELIM-NM and ELIM have lower correlation with NR. Thus, they can better separate LLM abilities as tie-breaking decisions.

3.2 AUC Agrees with Human Preferences

While user feedback is the north-star goal for LLM development (Ouyang et al., 2022), researchers often rely on cheaper offline proxies such as MCQA to guide training (Olmo et al., 2025). To test whether

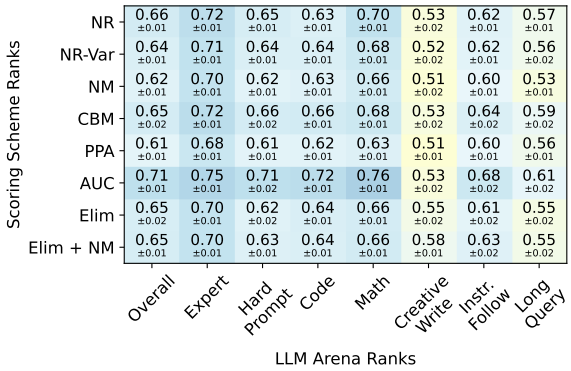

Figure 2: Spearman's correlation between LLM ranks from scoring schemes (dataset micro-average) versus user preferences of varied query types in LLM Arena. Answering until correct (AUC) scoring often best predicts model ranks in LLM Arena. Appendix A.4 decomposes results across each dataset.

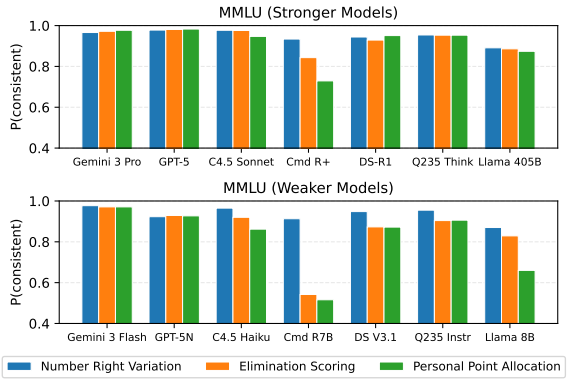

Figure 3: Logical consistency of scoring schemes with NR on MMLU. Weaker models (smaller, no reasoning) tend to be less consistent over schemes. Appendix A.5 has all datasets.

alternative scoring schemes improve MCQA as an online proxy, we compare LLM ranks under each scheme to LLM Arena ranks—a large-scale benchmark of user preferences via live voting that developers actively optimize for (Spangher et al., 2025).

AUC best agrees with LLM Arena (Figure 2)—more than NR—so AUC can better predict LLMs users prefer pre-deployment. This matches Shi et al. (2024) and Liu et al. (2025) who show users prefer LLMs that adapt well to feedback, which AUC captures by making LLMs self-correct; this trend holds in a scheme where LLMs always self-reflect (Appendix A.4). As expected, the gap between NR's and AUC's correlation with LLM Arena is wider on technical queries (e.g., Math, Expert) versus those unrelated to MCQA (e.g., Creative Writing).

3.3 Capturing Cross-Scheme Contradictions

We now show alternative scoring schemes' value in tracking logical consistency (Hase et al., 2023): a model should not contradict its belief of the answer

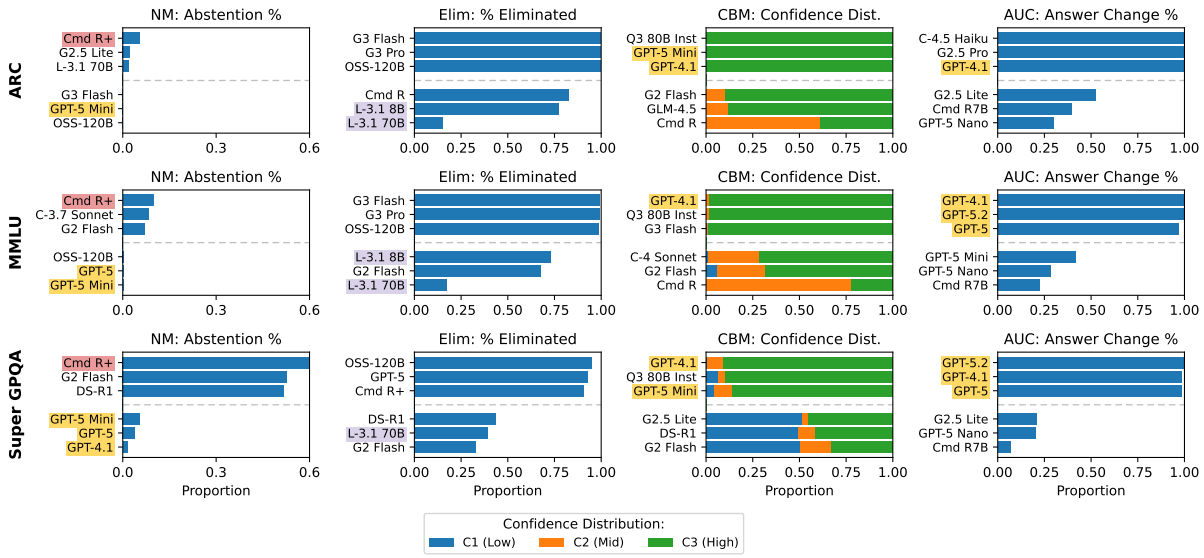

Figure 4: Analysis of abstention rates, option elimination rate, stated confidence, and answer change rates, surfaced via NM, ELIM, CBM, and AUC scoring schemes, respectively. We display the three highest and lowest scoring models under each facet for brevity. These schemes reveal model-specific behaviors: stronger GPT models are confident, rarely abstain, and quickly change answers, Command-R+ often abstains, and LLaMA-3.1 hesitates to eliminate. Appendix A.6 has all models and trends.

across schemes (Chern et al., 2024). We test consistency in two ways: 1) the answer a model picks in NR must not be eliminated in ELIM; and 2) NR's answer must have the most weight in PPA (or tied).

Figure 3 shows that for LLMs of the same family, scaling parameters or adding test-time reasoning improves consistency, aligning with prior work (Truong et al., 2023). Thus, running diverse scoring schemes for the same task like MCQA is a simple way to evaluate which interventions (e.g., scaling size or inference) can improve model consistency.

3.4 Scoring Schemes Reveal Model Behaviors

While NR mainly provides just one accuracy score to study models, alternative schemes also give finer-grained signals into model behavior, like abstention rates, confidence, and adaptability (Figure 4). Analyzing this yields interesting trends: 1) GPT models rarely abstain and state high confidence; 2) weak GPT models struggle to change answers after being told they are incorrect, while stronger GPT models readily do so; 3) weaker open-weight models like Command-R+ and Llama often abstain or hesitate to eliminate choices; and 4) All LLMs state high confidence in CBM and rarely abstain in NM on easier datasets like ARC and MMLU, but not Super GPQA, showing some awareness of MCQ difficulty (Guo et al., 2017). Only evaluating MCQA via NR scoring obscures these features of LLM behavior.

4 Related Work

Scoring Schemes: MCQA exams originally tested students via number right scoring (Kelly, 1916), but educators found limitations: it encourages students to guess when unsure (Guilford, 1936) and cannot discern partial knowledge if students answer identically (Lau et al., 2011). Educators have thus designed alternatives schemes to improve abilities MCQA measures (Cronbach and Meehl, 1955), so we extend them and assess their benefits in NLP.

MCQA Evaluation: MCQA is a keystone of NLP evaluation (Reddy, 1988) as it is simple and mirrors student testing (Clark et al., 2020). Given this dominance (Liang et al., 2023), work improves the task via open-ended formats (Myrzakhan et al., 2024; Chandak et al., 2025), shortcut reduction (Balepur and Rudinger, 2024; Balepur et al., 2026a), and quality control (Moore et al., 2024; Palta et al., 2024; Balepur et al., 2026c). Some of our MCQA schemes have been studied individually: Ma and Du (2023) and Balepur et al. (2024a) test process of elimination, Elhady et al. (2025) add an "I don't know" option for abstention, and Huang et al. (2024) assess self-correction. We instead compare schemes from education to study rank shifts, preference agreement, and behaviors. Work also studies how scoring can better adapt training and evaluation curricula (Hofmann et al., 2025; Meng et al., 2026, e.g., Item Response Theory), but our scoring schemes focus on evaluating individual examples.

5 Conclusion: Scheming Beyond MCQA

NLP often critiques MCQA since models can guess (Du et al., 2023; Balepur et al., 2024b) and the task poorly reflects downstream user needs (Saxon et al., 2024; Balepur et al., 2025), but it persists due to its simplicity. Educators face similar issues when testing students, so they design alternative scoring schemes. NLP can take the same path: easy fixes to prompts and metrics can curb guessing by design, reveal distinct model behaviors, and better predict user preferences, all without changing task inputs.

More broadly, we argue for human assessments to inform LLM scoring scheme design. In MCQA, we use educational testing, which also applies to programming (White, 1993) and writing (Tew and Guzdial, 2010) which have well-researched scoring rubrics and assignment creation protocols for students (Roumani, 2002). For NLP tasks sans human analogues (Bragg et al., 2025; Balepur et al., 2026b, e.g., deep research), a promising path is to study how humans finish them and which behaviors predict success. Lastly, the benefits of AUC's multi-turn setting with simulated user feedback suggests that more classic NLP tasks (e.g., natural language inference, summarization) could be run as multi-turn dialogues, forming user-inspired evaluations (Pan et al., 2025; Laban et al., 2026). Overall, researchers must treat scoring schemes as a key part of benchmark construction, just like educators do.

6 Limitations

LLMs are sensitive to prompting formats, especially in MCQA (Pezeshkpour and Hruschka, 2024), so altering prompts could shift model ranks further (Alzahrani et al., 2024). To design prompts fairly, we build off the MCQA prompts used in InspectAI (UK AISI, 2024): a standard evaluation framework from the United Kingdom AI Security Institute. This also applies to our analysis with LLM Arena (Zheng et al., 2023), as the platform does not disclose the prompts or inference parameters used, so different configurations could alter our results. Prompt sensitivity is another interesting aspect of LLM evaluation that future work can explore when considering alternative scoring schemes.

Second, while Answer Until Correct (AUC) is the scoring strategy with the highest agreement with LLM Arena, it is the most expensive, as it requires many turns (cost in Table 3). While still cheaper than crowdsourcing, future work could test alternate ways to reap the benefits of AUC with less attempts: in Appendix A.4, we show prompting LLMs to self-refine their answers just once improves agreement with LLM Arena. However, our goal is not to argue for a particular MCQA scoring scheme, but to motivate the benefits of evaluating with different scoring schemes, regardless of task.

7 Ethical Considerations

Our goal of testing different multiple-choice scoring regimes is to improve evaluation validity, but we never intend to argue that such evaluations are sufficient as pre-deployment checks. Models that excel in our schemes may still be prone to biases, hallucinations, and producing harmful outputs.

Generative AI (GenAI) was used in this project. We used Cursor[2] to design plots, refactor code, and debug errors, and we used ChatGPT to refine paper writing for brevity. AI tools did not directly write any parts of this paper. We take full responsibility for any GenAI errors. By discussing AI usage here, we encourage other researchers to do the same.

Acknowledgments

We would like to thank the CLIP lab at the University of Maryland for their support. In particular, we thank Yapei Chang, Jane Oh, and Ankit Aggarwal for discussions on earlier versions of this paper draft. This material is based upon work supported by the National Science Foundation under IIS-2339746 (Rudinger) IIS-2403436 (Boyd-Graber), and DGE-2236417 (Balepur). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Access to Cohere models (Command-R, Command-R Plus) was made possible via a Cohere for AI Research Grant.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

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

[2]https://cursor.com/agents

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.

Anthropic. 2025. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5. Accessed: 2025-11-29.

Nishant Balepur, Atrey Desai, and Rachel Rudinger. 2026a. Test-time reasoners are strategic multiple-choice test-takers. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 250–272, San Diego, California, United States. Association for Computational Linguistics.

Nishant Balepur, Malachi Hamada, Varsha Kishore, Sergey Feldman, Amanpreet Singh, Pao Siangliulue, Joseph Chee Chang, Eunsol Choi, Jordan Lee Boyd-Graber, and Aakanksha Naik. 2026b. Language models don't know what you want: Evaluating personalization in deep research needs real users. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15910–15944, San Diego, California, United States. Association for Computational Linguistics.

Nishant Balepur, Shramay Palta, and Rachel Rudinger. 2024a. It's not easy being wrong: Large language models struggle with process of elimination reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10143–10166, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Balepur, Bhavya Rajasekaran, Hyunjin Jane Oh, Michael Xie, Atrey Desai, Vipul Gupta, Steven James Moore, Eunsol Choi, Rachel Rudinger, and Jordan Lee Boyd-Graber. 2026c. BenchMarker: An education-inspired toolkit for highlighting flaws in multiple-choice benchmarks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15793–15824, San Diego, California, United States. Association for Computational Linguistics.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024b. Artifacts or abduction: How do llms answer multiple-choice questions without the question? In *Annual Meeting of the Association for Computational Linguistics*.

Nishant Balepur and Rachel Rudinger. 2024. Is your large language model knowledgeable or a choices-only cheater? In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 15–26, Bangkok, Thailand. Association for Computational Linguistics.

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.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D Hwang, Peter Jansen, Varsha Kishore, et al. 2025. Astabench: Rigorous benchmarking of ai agents with a scientific research suite. *arXiv preprint arXiv:2510.21652*.

Donald T Campbell and Donald W Fiske. 1959. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological bulletin*, 56(2):81.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. 2025. Eliminating discriminative shortcuts in multiple choice evaluations with answer matching. In *ICML 2025 Workshop on Assessing World Models*.

Jiahao Chen, Zitao Liu, Mingliang Hou, Xiangyu Zhao, and Weiqi Luo. 2024. Multi-turn classroom dialogue dataset: Assessing student performance from one-on-one conversations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5333–5337.

Steffi Chern, Zhulin Hu, Yuqing Yang, Ethan Chern, Yuan Guo, Jiahe Jin, Binjie Wang, and Pengfei Liu. 2024. Behonest: Benchmarking honesty in large language models. *arXiv preprint arXiv:2406.13261*.

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, et al. 2020. From 'f'to 'a'on the ny regents science exams: An overview of the aristo project. *Ai Magazine*, 41(4):39–53.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Clyde H Coombs, John Edgar Milholland, and Frank Burton Womer. 1956. The assessment of partial knowledge. *Educational and Psychological Measurement*, 16(1):13–37.

Lee J Cronbach and Paul E Meehl. 1955. Construct validity in psychological tests. *Psychological bulletin*, 52(4):281.

Bruno de Finetti. 1965. Methods for discriminating levels of partial knowledge concerning a test item. *British Journal of Mathematical and Statistical Psychology*, 18(1):87–123.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Commun. ACM*, 67(1):110–120.

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, Yizhi LI, Yunwen Li, dehua ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, yuelin bai, Xingyuan Bu, chenglin cai, Liang Chen, Yifan Chen, Cheng Chengtuo, Tianhao Cheng, Keyi Ding, Siming Huang, Huang Yun, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Z.Y. Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Owen Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, ChenghuaZhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. 2025. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Ahmed Elhady, Eneko Agirre, and Mikel Artetxe. 2025. WiCkeD: A simple method to make multiple choice benchmarks more challenging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1183–1192, Vienna, Austria. Association for Computational Linguistics.

Robert B. Frary. 1988. Formula scoring of multiple-choice tests (correction for guessing). *Educational Measurement: Issues and Practice*, 7:33–38.

Herbert L Fred. 2013. The diagnosis of exclusion: an ongoing uncertainty. *Texas Heart Institute Journal*, 40(4):379.

AR Gardner-Medwin. 2006. Confidence-based marking: towards deeper learning and better exams. In *Innovative assessment in higher education*, pages 161–169. Routledge.

Aidan Gomez. 2024. Command R: Retrieval-augmented generation at production scale. https://cohere.com/blog/command-r. Accessed: 2025-11-29.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. 2025. OLMES: A standard for language model evaluations. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5005–5033, Albuquerque, New Mexico. Association for Computational Linguistics.

JP Guilford. 1936. The determination of item difficulty when chance success is a factor. *Psychometrika*, 1(4):259–264.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023. Methods for measuring, updating, and visualizing factual beliefs in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2714–2731, Dubrovnik, Croatia. Association for Computational Linguistics.

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*.

Valentin Hofmann, David Heineman, Ian Magnusson, Kyle Lo, Jesse Dodge, Maarten Sap, Pang Wei Koh, Chun Wang, Hannaneh Hajishirzi, and Noah A. Smith. 2025. Fluid language model benchmarking. In *Second Conference on Language Modeling*.

Alan Holt. 2006. An analysis of negative marking in multiple-choice assessment. In *19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006)*, pages 115–118.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Amelie Friederike Kanzow, Dennis Schmidt, and Philipp Kanzow. 2023. Scoring single-response multiple-choice items: scoping review and comparison of different scoring methods. *JMIR Medical Education*, 9:e44084.

Frederick James Kelly. 1916. The kansas silent reading tests. *Journal of Educational Psychology*, 7(2):63.

Pieter Koele, Robert De Boo, and Paul Verschure. 1987. Scoring rules and probability testing. *Bulletin of the Psychonomic Society*, 25(4):280–282.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2026. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*.

Paul Ngee Kiong Lau, Sie Hoe Lau, Kian Sam Hong, and Hasbee Usop. 2011. Guessing, partial knowledge, and misconceptions in multiple-choice tests. *Journal of Educational Technology & Society*, 14(4):99–110.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.

Yuhan Liu, Michael JQ Zhang, and Eunsol Choi. 2025. User feedback in human-LLM dialogues: A lens to understand users but noisy as a learning signal. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2666–2681, Suzhou, China. Association for Computational Linguistics.

Chenkai Ma and Xinya Du. 2023. POE: Process of elimination for multiple choice reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4487–4496, Singapore. Association for Computational Linguistics.

Stephen MacNeil, Jonah Cooper, Eileen Wood, Michael Williams, and Fatma Arslantas. 2023. Examining students' performance with two partial-marks multiple-choice testing formats: Considering confidence and risk-taking. *Journal of Assessment and Institutional Effectiveness*, 13(1-2):145–171.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Guangyu Meng, Qingkai Zeng, John P. Lalor, and Hong Yu. 2026. A psychology-based unified dynamic framework for curriculum learning. *Computational Linguistics*, pages 1–49.

Steven Moore, Eamon Costello, Huy A Nguyen, and John Stamper. 2024. An automatic question usability evaluation toolkit. In *International Conference on Artificial Intelligence in Education*, pages 31–46. Springer.

Aidar Myrzakhan, Sondos Mahmoud Bsharat, and Zhiqiang Shen. 2024. Open-llm-leaderboard: From multi-choice to open-style questions for llms evaluation, benchmark, and arena. *arXiv preprint arXiv:2406.07545*.

Ikenna Kingsley Ndu, Uchenna Ekwochi, Chidiebere Di Osuorah, Isaac Nwabueze Asinobi, Michael Osita Nwaneri, Samuel Nkachukwu Uwaezuoke, Ogechukwu Franscesca Amadi, Ifeyinwa Bernadette Okeke, Josephat Maduabuchi Chinawa, and Casmir James Ginikanwa Orjioke. 2016. Negative marking and the student physician—a descriptive study of nigerian medical schools. *Journal of Medical Education and Curricular Development*, 3:JMECD–S40705.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. 2025. Olmo 3. *arXiv preprint arXiv:2512.13961*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Shramay Palta, Nishant Balepur, Peter Rankel, Sarah Wiegreffe, Marine Carpuat, and Rachel Rudinger. 2024. Plausibly problematic questions in multiple-choice benchmarks for commonsense reasoning. In *Conference on Empirical Methods in Natural Language Processing*.

Jane Pan, Ryan Shar, Jacob Pfau, Ameet Talwalkar, He He, and Valerie Chen. 2025. When benchmarks talk: Re-evaluating code LLMs with interactive feedback. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24672–24700, Vienna, Austria. Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Raj Reddy. 1988. Foundations and grand challenges of artificial intelligence: Aaai presidential address. *AI magazine*, 9(4):9–9.

Hamzeh Roumani. 2002. Design guidelines for the lab component of objects-first cs1. *SIGCSE Bull.*, 34(1):222–226.

Michael Saxon, Ari Holtzman, Peter West, William Yang Wang, and Naomi Saphra. 2024. Benchmarks as microscopes: A call for model metrology. In *First Conference on Language Modeling*.

Murray Shanahan. 2024. Talking about large language models. *Commun. ACM*, 67(2):68–79.

Weiyan Shi, Emily Dinan, Kurt Shuster, Jason Weston, and Jing Xu. 2024. When life gives you lemons, make cherryade: Converting feedback from bad responses into good labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3066–3082, Mexico City, Mexico. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Lucas Spangher, Tianle Li, William F. Arnold, Nick Masiewicki, Xerxes Dotiwalla, Rama Kumar Pasumarthi, Peter Grabowski, Eugene Ie, and Daniel Gruhl. 2025. Chatbot arena estimate: towards a generalized performance benchmark for LLM capabilities. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 1016–1025, Albuquerque, New Mexico. Association for Computational Linguistics.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Allison Elliott Tew and Mark Guzdial. 2010. Developing a validated assessment of fundamental cs1 concepts. In *Proceedings of the 41st ACM technical symposium on Computer science education*, pages 97–101.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 101–114, Toronto, Canada. Association for Computational Linguistics.

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

Edward M White. 1993. Holistic scoring: Past triumphs, future challenges. *Validating holistic scoring for writing assessment: Theoretical and empirical foundations*, 79:88.

Rand R Wilcox. 1982. Some new results on an answer-until-correct scoring procedure. *Journal of Educational Measurement*, pages 67–74.

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

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

A Appendix

A.1 Dataset Details

We select 1000 random examples from the testing splits of ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), and Super GPQA (Du et al., 2025). All datasets are publicly available, so our experiments are within their intended use. We did not collect any datasets, so we did not check for PII. To our knowledge, all questions are in English.

A.2 Experiment Details

We implement all LLMs in InspectAI via LiteLLM.[3] We access all closed-source LLMs via their native APIs (e.g. the OpenAI API for GPT-5). For open-weight LLMs, we use the Cohere API[4] for Cohere models and TogetherAI[5] for the rest. We allocate 24 CPU hours for each experiment. All results are reported from one run. The model endpoints are:

1. anthropic/claude-3-7-sonnet-20250219
2. anthropic/claude-haiku-4-5-20251001
3. anthropic/claude-sonnet-4-20250514
4. anthropic/claude-sonnet-4-5-20250929
5. openai/gpt-4.1-2025-04-14
6. openai/gpt-5-2025-08-07
7. openai/gpt-5.2-2025-12-11
8. openai/gpt-5-mini-2025-08-07
9. openai/gpt-5-nano-2025-08-07
10. google/gemini-2.0-flash
11. google/gemini-2.5-flash-lite
12. google/gemini-2.5-pro
13. google/gemini-3-flash-preview
14. google/gemini-3-pro-preview
15. openai-api/cohere/command-r-08-2024
16. openai-api/cohere/command-r7b-12-2024
17. openai-api/cohere/command-r-plus-08-2024
18. together/Qwen/Qwen3-235B-A22B-Instruct-2507-tput
19. together/Qwen/Qwen3-235B-A22B-Thinking-2507
20. together/Qwen/Qwen3-Next-80B-A3B-Instruct
21. together/Qwen/Qwen3-Next-80B-A3B-Thinking
22. together/meta-llama/Meta-Llama-3.1-405B-Instruct-Turbo
23. together/meta-llama/Meta-Llama-3.1-70B-Instruct-Turbo
24. together/meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo
25. together/deepseek-ai/DeepSeek-R1
26. together/deepseek-ai/DeepSeek-V3.1
27. together/openai/gpt-oss-120b
28. together/openai/gpt-oss-20b
29. together/moonshotai/Kimi-K2-Thinking
30. together/zai-org/GLM-4.5-Air-FP8
31. together/zai-org/GLM-4.6

A.3 Full Ranking Analysis

Tables 4, 6, and 8 have the rankings of all 31 LLMs for ARC, MMLU, and SuperGPQA, respectively, while Tables 5, 7, and 9 have the LLM scores of these rankings, respectively. To break ties in ranks, we use Standard Competition or "1224" ranking,[6] where tied models share the same rank and ensuing positions are skipped accordingly (e.g., 1, 2, 2, 4). GPT-5 is consistently the strongest across datasets and scoring regimes—ranking the highest overall.

On SuperGPQA, Command-R+ and the GPT-OSS models timed out after 24 hours under AUC scoring. Based on their current progress, we estimated that these models would need up to a week to finish, so we did not re-run them, and omitted them from any analysis that involved SuperGPQA.

We also evaluate Pearson's correlation between scores directly (instead of ranks) in Figure 6 which shows the same trend as §3.1: schemes like ELIM, PPA, and AUC reward behaviors distinct from NR.

A.4 Full LLM Arena Analysis

Figure 7 shows the correlation between rankings for each scoring strategy and prompt type in LLM Arena across all datasets. The trend in §3.2 holds over each dataset: the Answer Until Correct regime best correlates with preferences from LLM Arena.

To understand the higher agreement from LLM Arena further, we design an extension of AUC that executes two steps: 1) the model makes an initial prediction; and 2) the model reflects on its answer and decide whether it was correct or incorrect—following self-refine (Madaan et al., 2023). We run this scheme on ARC with GPT and open-weight models in §2.2. The self-refine ablation reaches an agreement of 0.89, AUC reaches 0.86, and NR only reaches 0.79—further indicating that the ability to critically self-reflect is the behavior that correlates well with human preferences.

[3] https://docs.litellm.ai/
[4] https://cohere.com/
[5] https://www.together.ai/
[6] https://en.wikipedia.org/wiki/Ranking#Standard_competition_ranking_("1224"_ranking)

A.5 Full Consistency Analysis

Figure 8 and 9 replicate Figure 3 but on ARC and Super GPQA which show a similar trend: smaller models without reasoning tend to be more inconsistent compared to their mode capable counterparts, except for select, strong fronteir models like GPT and Gemini. Figure 10, 12, and 14 have the consistency analysis results for all scoring strategies in §3.3 replicated across ARC, MMLU, and Super GPQA for closed-source models, respectively, while Figure 11, 13, 15 show consistency analyses for the same datasets and open-weight models, respectively. While closed-source and larger models are fairly consistent between scoring strategies, smaller and open-weight models lag further behind.

A.6 Full Model Behavior Analysis

Figures 16, 17, and 18 show the behavioral analysis from §3.4 across all datasets, models, and evaluated scoring regimes. Tables 5, 7, and 9 have the LLM scores of these rankings, respectively.

GPT models rarely abstain and state high confidence. Ranking models by NM abstention rate in Figures 16 and 17, GPT models occupy six of the ten lowest-abstaining positions; on the harder Super GPQA (Figure 18), six of the eight least-abstaining models are GPT variants. An exception is GPT-OSS-20B, a weaker open-sourced GPT model that falls near the upper half of the abstention ranking across all datasets. GPT models state high confidence as well: while most LLMs avoid low-confidence responses on easier datasets like ARC and MMLU, GPT models take the six positions with the fewest low-confidence responses on the harder Super GPQA (Figure 18).

Stronger GPT models readily revise answers after feedback; weaker ones struggle to do so. The answer-change behavior from the figures supports this directly: GPT-4.1, GPT-5, GPT-5.2, and GPT-OSS-120B fill four of the top six answer-change positions on ARC (Figure 16) and all four of the top four on MMLU (Figure 17). On Super GPQA, GPT-OSS-120B exceeded our time budget, yet the other three stronger GPT models still claim the top three positions (Figure 18). By contrast, GPT-5 Nano finishes last on ARC and second-to-last on Super GPQA in answer-change rate, and GPT-5 Mini ranks near the bottom on ARC and MMLU—the smallest GPT models are far less willing to revise answers upon feedback.

Weaker open-weight models often abstain or hesitate to eliminate choices. Command-R+ holds the single highest NM abstention rate on all three datasets (Figures 16–18), consistently the most likely model to pass rather than commit to an answer. Llama models show the same avoidance tendency, but in ELIM rather than abstention: across all three datasets, Llama 3.1 70B ranks last (31st on ARC and MMLU, 29th on Super GPQA), Llama 3.1 8B remains near the bottom (29th on ARC and MMLU, 28th on Super GPQA), and even Llama 3.1 405B falls in the lower quarter (24th–25th on ARC and MMLU, 23rd on Super GPQA)—indicating a systematic reluctance to rule out incorrect options for Llama models.

All LLMs show high confidence and low abstention on easier datasets, but not on Super GPQA. On ARC and MMLU, most models predominantly express high confidence and rarely abstain. On Super GPQA this changes markedly: abstention rates rise and models that predominantly expressed high confidence on easier tasks shift toward low and mid confidence (Figures 16–18). Even the ELIM elimination rates decline with difficulty, as models eliminate fewer choices on harder questions.

A.7 Prompts

We show the prompts for all eight scoring schemes in Prompts A.1–A.8. Each scheme adapts the base NR prompt (Prompt A.1)—initially drawn from InspectAI (UK AISI, 2024)—to alter the *response mode*: how the LLM responds to the MCQ. Afterwards, the scheme may also adjust the *scoring rule* via custom metrics.

Number Right (NR, Prompt A.1). This is the standard prompt used in MCQA: the response mode asks the model to select one answer letter, and the scoring rule rewards only correct answers (Kelly, 1916).

Control Variation (NR-VAR, Prompt A.2). We rephrase NR's task *instruction* with synonymous language while leaving both the response mode and scoring rule unchanged, to isolate sensitivity to prompt wording (Alzahrani et al., 2024).

Negative Marking (NM, Prompt A.3). We adapt NR's response mode to let LLMs abstain (via `ABSTAIN` in the answer field), and update the scoring rule to penalize wrong answers while rewarding abstaining over guessing (Holt, 2006).

Elimination (ELIM, Prompt A.4). Instead of selecting a correct answer, LLMs output a list of choices they believe are wrong. The scoring rule is updated to grant partial credit for each eliminated distractor, but revokes all credit if the correct answer is eliminated (Coombs et al., 1956).

Elimination with Negative Marking (ELIM-NM, Prompt A.5). We keep ELIM's elimination response mode but adapt the scoring rule to additionally penalize eliminating the correct answer, combining partial-knowledge credit with a guessing deterrent (Coombs et al., 1956).

Confidence-Based Marking (CBM, Prompt A.6). We extend NR's response mode to require a discrete confidence level alongside the answer, and update the scoring rule so that higher declared confidence amplifies both the reward for correct answers and the penalty for wrong ones (Gardner-Medwin, 2006).

Personal Point Allocation (PPA, Prompt A.7). We replace NR's single-answer response mode with a probability distribution over all choices, and replace the binary scoring rule with the Brier score to reward both accuracy and calibration (MacNeil et al., 2023).

Answer Until Correct (AUC, Prompt A.8). We adapt NR's response mode to allow multiple attempts, injecting a feedback block after each wrong answer that lists the LLM's prior incorrect choices, and update the scoring rule to reward answering correctly in fewer attempts (Wilcox, 1982).

Scheme	Agreement with Number Right
Rephrased Prompt	0.93
Shuffle Choices	0.96
Numbers as Choices	0.98
NM	0.94
ELIM	0.80
ELIM-NM	0.80
CBM	0.95
PPA	0.86
AUC	0.86

Table 2: Agreement between alternative scoring schemes with Number Right on ARC, contextualized by three prompt variation baselines for Number Right: rephrasing the text of the instructions, shuffling answer choices, and replacing letters with numbers in the choices.

Scoring Scheme	ARC (In)	ARC (Out)	MMLU (In)	MMLU (Out)	SuperGPQA (In)	SuperGPQA (Out)
NR	207	122	277	186	1,602	1,236
NR-VAR	206	101	271	151	1,297	923
NM	260	140	326	197	1,389	791
AUC	492	168	1,363	436	45,695	4,088
CBM	384	153	458	210	1,647	1,142
PPA	428	211	492	276	1,448	1,076
ELIM	298	179	373	259	1,489	980
ELIM-NM	297	184	379	278	1,462	984

Table 3: The median number of input and output tokens generated on each dataset across the distribution of models tested. While AUC benefits MCQA evaluation, we caveat that the scheme's multi-turn design increases evaluation costs—especially on difficult datasets like SuperGPQA with many choices.

Model	NR	NR-VAR	NM	AUC	CBM	PPA	ELIM	ELIM-NM	Overall
Claude 3.7 Sonnet	12	15	22	25	18	19	25	25	23
Claude Haiku 4.5	23	22	19	19	22	21	21	20	12
Claude Sonnet 4	1	4	5	13	3	10	13	12	11
Claude Sonnet 4.5	1	2	1	1	4	5	18	18	10
Gemini 2.0 Flash	20	21	21	18	21	23	26	26	25
Gemini 2.5 Flash Lite	25	25	25	26	26	24	23	24	26
Gemini 2.5 Pro	7	7	3	3	1	4	3	3	4
Gemini 3 Flash	3	1	2	4	9	1	1	1	2
Gemini 3 Pro	3	4	6	5	5	2	2	2	6
Command R	28	28	29	29	29	28	28	28	28
Command R+	26	27	28	28	28	25	27	27	27
Command R7B	31	29	31	31	31	29	30	31	30
GPT-4.1	17	8	15	7	15	11	11	9	16
GPT-5	3	2	4	2	2	3	3	3	1
GPT-5 Mini	6	6	7	16	6	6	7	5	3
GPT-5 Nano	20	23	17	20	19	18	15	15	8
GPT-5.2	8	9	9	6	8	8	9	13	13
Qwen3-235B Instruct	16	17	16	11	13	20	22	22	14
Qwen3-235B Thinking	12	12	8	8	12	13	8	7	9
Qwen3-80B Instruct	18	19	18	13	16	22	19	19	17
Qwen3-80B Thinking	15	16	13	17	14	15	12	10	15
DeepSeek-R1	10	9	13	13	10	12	9	8	18
DeepSeek-V3.1	19	17	24	23	19	17	20	21	22
Llama 3.1 405B	28	26	20	24	24	26	24	23	24
Llama 3.1 70B	27	31	27	27	27	31	31	30	31
Llama 3.1 8B	30	30	30	30	30	27	29	29	29
Kimi-K2 Thinking	9	9	12	10	11	7	5	6	7
GPT-OSS-120B	10	19	10	9	17	9	6	11	5
GPT-OSS-20B	24	24	26	22	25	14	14	17	19
GLM-4.5 Air	22	12	23	21	23	16	16	14	20
GLM-4.6	14	14	10	12	7	30	17	16	21

Table 4: Model ranks across scoring schemes for ARC

Model	NR	NR-VAR	NM	AUC	CBM	PPA	ELIM	ELIM-NM	Overall
Claude 3.7 Sonnet	0.967	0.963	0.938	0.95	0.958	0.943	0.851	0.791	0.681
Claude Haiku 4.5	0.952	0.956	0.941	0.979	0.954	0.931	0.897	0.834	0.76
Claude Sonnet 4	0.981	0.977	0.968	0.981	0.977	0.966	0.944	0.927	0.768
Claude Sonnet 4.5	0.981	0.978	0.975	0.991	0.976	0.973	0.904	0.881	0.772
Gemini 2.0 Flash	0.956	0.957	0.938	0.979	0.956	0.922	0.837	0.788	0.646
Gemini 2.5 Flash Lite	0.94	0.942	0.921	0.946	0.944	0.917	0.86	0.811	0.623
Gemini 2.5 Pro	0.975	0.971	0.971	0.99	0.981	0.974	0.973	0.959	0.832
Gemini 3 Flash	0.98	0.983	0.975	0.989	0.97	0.985	0.982	0.971	0.872
Gemini 3 Pro	0.98	0.977	0.968	0.987	0.976	0.983	0.98	0.965	0.801
Command R	0.863	0.862	0.797	0.914	0.856	0.716	0.702	0.593	0.552
Command R+	0.89	0.888	0.822	0.938	0.889	0.829	0.758	0.651	0.591
Command R7B	0.8	0.807	0.729	0.842	0.794	0.657	0.544	0.318	0.462
GPT-4.1	0.963	0.968	0.946	0.986	0.962	0.96	0.95	0.934	0.752
GPT-5	0.98	0.978	0.97	0.99	0.979	0.979	0.973	0.959	0.896
GPT-5 Mini	0.976	0.975	0.962	0.981	0.974	0.972	0.962	0.945	0.852
GPT-5 Nano	0.956	0.954	0.945	0.967	0.957	0.947	0.937	0.893	0.798
GPT-5.2	0.972	0.967	0.954	0.986	0.972	0.969	0.959	0.926	0.759
Qwen3-235B Instruct	0.964	0.961	0.945	0.983	0.965	0.937	0.887	0.821	0.758
Qwen3-235B Thinking	0.967	0.966	0.955	0.985	0.967	0.959	0.961	0.937	0.795
Qwen3-80B Instruct	0.962	0.96	0.943	0.981	0.961	0.931	0.901	0.844	0.743
Qwen3-80B Thinking	0.965	0.962	0.95	0.98	0.963	0.95	0.948	0.933	0.755
DeepSeek-R1	0.968	0.967	0.95	0.981	0.968	0.96	0.959	0.935	0.727
DeepSeek-V3.1	0.96	0.961	0.921	0.96	0.957	0.948	0.898	0.833	0.7
Llama 3.1 405B	0.863	0.914	0.94	0.955	0.95	0.809	0.857	0.819	0.647
Llama 3.1 70B	0.885	0.768	0.908	0.943	0.922	0	0.142	0.421	0.459
Llama 3.1 8B	0.804	0.804	0.756	0.894	0.816	0.748	0.684	0.56	0.52
Kimi-K2 Thinking	0.97	0.967	0.953	0.984	0.967	0.97	0.967	0.943	0.799
GPT-OSS-120B	0.968	0.96	0.953	0.984	0.96	0.969	0.963	0.929	0.824
GPT-OSS-20B	0.943	0.947	0.909	0.966	0.945	0.951	0.941	0.886	0.718
GLM-4.5 Air	0.954	0.966	0.933	0.966	0.954	0.949	0.917	0.908	0.713
GLM-4.6	0.966	0.965	0.953	0.982	0.972	0.483	0.904	0.888	0.706

Table 5: Model scores across scoring schemes for ARC

Model	NR	NR-VAR	NM	AUC	CBM	PPA	ELIM	ELIM-NM	Overall
Claude 3.7 Sonnet	15	14	24	26	14	15	23	23	23
Claude Haiku 4.5	19	17	15	14	18	21	19	18	12
Claude Sonnet 4	8	7	8	17	9	10	14	11	11
Claude Sonnet 4.5	5	5	6	5	7	6	11	7	10
Gemini 2.0 Flash	23	20	22	22	24	24	26	25	25
Gemini 2.5 Flash Lite	25	26	27	28	27	23	24	26	26
Gemini 2.5 Pro	4	4	5	3	5	4	4	4	4
Gemini 3 Flash	1	2	2	2	2	3	3	2	2
Gemini 3 Pro	3	3	3	4	3	2	1	1	6
Command R	29	29	29	29	29	26	28	30	28
Command R+	27	27	28	25	28	25	27	27	27
Command R7B	31	30	31	31	30	29	30	31	30
GPT-4.1	16	18	17	9	19	17	14	13	16
GPT-5	2	1	1	1	1	1	2	3	1
GPT-5 Mini	7	6	4	11	4	5	6	9	3
GPT-5 Nano	14	15	13	20	15	13	12	16	8
GPT-5.2	11	11	14	6	11	12	10	14	13
Qwen3-235B Instruct	17	19	18	13	16	18	22	22	14
Qwen3-235B Thinking	6	9	9	12	8	9	7	10	9
Qwen3-80B Instruct	20	21	21	15	23	22	21	21	17
Qwen3-80B Thinking	18	16	16	18	17	16	13	12	15
DeepSeek-R1	12	10	11	10	13	11	9	8	18
DeepSeek-V3.1	21	22	25	23	21	20	20	20	22
Llama 3.1 405B	28	25	19	27	20	27	25	24	24
Llama 3.1 70B	26	28	26	24	26	31	31	28	31
Llama 3.1 8B	30	31	30	30	31	28	29	29	29
Kimi-K2 Thinking	10	8	7	7	6	8	5	5	7
GPT-OSS-120B	9	11	10	8	10	7	8	6	5
GPT-OSS-20B	24	24	23	21	25	19	18	19	19
GLM-4.5 Air	22	23	20	19	22	14	17	17	20
GLM-4.6	13	13	12	16	12	30	16	15	21

Table 6: Model ranks across scoring schemes for MMLU

Model	NR	NR-VAR	NM	AUC	CBM	PPA	ELIM	ELIM-NM	Overall
Claude 3.7 Sonnet	0.867	0.867	0.767	0.83	0.876	0.868	0.703	0.572	0.681
Claude Haiku 4.5	0.857	0.858	0.819	0.92	0.862	0.823	0.785	0.673	0.76
Claude Sonnet 4	0.896	0.901	0.868	0.905	0.901	0.89	0.831	0.769	0.768
Claude Sonnet 4.5	0.915	0.907	0.878	0.945	0.902	0.904	0.844	0.805	0.772
Gemini 2.0 Flash	0.816	0.835	0.785	0.863	0.83	0.766	0.635	0.556	0.646
Gemini 2.5 Flash Lite	0.786	0.785	0.724	0.81	0.794	0.779	0.683	0.533	0.623
Gemini 2.5 Pro	0.921	0.913	0.879	0.953	0.905	0.919	0.913	0.849	0.832
Gemini 3 Flash	0.94	0.935	0.91	0.959	0.936	0.933	0.925	0.885	0.872
Gemini 3 Pro	0.935	0.934	0.89	0.951	0.927	0.933	0.934	0.896	0.801
Command R	0.667	0.67	0.55	0.777	0.746	0.628	0.551	0.307	0.552
Command R+	0.751	0.756	0.618	0.833	0.759	0.699	0.607	0.398	0.591
Command R7B	0.627	0.627	0.511	0.663	0.648	0.595	0.412	0.08	0.462
GPT-4.1	0.862	0.857	0.812	0.93	0.86	0.854	0.831	0.751	0.752
GPT-5	0.938	0.937	0.92	0.969	0.946	0.948	0.925	0.873	0.896
GPT-5 Mini	0.899	0.906	0.879	0.926	0.91	0.914	0.881	0.793	0.852
GPT-5 Nano	0.874	0.866	0.827	0.884	0.874	0.875	0.839	0.722	0.798
GPT-5.2	0.88	0.878	0.825	0.939	0.89	0.882	0.845	0.747	0.759
Qwen3-235B Instruct	0.861	0.853	0.81	0.925	0.866	0.85	0.731	0.587	0.758
Qwen3-235B Thinking	0.903	0.888	0.863	0.925	0.902	0.9	0.876	0.788	0.795
Qwen3-80B Instruct	0.844	0.834	0.789	0.915	0.834	0.804	0.744	0.599	0.743
Qwen3-80B Thinking	0.858	0.86	0.814	0.897	0.865	0.857	0.838	0.763	0.755
DeepSeek-R1	0.879	0.883	0.836	0.928	0.879	0.884	0.863	0.799	0.727
DeepSeek-V3.1	0.84	0.833	0.763	0.847	0.842	0.837	0.771	0.626	0.7
Llama 3.1 405B	0.689	0.794	0.796	0.826	0.848	0.626	0.652	0.56	0.647
Llama 3.1 70B	0.754	0.688	0.736	0.842	0.808	0.002	0.142	0.319	0.459
Llama 3.1 8B	0.628	0.625	0.516	0.737	0.642	0.624	0.537	0.318	0.52
Kimi-K2 Thinking	0.892	0.897	0.871	0.936	0.903	0.901	0.894	0.815	0.799
GPT-OSS-120B	0.893	0.878	0.836	0.935	0.895	0.902	0.873	0.806	0.824
GPT-OSS-20B	0.806	0.825	0.772	0.868	0.827	0.845	0.793	0.668	0.718
GLM-4.5 Air	0.829	0.832	0.791	0.891	0.842	0.875	0.81	0.681	0.713
GLM-4.6	0.875	0.872	0.831	0.91	0.889	0.461	0.818	0.726	0.706

Table 7: Model scores across scoring schemes for MMLU

Model	NR	NR-VAR	NM	AUC	CBM	PPA	ELIM	ELIM-NM	Overall
Claude 3.7 Sonnet	20	21	24	26	23	16	24	26	23
Claude Haiku 4.5	14	10	9	10	8	13	7	8	12
Claude Sonnet 4	11	11	11	19	12	11	14	12	11
Claude Sonnet 4.5	15	14	12	15	11	9	19	11	10
Gemini 2.0 Flash	26	26	25	25	27	28	26	25	25
Gemini 2.5 Flash Lite	30	30	27	27	30	30	25	24	26
Gemini 2.5 Pro	6	4	4	9	6	7	4	4	4
Gemini 3 Flash	3	3	2	6	3	3	2	2	2
Gemini 3 Pro	11	16	16	17	15	6	8	7	6
Command R	29	28	29	21	20	18	30	31	28
Command R+	25	25	30	—	24	29	27	29	27
Command R7B	31	31	31	28	31	12	31	30	30
GPT-4.1	17	19	18	4	22	21	11	14	16
GPT-5	1	1	1	1	1	1	1	1	1
GPT-5 Mini	2	2	3	8	2	2	3	3	3
GPT-5 Nano	4	5	6	13	5	5	6	5	8
GPT-5.2	18	18	17	3	14	15	18	22	13
Qwen3-235B Instruct	8	8	13	2	7	8	13	16	14
Qwen3-235B Thinking	7	7	7	11	9	14	10	9	9
Qwen3-80B Instruct	10	11	15	7	13	10	12	17	17
Qwen3-80B Thinking	13	13	10	12	17	22	20	13	15
DeepSeek-R1	21	22	20	16	26	25	22	20	18
DeepSeek-V3.1	19	17	22	23	19	23	21	21	22
Llama 3.1 405B	24	22	19	18	25	27	23	23	24
Llama 3.1 70B	27	27	26	22	28	31	29	27	31
Llama 3.1 8B	28	29	28	24	29	26	28	28	29
Kimi-K2 Thinking	8	9	8	5	10	17	9	10	7
GPT-OSS-120B	5	6	5	—	4	4	5	6	5
GPT-OSS-20B	22	20	21	—	21	19	17	18	19
GLM-4.5 Air	23	24	23	20	18	20	16	19	20
GLM-4.6	16	15	14	14	16	24	15	15	21

Table 8: Model ranks across scoring schemes for SuperGPQA. Dashes (—) denote models that could not complete the evaluation run in time, and were thus omitted from analyses.

Model	NR	NR-VAR	NM	AUC	CBM	PPA	ELIM	ELIM-NM	Overall
Claude 3.7 Sonnet	0.364	0.333	0.212	0.291	0.411	0.512	0.314	0.197	0.681
Claude Haiku 4.5	0.427	0.477	0.432	0.725	0.547	0.525	0.582	0.495	0.76
Claude Sonnet 4	0.469	0.466	0.406	0.5	0.523	0.535	0.445	0.401	0.768
Claude Sonnet 4.5	0.422	0.435	0.384	0.602	0.541	0.564	0.417	0.41	0.772
Gemini 2.0 Flash	0.217	0.228	0.183	0.398	0.272	0.314	0.265	0.205	0.646
Gemini 2.5 Flash Lite	0.179	0.193	0.16	0.251	0.197	0.268	0.31	0.212	0.623
Gemini 2.5 Pro	0.555	0.624	0.542	0.737	0.588	0.635	0.639	0.604	0.832
Gemini 3 Flash	0.673	0.66	0.651	0.775	0.716	0.717	0.76	0.728	0.872
Gemini 3 Pro	0.469	0.42	0.36	0.551	0.469	0.68	0.559	0.506	0.801
Command R	0.183	0.216	0.097	0.429	0.427	0.491	0.193	0.021	0.552
Command R+	0.218	0.231	0.095	—	0.365	0.295	0.217	0.079	0.591
Command R7B	0.153	0.14	0.085	0.166	0.158	0.529	0.172	0.027	0.462
GPT-4.1	0.394	0.377	0.306	0.805	0.414	0.463	0.492	0.381	0.752
GPT-5	0.765	0.748	0.707	0.923	0.791	0.774	0.788	0.749	0.896
GPT-5 Mini	0.674	0.69	0.642	0.769	0.73	0.733	0.698	0.646	0.852
GPT-5 Nano	0.597	0.618	0.527	0.611	0.643	0.693	0.6	0.552	0.798
GPT-5.2	0.393	0.41	0.309	0.83	0.474	0.514	0.426	0.271	0.759
Qwen3-235B Instruct	0.525	0.519	0.381	0.834	0.569	0.589	0.457	0.37	0.758
Qwen3-235B Thinking	0.549	0.55	0.505	0.674	0.546	0.521	0.5	0.484	0.795
Qwen3-80B Instruct	0.508	0.466	0.37	0.771	0.484	0.536	0.488	0.365	0.743
Qwen3-80B Thinking	0.459	0.458	0.421	0.67	0.446	0.437	0.416	0.399	0.755
DeepSeek-R1	0.312	0.316	0.28	0.561	0.337	0.349	0.343	0.309	0.727
DeepSeek-V3.1	0.391	0.412	0.259	0.418	0.434	0.427	0.368	0.301	0.7
Llama 3.1 405B	0.232	0.316	0.292	0.527	0.362	0.344	0.318	0.235	0.647
Llama 3.1 70B	0.208	0.221	0.164	0.428	0.256	0.062	0.2	0.184	0.459
Llama 3.1 8B	0.186	0.195	0.142	0.414	0.219	0.348	0.204	0.08	0.52
Kimi-K2 Thinking	0.525	0.518	0.498	0.79	0.545	0.503	0.501	0.465	0.799
GPT-OSS-120B	0.593	0.597	0.541	—	0.663	0.7	0.616	0.537	0.824
GPT-OSS-20B	0.28	0.368	0.277	—	0.422	0.482	0.433	0.357	0.718
GLM-4.5 Air	0.264	0.291	0.252	0.489	0.435	0.48	0.437	0.356	0.713
GLM-4.6	0.406	0.431	0.373	0.603	0.458	0.364	0.443	0.378	0.706

Table 9: Model scores across scoring schemes for SuperGPQA. Dashes (—) denote models that could not complete the evaluation run in time, and were thus omitted from analyses.

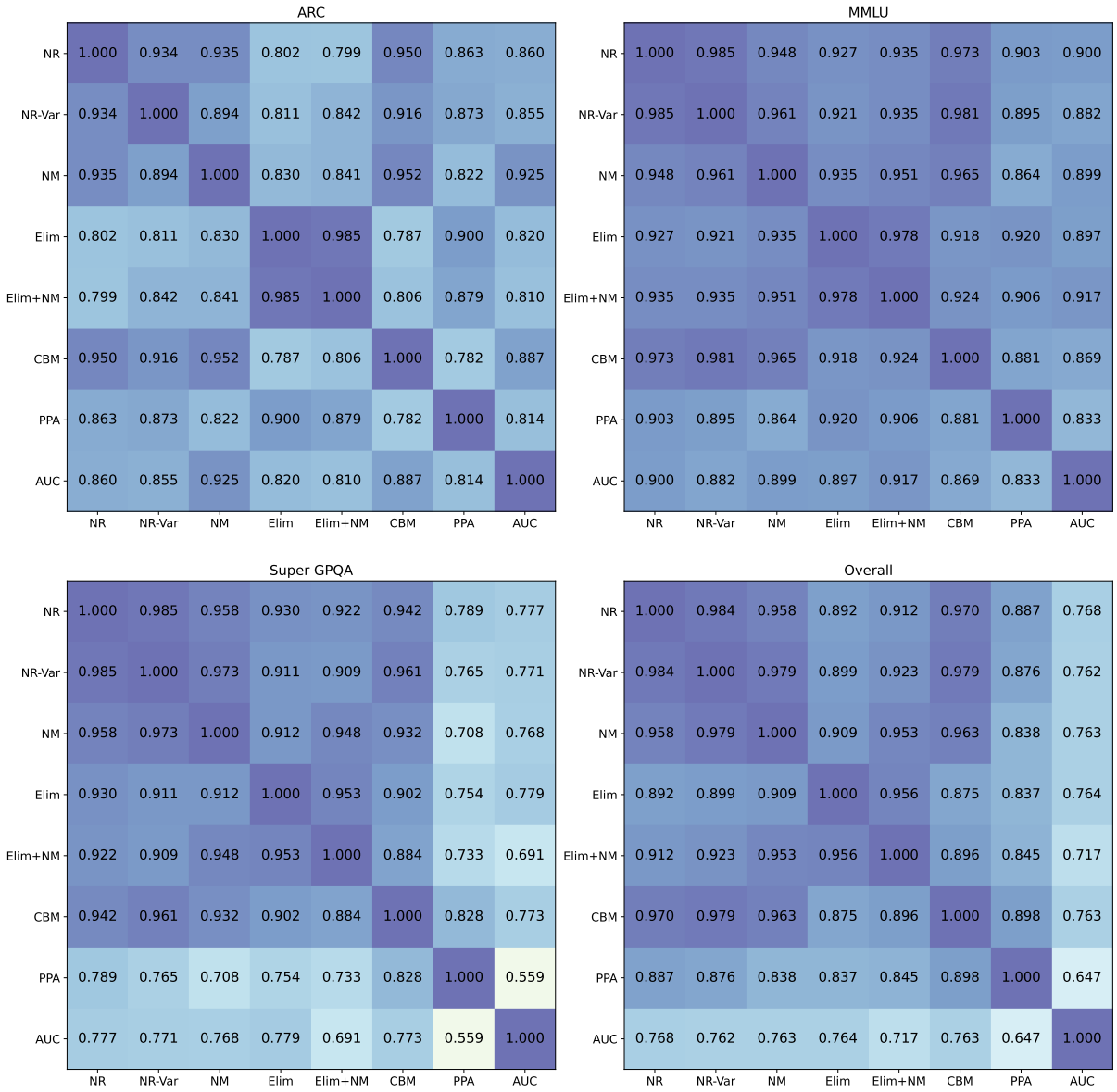

Figure 5: Spearman's correlation between model rankings across all scoring schemes and datasets.

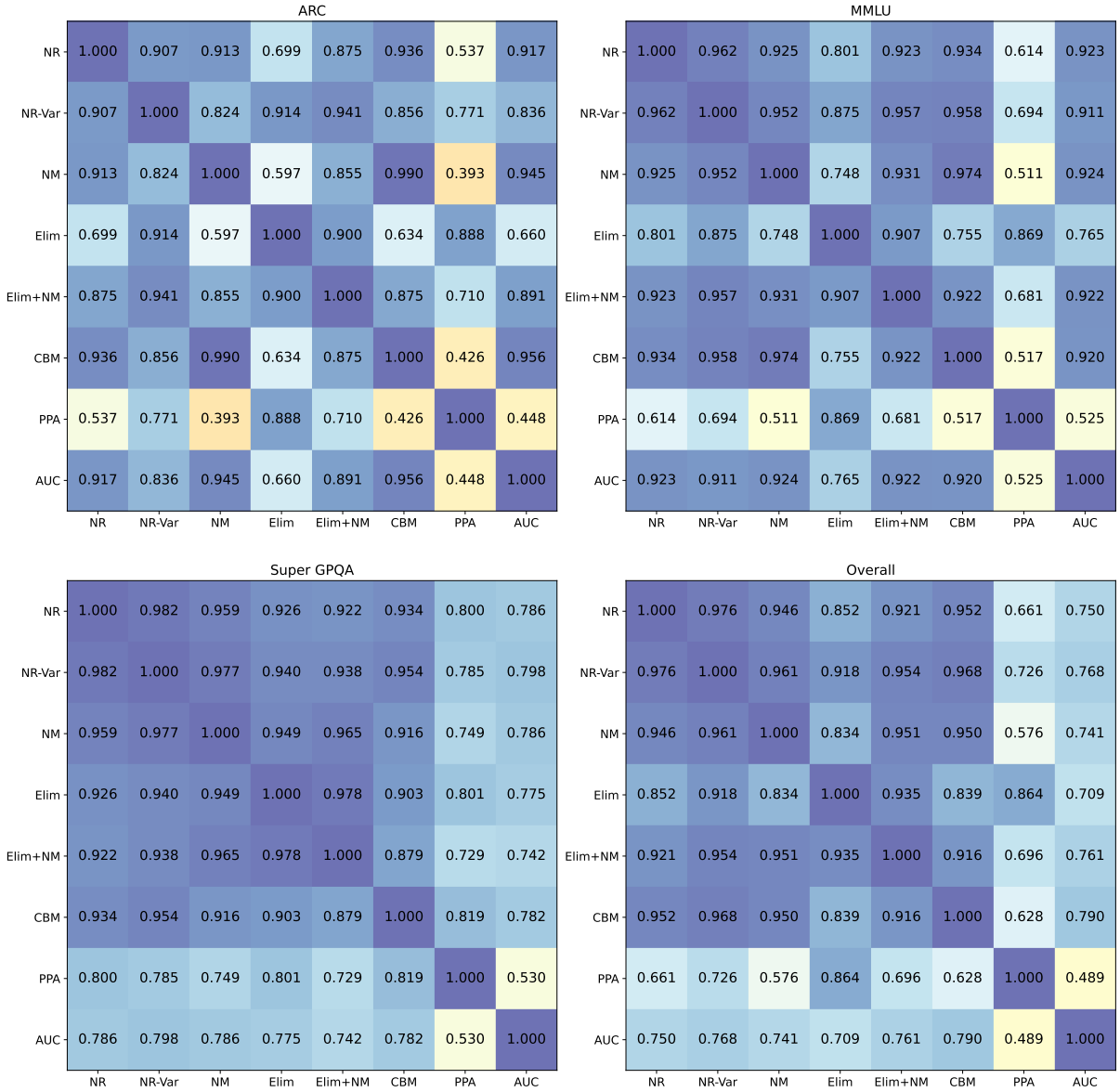

Figure 6: Pearson's correlation between scores across all schemes and datasets. We run Min-Max normalization on all scores. When considering disagreements at the score level, the trends in §3.1 become starker: certain schemes like ELIM, PPA, and AUC show large disagreements from NR, further suggesting that they test different abilities.

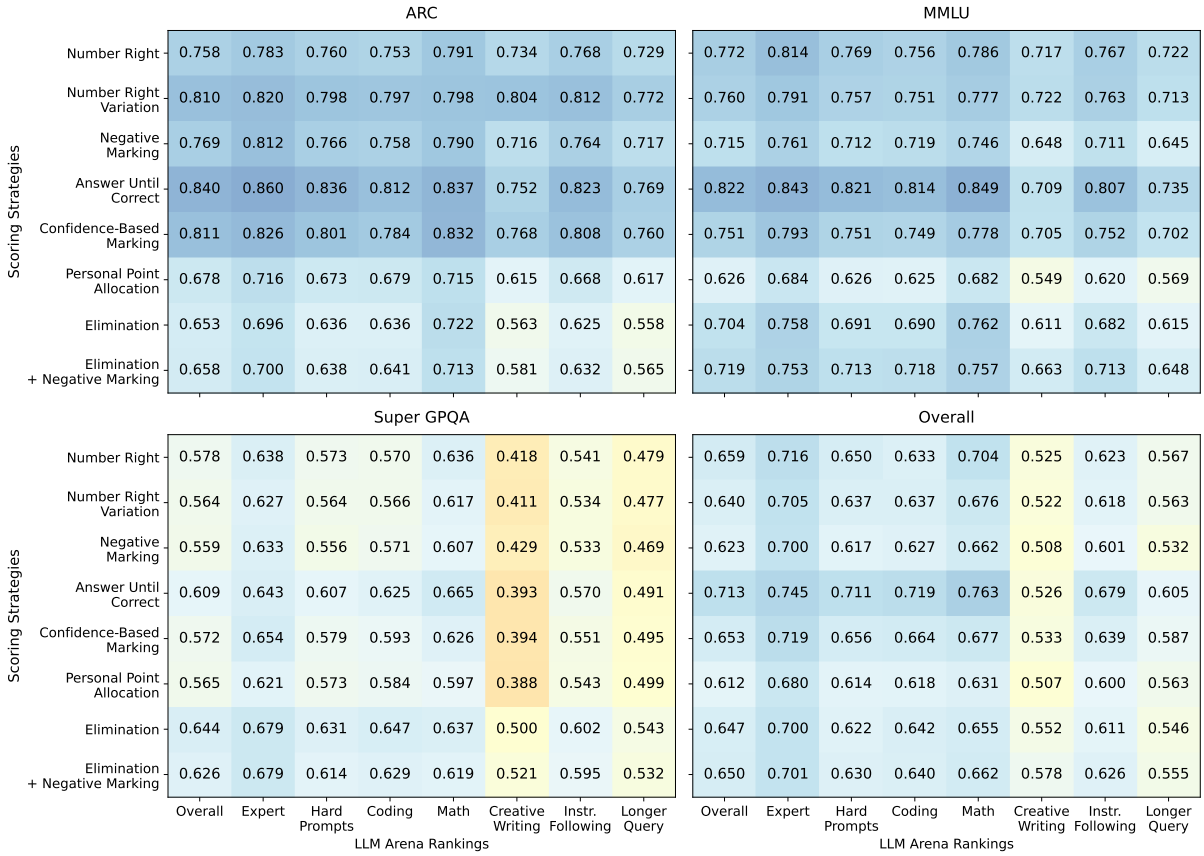

Figure 7: Correlation between scoring strategy and LLM Arena rankings across all datasets. Notably, Answer Until Correct has higher agreement with human preferences on LLM Arena across all datasets.

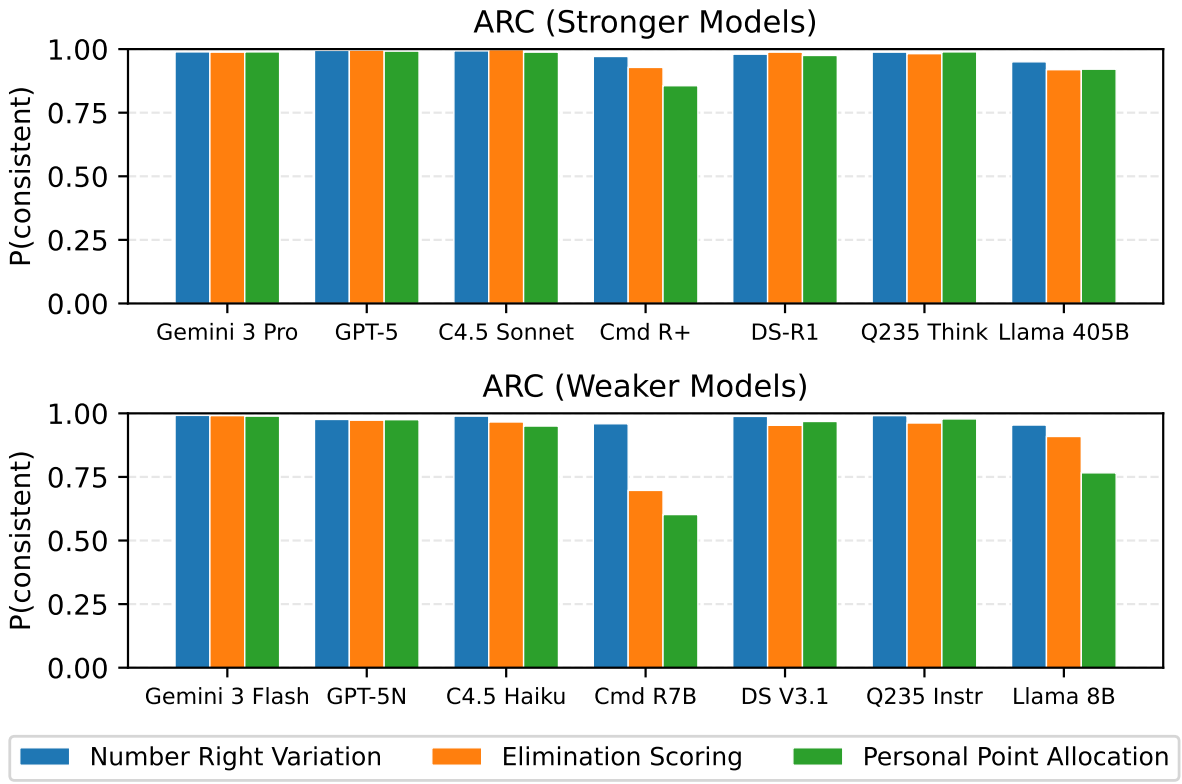

Figure 8: Consistency checks between weaker and stronger models between different scoring regimes and standard scoring on ARC. Weaker models (e.g., smaller, less reasoning) tend to be less consistent, except for strong closed-source models like Gemini and GPT.

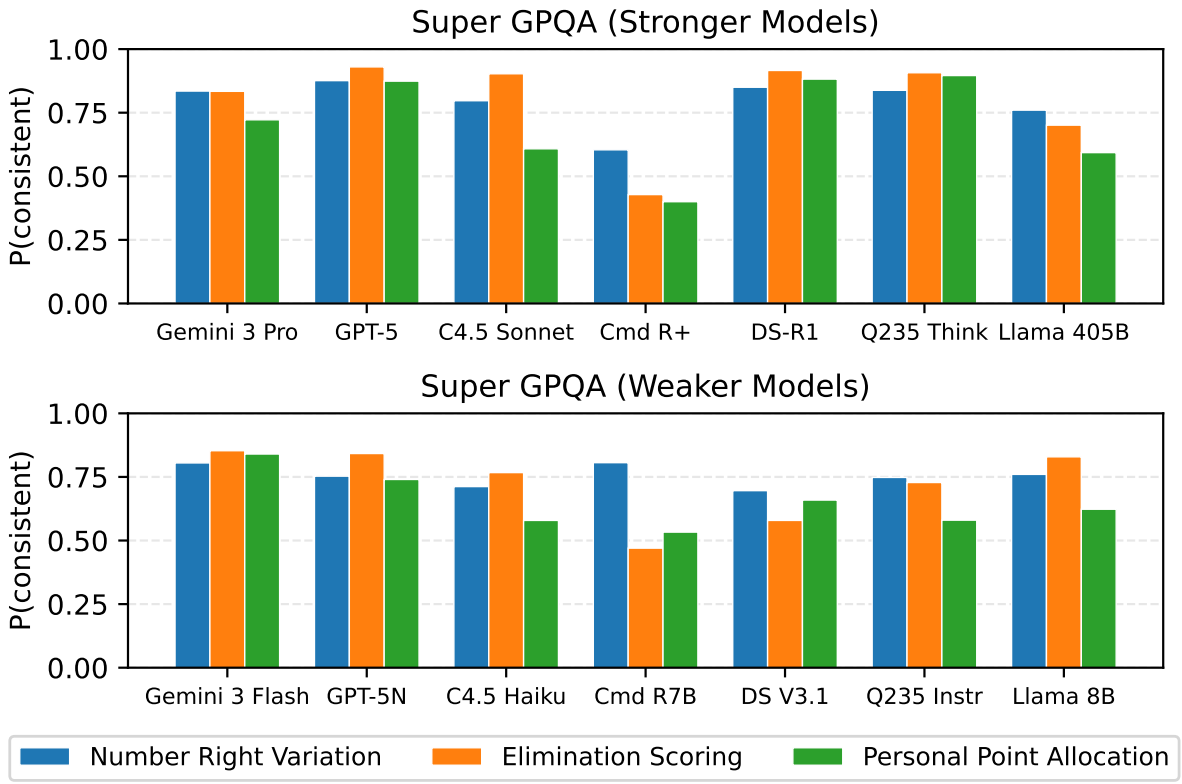

Figure 9: Consistency checks between weaker and stronger models between different scoring regimes and standard scoring on SuperGPQA. Weaker models (e.g., smaller, less reasoning) tend to be less consistent, except for strong closed-source models like Gemini and GPT.

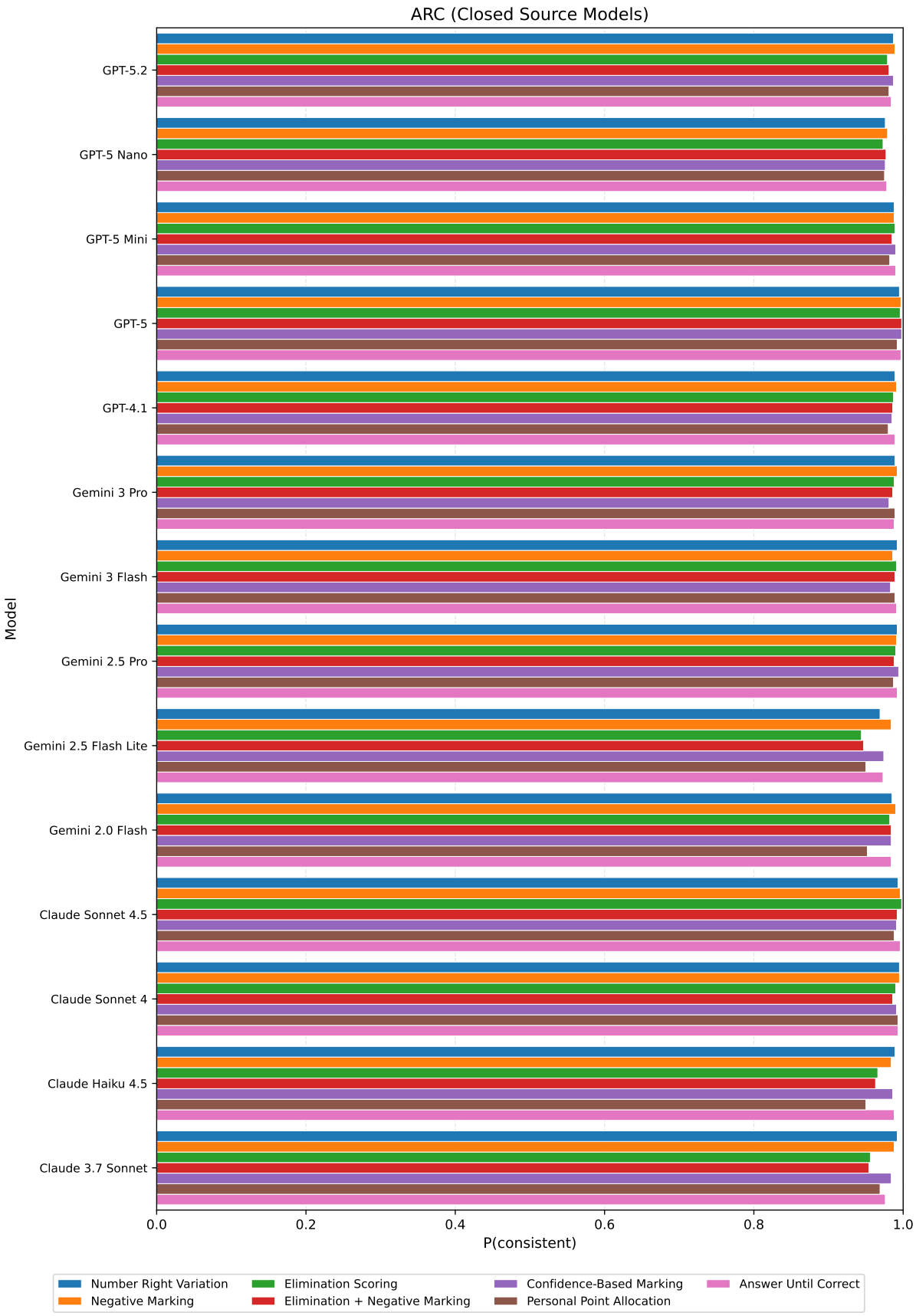

Figure 10: Consistency checks for closed-source models on ARC between all scoring strategies.

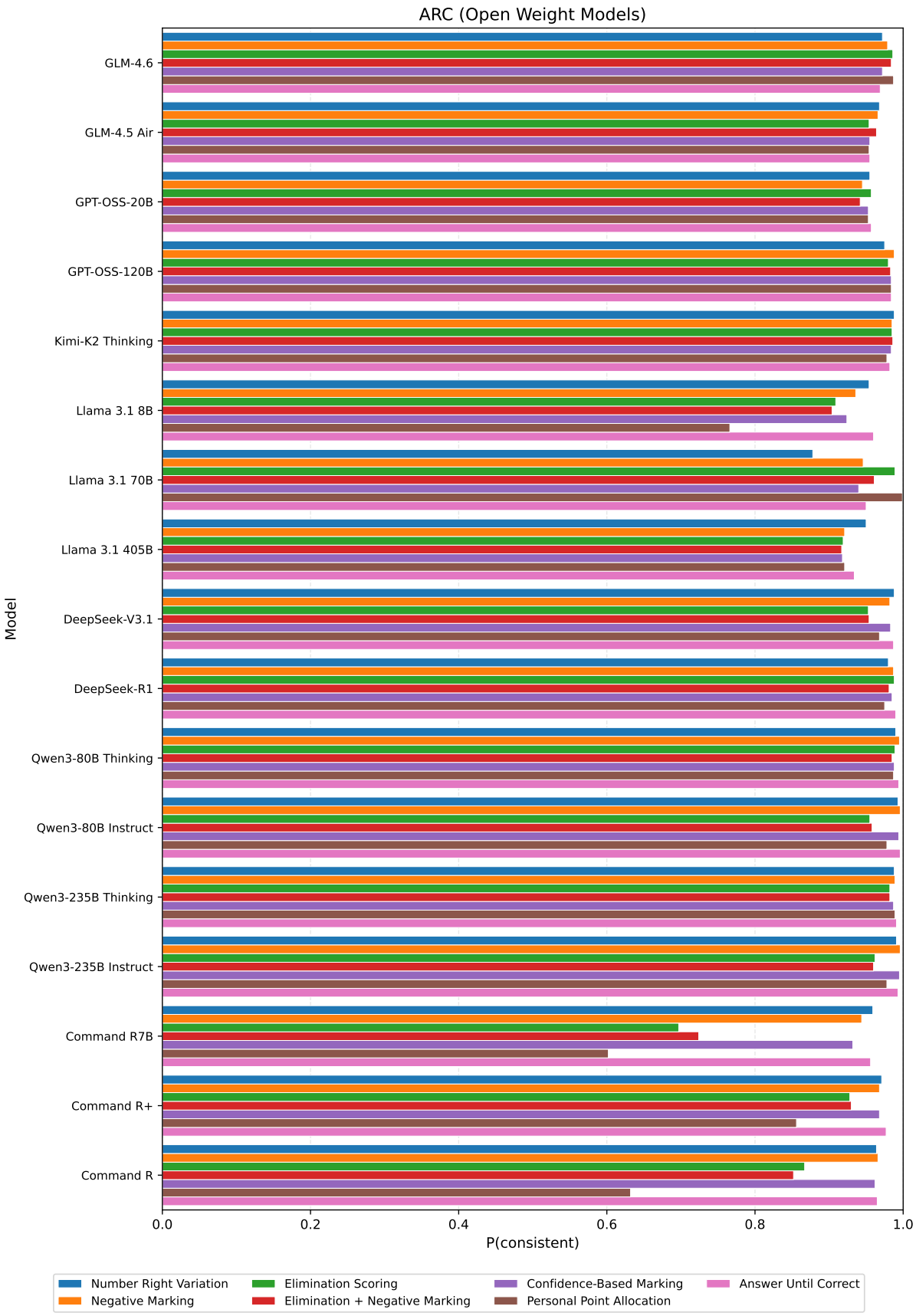

Figure 11: Consistency checks for open-weight models on ARC between all scoring strategies.

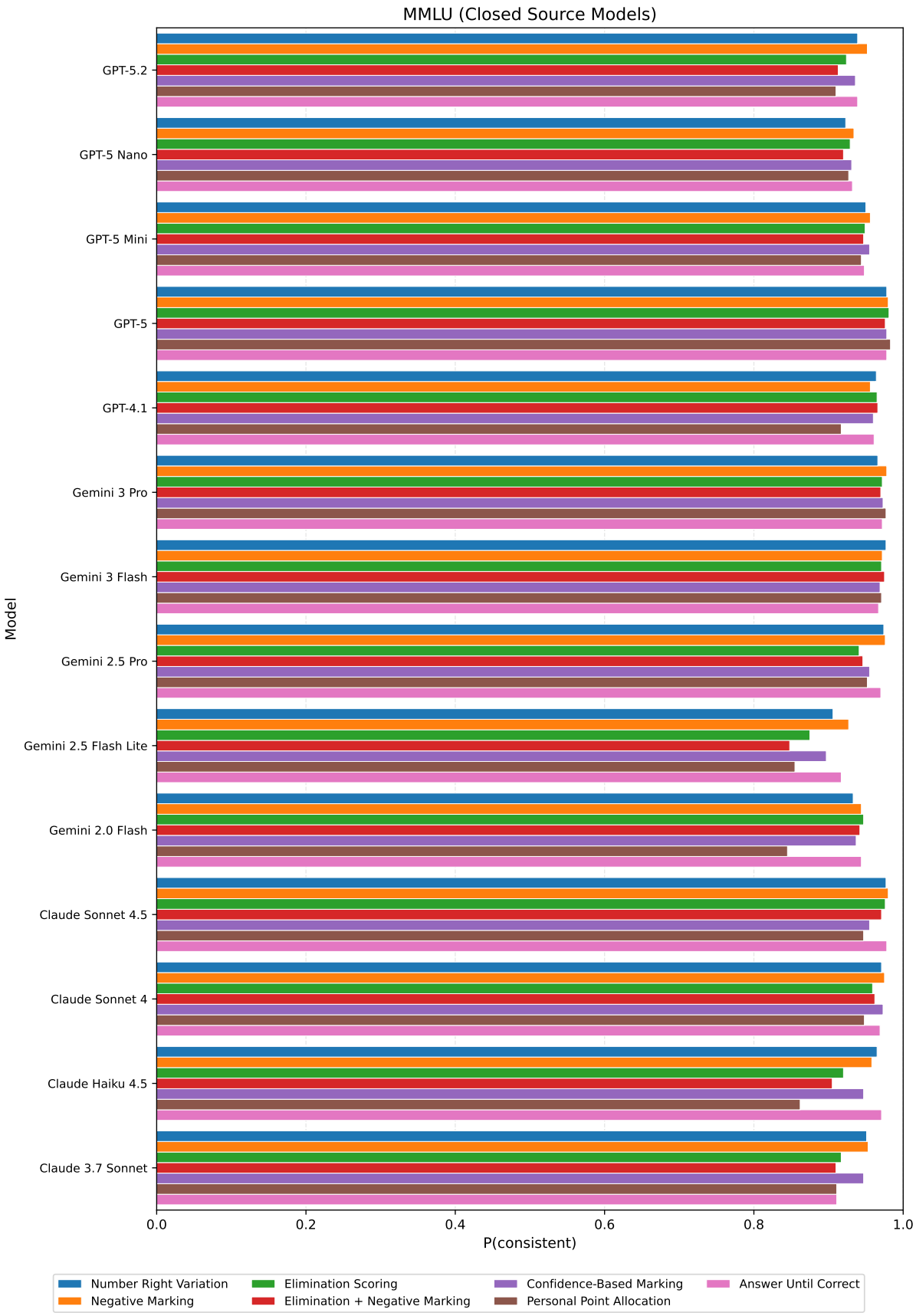

Figure 12: Consistency checks for closed-source models on MMLU between all scoring strategies.

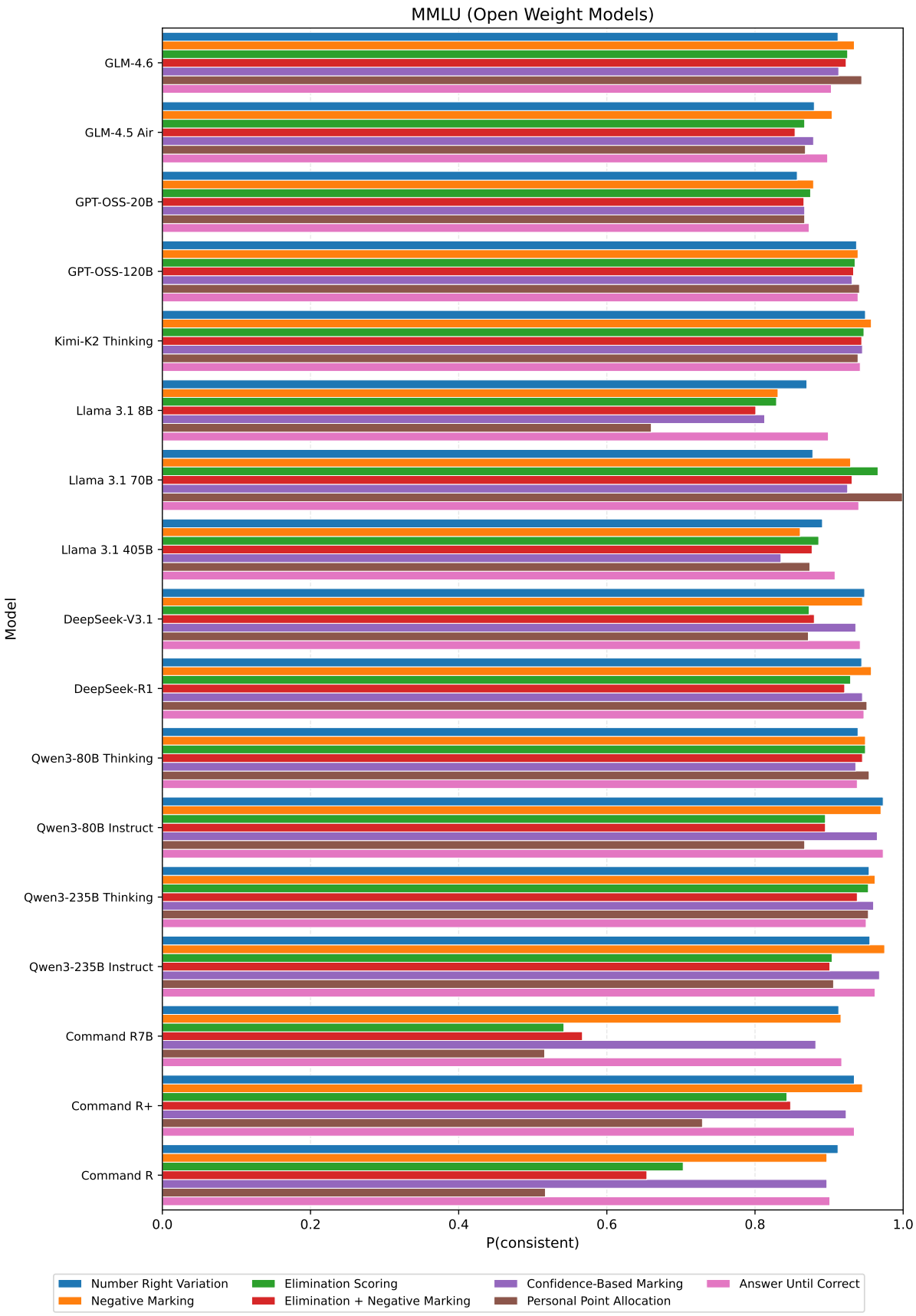

Figure 13: Consistency checks for open-weight models on MMLU between all scoring strategies.

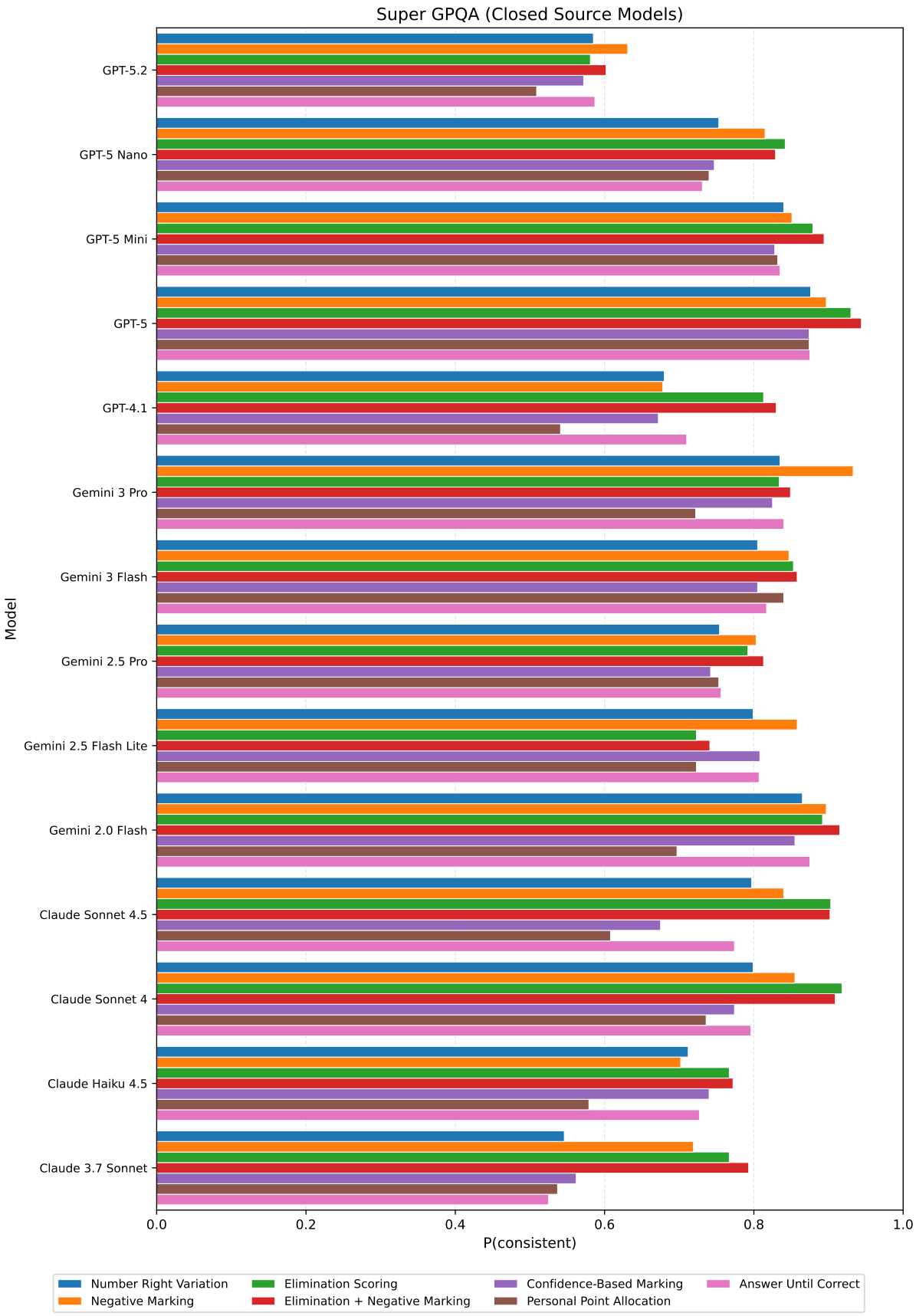

Figure 14: Consistency checks for closed-source models on SuperGPQA between all scoring strategies.

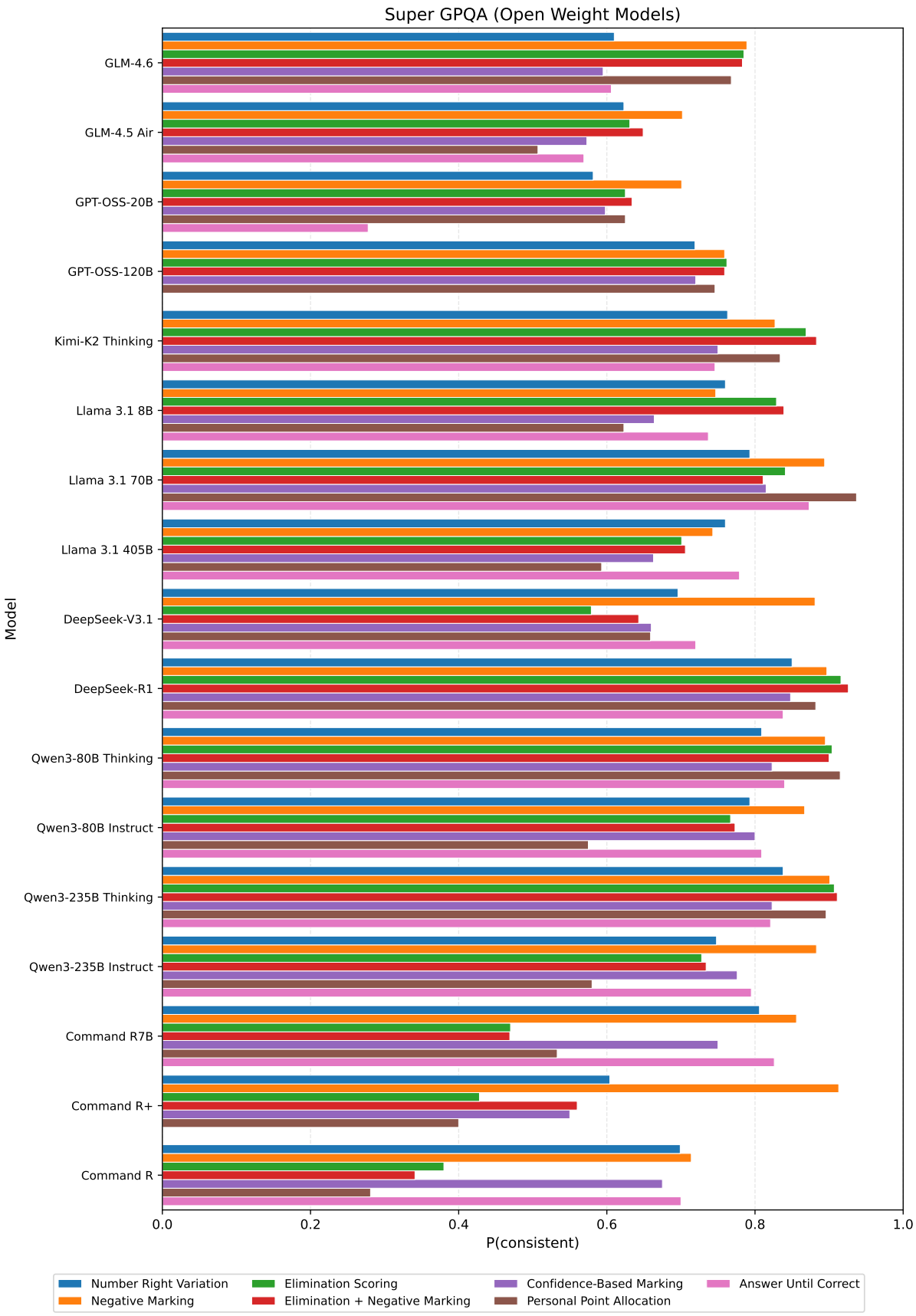

Figure 15: Consistency checks for open-weight models on SuperGPQA between all scoring strategies.

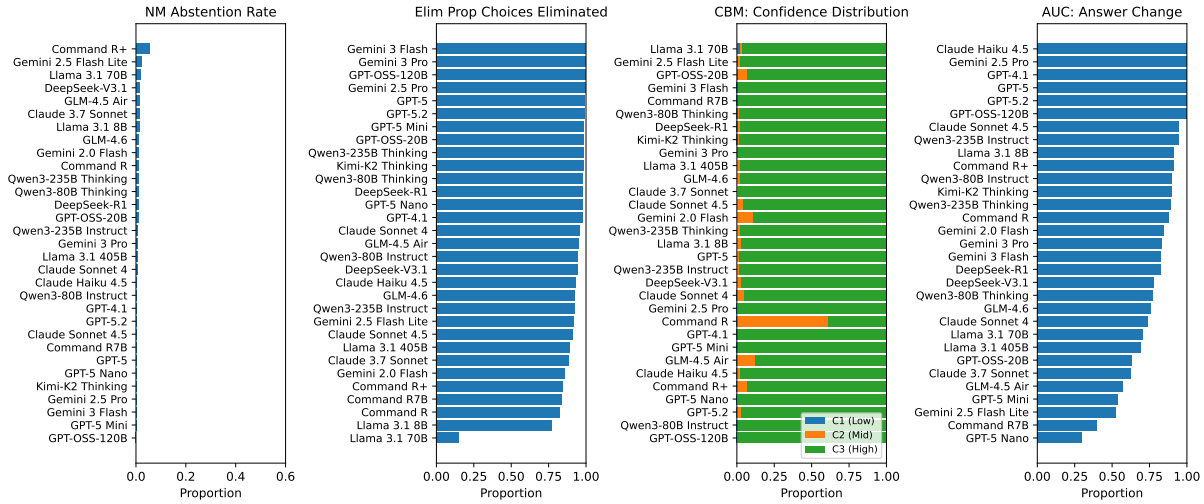

Figure 16: Model behavior analysis on ARC in Negative Marking, Elimination, Confidence-Based Marking, and Answer Until Correct scoring schemes.

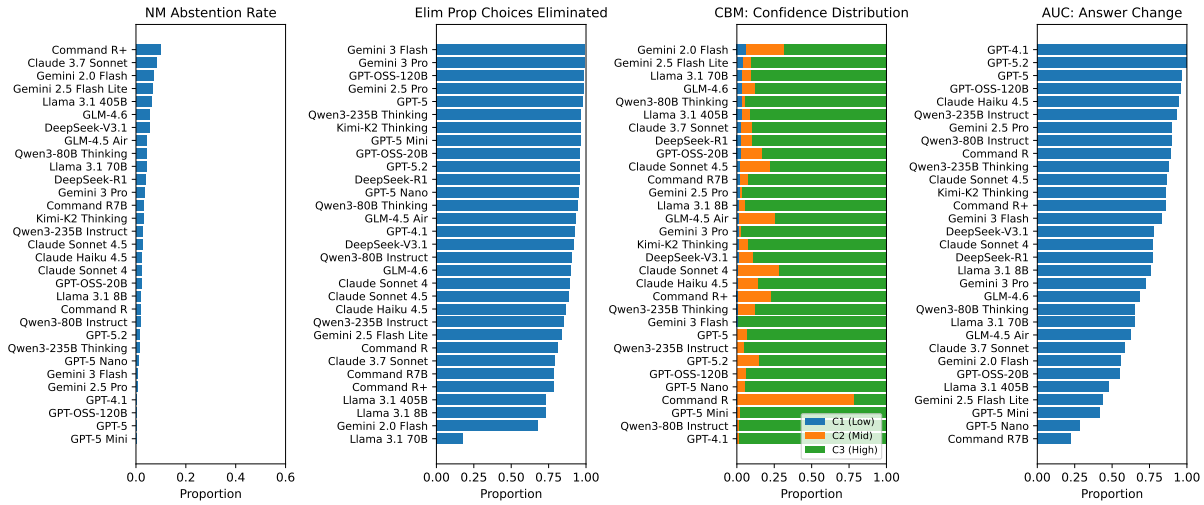

Figure 17: Model behavior analysis on MMLU in Negative Marking, Elimination, Confidence-Based Marking, and Answer Until Correct scoring schemes.

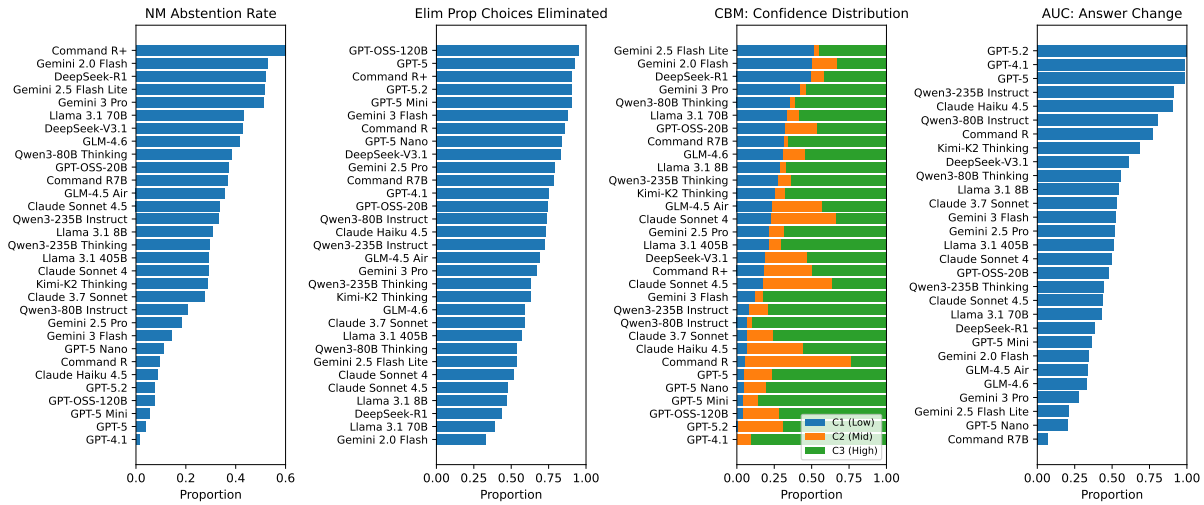

Figure 18: Model behavior analysis on Super GPQA in Negative Marking, Elimination, Confidence-Based Marking, and Answer Until Correct scoring schemes.

Prompt A.1: Number Right MCQA Prompt

<task>
Answer the following multiple choice question by selecting the letter of the correct answer:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
You will receive +1 point if your answer is correct, and 0 points if incorrect.
</scoring>

<format>
Generate your response as a JSON with two keys: 1) "answer" - the letter of the correct answer; and 2) "explanation" - an explanation for your decision. Do not generate anything else.
</format>

Prompt A.2: Number Right Variation Prompt

<task>
Please solve the following multiple choice question by choosing the correct option:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
Correct answers earn 1 point, while incorrect answers receive 0 points.
</scoring>

<format>
Provide your response as a JSON object containing two fields: 1) "answer" - the letter corresponding to your selected answer; and 2) "explanation" - a brief explanation of your reasoning. Do not generate anything else
</format>

Prompt A.3: Negative Marking Prompt

<task>
Answer the following multiple choice question by selecting the letter of the correct answer, or abstain if you are not confident:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
You will receive +1 point if your answer is correct, -1/(N-1) points if incorrect (where N is the total number of choices), and 0 points if you abstain.
</scoring>

<format>
Generate your response as a JSON with two keys: 1) "answer" - the letter of the correct answer or "ABSTAIN" if you choose to abstain; and 2) "explanation" - an explanation for your decision. Do not generate anything else
</format>

Prompt A.4: Elimination Prompt

<task>
Answer the following multiple choice question by eliminating the answers that you think are incorrect:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
You will receive +1 point for each incorrect answer you eliminate. However, if you eliminate the correct answer, you receive 0 points total (regardless of how many incorrect answers you eliminated). You may eliminate as many or as few answers as you want - there is no requirement to eliminate all answers or any specific number.
</scoring>

<format>
Generate your response as a JSON with two keys: 1) "eliminated" - a list of letters corresponding to the answers you have eliminated (e.g., ["A", "C", "D"]); and 2) "explanation" - an explanation for your eliminations. The "eliminated" list can contain any number of choices (including zero or all choices). Do not generate anything else
</format>

Prompt A.5: Elimination with Negative Marking Prompt

<task>
Answer the following multiple choice question by eliminating the answers that you think are incorrect:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
You will receive +1 point for each incorrect answer you eliminate, and -N points for each correct answer you eliminate (where N is the total number of choices). You may eliminate as many or as few answers as you want - there is no requirement to eliminate all answers or any specific number.
</scoring>

<format>
Generate your response as a JSON with two keys: 1) "eliminated" - a list of letters corresponding to the answers you have eliminated (e.g., ["A", "C", "D"]); and 2) "explanation" - an explanation for your eliminations. The "eliminated" list can contain any number of choices (including zero or all choices). Do not generate anything else
</format>

Prompt A.6: Confidence Marking Prompt

<task>
Answer the following multiple choice question by selecting the letter of the correct answer along with your confidence level:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
You will be scored using Confidence-Based Marking with discrete confidence levels. You must declare your confidence level as C=1 (low), C=2 (mid), or C=3 (high):
- C=1 (low confidence): +1 mark if correct, 0 marks (no penalty) if wrong
- C=2 (mid confidence): +2 marks if correct, -2 marks if wrong
- C=3 (high confidence): +3 marks if correct, -6 marks if wrong
Higher confidence levels yield greater rewards for correct answers but also incur greater penalties for incorrect answers. Choose your confidence level carefully based on how certain you are about your answer.
</scoring>

<format>
Generate your response as a JSON with three keys: 1) "answer" - the letter of the correct answer; 2) "confidence" - an integer confidence level of 1 (low), 2 (mid), or 3 (high); and 3) "explanation" - an explanation for your decision. Do not generate anything else
</format>

Prompt A.7: Personal Point Prompt

<task> Answer the following multiple choice question by allocating confidence scores (weights) to each choice, where all weights should sum to 1.0:
Question: q
Choices: $\mathcal{C}$
</task>

<scoring>
You will be scored based on how well your probability distribution matches the true outcome. Your score will be highest when you assign probability 1.0 to the correct answer and 0.0 to all incorrect answers. The more weight you allocate to the correct answer, the higher your score will be. For example, if the correct answer is "B", allocating "A": 0.2, "B": 0.6, "C": 0.15, "D": 0.05 will yield a better score than "A": 0.4, "B": 0.3, "C": 0.2, "D": 0.1 because more weight is placed on the correct answer "B".
</scoring>

<format>
Generate your response as a JSON with two keys: 1) "allocations" - a dictionary mapping each choice letter to its confidence weight (e.g., "A": 0.5, "B": 0.3, "C": 0.15, "D": 0.05); and 2) "explanation" - an explanation for your allocations. Do not generate anything else
</format>

Prompt A.8: Answer Until Correct

<task>
Answer the following multiple choice question by selecting the letter of the correct answer:
Question: q
Choices: $\mathcal{C}$
</task>

<feedback>
You have already tried answering with the following choices, which were incorrect: $\mathcal{A}'$. Do not select any of these choices as the correct answer.
</feedback>

<scoring>
You will have multiple attempts to answer correctly. Your score is calculated as (num_remaining - 1) / (N - 1), where num_remaining is the number of choices remaining after your attempts and N is the total number of choices. Answering correctly on your first attempt gives you the maximum score of 1.0. The more attempts you need, the lower your score.
</scoring>

<format>
Generate your response as a JSON with two keys: 1) "answer" - the letter of the correct answer; and 2) "explanation" - an explanation for your decision. Do not generate anything else.
</format>