how to represent code indentations, graph structures, tables, and other aspects of the CIs that did not have immediately clear text representations. We made the following key decisions:

- Line breaks in code are indicated with a newline character.
- One level of indentation is represented via four space characters.
- Questions/answers in table format are represented by listing each row individually, with the column title preceding each individual entry.
- Any tree diagrams in the questions are represented with textual descriptions.
- Questions with multiple correct answers (specifically, two questions on the BDSI) are excluded, as HELM does not currently support this answer format.

A more detailed document outlining the design decisions made for each CI is available as supplemental material to this paper.

We attempted to prevent LLM creators from using CIs to train future models but largely failed to do so. The only LLM developer that provided the capabilities to restrict data usage at the time of this study was OpenAI. However, this required an Enterprise account, which we did not have. We did avoid publicly sharing the CI questions and answers as a published benchmark. This mitigated the risk of test data leakage (CI questions and solutions cannot be scraped from the web), but did go against typical practices of publishing benchmarks for others to interrogate and use [9]. We obtained verification from SCS1 and BDSI maintainers that we had permission to use their CIs prior to running any CI questions through an LLM.

Finally, in order to enable few-shot runs of the model, we developed a set of in-context learning examples [24]: 5 for the SCS1 and 3 for the BDSI. These examples are not used to fine tune models for better performance; rather, they assist HELM in familiarizing itself with the correct input-output structure for evaluation items. This helps reduce situations where an LLM response is incorrectly evaluated due to a minor variation in the desired output format.

We chose to evaluate the following 10 LLMs created by four organizations:

(1) Anthropic Claude v1.3, Anthropic Claude 2.0, and Anthropic Claude 2.1 (by Anthropic)
(2) GPT-3.5 Turbo (0613) and GPT-4 (0613) (by OpenAI)
(3) Llama 2 (7B), Llama 2 (13B), and Llama 2 (70B) (by Meta)
(4) Mistral v0.1 (7B) and Mixtral (8x7B 32K seqlen) (by Mistral AI)

These models reflect a subset of models from HELM Lite [62] that were available at the time.

3.3 Analysis

3.3.1 Quantitative Analysis: Psychometric Properties to Compare LLM and Student Performance. We used difficulty and discrimination parameters defined in prior work ([86, 112]) to fit the SCS1 and BDSI to separate 2PL models. We then checked whether LLM response patterns were consistent with each 2PL model. We checked the person-fit statistic l_z, a standardization of the test-taker log likelihood function L to address the interaction of $ln(L)$ and θ [19, 25, 45]. l_Z is standardized, so a value of 0 denotes a perfectly expected or typical response pattern. Values above 2.0 could indicate overfitting (unexpectedly good fit) and below -2.0 could indicate noisy or unexpectedly poor fitting [45]. If the person-fit statistic was acceptable ($|l_z| < 2.0$), then we reported the latent knowledge level θ of each LLM run, effectively treating each one as independent test-takers and answering our first research question. θ is normalized, with 0 denoting an "average" test-taker, > 0 denoting an above average test-taker, and < 0 denoting a below average test-taker [19].

3.3.2 Qualitative Analysis: Informal Expert Panel Review. We identified unusual items in which LLM performance on these items deviated from expected student performance. To calculate expected student performance, we used the difficulty parameter (α) from the 2PL model for each item. α is effectively a z-score [19], with the proportion of the normal curve above that value representing the proportion of students with a $\geq$ 50% expected probability of getting the item correct. We then compared that percentage to the proportion of LLMs that got an item correct.

We considered two kinds of unusual items: those with high difficulty and those with low difficulty. Unusual high difficulty items are those with the greatest difficulty that had a greater proportion of LLM runs getting them correct when compared to the expected proportion of students. Therefore, unusual high difficulty items are those in which LLMs performed unexpectedly well. Unusual low difficulty items are those with the lowest difficulty in which the proportion of LLM runs getting them correct is less than the expected proportion of students getting it correct. Therefore, unusual low difficulty items are those in which LLMs performed unexpectedly poorly. We considered at most three high difficulty and low difficulty items that fulfilled this criteria.

Three authors participated in reviewing unusual items. These authors had experience in computing education research (three authors had collectively published over 12 papers to computing education research venues), teaching higher education computing courses (two authors had served as instructors; all four had experience as teaching assistants), psychometrics (one author had previously published multiple papers related to educational statistics and assessment design, and one author had developed materials for machine learning courses taught internationally), and LLMs (one author had published multiple papers on benchmarking and had industry experience developing deep learning algorithms).

The goal of this informal expert panel review [54] was to understand how LLM design, assessment design, and/or computing knowledge may explain deviations in LLM and expected test-taker performances. For each unusual question, experts considered whether the original question design, computing concept it assessed, or an aspect of the LLM design may have resulted in performance that differed from the expected question difficulty. We also considered whether the prompt structure could have been a confound. A previous psychometric evaluation of the SCS1 [112] included item trace plots as supplementary material [111]. These plots show the expected probability of selecting each multiple choice option for learners of varying knowledge levels. We used these plots to determine whether trends in incorrect LLM responses aligned with common student misconceptions.

4 RESULTS

4.1 RQ1: LLM Performance

Table 1 shows the results of person-fit statistics, with it and Figure 1 showing the knowledge level estimates for each LLM run with an acceptable fit. For the SCS1, we found that the 2PL model poorly fit the response patterns for Anthropic Claude v1.3 (zero and few shot), Llama 2 (7B) (zero and few shot), and Llama 2 (70B) (zero shot). The remaining models also had response patterns that reflected below average CS1 students, ranging from Llama 2 (7B) with few shot prompting ($\theta = -2.52 \pm 0.58$) to GPT-4 (0613) with zero shot prompting ($\theta = -0.33 \pm 0.22$).

Only five of the 20 LLM runs had responses that fit the 2PL model for the BDSI, with all other models being too noisy and poor of fits ($l_z < -2.0$). The LLMs with acceptable person-fit for the BDSI were all three Anthropic Claude models with few shot prompting and GPT-4 (0613) with zero and few shot prompting. These five instances produced responses that reflected CS2 students with data structures knowledge ranging from approximately average (GPT-4 (0613) with zero shot, $\theta = 0.12 \pm 0.44$ to above average (Anthropic Claude 2.0 with few shot, $\theta = 0.85 \pm 0.51$).

Table 1. Person fit statistics and knowledge estimates for LLMs with different prompting for the SCS1 and BDSI concept inventories. l_z is a person-fit statistic, with ** denoting unacceptable fit ($|l_z| > 2.0$). θ and standard error denote knowledge estimates of each LLM with an acceptable person fit.

		SCS1 (24 questions)			BDSI (11 questions)		
model name	prompting	l_z	θ	std. error	l_z	θ	std. error
claude-v1.3	zero shot	-2.31**	-	-	-4.05**	-	-
claude-v1.3	few shot	-2.79**	-	-	-0.76	0.66	0.48
claude-2.0	zero shot	-1.01	-1.06	0.32	-2.14**	-	-
claude-2.0	few shot	-1.13	-0.91	0.30	-1.59	0.46	0.46
claude-2.1	zero shot	-0.99	-1.01	0.31	-2.65**	-	-
claude-2.1	few shot	-0.36	-0.72	0.27	-0.31	0.85	0.51
gpt-3.5-turbo-0613	zero shot	-1.06	-1.45	0.39	-2.26**	-	-
gpt-3.5-turbo-0613	few shot	-0.26	-0.81	0.28	-2.26**	-	-
gpt-4-0613	zero shot	1.80	-0.33	0.22	-0.46	0.12	0.44
gpt-4-0613	few shot	0.62	-0.34	0.22	-0.31	0.13	0.44
llama-2-7b	zero shot	0.55	-2.52	0.58	-5.12**	-	-
llama-2-7b	few shot	-2.95**	-	-	-5.12**	-	-
llama-2-13b	zero shot	-2.34**	-	-	-4.35**	-	-
llama-2-13b	few shot	-0.71	-0.91	0.30	-4.91**	-	-
llama-2-70b	zero shot	-3.25**	-	-	-2.70**	-	-
llama-2-70b	few shot	-0.80	-1.11	0.33	-3.93**	-	-
mistral-7b-v0.1	zero shot	-0.42	-1.16	0.34	-4.50**	-	-
mistral-7b-v0.1	few shot	-0.58	-1.08	0.33	-2.26**	-	-
mixtral-8x7b-32kseqlen	zero shot	-1.34	-0.80	0.28	-3.09**	-	-
mixtral-8x7b-32kseqlen	few shot	-1.82	-0.61	0.25	-2.7**	-	-

Table 2. Number and percentage of invalid responses for LLM runs with zero and few shot in-context learning for each CI. Percentages calculated from 110 responses for each learning context for BDSI, and 240 responses for the SCS1.

	zero shot	few shot
BDSI	20 (18%)	0 (0%)
SCS1	46 (19%)	4 (2%)

Four runs produced responses that fit the 2PL models for both concept inventories: Anthropic Claude 2.0 and Anthropic Claude 2.1with few shot learning and GPT-4 (0613) with zero and few shot learning. In all four runs, response patterns for the SCS1 reflected below average CS1 knowledge ($\theta < 0$) but approximately average to above average knowledge of data structures ($\theta \geq 0$), typically considered a more advanced CS2 concept.

4.1.1 Validity Check: Number of Invalid Responses. LLM runs could still output invalid responses (e.g. a new line or word), which HELM would score as incorrect. We therefore attempted to correct for this using few shot in-context learning, as described in section 3.2.

Table 2 shows the frequency of invalid responses outputted from zero shot and few shot learning runs for each concept inventory. With zero-shot learning, we see that about 1 in 5 outputs are invalid. However, incorporating examples for few-shot learning resulted in no invalid responses for the BDSI and only 2% invalid responses for the SCS1 across all 10 LLMs. This suggests that for LLM runs for few-shot learning a valid answer is almost always outputted.

[80] Greg L Nelson and Amy J Ko. 2018. On Use of Theory in Computing Education Research. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*. ACM.

[81] Eng Lieh Ouh, Benjamin Kok Siew Gan, Kyong Jin Shim, and Swavek Wlodkowski. 2023. ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 54–60. https://doi.org/10.1145/3587102.3588794

[82] Miranda C Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, Validation, and Use of a Language Independent CS1 Knowledge Assessment. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16)*. ACM, New York, NY, USA, 93–101. https://doi.org/10.1145/2960310.2960316

[83] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 2086–2105. https://doi.org/10.18653/v1/2022.findings-acl.165

[84] Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite Task-Completion Dialogue Policy Learning via Hierarchical Deep Reinforcement Learning. (2017). https://doi.org/10.18653/v1/d17-1237

[85] Gustavo Pinto, Isadora Cardoso-Pereira, Danilo Monteiro, Danilo Lucena, Alberto Souza, and Kiev Gama. 2023. Large Language Models for Education: Grading Open-Ended Questions Using ChatGPT. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering* (Campo Grande, Brazil) *(SBES '23)*. Association for Computing Machinery, New York, NY, USA, 293–302. https://doi.org/10.1145/3613372.3614197

[86] Leo Porter, Daniel Zingaro, Soohyun Nam Liao, Cynthia Taylor, Kevin C Webb, Cynthia Lee, and Michael Clancy. 2019. BDSI: A Validated Concept Inventory for Basic Data Structures. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 111–119. https://doi.org/10.1145/3291279.3339404

[87] James Prather, Paul Denny, Juho Leinonen, Brett A Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, Stephen MacNeil, Andrew Petersen, Raymond Pettit, Brent N Reeves, and Jaromir Savelka. 2023. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education* (Turku, Finland) *(ITiCSE-WGR '23)*. Association for Computing Machinery, New York, NY, USA, 108–159. https://doi.org/10.1145/3623762.3633499

[88] Arif Rachmatullah, Bita Akram, Danielle Boulden, Bradford Mott, Kristy Boyer, James Lester, and Eric Wiebe. 2020. Development and validation of the middle grades computer science concept inventory (MG-CSCI) assessment. *EURASIA Journal of Mathematics, Science and Technology Education* 16, 5 (2020), em1841.

[89] Inioluwa Deborah Raji, Emily M Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. 2021. AI and the Everything in the Whole Wide World Benchmark. (Nov. 2021). arXiv:2111.15366 [cs.LG]

[90] Sam Saarinen, Shriram Krishnamurthi, Kathi Fisler, and Preston Tunnell Wilson. 2019. Harnessing the Wisdom of the Classes: Classsourcing and Machine Learning for Assessment Instrument Generation. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 606–612. https://doi.org/10.1145/3287324.3287504

[91] Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. (Oct. 2023). arXiv:2310.18018 [cs.CL]

[92] Eddie Antonio Santos, Prajish Prasad, and Brett A. Becker. 2023. Always Provide Context: The Effects of Code Context on Programming Error Message Enhancement. In *Proceedings of the ACM Conference on Global Computing Education Vol 1* (Hyderabad, India) *(CompEd 2023)*. Association for Computing Machinery, New York, NY, USA, 147–153. https://doi.org/10.1145/3576882.3617909

[93] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1* (Lugano and Virtual Event, Switzerland) *(ICER '22, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 27–43. https://doi.org/10.1145/3501385.3543957

[94] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1* (Lugano and Virtual Event, Switzerland) *(ICER '22)*. Association for Computing Machinery, New York, NY, USA, 27–43. https://doi.org/10.1145/3501385.3543957

[95] Andreas Säuberli and Simon Clematide. 2024. Automatic Generation and Evaluation of Reading Comprehension Test Items with Large Language Models. (April 2024). arXiv:2404.07720 [cs.CL]

[96] Jaromir Savelka, Arav Agarwal, Marshall An, Chris Bogart, and Majd Sakr. 2023. Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (Chicago, IL, USA,) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 78–92. https://doi.org/10.1145/3568813.3600142

[97] Jaromir Savelka, Arav Agarwal, Christopher Bogart, and Majd Sakr. 2023. Large Language Models (GPT) Struggle to Answer Multiple-Choice Questions about Code. (March 2023). arXiv:2303.08033 [cs.CL]

[98] Jaromir Savelka, Arav Agarwal, Christopher Bogart, Yifan Song, and Majd Sakr. 2023. Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses? (March 2023). arXiv:2303.09325 [cs.AI]

[99] Stanford Center for Research on Foundation Models. 2022. Ecosystem Graphs for Foundation Models. https://crfm.stanford.edu/ecosystem-graphs/index.html?mode=table. Accessed: 2024-3-12.

[100] Andrew Taylor, Alexandra Vassar, Jake Renzella, and Hammond Pearce. 2024. dcc –help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations with Large Language Models. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, Oregon, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1314–1320. https://doi.org/10.1145/3626252.3630822

[101] Cynthia Taylor, Daniel Zingaro, Leo Porter, Kevin C Webb, Cynthia Bailey Lee, and Mike Clancy. 2014. Computer science concept inventories: past and future. *Computer Science Education* 24, 4 (2014), 253–276.

[102] Allison Elliott Tew and Mark Guzdial. 2011. The FCS1: a language independent assessment of CS1 knowledge. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) *(SIGCSE '11)*. Association for Computing Machinery, New York, NY, USA, 111–116. https://doi.org/10.1145/1953163.1953200

[103] The National Science Foundation and The Institute of Education Sciences. 2018. *Companion Guidelines on Replication & Reproducibility in Education Research*. Technical Report. NSF and IES.

[104] Jan Vahrenhold and Wolfgang Paul. 2014. Developing and validating test items for first-year computer science courses. *Computer Science Education* 24, 4 (2014), 304–333. https://doi.org/10.1080/08993408.2014.970782 arXiv:https://doi.org/10.1080/08993408.2014.970782

[105] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.* 53, 3, Article 63 (jun 2020), 34 pages. https://doi.org/10.1145/3386252

[106] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C. Schmidt. 2023. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. arXiv:2302.11382 [cs.SE]

[107] R. Paul Wiegand, Anthony Bucci, Amruth N. Kumar, Jennifer L. Albert, and Alessio Gaspar. 2016. A Data-Driven Analysis of Informatively Hard Concepts in Introductory Programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (Memphis, Tennessee, USA) *(SIGCSE '16)*. Association for Computing Machinery, New York, NY, USA, 370–375. https://doi.org/10.1145/2839509.2844629

[108] Ben Williamson. 2024. AI in education is a public problem. https://codeactsineducation.wordpress.com/2024/02/22/ai-in-education-is-a-public-problem/. Accessed: 2024-5-30.

[109] Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. *IEEE/CAA Journal of Automatica Sinica* 10, 5 (2023), 1122–1136. https://doi.org/10.1109/JAS.2023.123618

[110] Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anaïs Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch (Eds.). Association for Computational Linguistics, Toronto, Canada, 610–625. https://doi.org/10.18653/v1/2023.bea-1.52

[111] Benjamin Xie. 2019. Supplementary Info for "An Item Response Theory Evaluation of a Language-Independent CS1 Knowledge Assessment" (Xie et al. SIGCSE 2019). https://github.com/codeandcognition/archive-2019sigcse-xie. Accessed: 2024-1-15.

[112] Benjamin Xie, Matthew J Davidson, Min Li, and Amy J Ko. 2019. An Item Response Theory Evaluation of a Language-Independent CS1 Knowledge Assessment. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, 699–705. https://doi.org/10.1145/3287324.3287370

[113] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on Large Language Model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing* (March 2024), 100211. https://doi.org/10.1016/j.hcc.2024.100211

[114] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. (2023). arXiv:2303.18223 [cs.CL]

[115] Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don't Make Your LLM an Evaluation Benchmark Cheater. (Nov. 2023). arXiv:2311.01964 [cs.CL]