Challenges for AI in Multimodal STEM Assessments: a Human-AI Comparison

Aymeric de Chillaz[1*] **Anna Sotnikova**[1*†] **Patrick Jermann**[1] **Antoine Bosselut**[1]

[1]EPFL

Abstract

Generative AI systems have rapidly advanced, with multimodal input capabilities enabling reasoning beyond text-based tasks. In education, these advancements could influence assessment design and question answering, presenting both opportunities and challenges. To investigate these effects, we introduce a high-quality dataset of 201 university-level STEM questions, manually annotated with features such as image type, role, problem complexity, and question format. Our study analyzes how these features affect generative AI performance compared to students. We evaluate four model families with five prompting strategies, comparing results to the average of 546 student responses per question. Although the best model correctly answers on average 58.5% of the questions using majority vote aggregation, human participants consistently outperform AI on questions involving visual components. Interestingly, human performance remains stable across question features but varies by subject, whereas AI performance is susceptible to both subject matter and question features. Finally, we provide actionable insights for educators, demonstrating how question design can enhance academic integrity by leveraging features that challenge current AI systems without increasing the cognitive burden for students.

1 Introduction

Generative AI has been widely tested in educational applications, including its ability to answer exam-level questions (Sallam, 2023; Lan et al., 2024; Wang et al., 2024a). There are two key challenges: AI can be misused in ways that undermine fair assessment, and its mistakes often appear convincing, potentially misleading students (Borges et al., 2024; Wang et al., 2023; Zhong et al., 2023; Arora et al., 2023). To better understand these risks, benchmarks were introduced to assess AI performance (Wang et al., 2024b). Recent advances in multimodal large language models (LLMs) have led to extensive efforts in developing image-based exam datasets, particularly in STEM. Anand et al. (2024) introduced a multimodal physics dataset, expanding from 300 manually created questions to 4,500 using LLMs; Liang et al. (2024) developed SceMQA, a dataset of 1,000+ scientific reasoning problems for students transitioning to college; Zhang et al. (2023) and Das et al. (2024) created multilingual, multimodal benchmarks across various subjects and difficulty levels.

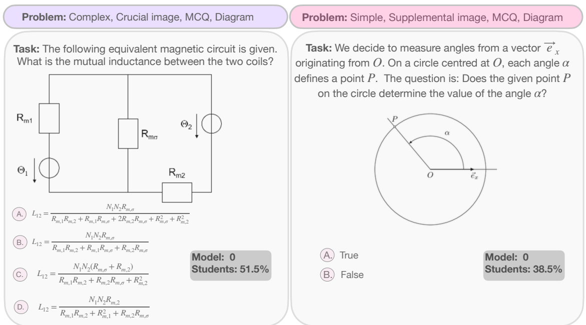


Figure 1: Example of STEM problems with average model performance (majority vote) compared to average student performance.

While these benchmarks provide insight into AI capabilities, they primarily evaluate models in isolation, without comparing their performance to humans. As a result, it is unclear whether a model's low performance stems from its limitations or if the problems themselves are inherently difficult for humans too. Understanding what makes a problem easier or harder for AI compared to humans would help warn students about potential risks and guide the design of fairer image-based assessments.

We compile 201 university-level STEM exam questions with images from Bachelor's and Master's programs across 11 subjects of varying complexity. To analyze model performance, each ques-

*Both authors contributed equally to this research.
†Corresponding author: aasotniko@gmail.com

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 279–293

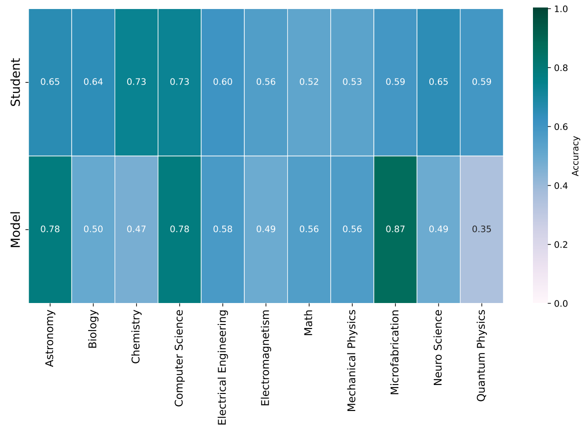


Figure 2: Average model and student accuracy per subject, with model results aggregated using the majority vote strategy.

tion is manually annotated with its image type, role, question type, and problem type. In addition, we collect student performance data, each question receiving at least five responses and an average of 546 respondents across the dataset. To evaluate AI performance, we implement five prompting strategies and test two models from GPT-family —`GPT-4o` and `o1-mini` (OpenAI, 2023) as performant models freely available to students, and `Qwen 2.5 72B VL` (Bai et al., 2025), `DeepSeek r1` (DeepSeek-AI et al., 2025), and `Claude 3.7 Sonnet, 2025` as performant models with visual capabilities.

Our results indicate that while LLMs perform well in text-based university assessments (Borges et al., 2024), they struggle with questions involving visual components. On average, models perform slightly worse than students. Student performance varies by subject, while model performance depends on question and image features. Based on the analysis, we provide recommendations for designing take-home assignments that maintain academic integrity by challenging models without increasing difficulty for students. These principles can also inform the development of more challenging benchmarks as models continue to improve.

2 Data Set Description

We manually collected 201 questions with images from exams and quizzes in 11 subjects from Bachelor's and Master's programs. Each question is paired with a gold answer provided by the educator who authored it. Questions were manually labeled with the following attributes[1]:

[1]The dataset is available on GitHub

Image Type: diagram, line plot, algorithm, and picture.
Image Purpose: An image is "supplemental" if all necessary information is in the text and can be inferred without it. It is "crucial" if required to solve the problem.
Question Type: multiple choice questions (MCQ), multiple choice questions multiple answers (MCQ-MA), and compound questions containing multiple sub-MCQ questions connected by the same question topic and having some related information in each other.
Complexity of Problem Conditions: "Complex" questions involve multiple subject concepts, while "Simple" ones require only one or two closely related concepts. This distinction does not indicate difficulty—a question may have one hard concept or multiple simple ones. Categorizing questions this way helped assess whether models struggled with interdependent conditions, as simple questions have fewer variables, while complex ones require integrating more information.

Student performance data was collected from historical course records as aggregated statistics, with 5 to 5,686 respondents per question (average: 546). Student performance also served as an indicator of problem difficulty. Our dataset includes 43 problems where fewer than 40% of the students answered correctly, 79 where 40–70% succeeded, and 79 where more than 70% solved the question.

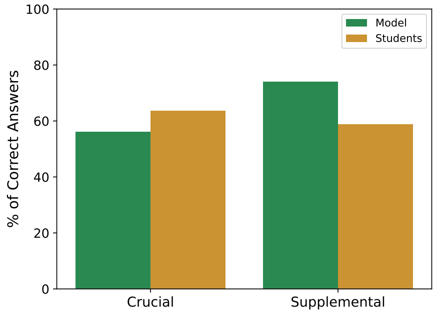
For detailed dataset statistics and student performance, see Appendices A.1 and A.3.

3 Experiments

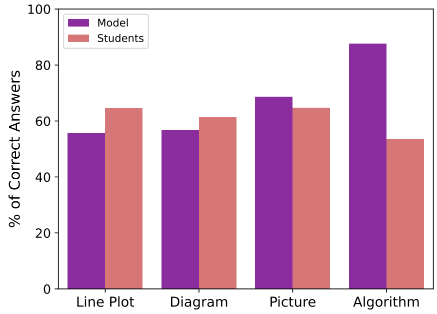
Our experiments assess model performance across five prompting strategies and compare it with human performance. Details on prompting strategies are provided in Appendix B. For multiple-choice (MCQ, MCQ-MA) and compound questions, we use exact match with the gold answer without partial credit. Model scores are aggregated using two methods: majority vote (assigning the most common score across strategies) and max (taking the highest achieved score). The max approach provides an upper bound estimate, highlighting if at least one strategy yields the correct answer. Model implementation details are in Appendix C.

4 Analysis

This section presents the experimental results, comparing the model and student performance across



(a) Image Property



(b) Image Type

Figure 3: (a) Effect of image properties on model performance aggregated by the majority vote strategy compared to average student performance. (b) Effect of image type on model performance aggregated by the majority vote strategy compared to average student performance.

various dimensions.

4.1 General performance on questions with images

Unless stated otherwise, we use majority vote aggregation. GPT-4o outperforms other models, and we focus on its results throughout. Detailed model comparisons are provided in Appendix D.1. As shown in Figure 9, all prompting strategies perform similarly and roughly match the average human student's performance on the task.

We analyze model and student accuracy on image-based questions across subjects (Figure 2). Both exhibit subject-specific strengths and weaknesses, but student accuracy varies less (0.52–0.73) than the model's (0.35–0.87). The model performs exceptionally well in Astronomy, Computer Science (CS), and Microfabrication, likely due to the structured nature of these questions and the model's ability to apply general concepts. Prior studies have shown that LLMs excel at CS-related tasks (Krüger and Gref, 2023; Song et al., 2024; Borges et al., 2024). In contrast, the model struggles with Quantum Physics, Chemistry, Neuroscience, and Electromagnetism, where complex, content-rich images may pose additional challenges.

4.2 Effect of image features

We examine the role of images in problem-solving, specifically whether they provide essential information absent from the text or if the problem can be solved without them. Figure 3a compares performance based on image necessity. As expected, student accuracy remains similar regardless of image importance, whereas models perform better on questions where images are non-essential. Our ablation study confirms this trend: removing supplemental images slightly improves model performance, though the effect is minimal (see Appendix D.2 for details).

Next, we analyze performance across image types (Figure 3b). Students perform similarly across line plots, diagrams, and pictures, and struggle the most with algorithm questions. Although the model has no difficulties in processing algorithms, it struggles the most with diagrams and line plots.

4.3 Effect of question features

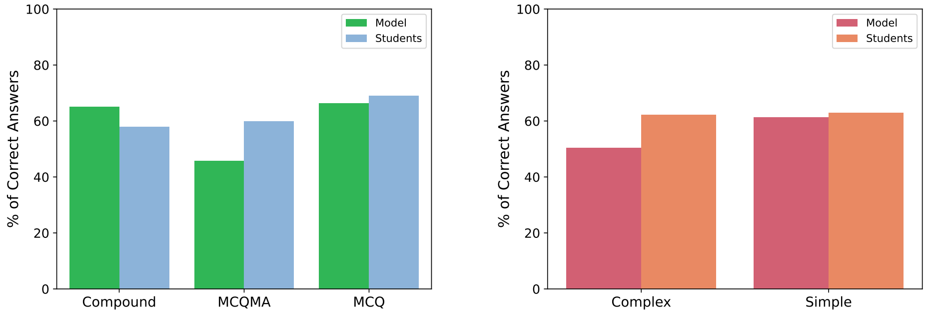
We observe that students perform similarly across all three question formats. Both students and the model get the best performance on MCQ questions. The model performs slightly better than students on compound questions, a subset of MCQs that are linked to represent steps of a larger problem. However, it struggles the most with MCQMA, often selecting some correct choices but failing to identify all (Figure 4a).

Figure 4b illustrates how the concept count influences performance. Although students perform consistently regardless of the number of concepts in a question, the model struggles when more than two concepts are involved.

We report statistical significance for the model's and students' results in Tables 5 and 6.

4.4 Error analysis

To assess the model's strengths and weaknesses, we analyzed 59 questions split into two sets: ones where the model outperformed students and ones where it underperformed.



(a) Question Type

(b) Problem Type

Figure 4: (a) Effect of question type on model performance aggregated by the majority vote strategy compared to average student performance. (b) Effect of problem type on model performance aggregated by the majority vote strategy compared to average student performance.

Questions easy for students, hard for the model We examined 31 questions where the model scored 0 using the max strategy—failing to produce a correct answer across five prompts—while students achieved over 40% accuracy. We find that humans more easily integrate common sense, domain-specific intuition, and experiential learning, whereas the model struggles to infer conditions or iterations that are not explicitly stated.

One notable category where students outperform the model involves physics-based reasoning and real-world conventions. These problems require understanding implicit relationships, precise numerical or symbolic extraction, and intuition-driven problem-solving. For instance, students effectively interpret diagrams, such as photonic crystal defects or force distributions in mechanical systems, while the model struggles with directional trends and recognizing constraints in visual data. Furthermore, the model has difficulty selecting the correct schema or plot from multiple options, a task that poses less challenge for humans.

Questions hard for students, easy for the model We analyze 28 questions where students' performance is below 40% while the model scores above 65%.

A key category where the model outperforms students includes problems requiring structured reasoning, precise pattern recognition, and large-scale knowledge retrieval. These problems follow well-defined rules, abstract mathematical principles, and algorithmic logic. The model's ability to detect structural patterns allows it to efficiently analyze periodicity in trigonometry, solve algorithmic network problems, and interpret simple electrical schematics with high accuracy. Unlike intuition-driven tasks, these problems follow clear logical steps. Students often struggle with multi-step reasoning due to cognitive load, whereas the model processes extended contexts effortlessly. As shown in Figure 10, student accuracy declines as question length increases, while the model maintains strong performance. Additionally, models excel in problems requiring abstraction and conceptual knowledge.

5 Conclusion

We show that questions requiring crucial images and multiple concepts, while remaining concise, pose a greater challenge for models without increasing difficulty for students. Additionally, models struggle more than humans in applying domain-specific intuition to problem-solving. However, our analysis reveals that models retain knowledge of the correct answer in 75.5% of questions across at least one prompting strategy but fail to retrieve it consistently. With a majority vote strategy, models achieve 58.5% accuracy, slightly below the human average of 62.7%. While overall performance appears similar, a closer analysis highlights the significant impact of the problem and image features on these results.

Finally, it is important to balance fair accessibility with preventing model misuse, as restrictive measures may inadvertently disadvantage students with vision impairments. For these students, problems with supplemental images are easier to understand through full-text descriptions, similar to how models rely on textual input over visual data.

Limitations

Our study explores how humans and models solve questions involving both images and text. However, it has several limitations.

First, our dataset is relatively small (201 examples). While we ensured high-quality data through manual collection and annotation and confirmed statistical significance, a larger dataset would improve reliability. We opted against automated data augmentation to maintain quality control. To facilitate further research, we publicly release our dataset with annotations.

Second, our grading method does not assign partial credit for multiple-choice multiple-answer (MCQMA) questions, leading to a stricter evaluation of model performance. Additionally, unlike humans, models do not employ elimination reasoning, as we do not adjust prompts for MCQMA responses, potentially disadvantaging them.

Third, when comparing course performance, we do not account for instructor influence, which may affect problem difficulty. This factor can introduce bias also for humans, as different instructors may present varying challenges for students within the same subject.

Acknowledgments

We are grateful to Dr. Jessica Dehler Zufferey for her valuable recommendations on question and image feature design, as well as her feedback on the interpretation of our results. We also thank Christian Vonarburg and Yves Renier for their assistance with data preparation. We appreciate the feedback provided by the EPFL NLP lab members throughout the development of this work. We also gratefully acknowledge the support of the Swiss National Science Foundation (No. 215390), Innosuisse (PFFS-21-29), the EPFL Center for Imaging, Sony Group Corporation, and a Meta LLM Evaluation Research Grant.

References

Avinash Anand, Janak Kapuriya, Apoorv Singh, Jay Saraf, Naman Lal, Astha Verma, Rushali Gupta, and Rajiv Shah. 2024. Mm-phyqa: Multimodal physics question-answering with multi-image cot prompting. *Preprint*, arXiv:2404.08704.

Daman Arora, Himanshu Gaurav Singh, and Mausam. 2023. Have llms advanced enough? a challenging problem solving benchmark for large language models. *Preprint*, arXiv:2305.15074.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Beatriz Borges, Negar Foroutan, Deniz Bayazit, Anna Sotnikova, Syrielle Montariol, Tanya Nazaretzky, Mohammadreza Banaei, Alireza Sakhaeirad, Philippe Servant, Seyed Parsa Neshaei, Jibril Frej, Angelika Romanou, Gail Weiss, Sepideh Mamooler, Zeming Chen, Simin Fan, Silin Gao, Mete Ismayilzada, Debjit Paul, Philippe Schwaller, Sacha Friedli, Patrick Jermann, Tanja Käser, Antoine Bosselut, EPFL Grader Consortium, and EPFL Data Consortium. 2024. Could chatgpt get an engineering degree? evaluating higher education vulnerability to ai assistants. *Proceedings of the National Academy of Sciences*, 121(49):e2414955121.

Rocktim Jyoti Das, Simeon Emilov Hristov, Haonan Li, Dimitar Iliyanov Dimitrov, Ivan Koychev, and Preslav Nakov. 2024. Exams-v: A multi-discipline multilingual multimodal exam benchmark for evaluating vision language models. *Preprint*, arXiv:2403.10378.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Tim Krüger and Michael Gref. 2023. Performance of large language models in a computer science degree program. *Preprint*, arXiv:2308.02432.

Yunshi Lan, Xinyuan Li, Hanyue Du, Xuesong Lu, Ming Gao, Weining Qian, and Aoying Zhou. 2024. Survey of natural language processing for education: Taxonomy, systematic review, and future trends. *Preprint*, arXiv:2401.07518.

Zhenwen Liang, Kehan Guo, Gang Liu, Taicheng Guo, Yujun Zhou, Tianyu Yang, Jiajun Jiao, Renjie Pi, Jipeng Zhang, and Xiangliang Zhang. 2024. Scemqa: A scientific college entrance level multimodal question answering benchmark. *Preprint*, arXiv:2402.05138.

OpenAI. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Malik Sallam. 2023. Chatgpt utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns. *Healthcare*, 11(6).

Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, Weihao Zeng, Yejie Wang, Zhuoma GongQue, Jianing Yu, Qiuna Tan, and Weiran Xu. 2024. Cs-bench: A comprehensive benchmark for large language models towards computer science mastery. *Preprint*, arXiv:2406.08587.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024a. Large language models for education: A survey and outlook. *Preprint*, arXiv:2403.18105.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *Preprint*, arXiv:2307.10635.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024b. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *Preprint*, arXiv:2307.10635.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Preprint*, arXiv:2306.05179.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *Preprint*, arXiv:2304.06364.

A Data Set Details

Here we present the data set statistics, and examples of questions for every data feature.

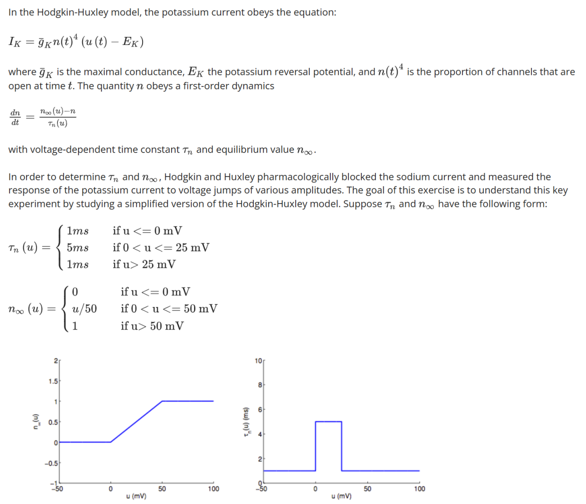
A.1 Data Format

Image Type: diagram, line plot, algorithm, and picture.

Image Purpose: supplemental, when the image is non-essential and all information about the problem is stated in the problem text or crucial, when the image is required to solve the problem. One can determine that the question in Figure 5 has a supplemental image since it could be inferred from the text.

Question Type: multiple choice questions (MCQ), multiple choice questions multiple answers (MCQ-MA), and compound questions containing multiple sub-MCQ questions connected by the same question topic and having some related information in each other.

Complexity of Problem Conditions: “Complex” means that the question involves multiple concepts of the subject, while “Simple” would require only one or two closely related concepts to solve the problem. This condition does not directly reflect problem difficulty; a question may involve a single difficult concept or multiple simple ones. Distinguishing between simple and complex questions



In the Hodgkin-Huxley model, the potassium current obeys the equation:

$$I_K = \bar{g}_K n(t)^4 \left(u\left(t\right) - E_K\right)$$

where $\bar{g}_K$ is the maximal conductance, E_K the potassium reversal potential, and $n(t)^4$ is the proportion of channels that are open at time t. The quantity n obeys a first-order dynamics

$$\frac{dn}{dt} = \frac{n_\infty(u) - n}{\tau_n(u)}$$

with voltage-dependent time constant τ_n and equilibrium value n_∞.

In order to determine τ_n and n_∞, Hodgkin and Huxley pharmacologically blocked the sodium current and measured the response of the potassium current to voltage jumps of various amplitudes. The goal of this exercise is to understand this key experiment by studying a simplified version of the Hodgkin-Huxley model. Suppose τ_n and n_∞ have the following form:

$$\tau_n\left(u\right) = \begin{cases} 1ms & \text{if u} <= 0 \text{ mV} \\ 5ms & \text{if } 0 < \text{u} <= 25 \text{ mV} \\ 1ms & \text{if u} > 25 \text{ mV} \end{cases}$$

$$n_\infty\left(u\right) = \begin{cases} 0 & \text{if u} <= 0 \text{ mV} \\ u/50 & \text{if } 0 < \text{u} <= 50 \text{ mV} \\ 1 & \text{if u} > 50 \text{ mV} \end{cases}$$

Figure 5: Example of a complex question with supplemental image.

allowed us to evaluate whether models struggled with interdependent conditions. Simple questions involve fewer variables, while complex ones require integrating multiple pieces of information.

One can determine that the question in Figure 5 is “Complex” and the image is “Supplemental”. The question is complex because it involves the Hodgkin-Huxley model, differential equations governing potassium channel dynamics, and voltage-dependent parameters, requiring knowledge of electrophysiology and mathematical modeling. The image is supplemental because it provides graphical representations of $n_\infty(u)$ and $\tau_n(u)$, but all necessary equations and definitions are clearly described in the text, making the image helpful but not essential.

A.1.1 Description of Labels

1. **Course_name**:
   - *Description*: The name or identifier of the course associated with the question.
   - *Example*: `"Calculus I"`, `"Physics 101"`
2. **Exercise_name**:
   - *Description*: The unique exercise id.
3. **Question**:
   - *Description*: The text of the question, may include LaTeX formatting and placeholders for images.
4. **Gold_answer**:
   - *Description*: The correct answer to the question.
5. **Question_type**:
   - *Description*: The format or type of the question.
   - *Possible Labels*:
     - `"MCQ"` (Multiple Choice Question)
     - `"MCQMA"` (MCQ Multiple Answers)
     - `"Compound"` (a non-open-ended question with multiple objectives)
6. **Image_type**:
   - *Description*: The type of images included in the question.
   - *Possible Labels (Others may be added as we manually label)*:
     - `"line plot"`
     - `"bar plot"`
     - `"scatter plot"`
     - `"histogram"`
     - `"pie chart"`
     - `"table"`
     - `"image"`
     - `"diagram"`
7. **Image_purpose**:
   - *Description*: The role of the image in the context of the question.
   - *Possible Labels*:
     - `"Crucial"` (Essential for solving the question)
     - `"Supplemental"` (Doesn’t provide additional context)
8. **Problem_conditions**:
   - *Description*: The complexity of the conditions within the problem.
   - `"Complex"` doesn’t necessarily mean that the problem is difficult. It simply means that many conditions are in play.
   - *Possible Labels*:
     - `"Simple"` (Conditions are straightforward and not interacting)
     - `"Complex"` (Multiple conditions interact to find the answer)
9. **Question_images**:
   - *Description*: A list of filenames or identifiers for images included in the question.

10. **Question_length_characters**:
    - *Description*: The length of the question text is measured in characters.
11. **Num_objectives**:
    - *Description*: The number of sub-questions within the question.
    - *Example*: `1`, `2`
12. **Language**:
    - *Description*: The language in which the question is written.
    - Always in `"English"` because we translated the French ones.
13. **Original_language**:
    - *Description*: The original language of the question before translation.
    - *Example*: `"French"`
14. **Was_translated**:
    - *Description*: Indicates whether the question was translated from another language.
    - *Possible Values*: `true` or `false`
15. **Image_file_type**:
    - *Description*: The file format of the images used.
    - *Example*: `"PNG"`, `"JPEG"`
16. **Answer_format**:
    - *Description*: The expected format of the answer.
    - *Possible Labels*:
      - `"Only MCQ Letter"` (previously called MCQ)
      - `"Only Numeric Answer"`
      - `"Derivation"`
      - `"Text"`
      - `"Code"`
      - `"Calculation"`
17. **Solution_type**:
    - *Description*: Indicates whether the question has a unique correct answer or multiple correct answers.
    - *Possible Labels*:
      - `"Unique answer"`
      - `"Multiple answers"`
18. **Type_of_text**:
    - *Description*: The formatting or typesetting used in the question text.
    - *Example*: `"LaTeX"`, `"Plain text"`, `"XML"`
19. **Objective_dependency**:
    - *Description*: Indicates whether the objectives in the question are independent or dependent on previous ones.
    - *Possible Labels*:
      - `"All Independent"` (Objectives can be solved separately)
      - `"Dependent"` (Some objectives rely on answers from previous parts)

A.2 Data Set Statistics

Data set statistic is presented in Table 1.

Feature	Options	Count
Question Type	MCQ	80
	Compound	59
	MCQMA	46
	Numeric and Formula	16
Image Type	Diagram	109
	Line Plot	45
	Picture	33
	Algorithm	8
	Other	6
Image Purpose	Crucial	162
	Supplemental	39
Problem Conditions	Simple	169
	Complex	32
Course Category	Astronomy	32
	Electrical Engineering	28
	Computer Science	20
	Math	19
	Electromagnetism	17
	Quantum Physics	26
	Mechanical Physics	16
	Neuroscience	15
	Microfabrication	10
	Chemistry	10
	Biology	8

Table 1: Data Statistics

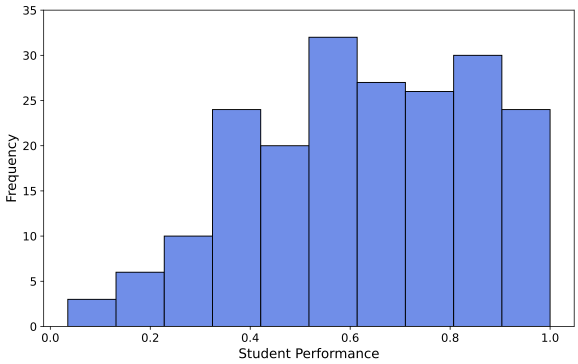


Figure 6: Student accuracy distribution.

A.3 Student Performance

Dataset difficulty illustrated by student performance is presented in Figure 6.

The distribution of students attempting a question is presented in Table 2.

Respondents	Questions
5-20	27
21-50	34
51-100	29
101-500	51
501-1000	33
1001-2000	21
2001-7000	6

Table 2: Distribution of questions by the number of respondents.

B Prompting strategies

B.1 Helper Functions

B.1.1 `question_type_prompt`

The `question_type_prompt` function creates a tailored instruction based on the type of question being posed. It supports several question types, each associated with a specific directive:

- **MCQ:** Instructs the model to select the correct option by returning only its letter.
- **MCQMA:** Similar to MCQ but expects multiple correct options, concatenated as a single string (e.g., `AB` rather than `A, B`).
- **Numeric Question:** Requests that the model output only the numerical answer.
- **Formula Question:** Expects the answer to be provided as a formula.
- **Open Ended:** Directs the model to comprehensively address all parts of the question.

For questions labeled as **Compound**, the function combines the individual instructions corresponding to each subquestion type. It first determines the number of subquestions and then appends the respective prompt text for each, ultimately guiding the model to return its answers as a JSON-formatted list.

B.1.2 `generate_format_instruction`

The `generate_format_instruction` function provides context-specific formatting advice based on the text's format:

- **XML:** The instruction reminds the model to interpret XML symbols correctly, ensuring that any formula or question components formatted in XML are properly understood.
- **LaTeX:** Advises careful interpretation of LaTeX expressions, especially for mathematical content.
- **Other:** When the text does not fall into the above categories, no extra formatting instruction is provided.

B.2 Prompting strategies

To generate question-answer pairs, we first conducted an experiment evaluating 12 different prompting strategies. Based on performance results, we selected five strategies for further analysis. Two of these serve as baselines: *direct zero-shot*, the model receives only the question and image without additional instructions or contextual information. *zero-shot chain-of-thought (CoT)* (Wei et al., 2023), the model is asked to produce intermediate reasoning steps before arriving to the final answer. Beyond the baselines, we investigated how the order of multimodal input affects performance. Specifically, we compared cases where the model processes the image at the beginning versus at the end of the text input. Our results indicate that presenting the *image first*, followed by the problem text, leads to better performance. Finally, for models with strong reasoning capabilities but lacking the multimodal component, we implemented a *two-stage* prompting strategy. We first use GPT-4o to generate a textual description of the image. This description is then passed, along with the problem text, to o1-mini and o1-preview models. The quality of the generated descriptions were manually verified.

B.2.1 Direct zero-shot

A straightforward prompt that presents the question to the model.

Direct zero-shot

You are an expert in STEM courses.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Your Answer:

B.2.2 Chain-of-Thought Prompt

This prompt encourages a step-by-step analytical approach, asking the model to think through the problem before answering.

Chain-of-Thought Prompt

You are an expert in STEM courses tasked with answering questions with step-by-step analysis. Examine both the image(s) and question text before answering.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Your Answer: Let's think step by step.

B.2.3 Image First Prompt

This prompt prioritizes image analysis by instructing the model to examine the image details before considering the text, and then synthesize a detailed answer.

Image-First Prompt

You are an expert in STEM courses tasked with answering questions. But, first, you must analyze the image(s), which you will follow with the textual analysis.

You will follow the next steps before providing an answer.
Step 1: Analyze the Image(s) First
- Describe elements, patterns, and relationships in the image(s).
Step 2: Use Observations to Analyze the Text
- Use the image understanding to find relevant textual information in the question.
Step 3: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.2.4 Two Stage Prompt

Image Description Prompt

This prompt requests a detailed description of the provided image, linking its elements to the question context for use by another model. It does not answer the question but aims to provide details that will enable another to answer it.

Image Description Prompt

I am going to provide you with a question with an image. I need you to describe this image in as many details as possible and link those details to the question and its context.

I will then share this description of the image with an LLM which doesn't have vision capabilities, but better reasoning skills than you. In other words, you will be the eyes for that second model. As such, it is primordial that you don't leave out any details!

Note that some details that you think might be useless, may not be, as such make sure that you focus on every aspect.

Here is the Image: <image_names >

Here is the question: <question text >

You may now provide your detailed description. Make sure to follow the instructions that were given to you.

Answer With Image Description Prompt

This prompt asks the model to answer a question based solely on an image description, with a reference to the detailed image description provided earlier.

Answer With Image Description Prompt

You are an expert in STEM courses and will answer a question that includes an image description..

Here is the description of the image:
<detailed image description >

[Refer to `generate_format_instruction`]
Here is the question that you need to answer: <question text >

[Refer to `question_type_prompt`]
Please, explain the solution and answer in the following format:

```
{
    "reasoning": "Your explanation.",
   "answer": "Your answer and nothing more."
}
```

Your Reasoning and Answer:

B.3 Selecting prompting strategies

Initially, we tested 12 prompting strategies on 10 questions to select the most effective ones for the subsequent experiments. Figure 7 shows a comparison across all strategies. We selected the *two-stage* strategy as the most effective, followed by two baseline strategies, and finally the best strategy for presenting a model with both text and image.

In the basic prompting category, the question was presented along with the image, allowing the models to interpret the visual data without additional instructions. In the second category, prompts directed the models to explicitly consider both the image and text, either together or sequentially, with varying emphasis on fine-grained versus coarse-grained details. Finally, in the third category, models lacking vision capabilities were provided with detailed descriptions of the image instead.

B.3.1 Simultaneous Prompt

This prompt asks the LLM to examine both image and text simultaneously, integrating insights from both modalities before answering. It emphasizes a holistic analysis that considers all available information concurrently.

Simultaneous Prompt

You are an expert in STEM courses tasked with answering questions. Examine both the image(s) and question text before answering.

You will follow the next steps before providing an answer.
Step 1: Analyze the Image(s) and Text Together
- Describe key elements, patterns, and relationships, integrating both sources.
Step 2: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.3.2 Text First Prompt

This prompt directs the LLM to analyze the question text initially and then examine the associated image, using the textual understanding to guide the image analysis. It ultimately expects the model to merge both insights into a coherent, well-informed answer.

Text First Prompt

You are an expert in STEM courses tasked with answering questions. But, first, you must analyze the text, which you will follow with the image analysis.

You will follow the next steps before providing an answer.
Step 1: Analyze the Question Text First
- Understand the question context.
Step 2: Use Observations to Analyze the Image
- Use textual understanding to find relevant visual information (elements, patterns, relationships, etc.)
Step 3: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.3.3 Dual Phase Prompt

This prompt divides the analysis into two distinct phases; first analyzing the image(s) and then the text, before synthesizing the information into a final answer. It ensures that each component is evaluated independently before being combined for a comprehensive response.

Dual Phase Prompt

You are an expert in STEM courses tasked with answering questions with a dual-phase approach.

You will follow the next steps before providing an answer.
Step 1: Analyze the Image(s) First
- Describe elements, patterns, and relationships in the image(s).
Step 2: Interpret the Question Text Separately
- Identify question context independently of your image findings.
Step 3: Synthesize Textual and Visual Information
- Combine insights from both phases.
Step 4: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

B.3.4 Recursive Prompt

This prompt directs the LLM to iteratively alternate between image and text analysis, refining its understanding with each pass until a complete picture is achieved. It is designed to produce a well-considered final answer by progressively integrating and re-evaluating both modalities.

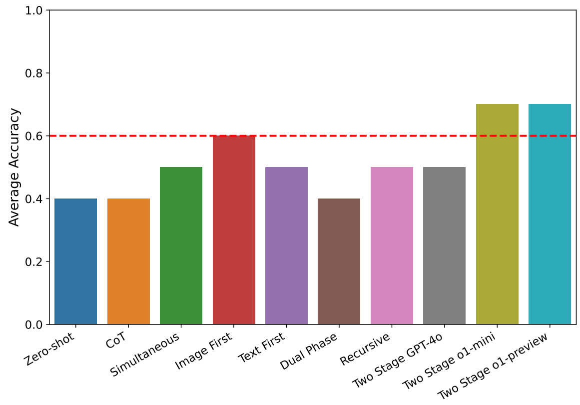


Figure 7: Model performance on the initial set of prompting strategies.

Recursive Prompt

You are an expert in STEM courses tasked with answering questions with recursive analysis.

You will follow the next steps before providing an answer.
Step 1: Analyze the Image(s) First
- Describe elements, patterns, and relationships in the image(s).
Step 2: Use Observations to Analyze the Text
- Use the image understanding to find relevant textual information in the question.
Step 3: Refine Analysis
- Alternate between image and text analysis, refining observations with each pass until a comprehensive understanding of the text and image is reached.
Step 4: Provide a Detailed Answer
- Synthesize observations into a complete answer.

Images: <image_names >

[Refer to `generate_format_instruction`]
Question: <question text >

[Refer to `question_type_prompt`]
Make sure to tackle every step mentioned above, before you answer.
Your Answer:

C Model Configuration

To evaluate performance, three OpenAI models were employed: GPT-4o with temperature 0.1, o1-mini-2024-09-12, and o1-preview-2024-09-12. GPT-4o was chosen as the baseline due to its strong vision capabilities. The o1-mini and o1-preview models, in contrast, lack native vision capabilities but exhibit strong reasoning abilities in text-based tasks. While GPT-4o allowed temperature adjustments, the o1 models did not support this feature. The primary focus was on GPT-family models as the ones that students can easily access models to

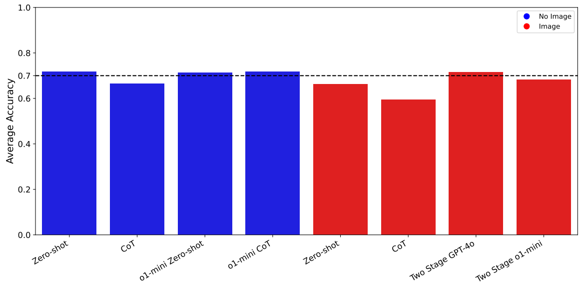


Figure 8: Model performance with and without supplemental image included.

run questions themselves while preparing a take-home assignment. We are trying to provide also some recommendations for educators on how to make such assignments less vulnerable to generative AI use. To explore the effect on different model families, we also test Qwen 2.5 72B VL, r1 Deepseek, and Claude 3.7 Sonnet, 2025.

We used `pandas`,[2] `json`,[3], `numpy`,[4] and `scikit-learn`[5] to process our results, compute accuracy scores, and compute statistical significance.

D Additional Experimental Results

D.1 Model performance comparison

We observe that the GPT model is the most performant one and that, in general, the models follow our findings. The results in various characteristics are presented in Table 4.

Looking at the performance per course in Table 3, we see that our findings hold. Also, sometimes, there are cases when Claude 3.7 or R1 outperform GPT model: in a subject like biology and mechanical physics.

D.2 Removing supplemental image

We tested the same prompts with and without supplemental images. For the *two stage* prompts we removed mentions of the image and didn't pass the descriptions. Figure 8 shows that the presence or absence of the image doesn't affect model performance.

D.3 Model performance vs student performance

In Figure 9, we compare model performance across five prompting strategies and two aggregation strategies with average student performance.

[2]https://pandas.pydata.org/docs/index.html
[3]https://docs.python.org/3/library/json.html
[4]https://numpy.org/doc/stable/index.html
[5]https://scikit-learn.org/stable/

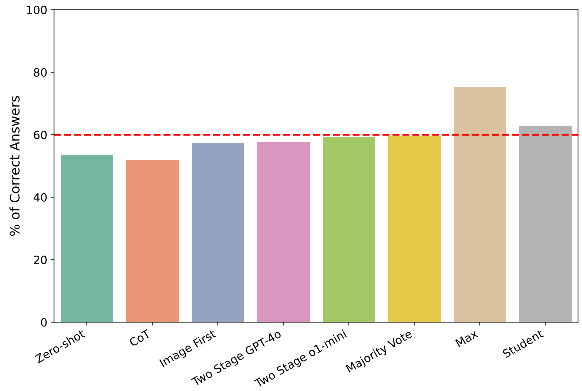


Figure 9: Average GPT-family models performance across five prompting strategies, aggregated results with the majority vote and maximum strategy and student performance.

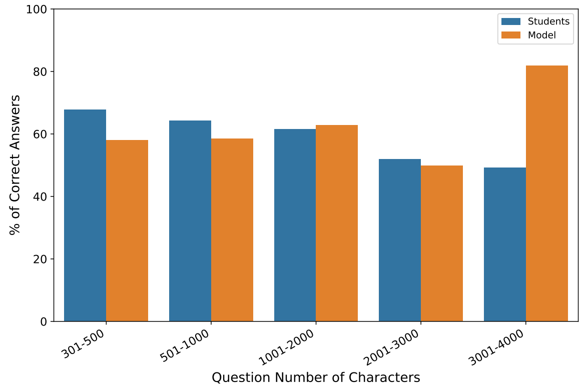


Figure 10: Comparison of student vs model accuracy depending on the question length in characters.

D.4 Student vs model accuracy depending on the question length

Figure 10 shows the comparison of the student and model accuracy (average majority vote) depending on the length of the question in characters.

D.5 Model performance across question and image features

Model and student performance per question and image features with 95% confidence intervals are presented in Tables 5 and 6.

Course Category	**GPT-4o**	**Claude 3.7**	**R1**	**Qwen 2.5-72B**
Astronomy	**0.78**	0.55	0.58	0.63
Biology	0.50	**0.75**	0.63	0.50
Chemistry	**0.47**	**0.47**	0.37	0.30
Computer Science	**0.78**	0.68	0.74	0.72
Electrical Engineering	**0.58**	0.48	0.37	0.43
Electromagnetism	**0.49**	0.45	0.45	0.45
Math	0.56	**0.57**	0.47	0.56
Mechanical Physics	0.56	0.56	**0.63**	0.56
Microfabrication	**0.87**	0.60	0.64	0.52
Neuro Science	**0.49**	0.31	0.29	0.36
Quantum Physics	**0.35**	0.29	0.18	0.24

Table 3: Average model performance across different course categories.

Category	**Label**	**GPT-4o**	**Claude 3.7**	**R1**	**Qwen 2.5-72 B**
Question feature	Simple	**0.613**	0.517	0.469	0.481
	Complex	**0.503**	0.454	0.459	0.502
	MCQ	**0.663**	0.563	0.538	0.575
	MCQMA	**0.457**	0.370	0.304	0.283
	Compound	**0.650**	0.566	0.533	0.538
Image feature	Crucial	**0.561**	0.473	0.426	0.448
	Supplemental	**0.740**	0.652	0.641	0.635

Table 4: Average model performance across different question and image features.

Category	Label	Model Accuracy and 95 % CI		
		Accuracy Mean	Lower Bound	Upper Bound
Question feature	Simple	0.613	0.54	0.68
	Complex	0.503	0.35	0.65
	MCQ	0.663	0.59	0.77
	MCQMA	0.457	0.68	0.84
	Compound	0.650	0.55	0.74
Image feature	Crucial	0.561	0.49	0.63
	Supplemental	0.74	0.60	0.86
	Algorithm	0.875	0.63	1.00
	Diagram	0.566	0.48	0.65
	Picture	0.687	0.54	0.84
	Line Plot	0.556	0.43	0.69

Table 5: Model accuracy means and 95% Confidence Intervals. CI is computed with non-parametric bootstrap using 1000 resamples.

Category	Label	Student Accuracy and 95 % CI		
		Accuracy Mean	Lower Bound	Upper Bound
Question feature	Simple	0.628	0.59	0.66
	Complex	0.622	0.54	0.70
	MCQ	0.689	0.64	0.74
	MCQMA	0.599	0.54	0.67
	Compound	0.579	0.52	0.64
Image feature	Crucial	0.637	0.60	0.67
	Supplemental	0.588	0.51	0.67
	Algorithm	0.534	0.40	0.66
	Diagram	0.613	0.57	0.66
	Picture	0.648	0.57	0.71
	Line Plot	0.645	0.58	0.71

Table 6: Student accuracy means and 95% Confidence Intervals for different image types. CI is computed with the non-parametric bootstrap using 1000 resamples