,

Language Models as Science Tutors

Abstract

NLP has recently made exciting progress toward training language models (LMs) with strong scientific problem-solving skills. However, model development has not focused on real-life use-cases of LMs for science, including applications in education that require processing long scientific documents. To address this, we introduce TutorEval and TutorChat. TutorEval is a diverse question-answering benchmark consisting of questions about long chapters from STEM textbooks, written by experts. TutorEval helps measure real-life usability of LMs as scientific assistants, and it is the first benchmark combining long contexts, free-form generation, and multi-disciplinary scientific knowledge. Moreover, we show that fine-tuning base models with existing dialogue datasets leads to poor performance on TutorEval. Therefore, we create TutorChat, a dataset of 80,000 long synthetic dialogues about textbooks. We use TutorChat to fine-tune Llemma models with 7B and 34B parameters. These LM tutors specialized in math have a 32K-token context window, and they excel at TutorEval while performing strongly on GSM8K and MATH. Our datasets build on open-source materials, and we release our models, data, and evaluations.

Keywords: language models, evaluation, science, education

1 Introduction

NLP has recently made exciting progress towards improving LM performance on problem-solving benchmarks such as GSM8K Cobbe et al. (2021) or MATH Hendrycks et al. (2021b) and on general knowledge benchmarks like MMLU Hendrycks et al. (2021a). One approach to constructing models for problem-solving consists in fine-tuning base models on math datasets, such as MetaMath Yu et al. (2023), MAmmoTH Yue et al. (2023), or WizardMath Luo et al. (2023). Other approaches train on web-scraped corpora, such as Llemma Azerbayev et al. (2023) with OpenWebMath Paster et al. (2023), or on large synthetic datasets, such as Phi-2 Gunasekar et al. (2023); Li et al. (2023c). This flurry of new LMs raises the question: how can we apply these problem-solving skills to build helpful LM tutors for science? The first challenge for answering this question is the lack of relevant LM evaluations.

Existing science evaluations are lacking in the following ways. First, they do not simulate real-life situations where we use LMs as assistants, such as asking to clarify hard concepts or asking for background information. Second, science evaluations only evaluate the final answer to a problem, not the reasoning process. Third, real-life situations often require LMs to process long scientific materials, which are not tackled by any existing evaluations.

To bridge these gaps, we introduce TutorEval, a realistic long-context question-answering benchmark requiring advanced scientific knowledge. TutorEval consists of over 800 questions written by experts about textbook chapters in math, physics, computer science, environmental science, and life sciences. TutorEval extends the LM-as-evaluator

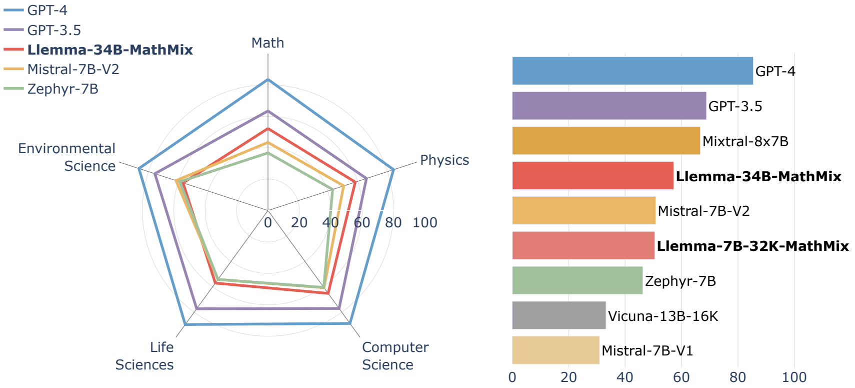


Figure 1: Left: performance breakdown on TutorEval by domains. Right: leaderboard of popular models on TutorEval. Our models are marked in bold.

framework Li et al. (2023a) by introducing *key points*: the experts write sketches of the correct answer, which an LM evaluator uses to grade the LM tutor's answers. See Figure 2.

We investigate LM training TutorEval. We find that fine-tuning with existing dialogue datasets delivers poor performance. Therefore, we create TutorChat, the first long-context dialogue dataset for science. TutorChat consists of 80,000 synthetic conversations about textbook chapters covering STEM topics, humanities and social sciences. We show that TutorChat is a rich resource for domain-specific fine-tuning and we construct a new data mixture, MathMix, to fine-tune our Llemma-MathMix models, two long-context LMs competitive on TutorEval, MATH and GSM8K.

In summary, we introduce TutorEval, the first long-context science benchmark, and TutorChat, the first dialogue dataset for science, and we conduct investigations into the importance of training and fine-tuning with scientific texts. We release our data, evaluations, and competitive long-context models specialized in science and math reasoning.

2 Related Work

LMs for science Minerva Lewkowycz et al. (2022) and Galactica Taylor et al. (2022) were two pioneering efforts to train LMs on large corpora of scientific documents. Llemma Azerbayev et al. (2023) has replicated Minerva approach using the OpenWebMath dataset Paster et al. (2023). MMLU Hendrycks et al. (2021a) and MATH Hendrycks et al. (2021b) are key benchmarks which have motivated this research direction. Other problem-solving benchmarks include SciBench Wang et al. (2023) and SciEval Sun et al. (2023).

Recent works have applied LMs to education. EduBot Li et al. (2023b) applies LMs to English-teaching and MathDial Macina et al. (2023) applied LMs to math. Various works study the expert-level capabilities of LMs: Huang et al. (2023) benchmarks GPT-4 on machine learning tasks and Wu et al. (2023) studies ChatGPT with retrieval-augmentation. Surveys include Peskoff and Stewart (2023), Frieder et al. (2023), and Collins et al. (2023).

Fine-tuning on model-generated dialogue A popular approach for improving LMs consists training smaller LMs on the high-quality outputs of larger LMs. Examples include the textbooks of Gunasekar et al. (2023) or Orca Mukherjee et al. (2023). This approach has led to an accumulation of dialogue datasets. ShareGPT and WildChat Zhao et al. (2024) are crowd-sourced from real-life ChatGPT conversations. UltraChat Ding et al. (2023) and OpenOrca Lian et al. (2023) are synthetic texts generated by ChatGPT.

LM as an evaluator NLP has recently explored new ways of evaluating LMs. One solution resorts to using pre-trained LMs to compare model outputs. AlpacaFarm Li et al. (2023a) uses LMs to simulate human evaluators by obtaining preference votes between model outputs. Chiang and Lee (2023), Zheng et al. (2023), Chen et al. (2023) analyse GPT-4's ability to simulate human preferences. LLMBar Zeng et al. (2023) evaluates LM evaluators for instruction-following. Most works find that GPT-4 is reliable, and we build on these findings to introduce TUTOREVAL.

3 TUTOREVAL

In Section 3.1, we show how TUTOREVAL was constructed with the help of experts and how TUTOREVAL can be used for both open-book and closed-book evaluation. In Section 3.2, we explain how we use an LM evaluator and we show that GPT-4 correlates well with human judgments.

3.1 Dataset construction

Question collection We work with 17 annotators, all STEM researchers with teaching experience. Each annotator was provided a set of textbook chapters and was instructed to write questions about topics which they are qualified to teach. These chapters come from open-source textbooks collected from libretexts.org, as discussed in Section 4.1.

Chapter:
[...] One of the key numbers we can extract from mass and momentum is the invariant mass, a number independent of the Lorentz frame: $W^2c^4 = (\sum_i E_i)^2 - (\sum_i \vec{p}_i)^2c^2$. This quantity takes its most transparent form in the center-of-mass, where [...].

Question:
How is the first equation in the chapter consistent with Einstein's $E = mc^2$ formula? It looks wrong.

LM Tutor:
The first equation you mentioned, $W^2c^4 = (\sum_i E_i)^2 - (\sum_i \vec{p}_i)^2c^2$, is indeed related to the famous $E = mc^2$ formula. [...] So, the first equation is indeed consistent with Einstein's formula.

……

Key points provided to GPT-4:

- The student is mistaken in saying the first equation is wrong.
- Einstein's $E = mc^2$ is valid only at rest. It can be obtained by setting the momenta $\sum_i \vec{p_i} = 0$, where W is the invariant mass m in $E = mc^2$.
- The second formula is what the student is looking for with $W = m$ and $E_{\mathrm{CM}} = E$.

GPT-4 Evaluator:
Presentation: The AI's response is structured in an informative manner [...]. Grade: 1.5/3.
Correctness: The AI's response is correct in stating that the first equation is related to Einstein's $E = mc^2$ formula [...]. However, the AI fails to address the key point that Einstein's formula is valid only when the object is at rest, which is crucial for correctness. [...] Grade: 1.5/3.

Figure 2: TUTOREVAL: given the chapter, LM Tutor answers the student question. GPT-4 grades the generation by referencing the key points.

In total, we collect 834 questions spanning math, physics, computer science, life sciences and environmental science. See Table 3 for statistics and Figure 2 and Appendix A for examples. We note that TUTOREVAL is biased toward math and computer science, but we hope to extend TUTOREVAL in the future as follow-up work.

Data categories TUTOREVAL is a diverse set of questions allowing us to evaluate models in different settings and to track fine-grained behavior. We make the following data categories:

1. *Closed-book*: can the *question* be understood without reading the chapter first?
2. *Answer in chapter*: is the *answer* contained in the chapter, or does it require substantial external knowledge?

3. *Misleading*: is the question phrased in a misleading way, such as including false assumptions or an having overly-confident tone?
4. *Difficulty*: how hard is the question relatively to the chapter's intended audience?

TutorEval contains 370 closed-book questions, meaning that the question does not refer explicitly to the chapter. This enables evaluation of short-context LMs, and we refer to this evaluation setting as TutorEval-ClosedBook.

Data validation To ensure the relevance of our data categories, we evaluate GPT-4 on different data categories both in open-book TutorEval and in the closed-book setting (Table 4). First, we see that GPT-4's results on open-book questions are significantly affected in closed-book evaluation, proving that open-book questions require the chapter. We also see that splitting questions by *answer in chapter* shows how well GPT-4 is able to utilize long contexts. This demonstrates that TutorEval is a rich benchmark which offers a thorough investigation of long-context understanding in the science domain.

3.2 LM as an Evaluator

Key points as ground-truths We ask human annotators to provide a ground-truth solution to each question by sketching the *key points* that a good answer should address. These key points make it possible to assess the correctness of an answer.

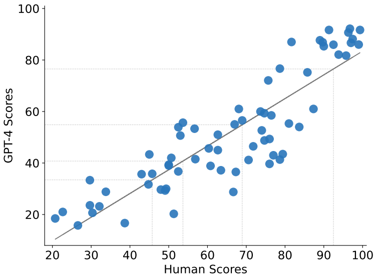


Figure 3: Score correlation between 17 annotators and GPT-4 for models evaluated in Table 5. Annotators evaluate their own set of 50 questions.

LM-powered evaluation At evaluation time, the LM tutor is prompted to answer each question based on the associated chapter. We let the LM generate up to 800 tokens. To rate the free-form model generations, we prompt an LM evaluator to assign grades to each output. The LM evaluator is instructed to follow the key points as a guide for judging the answer. We use the January 2024 version of GPT-4-Turbo OpenAI (2023). GPT-4 is prompted to grade the output's *presentation* and *correctness*. See Appendix C.1 for descriptions. Correctness is a more meaningful metric to benchmark LMs, so we usually report only the correctness scores. Nevertheless, presentation scores are informative, and we show that prompting GPT-4 for a presentation score improves human-GPT-4 correlation on correctness.

Human-GPT-4 agreement To show that GPT-4 can be used as a reliable evaluator, we ask each human annotator to grade the outputs of four anonymized models on the 50 questions they created. We report the final scores from humans and GPT-4 in Table 5. In Figure 3 we see that human and GPT-4 scrores correlate well. In Appendix C, we also compute 3 correlation metrics for human-GPT-4 gradings.

Additional analysis We run ablations to further analyse the effect of the TutorEval grading strategy on human-GPT-4 correlation. We consider (1) removing the key points from GPT-4's prompt, (2) removing the presentation score, and (3) using GPT-3.5 instead of GPT-4. Correlation deteriorates in all cases, indicating the necessity of each component. See Appendix C for details, where we also show in Figure 6 that the key points are particularly useful for GPT-4 to grade its own generations fairly.

3.3 Evaluating SOTA LMs on TutorEval

Figure 1 shows the leaderboard of TutorEval with state-of-the-art LMs and their domain breakdown. Detailed numbers can be found in Table 4 in the appendix. We see that GPT-4 dominates in all domains followed by GPT-3.5 and Mixtral. As shown in Table 5, human evaluators give 92% correctness to GPT-4 on TutorEval. Future work will investigate how to build a harder academic benchmark. In the remainder of this paper, we show that TutorEval performance for 7B and 34B models can be greatly improved with dedicated scientific training and fine-tuning.

4 TutorChat

In this section, we build the first dialogue dataset for science. We introduce TutorChat, a dataset of long, high quality, synthetic conversations about textbook materials. In Section 5, we will show that TutorChat is highly effective for improving performance on TutorEval, compared to general-purpose dialogue datasets like UltraChat.

4.1 Constructing TutorChat

We construct TutorChat in two steps: first we collect open-source textbooks which we process into a high-quality dataset of long documents, and secondly we generate dialogues about each chapter by simulating teacher-student interactions with GPT-3.5 and GPT-4.

Textbook collection We scrape the domain libretexts.org for open-source textbooks in all academic domains. We collect the HTML from each textbook page and convert it to latex using the OpenWebMath HTML-to-LaTeX processor. We apply additional filtering to each page to remove bibliographies and metadata. To structure our dataset, we concatenate short consecutive textbook chapters, so that each 'chapter' contains at least 512 words. The final dataset consists in 1,685 textbooks, 78,000 chapters, and covers all education domains, including STEM, humanities and social sciences. We release this dataset for future research.

Dialogue generation We generate TutorChat from our textbooks using GPT-3.5-Turbo Brown et al. (2020) and GPT-4-Turbo OpenAI (2023). We simulate *open-book* and *closed-book* conversations. Open-book dialogues simulate the student asking questions about the chapter to the teacher. Closed-book dialogues simulate the teacher conducting a class with the chapter. Using GPT-4, we also generate *textbook exams*: these are exam scripts with solutions based on a chapter. See Appendix E for more details. TutorChat consists of 78K dialogues. We create a validation split of 2.5K samples, which was used by TutorEval annotators to write questions.

4.2 Dataset Inspection

Quality control Empirically, we find that dialogues generated by GPT-4 are more factual and challenging than those written by GPT-3.5. GPT-3.5 dialogues are more error-prone and sometimes wander, but they are more conversational, realistic, and friendly. See Appendix F for examples of dialogues generated by GPT-3.5 and GPT-4.

Dataset experiments We run ablations to understand the impact of the TutorChat generation strategy. We sample 10K sequences of the following types: (1) dialogues generated by GPT-4 or GPT-3.5, (2) open-book or closed-book dialogues, (3) dialogues simulating strong or weak students, (4) a random sample of dialogues. We fine-tune Llemma-7B-32K (see Section 5.3) on each dataset and we report results in Figure 4 and Table 12 in Appendix G. On TutorEval, we find that the GPT-4 data gives better results than GPT-3.5 data. However, this performance gap is mostly due to the 13% of 'misleading' questions. Similarly, dialogues simulating weak students mostly help with misleading questions. Hence TutorChat helps mitigate sycophancy Perez et al. (2023).

Open-book dialogues are also more effective than closed-book dialogues, both for TutorEval and for ClosedBook. We believe that dialogues where an LM and a user discuss a common piece of text occur rarely in existing datasets, making TutorChat a valuable contribution. Finally, Table 12 shows that the random data sample has the most well-rounded performance. This shows that data diversity is key and that the engineering effort put into constructing TutorChat has produced a useful dataset.

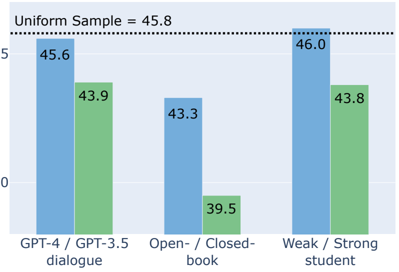


Figure 4: TutorEval results for fine-tuning Llemma-7B-32K with subsets of 10K TutorChat-STEM samples. See Table 12 for results.

5 Building LM Tutors

In this section, we investigate the main ingredients required for building strong and helpful LM science tutors, with a special focus on math. We ask the two following questions:

(1) How important is training with scientific text for TutorEval?

(2) How can we train an LM tutor that also has strong math problem-solving skills?

In Section 5.1, we begin by introducing our two best models: Llemma-7B-32K-MathMix and Llemma-34B-MathMix, and their training recipes. In Sections 5.2, 5.3, we answer the two above questions step-by-step, by finding the most suitable base model and the best curated fine-tuning data. As a result, our best models achieve competitive results on TutorEval and math problem-solving tasks.

5.1 Method

We present the two-stage fine-tuning pipeline which we use to obtain the best TutorEval results. We first extend the LM's context-window to 32K tokens, and then train with TutorChat. See Appendix H for hyperparameters.

Long-context training Llemma-7B has a context window of 4K tokens. This is too short for TutorEval, so we create a 5B-token dataset from OpenWebMath by sampling documents of up to 32K tokens. We fine-tune with an increased RoPE base Su et al. (2024), following Roziere et al. (2023). We name this model Llemma-7B-32K. Because of computational limitations, we are not able to run this context-extension phase on Llemma-34B, but we still increase the RoPE base before fine-tuning on TutorChat.

TutorChat fine-tuning We fine-tune on open-book dialogues by including the entire textbook chapter, and on closed-book dialogues by removing the textbook chapter. See Appendix H for details.

MathMix: a dataset for LM math tutors We find that we can further improve TutorEval performance by combining TutorChat-STEM and MetaMath. We process the original MetaMath dataset by randomly concatenating 10 question/answer pairs. We refer to the mix of TutorChat-STEM and MetaMath as MathMix.

5.2 Investigating Foundation Models

By comparing Llama-2 based models, CodeLlama and Llemma-7B, we find that TutorEval benefits significantly from training on scientific text.

Baselines We fine-tune Llama-2-7B and CodeLlama-7B with MathMix as in Section 5.1. Recall that Llemma was trained from CodeLlama, which was trained from Llama-2, so this helps isolating the effect of continued-training. To show the impact of long-context training, we also fine-tune a base Llemma model with MathMix directly, and refer to this model as Llemma-7B-MathMix. Finally, we compare these models with MAmmoTH, WizardMath, Llama-Pro-8B-Instruct Wu et al. (2024), Tulu Ivison et al. (2023), and other models.

Results See Table 1. Llama-2-based models do poorly on ClosedBook. Only Tulu-2-DPO-70B improves over our best 7B model. Llama-Pro-8B-Instruct was trained on the same data as Llemma. Comparing with Llemma-7B-MathMix shows the strength of our data.

CodeLlama-32K-MathMix ranks between our Llama-2 and Llemma models. This confirms the usefulness of training on code and on scientific data. Long-context training improves TutorEval performance by 4 points without harming ClosedBook, verifying that this helps long context understanding. Table 4 shows that Llemma-7B-32K-MathMix is stronger than Mistral-7B-V2 on math questions but weaker in other domains. Overall, Mistral performs slightly better than our best 7B Llemma model[1].

	Tutor-Eval	Closed-Book
Pre-trained Models		
MAmmoTH-7B	-	1.5
MAmmoTH-13B	-	24.2
MAmmoTH-70B	-	2.8
WizardMath-7B	-	20.1
WizardMath-13B	-	28.2
WizardMath-70B	-	0.5
Tulu-2-DPO-7B	-	33.4
Tulu-2-DPO-13B	-	42.6
Tulu-2-DPO-70B	-	59.5
Phi-2	-	28.6
Llama-Pro-8B-Instruct	-	31.6
Vicuna-13B-16K	33.2	37.7
Mistral-7B-V1	30.9	36.8
Zephyr-7B	46.3	50.4
Mistral-7B-V2	50.9	53.8
Ablations		
Llama-2-7B-32k-MathMix	30.8	31.7
CodeLlama-32k-MathMix	43.2	37.1
Llemma-7B-MathMix	46.4	46.4
Ours		
Llemma-7B-32K-MathMix	50.5	46.6
Llemma-34B-MathMix	57.2	57.6

Table 1: TutorEval results comparing different models. Most pre-trained models are not able to process long contexts so we only evaluate them on ClosedBook.

5.3 Investigating Fine-tuning Datasets

Having established that Llemma-7B-32K is a strong base model, we compare different fine-tuning datasets.

---

[1] Note that Mistral-7B-V2 is an industry-grade closed-source LM and that Llemma-7B-32K-MathMix is obtained from Llama-2 with open-source data.

Baselines We fine-tune Llemma-7B-32K with UltraChat (80K samples), TutorChat (75K samples), and TutorChat-STEM (40K samples). We also mix MetaMath into each of these datasets. Recall that MathMix is TutorChat-STEM + MetaMath and consists of 80K samples.

Results Table 2 shows that fine-tuning with TutorChat improves TutorEval by 9 points over UltraChat, showing the benefits of TutorChat.

MathMix gives the strongest model on TutorEval but is weaker than TutorChat for closed-book evaluation.

In Figure 5, we plot the TutorEval performance of different fine-tuned Llemma-7B-32K models against their average GSM8K and MATH scores (details in Table 13). We see that fine-tuning with TutorChat alone does not improve performance on math tasks, nor does combining TutorChat with MetaMath, surprisingly. However, TutorChat allows us to target science-specific dialogues, and TutorChat-STEM significantly improves math performance when combined with MetaMath to form MathMix.

Our model Llemma-7B-32K-MathMix outperforms Mistral-7B-V2 on math tasks, while matching its TutorEval performance. This illustrates the richness and usefulness of the TutorChat dataset.

6 Conclusion and Future Work

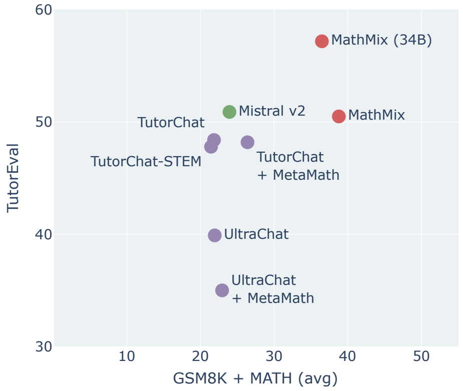
We have introduced TutorEval and TutorChat with the hope of encouraging the development of LMs as useful scientific assistants. Moreover, the methodology behind TutorEval opens up new avenues for creating specialised evaluations.

As follow-up work, we plan to keep building up TutorEval to cover more science disciplines, more levels of difficulty, and more diverse use-cases. We believe that the dataset presented in this paper can be the foundation of a large-scale crowd-sourcing effort for the research community to define how it would like LMs to be applied as a transformative technology.

Table 2: TutorEval results for fine-tuning Llemma-7B-32K with different dialogue datasets.

Fine-tuning data	TutorEval	Closed-Book
MetaMath	14.5	20.5
UltraChat	39.9	41.2
+ MetaMath	35.0	45.1
TutorChat	48.4	**50.9**
+ MetaMath	48.2	49.7
TutorChat-STEM	47.8	48.2
+ MetaMath (= MathMix)	**50.5**	46.6

Figure 5: Combined performance on TutorEval and GSM8K and MATH. Our MathMix models are in red. In purple are 7B-parameter baselines trained from Llemma-7B-32K. Mistral-7B-V2 is included in green.



References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics, 2023.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. Exploring the use of large language models for reference-free text quality evaluation: An empirical study, 2023.

Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.870. URL `https://aclanthology.org/2023.acl-long.870`.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL `https://aclanthology.org/N19-1300`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. URL `https://api.semanticscholar.org/CorpusID:3922816`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Katherine M. Collins, Albert Q. Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B. Tenenbaum, William Hart, Timothy Gowers, Wenda Li, Adrian Weller, and Mateja Jamnik. Evaluating language models for mathematics through interactions, 2023.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. 2023.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL `https://aclanthology.org/2023.emnlp-main.183`.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL `https://zenodo.org/records/10256836`.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021b. URL `https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf`.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Benchmarking large language models as ai research agents. *arXiv preprint arXiv:2310.03302*, 2023.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 452–466, 2019. doi: 10.1162/tacl_a_00276. URL `https://aclanthology.org/Q19-1026`.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023a.

Yu Li, Shang Qu, Jili Shen, Shangchao Min, and Zhou Yu. Curriculum-driven edubot: A framework for developing language learning chatbots through synthesizing conversational data, 2023b.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023c.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`, 2023.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.372. URL `https://aclanthology.org/2023.findings-emnlp.372`.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

OpenAI. Gpt-4 technical report, 2023.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL `https://aclanthology.org/P16-1144`.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. Openwebmath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*, 2023.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. Discovering language model behaviors with model-written evaluations. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.847. URL `https://aclanthology.org/2023.findings-acl.847`.

Denis Peskoff and Brandon Stewart. Credible without credit: Domain experts assess generative language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 427–438, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.37. URL `https://aclanthology.org/2023.acl-short.37`.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, aug 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL `https://doi.org/10.1145/3474381`.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. Scieval: A multi-level large language model evaluation benchmark for scientific research, 2023.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin, editors, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4413. URL `https://aclanthology.org/W17-4413`.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ping Luo, and Ying Shan. Llama pro: Progressive llama with block expansion, 2024.

Yongchao Wu, Aron Henriksson, Martin Duneld, and Jalal Nouri. Towards improving the reliability and transparency of chatgpt for educational question answering. In Olga Viberg, Ioana Jivet, Pedro J. Muñoz-Merino, Maria Perifanou, and Tina Papathoma, editors, *Responsive and Sustainable Educational Futures*, pages 475–488, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-42682-7.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL `https://aclanthology.org/P19-1472`.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*, 2023.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. (inthe)wildchat: 570k chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Bl8u7ZRlbM`.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685, 2023. URL `https://api.semanticscholar.org/CorpusID:259129398`.

Appendix A. Examples of TutorEval questions

Below we list some examples of TutorEval questions by domain.

A.1 Mathematics

Chapter url:
https://math.libretexts.org/Bookshelves/Algebra/Advanced_Algebra/09%3A_Sequences_Series_and_the_Binomial_Theorem/9.04%3A_Binomial_Theorem

Question: Can you walk me through the calculations for Exercise 1? I'd like to see another example worked out in the same level of detail as Example 2.

Key points:

- the question asks to calculate 8 choose 5
- the answer is 56, as stated in the chapter
- show the student how to use cancellations in the binomial coefficient formula to avoid carrying out large multiplications

Additional information

- closed-book: no
- answer in chapter: yes
- misleading: no
- difficulty: easy

A.2 Computer science

Chapter url: https://eng.libretexts.org/Bookshelves/Computer_Science/Applied_Programming/Book%3A_Neural_Networks_and_Deep_Learning_(Nielsen)/06%3A_Deep_Learning/6.04%3A_Recent_progress_in_image_recognition

Question: Is the dataset from ILSVRC-2012 a subset of the ImageNet dataset?

Key points:

- the full name of ILSVRC-2012 is ImageNet Large Scale Visual Recognition Challenge 2012
- ILSVRC-2012 contains 1000 categories and 1.2 million images

Additional information:

- closed-book: yes
- answer in chapter: yes
- misleading: no
- difficulty: easy

A.3 Physics

Chapter url:

https://phys.libretexts.org/Bookshelves/Classical_Mechanics/Classical_Mechanics_(Dourmashkin)/13%3A_Energy_Kinetic_Energy_and_Work/13.05%3A_Work_done_by_Non-Constant_Forces

Question: So the work done by a force F between x and y is simply the integral of F along the segment [x;y]?

Key points:

- The statement is imprecise.

- It is the integral of the component of F in the $y - x$ direction.

Additional information:

- closed-book: yes
- answer in chapter: yes
- misleading: yes
- difficulty: easy

A.4 Life sciences

Chapter url: https://med.libretexts.org/Bookshelves/Anatomy_and_Physiology/Anatomy_and_Physiology_1e_(OpenStax)/Unit_3%3A_Regulation_Integration_and_Control/17%3A_The_Endocrine_System/17.03%3A_The_Pituitary_Gland_and_Hypothalamus

Question: The section on antidiuretic hormone states that dehydration can result in high blood osmolarity. I don't really understand why that would happen. Will you please explain it to me?

Key points:

Key points:

- The student is confused about the relationship between blood osmolarity and hydration
- State that osmolarity is proportional to the amount of dissolved particles per unit volume
- Mention that the volume of consumed water influences blood volume
- State that lower blood volume given the same number of dissolved particles must result in higher osmolarity

Additional information:

- closed-book: no
- answer in chapter: no
- misleading: no
- difficulty: easy

A.5 Environmental science

Chapter url:

https://eng.libretexts.org/Bookshelves/Environmental_Engineering_(Sustainability_and_Conservation)/Book%3A_Essentials_of_Environmental_Science_(CK-12)/13%3A_Untitled_Chapter_13/13.03%3A_New_Page

Question: So eutrophication could lead to hypoxia due to explosive growth of algae that consume oxygen?

Key points:

- This is false.
- The student is confused about the causes of oxygen depletion.
- Need to clarify the logical chain of eutrophication.

Additional information:

- closed-book: no
- answer in chapter: yes
- misleading: no
- difficulty: hard

Appendix B. Detailed TutorEval results

Table 3: TutorEval question counts per domain and category. Question categories are described in Section 4.1.

	Math	CS	Physics	Env.	Life	Total
Hard	96	59	54	22	10	241
Misleading	55	21	28	4	5	113
Closed-book	159	100	71	20	20	370
Answer in chapter	128	98	56	20	6	308
Total	362	205	174	53	40	834

Table 4: Detailed TutorEval results for long-context models evaluated in this paper. We show scores by domain, difficulty, on the set of misleading questions, and in closed-book evaluation. Models are grouped as follows (top-down): (1) Strong pre-trained LLMs (2) Medium-sized pre-trained LMs (3) Our best models (4) Ablations of Llemma-7B-32K fine-tuned with different TutorChat-based datasets (4) Ablations of different base models fine-tuned with MathMix (5) Ablations of Llemma models fine-tuned without TutorChat.

Models	Domain					Difficulty		Misleading	TutorEval	Closed-Book
	Math	Physics	CS	Life	Env	Easy	Hard			
GPT-4	83.4	84.0	88.9	91.3	86.5	88.6	77.7	75.9	85.5	87.5
GPT-3.5	63.4	65.9	77.1	79.6	75.8	73.1	58.3	56.6	68.8	70.3
Mixtral-8x7B	61.8	63.1	75.0	72.9	74.2	71.3	55.3	54.4	66.7	68.7
Vicuna-13B-16K	25.6	28.5	47.2	46.3	36.8	38.1	21.3	23.4	33.2	37.7
Mistral-7B-V1	21.6	27.3	45.9	37.9	43.4	34.2	22.8	20.5	30.9	36.8
Zephyr-7B-Beta	36.7	43.9	60.5	53.8	59.1	50.5	35.9	33.5	46.3	50.4
Mistral-7B-V2	43.3	51.1	60.7	53.3	61.6	55.2	40.3	32.9	50.9	53.8
Llemma-7B-32K-MathMix	45.7	51.1	56.8	55.0	53.1	56.6	35.6	29.4	50.5	46.6
Llemma-34B-MathMix	52.2	58.8	65.2	55.4	56.9	62.0	45.4	40.2	57.2	57.6
Llemma-7B-32K-TutorChat-STEM	43.2	48.6	55.0	49.2	48.4	52.6	36.2	29.0	47.8	48.2
Llemma-7B-32K-TutorChat	45.3	48.6	55.0	46.3	45.0	54.0	34.7	27.8	48.4	50.9
Llemma-7B-32K-TutorChat+MetaMath	43.7	47.0	56.0	49.2	51.6	53.5	35.1	28.5	48.2	49.7
Llama-32K-MathMix	21.0	33.9	40.0	53.8	34.3	33.5	24.2	23.5	30.8	31.7
CodeLlama-32K-MathMix	38.4	38.6	56.3	42.1	41.2	48.4	30.4	30.7	43.2	37.1
Llemma-7B-MathMix	42.9	45.3	55.0	42.5	44.3	51.9	33.1	26.5	46.4	46.4
Llemma-7B-32K-MetaMath	14.4	14.0	17.8	9.2	7.6	16.3	9.9	7.2	14.5	20.5
Llemma-7B-32K-Ultrachat	34.3	37.4	50.8	41.3	43.1	44.6	28.2	22.1	39.9	41.2
Llemma-7B-32K-MetaMath-Ultrachat	29.8	29.9	47.9	32.1	39.6	38.1	25.4	21.2	35.0	45.1

Table 5: TutorEval scores of 4 models graded by humans and GPT-4. Models are introduced in Section 5. See Appendix B, Table 6 for detailed human evaluation results and Appendix C for data analysis about GPT-4/human agreement.

	Human TutorEval		**GPT-4 TutorEval**	
	Presentation	Correctness	Presentation	Correctness
Vicuna-13B-V1.5	51.0	49.0	44.5	32.9
Llemma-7B-32K-Ultrachat	51.4	59.4	46.2	39.4
Llemma-7B-32K-MathMix	67.8	67.9	62.4	50.0
GPT-4	92.3	91.9	89.5	85.2

Table 6: Detailed human TutorEval results. The 4 models below are graded blind by all annotators who contributed to TutorEval. Each annotator grades the model on their set of approximately 50 questions and we collect all results to compute the final TutorEval scores.

	Domain					Difficulty		Misleading	TutorEval
Models	Math	Physics	CS	Life	Env	Easy	Hard	Questions	
GPT-4	88.6	92.2	94.5	95.8	93.8	93.6	85.7	82.7	91.2
Llemma-7B-32K-MathMix	59.6	74.5	77.4	62.1	72.5	71.6	52.3	48.1	66.5
Llemma-7B-32K-Ultrachat	51.6	64.3	69.6	45.4	53.2	62.3	41.7	39.3	56.8
Vicuna-13B-16K	51.4	37.9	67.0	43.8	49.1	53.6	34.0	37.0	48.5

Table 7: We compare the GPT-4 and Llemma-34B-MathMix scores on different subsets of TutorEval questions, evaluated both open-book (TutorEval) and closed-book (ClosedBook). Note that the final ClosedBook score is the score achieved on closed-book questions only (370 questions) but that TutorEval includes all questions (834 questions). These results show that open-book questions cannot be evaluated with ClosedBook. These results also show how well LMs are able to utilize the added context to extract the answer when the question is marked as *answer in chapter*.

	GPT-4		Llemma-34B-MathMix		Number of
	TutorEval	ClosedBook	TutorEval	ClosedBook	Questions
Open-book Questions	83.4	67.2	54.9	42.9	464
Answer *not* in Chapter	81.4	67.1	50.0	40.5	261
Answer in Chapter	89.6	63.0	60.0	40.1	203
Closed-book Question	88.0	87.5	60.1	57.6	370
Answer *not* in Chapter	85.1	84.2	56.6	51.9	265
Answer in Chapter	93.9	89.8	68.8	63.4	105
Final score	85.5	87.5	57.2	57.6	834/370

Appendix C. TutorEval grading

C.1 Grading template

Below is the template used for prompting the LM-evaluator to assign grades to model outputs on TutorEval.

```
Your task is to evaluate the teaching abilities of a new AI system which is
interacting with a student about a science topic.  The student and AI system
are working together on a textbook chapter, and I would like you to rate how
well the AI system addressed the student's question.

You should give scores from 0 to 3 for PRESENTATION and CORRECTNESS. Half
points are allowed.  Please refer to the following descriptions:

PRESENTATION: the AI provides an engaging response which will make
the student want to learn more.  Examples of good presentation skills
include:  giving the response a clear and helpful structure, picking up on
positive aspects of the student's contributions, using examples to clarify
complicated ideas, explaining complicated arguments in detail, adding
follow-up and broadening remarks, etc.
CORRECTNESS: the AI correctly understands the question and the answer is
true and accurate.  The answer does not contain any false or misleading
statements.  The AI does not include any irrelevant information and does
not omit any essential reasoning steps.  The AI also correctly relates the
question to the chapter's content.  Pay particular attention to reasoning
and calculation mistakes.
Here is the textbook chapter used for this interaction:

"""
PASSAGE
"""

Here is the student's question:

"""
QUESTION
"""

To help you in your evaluation, we've compiled some ground-truth key points
which a good AI system should cover in its answer.  You MUST check that
the AI's answer agrees with these key points.  These key points have been
checked by experts and are 100% correct.  These key points are particularly
useful for spotting CORRECTNESS errors.

"""
Key points to cover:
KEY POINTS
"""

Here is the AI's answer:

"""
ANSWER
"""

Please present your scores as follows:
PRESENTATION: [explanation].  GRADE: x/3
CORRECTNESS: [explanation].  GRADE: x/3
```

C.2 Grading ablations

We collect statistics to help us understand how well TutorEval correlates with human for judging the 4 models evaluated in Table 5. For each of the 834 TutorEval questions, we compute 3 statistics: the Pearson, the Spearman, and the Kendall-τ coefficient between GPT-4 and human grades. The Pearson measures how close GPT-4's grades are to the human grades, and the Spearman and Kendall-τ measure how similar the rankings are. When computing the Spearman and Kendall-τ, we count it as 0 if either GPT-4 assigns same grades to all models while human grades vary, or if human grades are identical while GPT-4's grades differ.

We average each of these coefficients over the 834 questions. Correlation coefficients for correctness are reported in Table 8 and coefficients for presentation are in Table 9. We compare the final values with the statistics we obtain in three other evaluation configurations:

1. Evaluate without showing GPT-4 the ground-truth key points
2. Evaluate without prompting GPT-4 for a presentation grade
3. Evaluate with GPT-3.5 instead of GPT-4

In all cases, we see that at least one correlation coefficient drops compared to the standard TutorEval configuration. In Figure 6, we see that the key points are particularly useful for GPT-4 to grade its own generations fairly.

Correctness correlation	TutorEval grading	Grading without key-points	Grading without presentation	GPT-3.5 grading
Pearson	60.0	56.7	56.4	47.3
Spearman rank	58.1	55.1	58.0	46.6
Kendall-τ	54.8	52.2	54.9	44.6

Table 8: We report three correlation coefficients between the human and LLM-evaluator correctness scores for each question in TutorEval. The Pearson coefficient measures human/LLM correlation of grades, and the Spearman rank and Kendall-τ coefficients measure correlation of ranking. We run ablations on the standard TutorEval approach by (1) grading with GPT-4 without the key points, (2) grading with GPT-4 without a presentation score and (3) grading with GPT-3.5.

Table 9: Three correlation coefficients for presentation scores between human and LLM-evaluator.

Presentation correlation	TutorEval grading	Grading without key points	GPT-3.5 grading
Pearson	69.2	68.2	37.1
Spearman	67.6	67.2	36.5
Kendall-τ	63.2	62.8	34.2

In order to compare our GPT-4-grading approach to TutorEval with other evaluations using GPT-4 as an evalutor, such as Li et al. (2023a), we compute GPT-4 preference votes for three pairs of models:

1. Llemma-34B-MathMix and Mistral-7B-V2

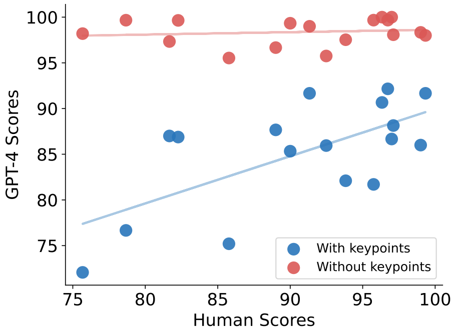


Figure 6: For each human annotator, we plot the score attributed to GPT-4 against the score attributed to GPT-4 by itself, with and without the key points. We see that the key points make it possible for GPT-4 to grade its own generations fairly.

2. Mistral-7B-V2 and Llemma-7B-32K-MathMix
3. Llemma-7B-32K-MathMix and Llemma-7B-32K-UltraChat.

For each pair of models A and B, we prompt GPT-4 with the outputs of the two models for each TutorEval question twice, switching the order between the two models. We prompt GPT-4 to decide if Model A or B is more correct, or if there is a tie. When GPT-4 changes does not identify the same model during both rounds, we count this as a tie.

Win and tie rates are collected in Table 10. We find that the ranking that results from preference votes coincides with the ranking of Table 1, even for models which rank relatively closely, like Mistral-7B-V2 and Llemma-7B-32K-MathMix. We conclude that grading is as reliable as preference voting, while being considerably cheaper and faster to run.

	win / Tie / win (%)
Llemma-34B-MathMix	**17.1**
VS	69.9
Mistral-7B-V2	12.9
Mistral-7B-V2	**17.9**
VS	68.0
Llemma-7B-32K-MathMix	14.1
Llemma-7B-32K-MathMix	**21.4**
VS	70.0
Llemma-7B-32K-Ultrachat	8.8

Table 10: Win-rate comparisons of various models on TutorEval questions, judged by GPT-4. The preference-based ranking matches the gradings-based ranking we obtain in Table 1.

Appendix D. TutorChat composition

Table 11: Summary statistics for TutorChat dialogues

Domains	Textbooks	Dialogues	Words
Biology	70	6K	9M
Business	77	7K	12M
Chemistry	119	12K	20M
Engineering	134	6K	10M
Geology	39	2K	4M
Human sciences	556	13K	21M
Mathematics	156	7K	12M
Medicine	91	4K	7M
Physics	71	5K	8M
Social sciences	248	12K	20M
Statistics	33	2K	3M
Workforce	91	2K	3M
Total STEM	713	45K	75M
Total	1685	78K	130M

Appendix E. TutorChat generation

We generate open-book conversations by simulating a student asking questions about this chapter. We generate closed-book conversations by showing the chapter only to the teacher and prompting the teacher to conduct a class based on the chapter. We diversify our templates by simulating different teacher and student personalities. For example, we sometimes prompt the student to be bright and keen or slow and badly behaved. In total we use 16 different templates.

TutorChat counts 15% of textbook exams, and the rest of TutorChat is evenly split between open-book and closed-book dialogues.

Below is an example of a template used for generating conversations for TutorChat. We initialise the ‘student’ and the ‘teacher’ with system prompts, and we start off the conversation by using a fixed template. We include an entire textbook chapter into this template.

Student system prompt

```
You are a student who struggles with staying focused and often finds
yourself distracted during lessons.  In your one-on-one tutoring
session, you are dealing with a topic you're not very familiar with.
You find yourself drifting off or not fully engaging with the material.
Your lack of focus sometimes leads you to ask about things you've
already covered or to miss key points in the explanation.
```

Teacher system prompt

```
You are a strict and detail-oriented teacher.  You insist on precision
and thoroughness in all answers.  Your teaching style is focused on
ensuring students understand the minutiae of every topic.  You engage
with your students actively but always emphasize the importance of
accuracy and detail in their responses.  While you are knowledgeable
and committed to student learning, your approach is rigid and exacting,
leaving no room for ambiguity or imprecise answers.
```

Template to start conversation

```
Here is a textbook chapter that you should learn for your upcoming
exam.  Read it carefully and let me know if you have any questions.
This is your main opportunity to clarify any misunderstandings, so make
the most of it!  Don't be afraid to ask silly questions, I'm here to
help!
##########
{text}
##########
Remember I'm your teacher so I'm here to answer all your questions
about the textbook!  Fire away!
```

Appendix F. TutorChat examples

Below we give two examples of dialogues in TutorChat. First we give a dialogue generated by GPT-4 about a linear algebra textbook chapter, and secondly we give an example of a GPT-3.5-generated dialogue about a biology textbook chapter.

F.1 GPT-4-generated dialogue

The following dialogue was generated using GPT-4. GPT-4 was presented with the following textbook chapter: **Click here** and was prompted to write a teacher-student dialogue simulating a class about its content.

<s>

assistant: Today we're going to discuss the concept of diagonalization of matrices. To start, let's consider a simple 2x2 matrix and determine whether it's diagonalizable. Take the matrix A = $\begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}$. Can you find the eigenvalues of this matrix?</s>

user: Uh, I'm not sure how to find eigenvalues.</s>

assistant: No problem. To find the eigenvalues, we need to solve the characteristic equation $det(A - \lambda I) = 0$, where I is the identity matrix and λ represents the eigenvalues. Can you calculate the determinant of $A - \lambda I$?</s>
user: I think so. I subtract λ from the diagonal entries, right? So it becomes $\begin{pmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{pmatrix}$.</s>
assistant: That's correct. Now find the determinant of that matrix.</s>
user: I multiply the diagonal entries and then subtract the product of the off-diagonal entries, which gives me $(3 - \lambda)(2 - \lambda) - (0)(1)$, so the determinant is $\lambda^2 - 5\lambda + 6$.</s>
assistant: Almost there, but be careful. The determinant of a 2x2 matrix is the product of the diagonal entries minus the product of the off-diagonal entries. Since the off-diagonal entries in the lower left are zero, you only need to consider the diagonal product. So what's the determinant now?</s>
user: Oh, I see. So the determinant is just $(3-\lambda)(2-\lambda)$, which simplifies to $\lambda^2-5\lambda+6$.</s>
assistant: Exactly. Now solve for λ when the determinant equals zero.</s>
user: So I set $\lambda^2-5\lambda+6 = 0$ and solve for λ. I think the solutions are $\lambda = 2$ and $\lambda = 3$.</s>
assistant: That's right, those are the eigenvalues of matrix A. Now, let's find the eigenvectors corresponding to each eigenvalue. Start with $\lambda = 2$.</s>
user: I plug $\lambda = 2$ back into the matrix $A - \lambda I$, right? So I get the matrix $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$.</s>
assistant: Correct. Now find a vector v such that $(A - 2I)v = 0$.</s>
user: So I'm looking for a vector that makes this matrix equation equal zero... I'm not sure how to do that.</s>
assistant: Think of it as a system of linear equations. What vector can you multiply by that matrix to get the zero vector?</s>
user: Well, if I take $v = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, then I get the zero vector when I multiply it by $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$.</s>
assistant: Exactly, so you found an eigenvector for $\lambda = 2$. Now find an eigenvector for $\lambda = 3$.</s>
user: I plug $\lambda = 3$ into $A - \lambda I$ and get $\begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}$. So I need a vector that will give me the zero vector when I multiply... I'm not sure what to pick.</s>
assistant: Look at the system of equations you get from that matrix. It tells you that for any vector v, the first component doesn't matter, but the second component must be the negative of the first to cancel out the 1. Can you find a vector that satisfies this condition?</s>
user: Oh, I think I get it. So if I choose $v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, it satisfies the condition.</s>
assistant: Exactly right. Now that we have the eigenvalues and eigenvectors, we can check the criteria for diagonalizability. A matrix is diagonalizable if it has enough linearly independent eigenvectors. Since our matrix is 2x2, we need 2 linearly independent eigenvectors. Do we have that?</s>
user: Yes, the vectors $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ are linearly independent.</s>

F.2 GPT-3.5-generated dialogue

<s>
assistant: Today, we will dive into the intricate world of chromatin, chromosomes, and DNA organization within the eukaryotic cell. We'll explore the distinctions between heterochromatin and euchromatin, the structure of chromosomes, the packing of DNA through supercoiling, the role of telomeres in DNA replication, and the fascinating mechanisms of chromatin loop formation. The goal is to gain a comprehensive understanding of these fundamental concepts in biochemistry.

Does this sound like a good plan for our session today? Can you follow along so far?</s>

user: I understand that you're covering some complex topics today, but I must admit that I'm not very familiar with them. I'll do my best to stay focused, but I may need some extra help to ensure I understand everything properly. Can we maybe start by revisiting the basics before delving into the more intricate details? That might help me grasp the concepts better.</s>

assistant: Absolutely, it's perfectly fine to revisit the basics before delving into the more complex details. Understanding the fundamentals is crucial for building a strong foundation in biochemistry. Let's start by revisiting the basics.

First, let's discuss the distinction between heterochromatin and euchromatin. Heterochromatin and euchromatin are two different structural forms of chromatin, which is a complex of DNA and proteins found in eukaryotic cells. Heterochromatin appears as dark, densely packed regions in the cell nucleus, while euchromatin appears as lighter, less densely packed regions.

Can you explain the differences between heterochromatin and euchromatin based on what we've just discussed?</s>

user: So, from what you've just explained, it seems that heterochromatin is the densely packed form of chromatin, while euchromatin is less densely packed. Heterochromatin appears as dark regions in the cell nucleus, and euchromatin appears as lighter regions. Heterochromatin is typically associated with gene silencing, while euchromatin is associated with active gene expression.

I hope I got that right. It's a bit tricky for me to grasp, but I'm doing my best to follow along.</s>

assistant: You've grasped the basic concepts well! Heterochromatin is indeed densely packed and associated with gene silencing, while euchromatin is less densely packed and associated with active gene expression. Understanding these distinctions is fundamental to comprehending the organization and regulation of genetic material within eukaryotic cells.

Now, let's move on to the structure of chromosomes and the organization of DNA within them. Can you explain what a chromosome is and how DNA is organized within it based on the information we've covered so far?</s>

user: I remember that chromosomes are structures made of DNA and proteins, and they contain the genetic information of an organism. The DNA within chromosomes is organized through a process called supercoiling, where the DNA strands are tightly coiled around proteins called histones. This coiling allows for the efficient packaging of the long DNA molecules into the compact structure of the chromosome.

I'm trying to connect the dots based on what we've discussed earlier. It's a bit challenging for me, but I'm doing my best to keep up.</s>

assistant: You're absolutely correct! Chromosomes are indeed structures made of DNA and proteins, and they contain the genetic information of an organism. The organization of DNA within chromosomes involves supercoiling, where the DNA strands are tightly coiled around histone proteins, allowing for efficient packaging of the genetic material.

Understanding these fundamental concepts is vital for appreciating the intricacies of DNA organization and its impact on genetic expression. Now, let's delve into the details of supercoiling and its significance in DNA metabolism. Can you explain the concept of supercoiling and its role in DNA packaging and regulation of genetic code based on our discussion?</s>

user: I think supercoiling refers to the coiling of DNA around histone proteins, which allows for the efficient packaging of the DNA into the compact structure of the chromosome. This coiling helps regulate gene expression by controlling the accessibility of the DNA to the cellular machinery that reads and transcribes the genetic code.

I hope I've understood the concept correctly. It's quite challenging for me to grasp these intricate details, but I'm trying my best to follow along.</s>

assistant: You're on the right track! Supercoiling does indeed involve the coiling of DNA around histone proteins to efficiently package the DNA into the compact structure of the chromosome. This packaging plays a crucial role in regulating gene expression by controlling the accessibility of the DNA to the cellular machinery responsible for reading and transcribing the genetic code.

Understanding the impact of supercoiling on gene regulation is essential for comprehending the intricate mechanisms of DNA organization and genetic expression.

Now, let's move on to the next topic: telomeres. Can you explain what telomeres are and their significance in DNA replication and cell division, based on what we've discussed so far?

user: Telomeres are repetitive nucleotide sequences located at the ends of chromosomes. They play a crucial role in DNA replication and cell division by protecting the ends of the chromosomes from deterioration or fusion with neighboring chromosomes. Telomeres also help in preventing the loss of genetic information during DNA replication, as the replication machinery cannot fully replicate the ends of linear DNA molecules.

I'm trying to connect the dots based on our discussion. It's challenging for me to fully engage with the material, but I'm doing my best to grasp the concepts.</s>

assistant: You're absolutely correct! Telomeres are repetitive nucleotide sequences located at the ends of chromosomes, and they serve the crucial role of protecting the chromosome ends from deterioration or fusion with neighboring chromosomes. Additionally, telomeres help prevent the loss of genetic information during DNA replication, as the replication machinery cannot fully replicate the ends of linear DNA molecules.

Understanding the significance of telomeres in DNA replication and chromosome stability is pivotal for comprehending the intricate processes that ensure genetic integrity.

Now, let's move on to the next topic: chromatin structure. Can you explain the concept of chromatin structure and its relevance to gene regulation and DNA organization, based on our discussion so far?</s>

user: I think chromatin structure refers to the organization of DNA and proteins in the nucleus of a cell. It plays a crucial role in gene regulation, as the accessibility of the DNA to the cellular machinery is influenced by the packing and modification of the chromatin structure. Changes in the chromatin structure can impact gene expression by either allowing or preventing the transcription machinery from accessing specific regions of the DNA.

I'm trying to piece together the information we've discussed. It's quite challenging for me to fully engage with the material, but I'm doing my best to understand the concepts.</s>

assistant:You've captured the essence of chromatin structure very well! Chromatin structure indeed encompasses the organization of DNA and proteins within the nucleus of a cell, and it significantly influences gene regulation. The packing and modification of chromatin structure play a crucial role in determining the accessibility of DNA to the cellular machinery, ultimately impacting gene expression.

Understanding the intricate details of chromatin structure and its impact on gene regulation is fundamental for comprehending the complexities of DNA organization and genetic expression.

Thank you for your diligent effort in grasping these challenging concepts. Your dedication to understanding the material is commendable.

Thank you! That's all for today!

Appendix G. TUTORCHAT ablations

Table 12: We control the usefulness of various sub-splits of TUTORCHAT-STEM by fine-tuning LLEMMA-7B-32K and evaluating on TUTOREVAL. All datasets contain 10K samples. The set of misleading questions in TUTOREVAL accounts for most of the performance gaps, suggesting that the model becomes sycophantic when training on certain splits.

	Non-misleading questions	Misleading questions	TUTOREVAL	CLOSED-BOOK
GPT-4 generations	46.7	33.4	45.6	42.6
GPT-3.5 generations	46.8	25.8	43.9	38.4
Open-book dialogues	45.5	27.0	43.3	41.6
Closed-book dialogues	42.2	22.0	39.5	40.1
Strong student	47.2	20.1	43.8	43.3
Weak student	48.0	32.5	46.0	41.6
Random sample	47.9	30.3	45.8	45.9

Appendix H. Training details

We fine-tune Llama Touvron et al. (2023) modes, based on the Transformer architecture Vaswani et al. (2017). We use FlashAttention Dao et al. (2022); Dao (2023) as an efficient implementation of attention.

H.1 Long context training

For long-context training, we fine-tune base models for one epoch on 5B tokens from OpenWebMath with up to 32K tokens per sample. We use a batch-size of 512, a learning rate 2e-5 with a 10% warm-up, and the Adam optimizer Kingma and Ba (2015). Following Roziere et al. (2023), we set the RoPE base to 1M to facilitate long-context understanding.

We use 16 H100 GPUs to fine-tune Llemma-7B-32K on this dataset.

H.2 Dialogue fine-tuning

All base models in this paper follow the same hyperparameters for dialogue tuning. We always fine-tune for two epochs, with a batch size of 16 and a learning rate of 1e-5 and a 10% warm-up.

For fine-tuning with TutorChat, we process the GPT-generated dialogues as follows. Teacher/student dialogues are processed as assistant/user dialogues and we remove recurrent text coming from the templates used to generate dialogues. For open-book dialogues we include the chapter before the start-token `<s>` but we do not train on the text. For closed-book dialogues, we remove the chapter entirely. For textbook-exam generations, we remove the chapter and we randomly re-label the question/answer sequence as either user/assistant, or assistant/user, or we keep the question/answer labels.

In most cases, we only train the LM on the assistant tokens during loss back-propagation with next-token prediction. For textbook exams and GPT-4-generated dialogues featuring a strong student, we train on the entire conversation. This is to encourage strong question-answering capabilities. In preliminary experiments (not included here), we found that these processing choices improve question TutorEval performance by several points.

To fine-tune with MetaMath, we process the dataset by randomly concatenating 10 question/answer pairs. This forms longer documents and allows us to balance batch sizes when combining MetaMath and TutorChat. We randomly process question/answer pairs as either assistant/user, user/assistant, question/answer, or problem/solution pairs. We fine-tune the LM by backpropagating the loss on the entire sequence.

To fine-tune Llemma-7B-32K, we use one A100 GPU with 80GB memory. To fine-tune Llemma-34B, we use 32 H100 GPUs.

Appendix I. Expanded model evaluations

In this section we collect additional evaluation results for some of the models evaluated in this paper. These results give a comprehensive over-view of the strengths and weaknesses of these models. Our goal in this paper is to further the study of LMs that can be used in real-life settings, so we give a holistic over-view of the impact of training with TutorChat.

We evaluate on MBPP Austin et al. (2021) and HumanEval Chen et al. (2021) for coding tasks. We use the LM Evaluation Harness Gao et al. (2023) for all non-coding tasks.

Table 13: Benchmark results for different fine-tuned versions of LLEMMA-7B-32K. † Fine-tuning with MetaMath yields 0% on MATH because the model is unable to output answers which are detected by the official MATH parsing script. The model has over-fit to GSM8K and is unusable for other tasks.

	GSM8K (8)	MATH (4)	Average
LLEMMA-7B	31.9	12.9	22.4
LLEMMA-7B-32K	28.8	13.0	20.9
+ MetaMath	60.0	$0^{\dagger}$	30.0
+ UltraChat	30.0	13.9	22.0
+ UltraChat + MetaMath	31.1	14.8	23.0
+ TUTORCHAT	28.9	14.8	21.9
+ TUTORCHAT + MetaMath	36.3	16.5	26.4
+ TUTORCHAT-STEM	28.3	14.6	21.5
+ MathMix	55.9	21.7	38.8
Llemma-34B-MathMix	51.1	21.9	36.5
Mistral-7B-V2	38.9	9.0	24.0

Table 14: Coding benchmark results for various fine-tuned versions of LLEMMA-7B-32K and LLEMMA-34B-32K models. We apply greedy decoding for pass@1 scores and temperature 0.8 for pass@10 on HumanEval and MBPP. Following the evaluation in Roziere et al. (2023), we conducted zero-shot evaluations on HumanEval and 3-shot evaluations on MBPP

	HumanEval		MBPP		
	Pass@1	Pass@10	Pass@1	Pass@10	Average
LLEMMA-7B	28.1	47.6	40.6	61.0	44.3
LLEMMA-7B-32K	28.7	54.3	40.6	59.8	45.8
+ UltraChat	26.2	51.2	31.2	55.6	41.1
+ TUTORCHAT	25	51.2	40.8	61.4	44.8
+MathMix	34.2	56.1	40.8	61.6	48.2
Mistral-7B-V2	38.1	59.1	38.2	57	48.1
CodeLlama 7B	33.5	59.6	41.4	66.7	50.3
Llemma-34B-MathMix	45.1	77.4	52.6	74.0	62.3
CodeLlama 34B	48.8	76.8	55.0	76.2	64.2

Table 15: Evaluation results for various models on non-scientific tasks. We evaluate on ScienceQA Welbl et al. (2017), PIQA Bisk et al. (2020), WinoGrande (WG) Sakaguchi et al. (2021), ARC Clark et al. (2018), HellaSwag (HS) Zellers et al. (2019), BoolQ Clark et al. (2019), LAMBADA (LBD) Paperno et al. (2016), and Natural Questions Kwiatkowski et al. (2019). We see that long-context training and fine-tuning with TutorChat and MathMix affects overall performance slightly compared to the base Llemma-7B model. We see also that Mistral is a much more well-rounded model due to its generalist training.

	SciQ	PIQA	WG	ARC-E	ARC-C	HS	LogiQA	BoolQ	LBD	NQ	MMLU	Average
k-shot	0	0	0	0	25	10	0	32	0	32	5	
Llemma-7B	94	72	62	71	47	62	30	76	61	12	46	57.6
Llemma-7B-32K	93	72	61	71	47	62	31	72	60	11	44	56.7
+ UltraChat	94	72	61	71	48	63	30	75	61	10	44	57.2
+ TutorChat	92	72	62	67	46	61	31	71	60	10	44	56.0
+ MathMix	91	71	59	65	47	60	28	76	56	10	44	55.2
Mistral-7B-V2	95	79	73	82	63	84	33	85	70	26	58	68.0
Llemma-34B-MathMix	94	75	70	72	52	74	34	86	72	20	56	64.1