ClozeMath: Improving Mathematical Reasoning in Language Models by Learning to Fill Equations[*]

Quang Hieu Pham[1†], Thuy Duong Nguyen[1†], Tung Pham[1], Anh Tuan Luu[2], Dat Quoc Nguyen[1]
[1]Qualcomm AI Research[‡]; [2]Nanyang Technological University, Singapore
[1]{hieupq, duonnguy, tungp, datnq}@qti.qualcomm.com; [2]anhtuan.luu@ntu.edu.sg

Abstract

The capabilities of large language models (LLMs) have been enhanced by training on data that reflects human thought processes, such as the Chain-of-Thought format. However, evidence suggests that the conventional scheme of next-word prediction may not fully capture how humans learn to think. Inspired by how humans generalize mathematical reasoning, we propose a new approach named ClozeMath to fine-tune LLMs for mathematical reasoning. Our ClozeMath involves a text-infilling task that predicts masked equations from a given solution, analogous to cloze exercises used in human learning. Experiments on GSM8K, MATH, and GSM-Symbolic show that ClozeMath surpasses the strong baseline Masked Thought in performance and robustness, with two test-time scaling decoding algorithms, Beam Search and Chain-of-Thought decoding. Additionally, we conduct an ablation study to analyze the effects of various architectural and implementation choices on our approach.

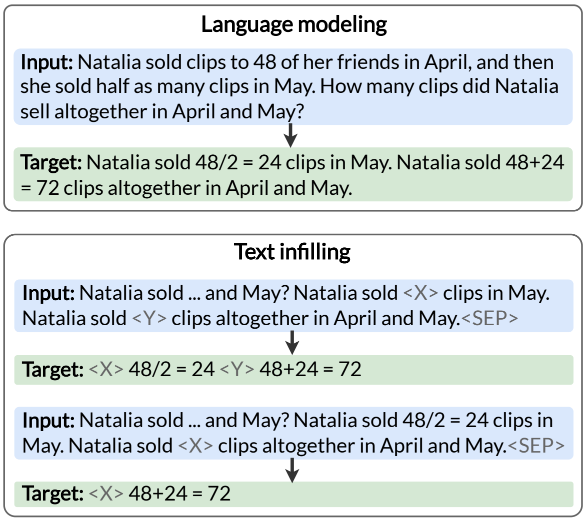

Figure 1: ClozeMath's mixture of training objectives. In text infilling, masked equation spans are replaced with special mask tokens (<X>, <Y>) and are gradually predicted from left to right. These mask tokens ensure alignment of token positions during training. A special <SEP> token separates the input and target parts, using bidirectional and causal attention masks, respectively.

1 Introduction

To mimic human reasoning in mathematical settings, current Large Language Models (LLMs) are prompted or trained to generate intermediate thinking steps before reaching a conclusion (Wei et al., 2022; Kojima et al., 2022). Some efforts focus on generating data in this format by collecting responses from humans or LLMs (Lightman et al., 2024; Yue et al., 2024) or by implicitly learning task-specific representations (Wang et al., 2024). However, while these outputs resemble the surface form of human thought, the current language modeling objective may not align with how humans learn to think (Bachmann and Nagarajan, 2024; Bender et al., 2021). When learning from a mathematical solution, it is easier for a human to generalize the main approach before working on the details, rather than memorizing which step follows another (Mason et al., 2010). Considering the input problem in Figure 1, the textual part of the target solution describes its general approach (i.e., rationales), while the equations are problem-specific.

In light of this observation, we propose enhancing the training of language models for mathematical reasoning by introducing an additional text infilling objective (Raffel et al., 2020; Tay et al., 2023b,a). In this approach, models learn to predict masked equations alongside the standard language modeling objective. This method reinforces the model's ability to infer mathematical relationships by predicting equations based on the surrounding

[*]This work was completed while all authors, except the fourth, were at Movian AI, Vietnam. All datasets and models were downloaded, trained, and evaluated using Movian AI's resources.

[†]The first two authors contributed equally to this work.

[‡]Qualcomm Vietnam Company Limited.

Findings of the Association for Computational Linguistics: ACL 2025, pages 14322–14329
July 27 - August 1, 2025

rationale, encouraging a more structured understanding of problem-solving steps.

While traditional text infilling objectives involve masking and predicting random text spans, we find that this approach disrupts the logical coherence of the unmasked parts available to the model, ultimately harming reasoning performance. Regarding model architecture, PrefixLM (Liu et al., 2018), which applies a bidirectional attention mask to the model's prompt, has been shown to work well with text infilling in T5 (Raffel et al., 2020). However, in the context of mathematical reasoning, we demonstrate that only our equation masking strategy benefits from this architectural choice.

From a different perspective, Masked Thought (Chen et al., 2024), a recent method with strong empirical results, regularizes a model to look further back into a math problem's definition by randomly corrupting the problem's solution during language model training. However, mathematical solutions may include auxiliary derived steps that are not directly connected to the problem definition. We demonstrate a common example in which Masked Thought may teach the model spurious correlations in Section 3.2. Empirically, our method also substantially outperforms Masked Thought.

In summary, our contributions are as follows: **(1)** We introduce a simple yet highly effective fine-tuning strategy, namely ClozeMath, that enhances the mathematical reasoning abilities of LLMs. **(2)** We conduct extensive experiments on two standard benchmarks, GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), including a study on ClozeMath scalability on test-time compute, and perform robustness analyses on the newly released benchmark GSM-Symbolic (Mirzadeh et al., 2025). We also show that ClozeMath outperforms the recent strong baseline Masked Thought. **(3)** We present a comprehensive ablation study that explores the impact of various architectural choices on the effectiveness of our proposed method.

2 Our ClozeMath approach

Let $\{X^i, Y^i\}_{i=1}^n$ be a training set, in which X^i is a text-based mathematical problem (e.g., X^i is the Input in the Language modeling part in Figure 1), and $Y^i = y_1^i\ y_2^i\ \ldots\ y_{m_i}^i$, where each y_m^i is a text segment or a mathematical equation (e.g., Y^i is the Target in the Language modeling part in Figure 1; then $m_i = 5$ and y_1^i is "Natalia sold", y_2^i is "48/2 = 24", y_3^i is "clips in May. Natalia sold", y_4^i is "48+24 = 72" and y_5^i is "clips altogether in April and May."). The language modeling objective is formulated as:

$$\mathcal{L}_{\text{lm}} = \sum_{i=1}^{n} P(Y^i|X^i) \tag{1}$$

Let $F^i = \{f_1^i, f_2^i, \ldots f_{|F^i|}^i\} \subset \{1, 2, ..., m_i\}$ be the set of indices such that $\forall m \in F^i$, y_m^i is a mathematical equation (e.g., from the example in the paragraph above, $F^i = \{2, 4\}$). Our text-infilling objective is formulated as follows:

$$\mathcal{L}_{\text{tf}} = \sum_{i=1}^{n} \sum_{\mathrm{M} \in \mathcal{M}(F^i)} P(y^i_{m\in\mathrm{M}}|X^i, Y^i \setminus y^i_{m\in\mathrm{M}}) \tag{2}$$

where $\mathcal{M}(F^i) = \{\{f^i_{t:|F^i|}\} \mid t = 1, \ldots, |F^i|\}$ contains combinations of equations to be masked (e.g., given $F^i = \{2, 4\}$, we have $\mathcal{M}(F^i) = \{\{2, 4\}, \{4\}\}$). This involves masking all equations and then gradually unmasking them from the first to the last, as each equation only involves computations from its previous counterparts. For groups of simultaneous mathematical transformations, such as systems of equations, we randomly mask 50% of each group.

Finally, our ClozeMath approach trains to minimize the two objectives simultaneously:

$$\mathcal{L}_{\text{ClozeMath}} = \mathcal{L}_{\text{lm}} + \mathcal{L}_{\text{tf}} \tag{3}$$

Since the number of terms in $\mathcal{L}_{\text{tf}}$ depends on the number of equations to be masked within each dataset, in practice, we balance the two training objectives by duplicating samples optimized for $\mathcal{L}_{\text{lm}}$ to maintain a roughly 50:50 sample ratio.

We implement ClozeMath with PrefixLM (Liu et al., 2018) on pre-trained decoder-only models. In PrefixLM, the prompt part's attention is bidirectional, while the target sequence uses a causal mask. To let the model learn to use the two attention patterns correctly, there is a special *<SEP>* token separating the two parts. During inference, we employ the conventional next-token prediction.

3 Experiments

3.1 Experimental setup

Datasets: We conduct experiments on two standard benchmarks for mathematical problem solving, GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). To demonstrate the context-understanding ability of our method, we

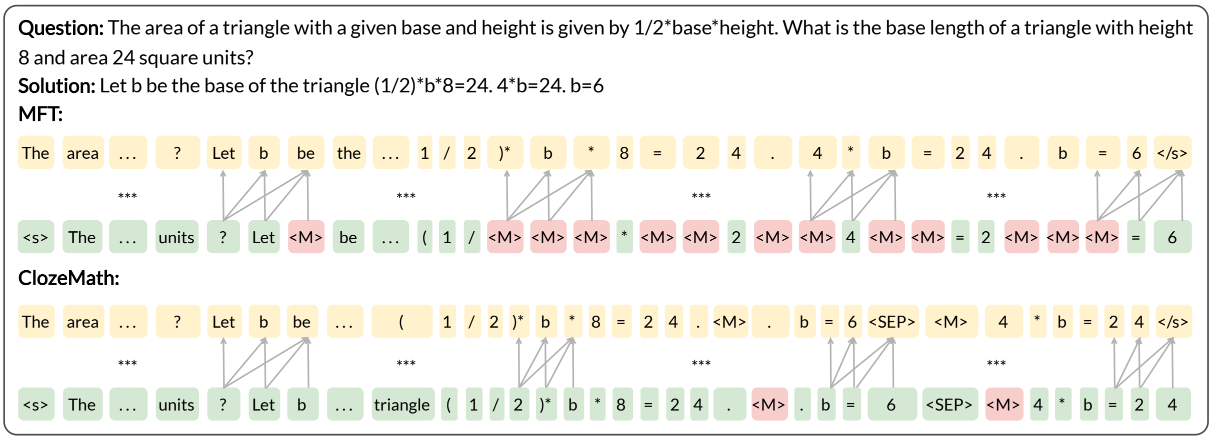

Figure 2: Implementation of MFT (Chen et al., 2024) and ClozeMath on the same sample. Here, *** denotes the unshown causal attention mask in this figure. MFT aims at learning longer-range dependencies by randomly inputting noisy mask (<M>) tokens into the solution's parts while keeping the problem definition untouched. MFT is prone to learning spurious correlation; in this example, a derived transformation might not fully perceive its predecessors.

also include the GSM-Symbolic evaluation collection (Mirzadeh et al., 2025), which varies problems from the GSM8K test set by: (1) changing names, nouns, and numbers (GSM-Sym) (2) adding a new constraint (GSM-P1), and (3) adding two new constraints (GSM-P2), as additional evaluation sets. See Appendix A for more details.

Implementation: To verify ClozeMath under a consistent fine-tuning condition, instead of continually fine-tune instruction following models, we fine-tune recent strong base LLMs, i.e models without the instruction following ability, including three generic models "Llama-3.1-8B", "Llama-3.2-3B", "Llama-3.2-1B" (Grattafiori et al., 2024), and a mathematics-specific model "DeepSeek-Math-7B-base" (Shao et al., 2024), with low-rank adaptation LoRA (rank=32) (Hu et al., 2022); and other hyper-parameters are specified in Apppendix B. Without specification, we report results of all our models and baselines obtained by **beam search** decoding with num_beams = 5.

Here, we also extend the foundation models' vocabulary and fine-tune their available reserved tokens for <*SEP*> and mask tokens.

Baseline: Our strong baseline approach for mathematical reasoning is Masked Thought Fine-tuning (Chen et al., 2024) using the same LoRA setting.

3.2 Main results

Comparison with Masked Thought Fine-tuning (MFT): Figure 2 demonstrates the MFT approach. When learning from a math problem, MFT randomly masks solution tokens while preserving their corresponding labels and the problem definition unmasked. This disruption to the solution's logical flow forces the model to generate tokens by focusing on previous unmasked information, where the problem definition serves as the largest consecutive informative segment. In the example in Figure 2, the solution's first step formulates all information from the problem definition, with later steps being direct transformations of this initial step. As a result, MFT's masking strategy might learn spurious correlations when transformation steps are closely interconnected without direct links to the problem definition. For instance, the model is forced to predict "$4 * b = 24$" while the definition of variable "b" is masked in the prior context.

In contrast, ClozeMath's equation text-infilling strategy provides the model with a general solution plan and teaches it the correlation between consecutive transformation steps, which is more reasonable. Empirically, Table 1 demonstrates that

	Settings	DeepSeek-Math	LLama-3.1-8B	LLama-3.2-3B	LLama-3.2-1B
GSM8K	Base	59.21	49.58	17.66	4.62
	MFT	70.20	64.82	45.03	21.15
	ClozeMath	**74.22**	**70.00**	**53.15**	**27.89**
MATH	Base	31.68	18.06	4.54	3.84
	MFT	33.42	20.94	10.14	4.52
	ClozeMath	**36.90**	**22.88**	**11.18**	**5.00**

Table 1: Overall results on GSM8K and MATH datasets. "DeepSeek-Math" and "MFT" stand for "DeepSeek-Math-7B-base" and the baseline Masked Thought Fine-tuning, respectively. "Base" denotes the results obtained with few-shot prompting (5-shot for GSM8K and 4-shot for MATH) on the base models (without fine-tuning) using the well-known "lm-evaluation-harness" framework (Gao et al., 2024).

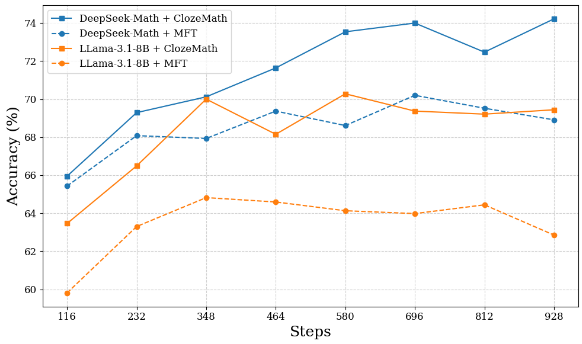

Figure 3: GSM8K test accuracies w.r.t. the numbers of training steps.

Model	MFT	ClozeMath
DeepSeek-Math	76.50	**77.10**
LLama-3.1-8B	74.30	**75.97**
Llama-3.2-3B	55.27	**57.39**
Llama-3.2-1B	28.80	**31.84**

Table 2: Experimental results on GSM8K with Chain-of-Thought decoding, a type of self-consistency decoding with the number of generated samples = 9 (Wang and Zhou, 2024).

ClozeMath substantially outperforms MFT on both GSM8K and MATH datasets. Furthermore, Figure 3 shows that ClozeMath is also more sample efficient than MFT, consistently showing better performance across all recorded training checkpoints. We put some qualitative examples in Appendix C.

Model's generation to test-time scaling: Besides sample efficiency during training, the ability to leverage more computational budget during inference time has been proven to be an important aspect that contributes to the success of recent reasoning models (OpenAI, 2024; Snell et al., 2025). To prove the efficiency of our method during inference time, we implement a type of self-consistency decoding method, namely Chain-of-thought (CoT) decoding (Wang and Zhou, 2024). For each test example, CoT decoding samples k answers starting with the first k most probable tokens. CoT decoding then computes and aggregates by summing sampled answers' scores based on the model's confidence in answer tokens. Specifically, on GSM8K, answer tokens are numerical tokens after the annotation string *####*. The answer with the highest aggregated score is the model's final answer. However, due to inconsistencies in models' tokenizer decoding and encoding processes, where a sequence of token IDs may be re-encoded differently after decoding, the default output format of the MATH dataset poses a compatibility issue. Specifically, numerical answers in MATH are wrapped inside `\boxed{}`, which may not always appear at the end of the output sequence. This introduces a risk of incorrect alignment for scoring during CoT decoding, making the format incompatible with the method's confidence score computations. Consequently, we limit our evaluation on GSM8K only, with $k = 9$ sampled answers per question. As shown in Table 2, the results consistently demonstrate that ClozeMath achieves better scaling than MFT with the same inference-time computation budget.

Models' robustness on GSM-Symbolic: We further evaluate the models fine-tuned using GSM8K and MATH with MFT and ClozeMath on 5K GSM-Symbolic examples and present the results in Table 3. Overall, ClozeMath notably outperforms MFT across different evaluation sets. For DeepSeek-Math and Llama-3.1-8B trained on GSM8K, ClozeMath achieves a 5.0% improvement (49.25% vs. 44.25%) and a 6.35% improvement (47.65% vs. 41.30%) over MFT on GSM-P1, where an added constraint shifts the problem distribution from GSM8K. Similarly, the same trends are observed with Llama-3.2-3B and Llama-3.2-1B, indicating that ClozeMath generalizes well across different model scales. On GSM-P2, which is considerably harder than GSM8K, the models fine-tuned with both methods on GSM8K show only a slight performance gap. However, when trained on MATH, which is less similar to GSM-Symbolic than GSM8K, the gap between ClozeMath and MFT becomes clearer on GSM-P2: 2.2% (16.9% vs. 14.7%) for DeepSeek-Math and 1.0% (8.5% vs. 7.5%) for Llama-3.1-8B. We hypothesize that the small performance gap between ClozeMath and MFT on Llama-3.2-1B, trained on the MATH dataset, is due to its high complexity and broad domain coverage, making it challenging for a 1B model to learn effectively.

3.3 Ablation study

We examine the effectiveness of our equation masking strategy and architectural choices, as described in Section 2, using DeepSeek-Math-7B on GSM8K, and present obtained results in Table 4.

W/o Text-infilling (71.79%) refers to the variant optimized with only the language modeling objective $\mathcal{L}_{\text{lm}}$ using PrefixLM, without the text filling objective. Table 4 shows the notable contribution of our text infilling objective (i.e., improving the

Train.	Evaluation	Settings	DeepSeek-Math	Llama-3.1-8B	Llama-3.2-3B	Llama-3.2-1B
GSM8K training set	GSM-Sym	MFT	63.05	60.90	40.45	18.55
		ClozeMath	**65.65**	**63.40**	**46.70**	**23.95**
	GSM-P1	MFT	44.25	41.30	22.5	8.30
		ClozeMath	**49.25**	**47.65**	**28.6**	**11.25**
	GSM-P2	MFT	**20.20**	22.30	6.80	1.30
		ClozeMath	20.00	**22.50**	**10.20**	**3.20**
	Overall	MFT	46.96	45.34	26.54	11.00
		ClozeMath	**50.08**	**48.92**	**32.16**	**14.72**
MATH training set	GSM-Sym	MFT	55.50	39.85	17.40	4.10
		ClozeMath	**58.35**	**45.85**	**25.10**	4.05
	GSM-P1	MFT	35.00	21.80	7.85	1.40
		ClozeMath	**40.25**	**23.90**	**8.45**	**1.60**
	GSM-P2	MFT	14.70	7.50	2.30	1.00
		ClozeMath	**16.90**	**8.50**	1.80	1.00
	Overall	MFT	39.14	26.16	10.56	2.40
		ClozeMath	**42.82**	**29.60**	**13.78**	**2.46**

Table 3: Results on GSM-Symbolic evaluation sets for the fine-tuned models used to report scores in Table 1.

Model	GSM8K
$\text{ClozeMath}_{\text{DeepSeek-Math}}$	**74.22**
W/o Text infilling	71.79
W/o PrefixLM	72.71
W/o Text-infilling & W/o PrefixLM	71.57
W/o Equation masking	71.19

Table 4: Ablation study results.

performance from 71.79% to 74.22%). **W/o PrefixLM** (72.71%) refers to the variant that employs causal language modeling (CausalLM) instead of PrefixLM. Therefore, **W/o PrefixLM & W/o Text-infilling** (71.57%) represents the conventional **instruction tuning** setting (**IT**), where only $\mathcal{L}_{\text{lm}}$ is optimized with CausalLM.

Raffel et al. (2020) point out that a causal attention mask limits the model's state representation because the prefix, such as a problem definition, is always included in the model's context. However, when trained exclusively with IT, PrefixLM only slightly improves the model's accuracy from 71.57% to 71.79%. This suggests that the model struggles to comprehend problem definitions. In contrast, when combined with our equation text-infilling objective, PrefixLM substantially boosts the model's performance, surpassing the standard IT setup by 2.65% (71.57% $\rightarrow$ 74.22%). Note that without PrefixLM, the result obtained with our equation infilling objective is still higher than that of the IT setting by 1.14% (72.71% vs. 71.57%). This confirms that the presence of textual rationales alongside the problem definition in the model's prefix during training makes learning to fill equations easier. As a result, the model learns to fill in equations more effectively and develops a deeper and more structured understanding of each problem.

W/o Equation masking: To support our claim, we compare our masking strategy to a setting where the text infilling objective is optimized over randomly masked spans instead of mathematical equations. Specifically, we randomly mask both short and long token spans in each example. Depending on the span length used, we mask out either 15% (for short spans) or 50% (for long spans) of each solution. This approach mimics the equation masking strategy by considering the number of masked tokens in each solution. Table 4 shows that random masking disrupts the logical meaning of the textual parts of mathematical solutions, substantially reducing the model's performance from 74.22% to 71.19%.

4 Conclusion

To better align with human-like mathematical reasoning, we introduce ClozeMath, an approach that trains language models to predict masked equations using a text-infilling objective alongside the standard language modeling objective. We highlight the pitfall of learning spurious correlations with MaskedThought, a strong current baseline, and demonstrate the consistently superior performance of ClozeMath across various tests of performance and robustness.

Limitations

In this work, we focus specifically on mathematical reasoning. However, we believe our approach has potential applications beyond this domain. As models are increasingly trained to leverage external tools (Paranjape et al., 2023), they begin to exhibit human-like reasoning patterns that incorporate tool usage. Investigating the effectiveness of our method in these broader contexts would be a promising direction for future research. Due to limited computational resources, and in line with prior work such as Masked Thought (Chen et al., 2024), we evaluate our approach using LLMs with fewer than 10 billion parameters. Future studies could explore how our method scales when applied to larger foundational models.

Acknowledgement

We extend our thanks to Linh The Nguyen (v.linhnt140@vinai.io) for his help in setting up the evaluation framework.

References

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The Pitfalls of Next-Token Prediction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 2296–2318.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, page 610–623.

Changyu Chen, Xiting Wang, Ting-En Lin, Ang Lv, Yuchuan Wu, Xin Gao, Ji-Rong Wen, Rui Yan, and Yongbin Li. 2024. Masked Thought: Simply Masking Partial Reasoning Steps Can Improve Mathematical Reasoning Learning of Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5872–5900.

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. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint*, arXiv: 2110.14168.

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. 2024. A framework for few-shot language model evaluation.

Aaron Grattafiori et al. 2024. The Llama 3 Herd of Models. *arXiv preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems: Datasets and Benchmarks Track*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the Tenth International Conference on Learning Representations*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Proceedings of the Thirty-sixth Conference on Neural Information Processing Systems*, pages 22199–22213.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by Summarizing Long Sequences. In *Proceedings of The Sixth International Conference on Learning Representations*.

J. Mason, L. Burton, and K. Stacey. 2010. *Thinking Mathematically*. Pearson.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. In *Proceedings of the Thirteenth International Conference on Learning Representations*.

OpenAI. 2024. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`. Accessed May 18, 2025.

Bhargavi Paranjape, Scott M. Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *ArXiv*, abs/2303.09014.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint*, arXiv:2402.03300.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *Proceedings of The Thirteenth International Conference on Learning Representations*.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023a. UL2: unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Yi Tay, Jason Wei, Hyung Chung, Vinh Tran, David So, Siamak Shakeri, Xavier Garcia, Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc Le, and Mostafa Dehghani. 2023b. Transcending Scaling Laws with 0.1% Extra Compute. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1471–1486.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. 2024. Guiding Language Model Reasoning with Planning Tokens. In *Proceedings of the First Conference on Language Modeling*.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *Proceedings of The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the Thirty-sixth Conference on Neural Information Processing Systems*, pages 24824–24837.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning. In *Proceedings of the Twelfth International Conference on Learning Representations*.

A Datasets

GSM8K: The dataset contains $8,752$ middle school math problems ($7,473$ training, $1,319$ test). In general, problems in GSM8K involve using only basic arithmetic operations ($+, -, \times, \div$). The dataset's *main* configuration features step-by-step textual solutions with annotations of calculations provided in each solution's reasoning step.

MATH: The dataset consists of $12,500$ problems ($7,500$ training, $5,000$ test) and is significantly more challenging than GSM8K, covering topics like calculus, geometry, probability, and number theory. Its solutions include equations with LaTeX annotations and Asymptote vector graphics.

GSM-Symbolic: The dataset is constructed from 100 templates based on GSM8K's test problems by introducing common names, nouns, and numbers in both problems and solutions as symbolic variables to generate a diverse set of problems. The initial release includes three difficulty levels: (i) GSM-Sym, which only replaces variables in the original problems; (ii) GSM-P1, with a new constraint added to the problems before replacing variables; and similarly (iii) GSM-P2, with two new constraints. To balance cost and diversity, we sample 5000 problems from the GSM-Symbolic dataset for evaluation.

B Training & Inference configurations

We train all of our models and the baseline with LoRA adapters (rank=32), learning_rate=5e-5, and a cosine learning rate scheduler, on a machine with a single 40GB NVIDIA A100 GPU.

C Qualitative examples

We select some representative examples to demonstrate cases where ClozeMath performs better than MFT, such as a better understanding of numeric values, as illustrated in Figures 4 and 5, and the prevention of hallucination, as shown in Figure 6.

Demonstration for ClozeMath helps prevent Information Misinterpretation Error

Question input: Kayla is having her birthday party at a movie theater. The fee to rent the theater is $125 for a party of 20, plus $6 for each additional guest. Kayla invited her 25 classmates and the 7 girls in her dance class, as well as 13 family members. Only 4 people said they could not come. How much will the party cost?

Gold target solution: Kayla invited 25 classmates + 7 dance class friends + 13 family members = 45 people invited. Of the 45 people invited – 4 people cannot come = 41 people attending the party. Of the 41 people attending – 20 are included in the fee = 21 additional guests not included in the fee. The cost for 21 additional guests is 21 x $6 = $126. The total cost for the party is the $125 fee + $126 for additional guests = $251. #### 251

MFT solution output: Kayla invited 25 classmates + 7 girls in her dance class + 13 family members = 45 people. 45 people - 4 people who could not come = 41 people came to the party. 41 people x $6 per person = $246 for additional guests. 20 people x $125 per person = $2500 to rent the theater. 246 + $2500 = $2746 for the party. #### 2746

ClozeMath solution output: Kayla invited 25 + 7 + 13 = 45 people. There will be 45 - 4 = 41 people at the party. There will be 41 - 20 = 21 additional people at the party. The additional people will cost 21 x $6 = $126. The party will cost $125 + $126 = $251. #### 251

Figure 4: In this example, the Deepseek-Math model trained with MFT does not understand that the $6 prices are only applied for the number of people exceeding 20 (highlighted in green in the "Question input:").

Demonstration for ClozeMath helps prevent Logical Reasoning Error

Question input: Toulouse has twice as many sheep as Charleston. Charleston has 4 times as many sheep as Seattle. How many sheep do Toulouse, Charleston, and Seattle have together if Seattle has 20 sheep?

Gold target solution: If Seattle has 20 sheep, Charleston has 4 * 20 sheep = 80 sheep. Toulouse has twice as many sheep as Charleston, which is 2 * 80 sheep = 160 sheep. Together, the three has 20 sheep + 160 sheep + 80 sheep = 260 sheep #### 260

MFT solution output: Toulouse has 2 * 20 = 40 sheep. Charleston has 4 * 20 = 80 sheep. Together, Toulouse, Charleston, and Seattle have 40 + 80 + 20 = 140 sheep. #### 140

ClozeMath solution output: Charleston has 4 * 20 sheep = 80 sheep. Toulouse has 2 * 80 sheep = 160 sheep. Together, Toulouse, Charleston, and Seattle have 160 sheep + 80 sheep + 20 sheep = 260 sheep. #### 260

Figure 5: In this example, the Deepseek-Math model trained with MFT misunderstood that "Toulouse" has twice the number of sheep as "Seattle," while the double relationship is actually between Toulouse's number of sheep and Charleston's number of sheep.

Demonstration for ClozeMath helps prevent Irrelevant Information Error

Question input: Claire makes a 3 egg omelet every morning for breakfast. How many dozens of eggs will she eat in 4 weeks?

Gold target solution: She eats 3 eggs every day and there are 7 days in a week so she eats 3*7 = 21 eggs a week. After 4 weeks she will have eaten 4*21 = 84 eggs. There are 12 eggs in 1 dozen and she'll eat 84 eggs so that's 84/12 = 7 dozen eggs. #### 7

MFT solution output: She eats 3 eggs a day and there are 7 days in a week so she eats 3*7 = 21 eggs a week. There are 4 weeks in a month so she eats 21*4 = 84 eggs a month. There are 12 months in a year so she eats 84*12 = 1008 eggs a year. There are 12 eggs in a dozen so she eats 1008/12 = 84 dozens of eggs a year #### 84

ClozeMath solution output: She eats 3 eggs a day and there are 7 days in a week so she eats 3*7 = 21 eggs in a week. She eats 21 eggs in a week and there are 4 weeks in 4 weeks so she eats 21*4 = 84 eggs. There are 12 eggs in a dozen and she eats 84 eggs so she eats 84/12 = 7 dozens of eggs. #### 7

Figure 6: In this example, the Deepseek-Math model trained with MFT incorrectly hallucinates that "There are 12 months in a year..." even though the question does not ask for this information, leading to an incorrect final calculation.