Logical Reasoning with Outcome Reward Models for Test-Time Scaling

Ramya Keerthy Thatikonda[∀] **Wray Buntine**[∀,∃] **Ehsan Shareghi**[∀]

[∀]Department of Data Science & AI, Monash University
[∃]College of Engineering and Computer Science, VinUniversity

Abstract

Logical reasoning is a critical benchmark for evaluating the capabilities of large language models (LLMs), as it reflects their ability to derive valid conclusions from given premises. While the combination of test-time scaling with dedicated outcome or process reward models has opened up new avenues to enhance LLMs performance in complex reasoning tasks, this space is under-explored in deductive logical reasoning. We present a set of Outcome Reward Models (ORMs) for deductive reasoning. To train the ORMs we mainly generate data using Chain-of-Thought (CoT) with single and multiple samples. Additionally, we propose a novel tactic to further expand the type of errors covered in the training dataset of the ORM. In particular, we propose an echo generation technique that leverages LLMs' tendency to reflect incorrect assumptions made in prompts to extract additional training data, covering previously unexplored error types. While a standard CoT chain may contain errors likely to be made by the reasoner, the echo strategy deliberately steers the model toward incorrect reasoning. We show that ORMs trained on CoT and echo-augmented data demonstrate improved performance on the FOLIO, JustLogic, and ProverQA datasets across four different LLMs.[1]

1 Introduction

Logical reasoning in large language models (LLMs) has primarily been studied as a symbolic task using in-context learning (Matthew Lam et al., 2024; Pan et al., 2023; Ye et al., 2023; Olausson et al., 2023), and fine-tuning (Thatikonda et al., 2024; Qi et al., 2025). Current state-of-the-art techniques in reasoning which bundle test-time scaling (Brown et al., 2024; Snell et al., 2024) with Process or Outcome Reward Models (Wang et al., 2024; Lightman et al., 2024; Uesato et al., 2022), proven effective in math and coding, while remaining heavily underexplored for logical reasoning. This presents a significant opportunity to assess and enhance LLMs' reasoning capabilities using reward models at test-time with text-based reasoning. In this paper we explore logical reasoning with test-time scaling, demonstrating performance gains on three datasets; FOLIO (Han et al., 2024), ProverQA (Qi et al., 2025), and JustLogic (Chen et al., 2025) when combined with verification via Outcome Reward Models (ORMs).

Outcome Reward Models (ORMs) enable verification of the entire reasoning sequence by assigning a confidence score to the final output. A central challenge in training ORMs is acquiring high-quality, diverse training data. To address this, we generate multiple Chain-of-Thought (Wei et al., 2022) reasoning candidates via sampling. Compared to an ORM trained on a single sample per reasoning question, this facilitates more effective re-ranking of solutions during inference.

To further enrich the training data, we leverage the echoing behavior of LLMs - where models tend to align their reasoning with user-provided answers. This phenomenon can introduce hallucinated reasoning, which, when subsequently flagged as incorrect through a second-level filtering process, results in a diverse set of flawed reasoning paths. We show that incorporating these echoed errors into the training set helps the ORM learn to better distinguish between valid and invalid reasoning trajectories during use with test-time scaling.

Our contributions are as follows: (1) We demonstrate that training ORMs with multiple sampled CoT candidates (per example) significantly improves the reliability of the ORM on deductive reasoning tasks. (2) We show that augmenting CoT training data with data generated by echoed-error further enhances the resulting ORMs' accuracy. (3) We compare ORM models on training data size, format, and reward distribution.

[1]Code is available at https://github.com/RamyaKeerthy/LogicORM

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26114–26124
November 4-9, 2025

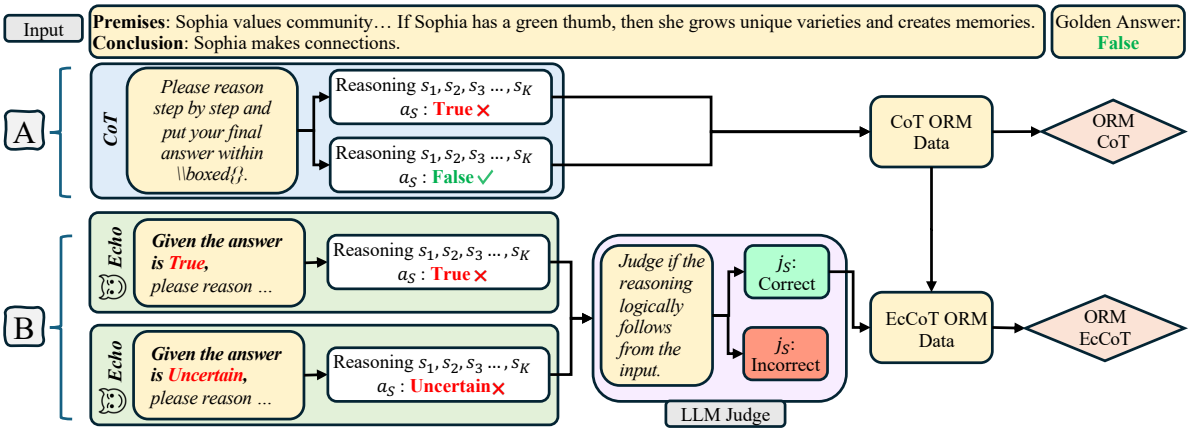

Figure 1: Methodology for generating training data for CoT and EcCoT ORMs. **A.** The LLM is prompted to generate CoT reasoning, and both correct and incorrect answers are used to construct the ORM dataset. **B.** The LLM is provided with a misleading answer to elicit reasoning. The resulting incorrect reasoning trajectories are then passed back to the LLM for evaluation. If the LLM fails to recognize its error, the trajectory is added to the dataset.

2 Outcome Reward Model for Logic

In outcome supervision, the reward model evaluates the entire reasoning sequence and assigns a final score reflecting the quality of the outcome. During inference, this score can be used to re-rank the candidate solutions generated by a large language model (LLM). Following Lightman et al. (2024) recipe for mathematical reasoning, to train an Outcome Reward Model (ORM), each reasoning trace is compared against a gold-standard label: a positive reward is given when the final output matches the correct answer, and a negative reward is assigned otherwise. The ORM learns to map reasoning sequences to these rewards, producing logits that can be used to rank candidate outputs during inference.

We explore two types of data generation strategies for training ORMs: Outcome supervision on standard Chain-of-Thought (CoT) reasoning, and Outcome supervision on Echo Chain-of-Thought (EcCoT) - a variant that incorporates the LLM's echoing behavior to produce additional reasoning paths. See Figure 1 for an overview.

CoT ORM Data Generation. For standard CoT data generation, we prompt an LLM with the context and question to generate step-by-step reasoning, with the final answer appended at the end. The LLM generates multiple reasoning candidates for a given question. These candidates are then parsed into individual CoT traces. We used "`Please reason step by step, and put your final answer within \\boxed{}`" as prompt.

Each candidate is labeled with a reward: positive if the final answer matches the gold label, and negative otherwise. This data is used to train a second LLM as a classifier that predicts the reward score based on the reasoning sequence. Our approach builds on the ideas presented in Wang et al. (2024), but differs by applying outcome supervision rather than process supervision, and by using a straightforward automated annotation to label the data.

EcCoT ORM Data Generation. LLMs often exhibit a tendency to follow user-provided answers uncritically, sometimes hallucinating or forcefully aligning their reasoning to fit an incorrect answer. We exploit this behavior to generate challenging negative examples.

In contrast to Li et al. (2025), we prompt the LLM with the instruction "`Given the answer is True, please reason step by step, and put your final answer within \\boxed{}`" for all reasoning questions. This coerces the model into producing reasoning that unjustifiably supports the provided answer (where we deliberately provided an incorrect answer, e.g. "True" where the correct answer was "False") resulting in flawed reasoning trajectories. These incorrect yet plausible reasoning sequences, referred to as echoes, are valuable for training ORMs to penalize invalid rationales.

While echoing can encourage LLMs to commit to flawed reasoning paths, we further filtered the collected echoes by prompting the LLM (i.e.,

"Judge if the reasoning logically follows from the input; respond only with Correct or Incorrect.") itself to evaluate whether the reasoning trajectories were correct. We discarded the echoed examples that the LLM identified as incorrect, as these were deemed too obvious and unlikely to occur during inference. The remaining echoes (those involving incorrect reasoning and not easily recognized as such) were retained and added to the training data alongside the CoT examples.

The statistics for CoT and EcCoT, together with sample prompt outputs, are provided in Appendix A.

3 Experimental Setup

The ORMs utilized in this study were trained using the Qwen 2.5 7B Instruct models (Hui et al., 2024) on a single A100 GPU for three epochs, with a batch size of 64 and a learning rate of 5×10^{-4}. Training was conducted using a LoRA-based PEFT configuration (Hu et al., 2022). For result annotation, we apply a step tag <extra_0>, with positive and negative outcomes indicated by '+' and '-', respectively following Zhang et al. (2025).

Logical Reasoning Datasets. We use ProverQA (Qi et al., 2025), JustLogic (Chen et al., 2025) and FOLIO (Han et al., 2024) training sets to generate data for training ORMs, sampling 8, 8 and 10 reasoning candidates per instance, respectively. The ProverQA training set comprises of 5,000 logical reasoning questions across three difficulty levels (easy, medium, and hard) with additional noise-premise variations. JustLogic training set consists of 4900 synthetically generated logical reasoning questions with difficulty spanning across 7 reasoning depths. FOLIO contains around 1,000 human-annotated deductive reasoning questions. All datasets involve deriving a conclusion based on a given premise. A well-known example of deductive reasoning is:

> Premises: All humans are mortal. Jack is a human. Conclusion: Jack is mortal. Labels: True/False/Uncertain

3.1 Data Generator

We refer to the the model used to generate the training data as the *generator*. We employ two types of generators with varying sample size.

Qwen2.5 Data Generator (*Qwen-as-a-Generator*). Our initial experiments aim to validate the hypothesis that training with 8 samples from a generator improves ORM performance compared to training with a single sample. We use the Qwen2.5-7B Instruct model to generate both small (1-generator sample) and large (8-generator samples) CoT datasets on ProverQA, resulting in two models: ORM-CoT^{small} and ORM-CoT^{large}. We further augment the large CoT dataset with Echo Chain-of-Thought (EcCoT) data, using the same LLM, to produce ORM-EcCoT^{large}.

GPT-4o Data Generator (*GPT4o-as-a-Generator*). We run a set of experiments by using GPT-4o to generate CoT and EcCoT data for ProverQA, FOLIO and JustLogic[2]. This results in six ORM variants, two for each dataset.

For details on the statistics of the data generated for ORMs, see Appendix A.

3.2 Reasoner

We refer to the model that generates reasoning samples at inference time using test data as the *reasoner*. The reasoner generates N samples per query using a temperature of 0.6, which are evaluated using Best-of-N sampling (Lightman et al., 2024) where out of the N samples, the sample receiving the highest score from the ORM is selected. We first evaluate the ORM models from *Qwen-as-a-Generator* using two reasoners, Qwen2.5-7B Instruct, and GPT-4o, on the ProverQA (hard) test set. Encouraged by these results, we extend our evaluation to the FOLIO and JustLogic datasets, incorporating additional reasoners (LLaMA-3.1-8B (Dubey et al., 2024) and Qwen3-8B (Yang et al., 2025)) and leveraging ORM variants trained with data from *GPT4o-as-a-Generator*.

4 Results and Discussion

4.1 Preliminary Experiments

We first establish a baseline to justify the selection of ORM training data parameters, such as generator sample size (small vs large), data generation method (CoT vs EcCoT), and generator model choice (GPT4o vs Qwen).

Sample Size. Figure 3 illustrates the performance variation of an ORM trained using CoT data generated by the Qwen model. The primary difference across configurations lies in the number of samples

[2]We apply resampling to address the large volume of echoes generated by GPT-4o for the JustLogic dataset. See Appendix B for details of the sampling and analysis.

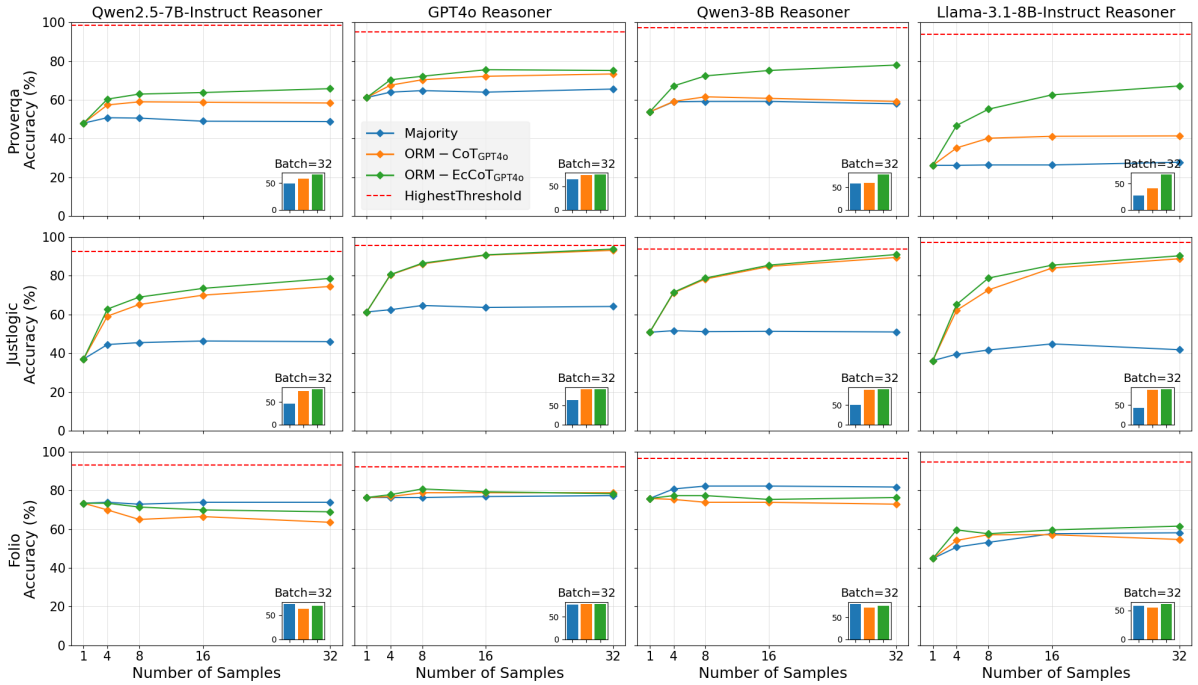

Figure 2: Performance of the ORM trained on GPT-4o-generated data using Chain-of-Thought (CoT) and Echoed formats for ProverQA, JustLogic, and FOLIO. ORM inference was performed using the Best-of-N method for both CoT and Echoed generations. The red dashed line denotes the maximum achievable accuracy assuming at least one correct rationale among the N sampled responses.

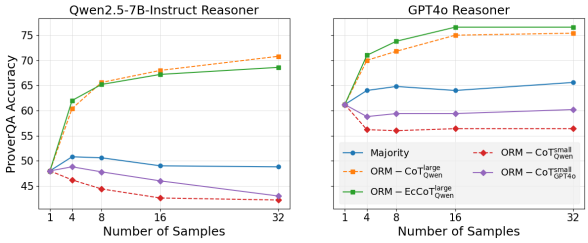

Figure 3: Performance comparison of ORM-trained models based on training data size, data generator, and reasoning generator for ProverQA dataset.

generated per question. As the number of samples increases from 1 to 8, the ORM's performance surpasses that of majority voting. This result suggests that generating multiple samples per question, while keeping the training set and questions fixed, can yield more effective training data compared to relying on a single sample.

Augmenting Echo CoT. Figure 3 presents the performance comparison between EcCoT and standard CoT using 8 generated samples. Incorporating Echo data enhances the ORM and consequently improves the performance of the GPT-4o reasoner, while the improvement is less pronounced for the Qwen reasoner. These results highlight the potential value of Echo, motivating further investigation into its effects.

Generator Model. Figure 3 examines the differences between generator models. A single-sample CoT generated by Qwen and GPT-4o reveals noticeable variation, with GPT-4o consistently outperforming Qwen. These results motivate the selection of GPT-4o as the preferred generator model (explored next).

4.2 Main Results and Analysis

Based on the observations from above, we opt for GPT4o as a generator and train ORMs for ProverQA, JustLogic, and FOLIO datasets.

ORM with Deductive Reasoning. We use two ORMs: CoT and EcCoT for all the benchmarks. These models are evaluated using test-time scaling outputs generated by four different reasoners.

For FOLIO, the relatively small training set (1,000 records) is reflected in the performance trends shown in Figure 2. Majority voting yields a wide accuracy range, from 40% to 80%, depending on the model. Qwen, and GPT family of models demonstrate strong performance on FOLIO without requiring additional verification. In contrast, LLaMA begins with lower majority voting accuracy, underscoring the potential benefits of ORM methods on this dataset. While ORM-CoT improves upon majority voting with up to 16 samples,

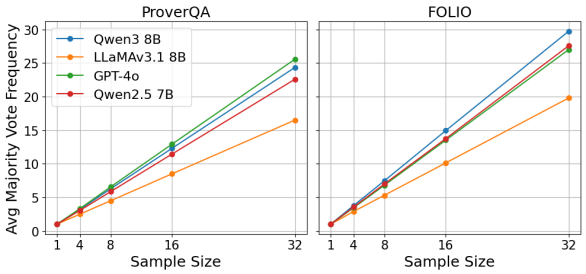

Figure 4: Average majority vote performance across varying sample sizes and benchmarks for different LLM reasoners. Notably, FOLIO achieves close to 32 correct answers on average, suggesting limited room for ORM-based improvement due to the high confidence and consistency of the reasoners.

its performance declines at higher sample sizes. In comparison, ORM-EcCoT consistently outperforms both majority voting and ORM-CoT across all sample sizes, demonstrating its robustness and effectiveness on smaller datasets like FOLIO.

The large volume of training data in the ProverQA and JustLogic datasets contributed to notable performance gains for both the CoT and Echo CoT models. These models consistently outperformed the majority vote baseline, with the most substantial improvements observed on logical reasoning benchmarks. Notably, ORMs trained with EcCoT consistently outperformed those trained with CoT on ProverQA, reinforcing our hypothesis that incorrect rationales echoed by the LLM can be leveraged effectively.

Highest Threshold (HT). To further analyze performance limits, we measure HT, representing the maximum achievable accuracy assuming at least one correct rationale exists among the N sampled responses. For JustLogic, the HT is nearly equivalent to the performance of the CoT-trained ORM, which explains the limited impact of EcCoT in this case, as the CoT ORM has already reached a performance ceiling. In contrast, the other two benchmarks demonstrate a noticeable gap between ORM performance and their respective HT values. This indicates untapped potential and suggests that a well-designed verification mechanism alone could drive substantial gains in reasoning accuracy, making it a promising direction for future research.

Majority Vote Frequency. To address discrepancies observed in the results, we analyzed the majority vote frequency across different sample sizes. This metric captures the average number of correct rationales generated by the reasoner across N samples. Figure 4 presents the majority vote frequency for the three benchmarks, which we directly relate to the performance trends shown in Figure 2. In the case of FOLIO, the reasoning paths are correct in nearly 90% of the samples, suggesting that ORM has limited room for improvement over the majority vote. In contrast, the other two benchmarks show a lower proportion of correct answers per sample, providing ORM with a more diverse set of reasoning paths to select from.

In addition to these results, we present two ablation studies (Appendix C). The first examines the effect of using EcCoT versus CoT with larger sample sizes. The second analyzes the impact of ORM on reasoners of different sizes (i.e., Gemma3- 1, 4, and 12B variants (Kamath et al., 2025)), highlighting the benefits of reward models for smaller language models in test-time settings.

5 Conclusion

In this work, we propose the use of outcome reward models (ORMs) supervised on the final outputs of reasoning paths as a framework for exploring test-time scaling in text-based reasoning. We present a diverse set of ORMs trained on varying model sizes and configurations, and evaluate their performance on three logical reasoning benchmarks - FOLIO, JustLogic, and ProverQA. To enrich training data, we advocate for sampling multiple Chain-of-Thought (CoT) responses and incorporating Echo-based augmentations. Our results provide strong empirical support for this approach. Future work may explore process reward models to assess the correctness of intermediate reasoning steps.

6 Limitations

Considering the use of GPT-4o models, we acknowledge the inherent uncertainty associated with data generation. While these models are capable of producing high-quality outputs, they remain susceptible to hallucinations, inconsistencies, and spurious correlations, especially when prompted to generate complex reasoning chains.

In this work, we explored Outcome supervision, focusing only on the correctness of the final answer without explicitly verifying the validity or faithfulness of the entire reasoning process. This approach can overlook intermediate errors that may still lead to the correct final answer, thus introducing a risk of reinforcing flawed or superficial reasoning patterns during training.

References

Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.

Michael K Chen, Xikun Zhang, and Dacheng Tao. 2025. Justlogic: A comprehensive benchmark for evaluating deductive reasoning in large language models. *arXiv preprint arXiv:2501.14851*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander R. Fabbri, Wojciech Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22017–22031. Association for Computational Linguistics.

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 *International Conference on Learning Representations*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-coder technical report. *CoRR*, abs/2409.12186.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Róbert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucinska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, and Ivan Nardini. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. 2025. START: self-taught reasoner with tools. *CoRR*, abs/2503.04625.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Long Hei Matthew Lam, Ramya Keerthy Thatikonda, and Ehsan Shareghi. 2024. A closer look at tool-based logical reasoning with LLMs: The choice of tool matters. In *Proceedings of the 22nd Annual Workshop of the Australasian Language Technology*

Association, pages 41–63, Canberra, Australia. Association for Computational Linguistics.

Theo Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5153–5176. Association for Computational Linguistics.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 3806–3824. Association for Computational Linguistics.

Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. 2025. Large language models meet symbolic provers for logical reasoning evaluation. *CoRR*, abs/2502.06563.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.

Ramya Keerthy Thatikonda, Jiuzhou Han, Wray L. Buntine, and Ehsan Shareghi. 2024. Strategies for improving nl-to-fol translation with llms: Data generation, incremental fine-tuning, and verification. *CoRR*, abs/2409.16461.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *CoRR*, abs/2211.14275.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.

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. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. Satlm: Satisfiability-aided language models using declarative prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

A Data Generation

For CoT (Chain-of-Thought) data generation, we use a single prompt: "`Please reason step by step, and put your final answer within \\boxed{}`". This elicits a reasoning path with the final answer appearing at the end. For Echo generation, we slightly modify the prompt to: "`Given the answer is True, please reason step by step, and put your final answer within \\boxed{}`". In Tables 1-4, the Echo column displays generations for each label, regardless of the ground truth. When prompted to Echo, the number of incorrect outputs increases, suggesting not only echoed reasoning but also the presence of other ambiguous or flawed reasoning paths.

A sample for comparison of reasoning outcomes for a echoed prompt and regular CoT is presented in Table 6.

Mode	Echo	Total	Correct	Incorrect
CoT 0-shot	–	39919	29962 (75%)	9957 (25%)
Echo 0-shot	True	39935	29887 (75%)	10048 (25%)
	False	39925	29081 (73%)	10844 (27%)
	Uncertain	39961	27291 (68%)	12670 (32%)
Echo-CoT	–	46469	29962 (64%)	16507 (36%)

Table 1: ORM training data for ProverQA dataset using Qwen.

Mode	Echo	Total	Correct	Incorrect
CoT 0-shot	–	10009	7383 (74%)	2626 (26%)
Echo 0-shot	True	10010	6552 (65%)	3458 (35%)
	False	10010	5459 (55%)	4551 (45%)
	Uncertain	10004	4918 (49%)	5086 (51%)
Echo-CoT	–	19105	7383 (39%)	11722 (61%)

Table 2: ORM training data for FOLIO dataset using GPT4o.

Mode	Echo	Total	Correct	Incorrect
CoT 0-shot	–	39998	34486 (86%)	5512 (14%)
Echo 0-shot	True	40000	32865 (82%)	7135 (18%)
	False	39996	25725 (64%)	14271 (36%)
	Uncertain	39968	19385 (49%)	20583 (51%)
Echo-CoT	–	63278	34486 (54%)	28792 (46%)

Table 3: ORM training data for ProverQA dataset using GPT4o.

B Resampling of Echo Data

Unlike ProverQA and FOLIO, GPT-4o produces a significantly larger number of echoed rationales in the JustLogic dataset (Table 4), resulting in a pronounced imbalance between correct and incorrect rationales. To address this, we adopt a weighted resampling approach that promotes both diversity and representational balance. Resampling is guided by two criteria: (1) BLEU-based diversity and (2) frequency of rationales per question.

To quantify the diversity of Echo-generated outputs relative to Chain-of-Thought (CoT) generations, we introduce a BLEU-based relative ranking metric. For each Echo response, we compute the BLEU score against its corresponding CoT responses. We then derive a percentile rank to assess the diversity of each Echo response within the group of candidates generated for the same question.

Mode	Echo	Total	Correct	Incorrect
CoT 0-shot	–	39197	27918 (74%)	11279 (26%)
Echo 0-shot	True	39199	21575 (55%)	17624 (45%)
	False	39199	20182 (51%)	19017 (49%)
	Uncertain	39188	14407 (37%)	24781 (63%)
Echo-CoT	–	49197	27918 (40%)	41279 (60%)

Table 4: ORM training data for JustLogic dataset using GPT4o. 10,000 records are sampled from the total echo records to preserve the final distribution of correct and incorrect values.

Formally, let the dataset comprise:

- **Echo**: A set of generated hypotheses $\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$
- **CoT**: A set of reference generations $R_i = \{r_{i1}, r_{i2}, \ldots, r_{iM}\}$ corresponding to the same input x_i with sample size M.

For each Echo record $h_i \in \mathcal{H}$, where h_i is associated with input x_i, compute the BLEU score B_i using the set of CoT generations R_i as references:

$$B_i = \text{BLEU}(h_i, R_i)$$

Let $\mathcal{G}_i \subseteq \mathcal{H}$ denote the set of all Echo hypotheses associated with a shared record identifier i (i.e., same input x_i). Define the group-wise percentile rank of each BLEU score B_i within group $\mathcal{G}_i$ as:

$$P_{\text{BLEU},i} = 1 - \frac{\text{rank}(B_i \mid \mathcal{G}_i)}{|\mathcal{G}_i|}$$

where $\text{rank}(B_i \mid \mathcal{G}_i)$ is the ascending rank of B_i within group $\mathcal{G}_i$, and $|\mathcal{G}i|$ is the group size. A higher $P\text{BLEU}, i \in [0, 1]$ indicates greater diversity relative to other hypotheses in the same group.

To reduce the sampling bias introduced by over-represented questions, we apply a frequency-based weighting scheme. For each input i, the frequency weight is defined as:

$$W_{\text{freq},i} = 1 - \frac{f_i - f_{\min}}{f_{\max} - f_{\min}}$$

where f_i is the frequency of a given record i, and $f_{\min}, f_{\max}$ denote the minimum and maximum frequencies across all records, respectively.This penalizes over-represented records during sampling.

The final sampling weight w_i is a linear combination of diversity and frequency-based weights:

$$w_i = \alpha \cdot P_{\text{BLEU},i} + \beta \cdot W_{\text{freq},i}$$

where α and β are hyperparameters controlling the relative importance of each component. We set $\alpha = 0.8$ and $\beta = 0.2$, and sample 10,000 hypotheses from the full set of 40,000.

Sample Size Analysis. To justify the selection of 10,000 samples, we conduct an empirical analysis using subsets of 10k, 20k, and 30k examples, each drawn via the proposed weighted sampling procedure. As shown in Figure 5, performance degrades with larger sample sizes due to the increased inclusion of low-quality or redundant hypotheses. This validates our choice of 10k as a balanced point between coverage and quality.

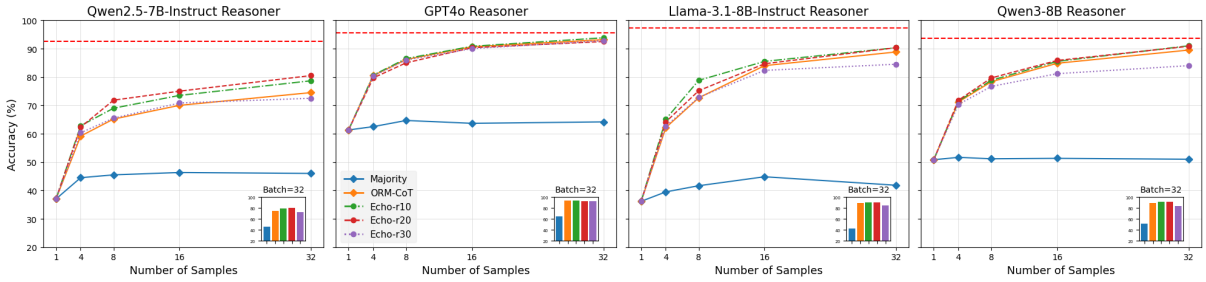

Figure 5: Performance of the ORM trained on GPT-4o-generated Echoed data with varying sample sizes. Sampling 10k examples yields the best performance, reflecting a balance between diversity and quality. Increasing the sample size to 30k degrades performance, likely due to the inclusion of lower-quality or redundant samples.

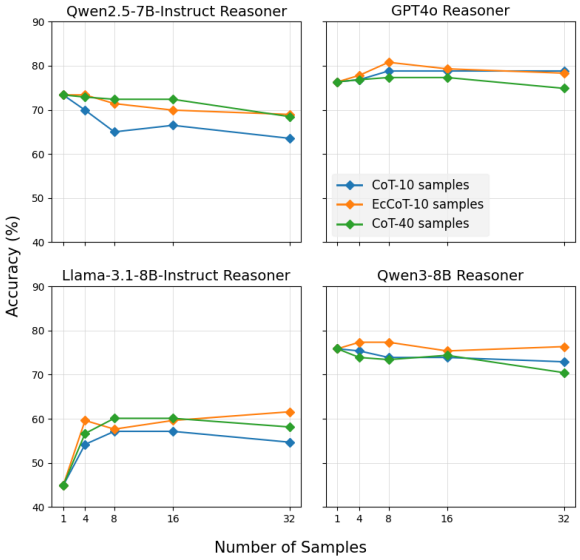

Figure 6: Performance of FOLIO with 10 samples using Echo and CoT vs 40 samples using simple CoT.

C Ablation Studies

Sample Size of CoT vs EcCoT We aimed to measure the effect of sample size as determined by the number of training samples used. For FOLIO, we initially selected 10 samples per example, resulting in an Echo configuration of 10 samples per label. To determine whether the performance gains were due to the Echo method or simply the increased number of samples, we instead sampled 40 training examples using CoT for FOLIO and compared the performance of 10 Echo samples against 40 CoT samples. Figure 6 shows that increasing the number of ORM samples does not outperform the Echo samples for both LLaMA and Qwen reasoners, where the performance was lower compared to CoT. This reinforces our motivation to use the Echo method rather than simply increasing the number of training records. We attribute this, in part, to the difference in diversity among the records, as discussed in detail in the Appendix D.

Impact of Reasoner Size We conducted an additional ablation study to examine the effect of LLM size on ORM performance. To this end, we evaluated Gemma models (Kamath et al., 2025) of varying sizes (see Figure 7). Notably, the smallest model, Gemma 1B, exhibited the most significant improvement—achieving nearly a 30% increase in accuracy with the EcCot model compared to the 4B and 12B variants for ProverQA. These results highlight the effectiveness of using a simple CoT-based verification approach for smaller models, serving as a promising first step in validating reasoning paths.

D Echo Generation Diversity

We evaluate generation diversity using self-BLEU scores, where lower values indicate higher diversity. Table 5 reports average self-BLEU scores for both standard CoT and EcCoT generations. Across datasets, EcCoT consistently produces more diverse reasoning paths. Notably, increasing the number of standard CoT samples (e.g., from 10 to 40 in FOLIO) does not yield improvements in diversity.

E System Requirements for Experimentation

The Qwen and LLaMA models were accessed via the Hugging Face interface at https://huggingface.co/Qwen/ and https://huggingface.co/meta-llama/, respectively. All models are gated and require access approval. GPT models were accessed through their API using batch calls to the /v1/chat/completions endpoint. Data generation, training, and inference were per-

Dataset	Sample Size	CoT Self-BLEU	EcCoT Self-BLEU
ProverQA	8	0.92	0.77
FOLIO	10	0.92	0.83
FOLIO	40	0.92	0.93

Table 5: Self-BLEU scores (lower is better) for standard CoT and EcCoT generations across datasets.

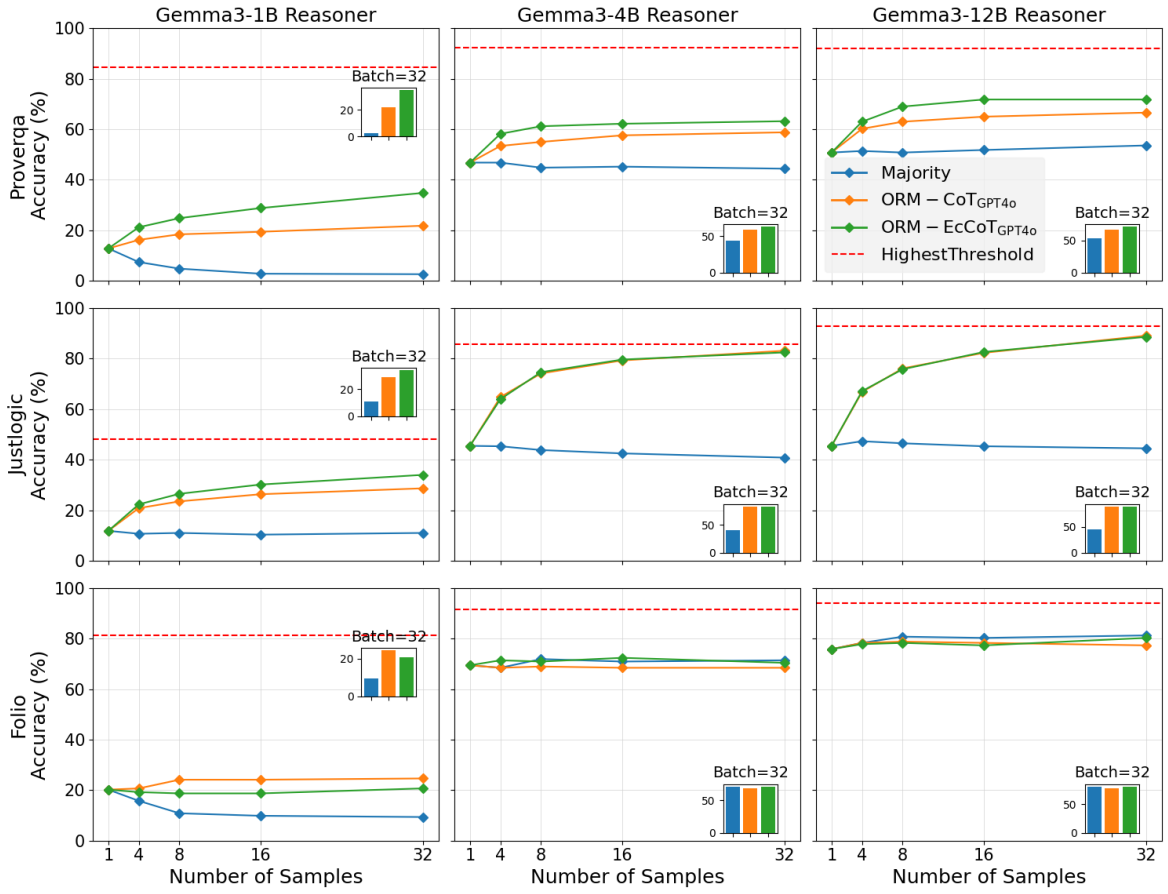

Figure 7: Performance of Gemma-based models of varying sizes on the ProverQA dataset shows a significant improvement with ORM, particularly benefiting the smaller models.

formed on a single A100 GPU, except for the API-based interactions.

Echoing Prompt: Given the answer is Uncertain (C), please reason step by step, and put your final answer within \\boxed{}.	**CoT Prompt:**Please reason step by step, and put your final answer within \\boxed{}.
Input Context and Question: William Dickinson was a British politician who sat in the House of Commons... William Dickinson supported the Portland Whigs. People who supported the Portland Whigs did not get a seat in the Parliament. **Question:** Is the statement "William Dickinson did not get a seat in Parliament" true, false, or uncertain? **Answer Options:** A) True B) False C) Uncertain	**Input Context and Question:** William Dickinson was a British politician who sat in the House of Commons... William Dickinson supported the Portland Whigs. People who supported the Portland Whigs did not get a seat in the Parliament. **Question:** Is the statement "William Dickinson did not get a seat in Parliament" true, false, or uncertain? **Answer Options:** A) True B) False C) Uncertain
Reasoning: 1. William Dickinson sat in the House of Commons $\Rightarrow$ had a seat in Parliament. 2. He supported the Portland Whigs $\Rightarrow$ such supporters didn't get a seat. These are contradictory. Cannot resolve definitively. **Final Answer:** \\boxed{C} (Uncertain)	**Reasoning:** 1. William Dickinson supported the Portland Whigs. 2. Portland Whigs' supporters did not get a seat. $\Rightarrow$ William Dickinson didn't get a seat. **Final Answer:** \\boxed{A} (True)
Predicted Answer: C (Uncertain) **Judgment:** Correct	**Predicted Answer:** A (True) **Judgment:** Correct

Table 6: Comparison of reasoning outcomes for an echoed prompt vs. a regular CoT.