Test-Time Scaling for Multistep Reasoning in Small Language Models via A* Search

Alexander Braverman
Cornell University
ab3349@cornell.edu

Weitong Zhang
University of North Carolina at Chapel Hill
weitongz@unc.edu

Quanquan Gu
University of California, Los Angeles
qgu@cs.ucla.edu

Abstract

Large language models (LLMs) have demonstrated strong abilities across various tasks but are costly in computation and memory. In contrast, Small Language Models (SLMs) offer significant advantages in efficiency and deployability but usually struggle with complex mathematical reasoning tasks. To tackle this issue, we present the Test-time A* Search (TTA*), a test-time scaling framework that casts reasoning as a goal-directed search over a tree of partial solutions in this paper. TTA* is training-free and requires no external supervision or multi-model structure, making it practical in resource-constrained settings. As a drop-in decoding wrapper for SLMs, TTA* systematically explores, critiques, and refines candidate solution paths via its own self-reflection capability. Extensive experiments on popular mathematical reasoning benchmarks and a variety of base models show that TTA* consistently improves accuracy and robustness, indicating broad applicability to general mathematical reasoning tasks.

1 Introduction

Large Language Models (LLMs) such as GPT5 [16] have showcased remarkable capabilities across a wide range of AI tasks. However, their superior performance comes with substantial computational and memory demands, which complicates personalization and hinders deployment in resource-constrained settings. In contrast, Small Language Models (SLMs) offer attractive potential due to their efficiency, lower latency, and ability to run locally—features that are particularly valuable for remote or resource-limited scenarios [3, 19, 2]. Yet SLMs continue to struggle with complex reasoning, especially in high-stakes domains such as mathematics and healthcare [12, 1]. In particular, SLMs often underperform on multi-step reasoning and are prone to hallucination—both critical concerns for safety and reliability [9, 29, 5, 10].

A growing body of work seeks to enhance the reasoning capability of small models. Many approaches rely on additional training, for example with preference data from human feedback or guidance from larger teacher models, to instill better stepwise reasoning [15]. To avoid the cost and complexity of retraining, *test-time scaling* methods have been proposed, in which the model allocates extra inference-time computation—often via structured search—to improve its answers. However, most existing test-time scaling techniques depend on ancillary components such as progress reward models (PRMs) or other external guidance, which undermines deployability and increases engineering burden. A complementary line of work explores self-rewarding or self-reflection strategies, where the model critiques its own intermediate outputs. While promising, these methods can compound errors over multiple steps due to the limited evaluative capacity of small models.

In this paper, we pursue a practical *test-time scaling* approach with self-reflection for small language models, aiming to boost reliability without additional, infeasible training costs [22]. We introduce

Test-time A* Search (TTA*), a framework that integrates heuristic search with SLMs to improve solution quality on complex reasoning tasks. TTA* casts reasoning as a goal-directed search over a tree of partial or imperfect solutions: nodes encode candidate derivations, edges apply iterative refinements, and expansions are prioritized by an A*-style score [8] that blends a cost function from the length of the search path (cost-to-come) with a heuristic of future potential from model's self-evaluation (cost-to-go). The procedure is *training-free*, requires no external supervision or multi-model orchestration, and supports explicit compute budgets with anytime behavior—making it well suited for resource-constrained deployments. As illustrated in Figure 1, our contributions are as follows:

- We formulate multi-step reasoning as a guided tree search and leverage the model's self-reflection to define the heuristic, prioritizing expansions toward high-quality solutions.
- To mitigate hallucinations and compounding errors, we design an A*-style cost function that balances exploration with skepticism about self-assessments, avoiding overuse of self-reflection while improving efficiency under fixed compute budgets.
- We introduce a calibration-aware scoring mechanism that enables consistent evaluation of partial and final answers, supporting robust multi-step reasoning.
- We conduct comprehensive experiments across a variety of benchmarks and SLM backbones. Results show that TTA* improves accuracy and compute efficiency without additional training, external supervision, progress-reward models, or multi-model orchestration.

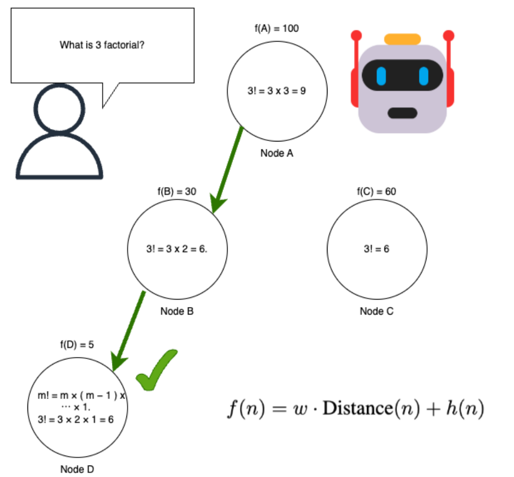

Figure 1: An example of the TTA* method for the given question. Green edges represent the chosen reasoning path. Each node is a candidate answer, and the tree structure shows how answers improve through progressive exploration.

2 Related Work

Multi-step reasoning in LLMs. Recent advances in LLMs highlight their ability to perform multi-step reasoning. Chain-of-thought (CoT) prompting [20] exposes intermediate steps and improves performance on complex tasks such as mathematical problem solving. Beyond single-path CoT, search-based methods explore multiple reasoning trajectories before committing to an answer. In particular, tree search techniques integrate with LLMs to expand and prune derivations: Monte Carlo Tree Search (MCTS) has been applied to explore intermediate steps, evaluate partial solutions, and backtrack from low-value branches [23, 27]. These approaches often rely on external reward models or curated supervision (e.g., outcome-based or process-based rewards), which can limit practicality in constrained deployment settings.

Self-reflection and self-rewarding. An alternative line of work equips LLMs with *self-evaluation* signals to guide reasoning. Here, the model generates candidate derivations and attaches lightweight diagnostics (e.g., confidence estimates, critiques, or partial checks), which inform revisions, back-tracking, or reordering. Such mechanisms can reduce ungrounded derivations and prioritize promising branches without ground-truth labels, though they risk amplifying errors when the evaluator is weak, especially in small models. Recent tree-search systems incorporate internal assessments as *progress heuristics* or *value estimates* to steer exploration, sometimes augmented with verifier-like signals or learned progress models [23, 27]. Our work builds on this direction but explicitly *binds* self-evaluation to an A*-style score that balances cost-to-come and cost-to-go, mitigating overconfidence and limiting error propagation during search.

Small Language Models. Improving mathematical reasoning in small language models (SLMs) has been approached via knowledge distillation and additional supervision [30, 28], as well as inference-time assistance from larger models. For example, *Speculative Thinking* pairs a small model with a stronger "guide" at test time to handle difficult reflective steps, improving accuracy but introducing a two-model dependency [25]. Other pipelines demonstrate impressive results with search plus process-level supervision: *rStar-Math* uses an SLM policy together with an SLM reward model to drive MCTS and iterative self-evolution; the approach attains high math accuracy but requires reward-model training and multi-round data generation/retraining [7]. While effective, these strategies increase engineering and compute complexity (e.g., extra reward models, fine-tuning, or multi-model orchestration). In contrast, our focus is a *single-model, training-free* test-time method: we cast multi-step reasoning as heuristic search and instantiate the heuristic via the model's own critiques and self-evaluation, avoiding external supervision, progress-reward models, or teacher models while retaining the benefits of structured exploration.

3 Test Time Scaling with A*-Search

3.1 Motivation

LLM reasoning frequently spans long, multi-step derivations [18]. When treated as a single forward pass, early mistakes propagate and compound, degrading reliability in tasks such as mathematical problem solving. Therefore, tree search such as Monte-Carlo Tree Search (MCTS, [27, 1, 23]) has emerged where the nodes encode the intermediate reasoning states and the edge apply the improvement of the reasoning. This structure naturally supports branching, backtracking, and iterative refinement [21], enabling exploration of diverse solution paths and more stable convergence to correct answers.

However, many of these tree search approaches for LLMs rely on auxiliary components—such as reward/value models, progress reward models, or additional fine-tuning—to evaluate partial solutions and guide exploration, which undermines deployability in resource-constrained environments. If we would like to scale the reasoning ability of the small language model during the test time, we have to rely on the self-reflection [17] mechanism of the language models. Since the self-reflection ability of small language models are usually limited, a longer reasoning path as in MCTS will hallucinate and finally harm the reasoning performance. Therefore, it would be necessary to limit the depth of the search while keep exploring the new nodes.

3.2 Tree-Based Reasoning

In this subsection, we begin with the formulation of the tree-based reasoning as illustrated in Figure 1. In particular, each node n represents a partial, intermediate or imperfect reasoning result and edge denoted by $e(n_1, n_2)$ describes the self-reflection and refinement between node n_1 to node n_2.

A* search. Based on this tree-based reasoning structure, the A* search is a best-first algorithm that balances the cost to reach a node with a heuristic estimate of the cost to the goal [8] defined as:

$$n_{\text{next}} = \arg\min_{n \in \mathcal{N}_{\text{visited}}} f(n) = g(n) + h(n), \tag{1}$$

where $\mathcal{N}_{\text{visited}}$ is the collection of the *visited* node in a tree search task, $g(n)$ is the cost of reaching node n and $h(n)$ is the heuristic function defined by the self-critic / self-reflection of the language models. As suggested in [8], the path of the A* search is guaranteed to be the shortest path as long as the heuristic function $h(\cdot)$ is *admissible* (i.e. $h(\cdot)$ never overestimates the cost of reaching the goal).

3.3 Adapting A* Search to Tree-based Reasoning

We adapt A* to LLM reasoning by treating each node as a solution candidate and define g and h as

$$g(n) = w \cdot \text{Distance}(n) \quad h(n) = 100 - \text{Reward}(n); \quad f(n) = g(n) + h(n), \tag{2}$$

where Distance(n) is the depth of the node from the root, encouraging broad exploration. w controls the exploration-exploitation tradeoff, and Reward is derived from correctness, self-consistency, or critiques using the same LLMs.

This formulation favors nodes that are either promising or close to the root, guiding the model to refine answers iteratively and reliably.

3.4 Proposed Methods

Now we can provide a detailed description of the proposed algorithm for TTA* for SLMs presented in Algorithm 1. The algorithm starts from a root node by prompting the model with the classical CoT prompt *Let's thing step by step* as well as the input question. The TTA* traverses the solutions for `max_iteration` (set to 8 in our experiments) steps, where it first uses it's own self-evaluation to assign a 0-100 score for the correctness, coherence and completeness of the response. This critique will be further used as the heuristic function defined in (2) to expand child nodes. After all iterations, the answers with the highest self-evaluation score is then selected and reported. We release our implementation on GitHub.[1]

Algorithm 1 Test-Time A* Search (TTA*)

Require: Problem prompt (e.g., "What is 4 times 3?")
Ensure: Answer with maximum LLM evaluation score

```
root_node ← LLM(prompt), AnswerStorage ← {}            ▷ Stores (answer, score)
Critique and score root_node; store result
for i = 1 to max_iterations do
    Expand current_node into two children per critique
    for each child do
        Critique, score (0–100), and store
    end for
    Select next current_node via f(n) defined in (2)
end for
return answer with highest score
```

4 Experiments

We conduct experiments using LLaMA-3-8B [6], LLaMA-3.1-8B [14], and Qwen2.5-Math-7B [24]. We evaluate the performance of the TTA* algorithm on a various of benchmarks including GSM8K [4], MATH500 [11], AIME (2024) [13], and MATH401 [26], covering problems from grade-school to competition level. The detailed prompt template is deferred into Appendix A. As presented in Table 1, TTA* demonstrates consistent improvements across all three models and benchmarks. On AIME (2024), the most challenging benchmark, TTA* achieves exceptional gains with Llama-3.1-8B showing remarkable improvement from 3.3% to 10.0%, which is a 203% relative improvement.

Name	**GSM8K**	**MATH500**	**AIME (2024)**	**MATH401**
Qwen Models				
Qwen2.5-Math-7B	73.3	46.8	6.7	65.5
w/ TTA*	87.1	74.2	10.0	78.9
Improvement Δ	+13.8 ($\uparrow$ 18.9%)	+27.4 ($\uparrow$ 58.5%)	+3.3 ($\uparrow$ 49.3%)	+13.4 ($\uparrow$ 20.5%)
Llama Models				
Llama-3-8B	75.4	26.2	3.3	62.8
w/ TTA*	87.1	44.2	6.7	78.3
Improvement Δ	+11.7 ($\uparrow$ 15.5%)	+18.0 ($\uparrow$ 68.7%)	+3.4 ($\uparrow$ 103.0%)	+15.5 ($\uparrow$ 24.7%)
Llama-3.1-8B	77.2	52.2	3.3	68.8
w/ TTA*	90.2	66.6	10.0	80.1
Improvement Δ	+13.0 ($\uparrow$ 16.8%)	+14.4 ($\uparrow$ 27.6%)	+6.7 ($\uparrow$ 203.0%)	+11.3 ($\uparrow$ 16.4%)

Table 1: Reasoning accuracy for TTA*. Baseline comparison is made with zero-shot COT [20]. The improvement Δ is the absolute and the relative percentage improvement in terms of the accuracy.

5 Conclusion

We propose Test-Time A* Search (TTA*), a framework that equips language models with structured, iterative reasoning via tree-based search. TTA* consistently outperforms zero-shot chain-of-thought on multiple mathematical reasoning tasks, validating the effectiveness of combining classical search with modern LLMs to systematically discover high-quality solutions.

However, our evaluation is limited to math problems, and TTA*'s generalizability to broader reasoning domains remains untested. Moreover, its reliance on multiple LLM calls may hinder deployment in latency-sensitive settings.

[1] `https://github.com/astarllmpaper/Test-Time-A-Search/tree/main`

References

[1] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges, 2024.

[2] Mohammed Al-Garadi, Tushar Mungle, Abdulaziz Ahmed, Abeed Sarker, Zhuqi Miao, and Michael E. Matheny. Large language models in healthcare, 2025.

[3] Yihang Cheng, Lan Zhang, Junyang Wang, Mu Yuan, and Yunhao Yao. Remoterag: A privacy-preserving llm cloud rag service, 2024.

[4] 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, 2021.

[5] Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2024.

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, 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, Danny Wyatt, 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, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, 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, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine

Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[7] Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking, 2025.

[8] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[9] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, January 2025.

[10] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022.

[11] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[12] Jingyuan Ma, Damai Dai, Zihang Yuan, Rui li, Weilin Luo, Bin Wang, Qun Liu, Lei Sha, and Zhifang Sui. Large language models struggle with unreasonability in math problems, 2025.

[13] Mathematical Association of America. 2024 aime i and ii problems. `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024`, 2024. `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024`.

[14] Meta AI. Introducing llama 3.1: Our most capable models to date. `https://ai.meta.com/blog/meta-llama-3-1/`, July 2024. Accessed: 2025-07-16.

[15] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025.

[16] OpenAI. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`, August 2025. Accessed on 2025-08-11.

[17] Matthew Renze and Erhan Guven. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, page 476–483. IEEE, November 2024.

[18] Si Shen, Fei Huang, Zhixiao Zhao, Chang Liu, Tiansheng Zheng, and Danhao Zhu. Long is more important than difficult for training reasoning models, 2025.

[19] Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. Large language models for education: A survey and outlook, 2024.

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

[21] Xue Wu and Kostas Tsioutsiouliklis. Thinking with knowledge graphs: Enhancing llm reasoning through structured data, 2024.

[22] Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Hao, and Nishil Talati. Understanding the performance and estimating the cost of llm fine-tuning, 2024.

[23] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning, 2024.

[24] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024.

[25] Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time, 2025.

[26] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. How well do large language models perform in arithmetic tasks?, 2023.

[27] Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b, 2024.

[28] Yong Zhang, Bingyuan Zhang, Zhitao Li, Ming Li, Ning Cheng, Minchuan Chen, Tao Wei, Jun Ma, Shaojun Wang, and Jing Xiao. Self-enhanced reasoning training: Activating latent reasoning in small models for enhanced reasoning distillation, 2025.

[29] Shuang Zhou, Zidu Xu, Mian Zhang, Chunpu Xu, Yawen Guo, Zaifu Zhan, Yi Fang, Sirui Ding, Jiashuo Wang, Kaishuai Xu, Liqiao Xia, Jeremy Yeung, Daochen Zha, Dongming Cai, Genevieve B. Melton, Mingquan Lin, and Rui Zhang. Large language models for disease diagnosis: A scoping review, 2025.

[30] Xunyu Zhu, Jian Li, Can Ma, and Weiping Wang. Improving mathematical reasoning capabilities of small language models via feedback-driven distillation, 2024.

A Appendix

A.1 Prompts

Prompt for Critique

Question: question
Answer: answer
Please provide detailed constructive criticism, yet highlight what is already correct.
Point the student in the right direction. Do not solve the problem.
Provide a grade (out of 100) in the format 'Grade: xx'.

Prompt for Generating Child Nodes

Question: question
Previous Answer: previous answer
Critique: critique
Given the feedback above, please try to solve the original problem again, step by step.