
Socratic RL: A Novel Framework for Efficient Knowledge Acquisition through Iterative Reflection and Viewpoint Distillation

Xiangfan Wu

June 17, 2025

Abstract

Current Reinforcement Learning (RL) methodologies for Large Language Models (LLMs) often rely on simplistic, outcome-based reward signals (e.g., final answer correctness), which limits the depth of learning from each interaction. This paper introduces Socratic Reinforcement Learning (Socratic-RL), a novel, process-oriented framework designed to address this limitation. Socratic-RL operates on the principle that deeper understanding is achieved by reflecting on the causal reasons for errors and successes within the reasoning process itself. The framework employs a decoupled "Teacher-Student" architecture, where a "Teacher AI" analyzes interaction histories, extracts causal insights, and formulates them into structured "viewpoints." These viewpoints, acting as distilled guidance, are then used by a "Student AI" to enhance its subsequent reasoning. A key innovation is the iterative self-improvement of the Teacher AI, enabling its reflective capabilities to evolve through a meta-learning loop. To manage the accumulation of knowledge, a distillation mechanism compresses learned viewpoints into the Student's parameters. By focusing on process rather than just outcome, Socratic-RL presents a pathway toward enhanced sample efficiency, superior interpretability, and a more scalable architecture for self-improving AI systems. This paper details the foundational concepts, formal mechanisms, synergies, challenges, and a concrete research roadmap for this proposed framework.

1 Introduction

Reinforcement learning (RL) has emerged as a key technology for fine-tuning large language models (LLMs) [1, 2, 3, 4, 5, 6, 7, 8, 6, 9, 10, 11] to improve their reasoning and accuracy on complex tasks. Leading systems, such as OpenAI's GPT-4o and o1 [12], Google's Gemini [13], Anthropic's Claude 3 Opus [14], and DeepSeek [15, 1, 2], all leverage RL techniques to enhance their capabilities beyond what is possible with supervised learning alone. In domains requiring sophisticated reasoning, RL serves as the core mechanism driving their remarkable performance.

However, prevailing RL methodologies for LLMs often rely on simplistic and outcome-oriented reward signals (e.g., final answer correctness). This coarse-grained feedback mechanism significantly limits the depth and breadth of learning from each interaction.

To address this limitation, this paper introduces Socratic Reinforcement Learning (Socratic-RL), a novel, process-oriented framework. Its core principle is that deeper

understanding is achieved by reflecting on the causal chain of successes and failures within the reasoning process itself, rather than merely observing the final outcome. Socratic-RL employs a decoupled “Teacher-Student” architecture. A “Teacher AI” is tasked with analyzing interaction histories, extracting the underlying causal relationships for errors and successes, and formulating them into structured “viewpoints.” These viewpoints, serving as distilled guidance, are then used by a “Student AI” to guide and enhance its subsequent reasoning.

A key innovation of the framework is the iterative self-improvement of the Teacher AI. Through a meta-learning loop, the Teacher’s reflective and analytical capabilities are designed to evolve and strengthen over time. To manage the continuous accumulation of knowledge, a knowledge distillation mechanism is introduced to efficiently compress and integrate the learned viewpoints into the Student AI’s parameters.

By focusing on the process rather than merely the outcome, Socratic-RL presents a pathway toward AI systems with enhanced sample efficiency, superior interpretability, and a more scalable architecture for self-improvement. This paper provides a detailed account of the framework’s foundational conc

2 Related Work and Theoretical Foundations

Outcome vs. Process Supervision. A cornerstone of our work is the distinction between Outcome Supervision and Process Supervision. While Reinforcement Learning from Human Feedback (RLHF) [16, 17] is a form of outcome supervision, rewarding final outputs, recent work has shown the power of process supervision, which rewards the intermediate steps of a model’s reasoning. Socratic-RL can be seen as a novel form of automated process supervision. Instead of requiring human annotation for each reasoning step, the Teacher AI learns to automatically generate process-level feedback (viewpoints) and improves this ability over time.

AI Feedback and Self-Improvement. The framework’s architecture shares a conceptual lineage with Reinforcement Learning from AI Feedback (RLAIF) [18]. However, RLAIF systems typically use a static AI critiquer. They lack a mechanism for the critiquer to learn from the outcomes of its own critiques. Socratic-RL directly addresses this gap by introducing a meta-learning loop where the Teacher AI’s ability to generate insightful ‘viewpoints’ is itself refined based on the Student’s subsequent performance, creating a system that not only learns, but learns how to teach. This also differentiates it from frameworks like Self-Refine [19], where the same model performs both generation and critique. By decoupling the Teacher and Student, Socratic-RL allows for specialized optimization: the Student becomes an expert at solving tasks, while the Teacher becomes an expert at reflection and causal analysis.

Efficiency in RL Fine-Tuning. Our work is also situated within a growing body of research focused on improving the efficiency of RL fine-tuning. Methods like GPRO [1] and its variants aim to optimize the learning process at a granular level by identifying and selectively updating only the most salient sub-networks or parameters (e.g., LoRA modules) for a given task. This focus on *parameter-level saliency*, which identifies *where* in the network to apply updates, is complementary to Socratic-RL’s approach. Instead of optimizing the mechanics of the gradient update, Socratic-RL focuses on *semantic-level*

saliency. It identifies the most critical conceptual flaw in a reasoning trace and generates a high-level "viewpoint" as feedback. In essence, while GPRO makes the learning process more efficient by optimizing the *update mechanism*, Socratic-RL aims to achieve efficiency by improving the *quality and abstraction of the learning signal itself*, addressing the *why* of the error, not just the *how* of the correction.

Prompting, Distillation, and Meta-Learning. The "viewpoints" generated by our Teacher AI are a form of dynamic, contextualized Prompting. Unlike static, manually-crafted prompts, viewpoints are automatically generated based on a deep analysis of past performance, targeting specific identified weaknesses in the Student's reasoning. Furthermore, the knowledge compression cycle is a direct application of Knowledge Distillation (KD), and the evolving Teacher AI firmly places our framework within the domain of Meta-Learning ("learning to learn"). By integrating these diverse concepts, Socratic-RL aims to create a more holistic, efficient, and intelligent learning paradigm.

3 The Socratic-RL Framework: Architecture and Formalism

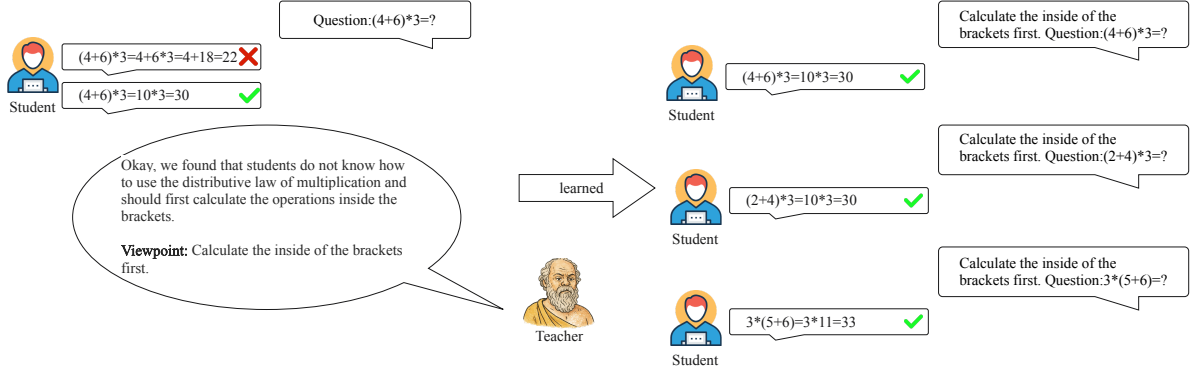


Figure 1: A high-level overview of the Socratic-RL framework.

3.1 System Overview and Formalism

The Socratic-RL framework is conceptualized as a bi-level optimization process involving two primary agents: a Student AI tasked with solving problems and a Teacher AI tasked with generating pedagogical feedback. The system's objective is to iteratively refine both agents to maximize the Student's ultimate task performance. We formalize the core components in the context of autoregressive language models, where tasks involve generating a sequence of tokens $y = (y_1, \dots, y_T)$.

- **State and Action Space:** In the LLM context, a state s_t represents the partial context for generation at step t , comprising the initial prompt x_{prompt} and the sequence of previously generated tokens: $s_t = (x_{\text{prompt}}, y_1, \dots, y_{t-1})$. The action a_t corresponds to generating the next token, y_t . The full generated sequence is the trajectory of actions.

- **Environment Interaction Trace (τ):** A trace is a complete record of the Student’s interaction with a task. It is a sequence $\tau = (s_0, a_0, \dots, s_T, a_T, R)$, where s_0 is the initial prompt, a_T is the final action (e.g., an end-of-sequence token), and R is a scalar reward signal received upon completion. For complex reasoning tasks, intermediate rewards are typically zero ($r_0, \dots, r_{T-1} = 0$), and only a final outcome-based reward R is provided, making credit assignment a significant challenge.
- **Student Policy (π_S):** The Student AI is an autoregressive policy, parameterized by θ_S , which generates an action (token) given a state and a set of active viewpoints. Its objective is to maximize the expected final reward. The policy is formally written as $\pi_S(a_t|s_t, V; \theta_S)$, where V is the set of active viewpoints, typically prepended to the input context s_t .
- **Viewpoint (v):** A viewpoint is a piece of structured, human-readable text representing a generalizable principle, a heuristic, a causal explanation, or a counter-example. It is designed to be a portable piece of knowledge that can guide the Student’s reasoning process. For instance, a viewpoint could be: *"Principle: In multi-step arithmetic, always resolve expressions within parentheses before applying external operators."*
- **Teacher Policy (π_T):** The Teacher AI is a generative model, parameterized by θ_T , that takes a full interaction trace τ as input and performs a form of automated causal analysis. It aims to identify the root cause of failure (or success) within the Student’s reasoning trace and synthesizes this insight into a new viewpoint v . The policy is thus defined as $v \sim \pi_T(\tau; \theta_T)$. The Teacher’s goal is not to solve the task itself, but to produce viewpoints that are maximally effective for improving the Student’s policy π_S .

The overarching goal is to find optimal parameters θ_S^* for the Student. This is achieved by iteratively improving the Teacher’s parameters θ_T such that the viewpoints it generates accelerate the learning of the Student. The system thus solves a meta-learning problem where the Teacher learns how to teach effectively.

3.2 The Core Loop: From Interaction to Viewpoint

The Student AI interacts with the environment. The Teacher AI observes the Student’s interaction trace τ to perform a causal analysis, identifying specific failure modes like **logical fallacies, procedural errors, or flawed assumptions**. Based on this analysis, the Teacher generates a "viewpoint." For example, if a Student fails ‘ $(4 + 6) * 3$ ’ by calculating ‘ $4 + 18$ ’, the Teacher might generate the viewpoint: *"In arithmetic, operations inside parentheses must be evaluated first."* This viewpoint is then added to the set of active viewpoints V to guide the Student.

Algorithm 1 The Socratic-RL Core Loop

```
1: Initialize Student policy  $\pi_S$  and Teacher policy  $\pi_T$ .
2: Initialize an empty set of active viewpoints  $V = \emptyset$ .
3: Initialize a knowledge base of all learned viewpoints  $\mathcal{V}_{KB} = \emptyset$ .
4: for each episode  $k = 1, 2, \dots$  do
5:                                      $\triangleright$  Phase 1: Student Interaction
6:   Sample a task and generate an interaction trace  $\tau_k \sim \pi_S(\cdot|s, V)$ .
7:   Obtain outcome (e.g., success/failure, final reward  $R_k$ ).
8:
9:                                      $\triangleright$  Phase 2: Teacher Reflection
10:  if outcome is suboptimal or meets generation criteria then
11:    Generate a new viewpoint  $v_k \sim \pi_T(\tau_k)$ .
12:    Add  $v_k$  to the active viewpoints:  $V \leftarrow V \cup \{v_k\}$ .
13:    Add  $v_k$  to the knowledge base:  $\mathcal{V}_{KB} \leftarrow \mathcal{V}_{KB} \cup \{v_k\}$ .
14:  end if
15:
16:                                      $\triangleright$  Phase 3: Meta-Learning (Teacher Evolution)
17:  Update  $\pi_T$  using feedback on the utility of past viewpoints from  $\mathcal{V}_{KB}$  (see Section
    3.3).
18:
19:                                      $\triangleright$  Phase 4: Knowledge Distillation
20:  if a distillation condition is met (e.g., fixed interval) then
21:    Fine-tune a new Student  $\pi'_S$  to internalize viewpoints in  $\mathcal{V}_{KB}$ .
22:    Set  $\pi_S \leftarrow \pi'_S$  (the new Student becomes the current one).
23:    Reset active viewpoints:  $V \leftarrow \emptyset$ .
24:  end if
25: end for
```

3.3 The Meta-Learning Engine: Evolving the Teacher AI

We posit that the Teacher AI’s capabilities are context-dependent, varying across different environments. The Teacher AI’s ability to generate effective "viewpoints" is a learnable skill, and its evolution is driven by a meta-objective: to learn how to generate viewpoints that construct the most effective prompts for the Student, thereby maximizing its learning progress. We formalize the quality of a viewpoint v with a utility score $U(v)$, which measures the performance uplift it provides when added to the Student’s context in a set of probe tasks $\mathcal{P}_{\text{probe}}$:

$$U(v) = \mathbb{E}_{p \sim \mathcal{P}_{\text{probe}}} [\text{Score}(\pi_S(\cdot|p, V \cup \{v\}))] - \mathbb{E}_{p \sim \mathcal{P}_{\text{probe}}} [\text{Score}(\pi_S(\cdot|p, V))]$$

Where $\text{Score}(\cdot)$ is the task success metric. Through this mechanism, we can obtain a Teacher AI that continuously improves its own prompts.

3.4 Knowledge Distillation and Scalability

A core challenge for scalability is that relying on an ever-growing set of explicit viewpoints V in the prompt context is computationally inefficient and has a finite limit. To ensure long-term learning and create a more capable standalone model, Socratic-RL incorporates a modular knowledge distillation mechanism. This mechanism’s purpose is to compress

the procedural knowledge encapsulated in the viewpoint-guided interactions into the parameters θ_S of a new Student model, π'_S .

The default method for this is **policy distillation** (or behavioral cloning). Here, the new Student π'_S is trained to mimic the output distribution of the original, viewpoint-guided Student π_S . This is achieved by minimizing the Kullback-Leibler (KL) divergence between the two policies over a dataset \mathcal{D} of inputs and viewpoints from the knowledge base \mathcal{V}_{KB} :

$$\mathcal{L}_{\text{distill}} = \mathbb{E}_{(\text{Input}, v) \sim \mathcal{D} \times \mathcal{V}_{KB}} [D_{KL}(\pi_S(\cdot | \text{Input}, v; \theta_S) \parallel \pi'_S(\cdot | \text{Input}; \theta'_S))]$$

This effectively trains the new Student to behave as if it "knows" the principles from the viewpoints without needing to see them.

However, the distillation module is designed to be plug-and-play, allowing for more advanced techniques to be employed. Alternative strategies include:

- **Direct Preference Optimization (DPO):** The viewpoints can be used to generate preference pairs. For a given problem, we can generate one response using a relevant helpful viewpoint ($y_{\text{preferred}}$) and another response without it, or with a deliberately unhelpful "negative viewpoint" (y_{rejected}). This creates a dataset of triplets $(x_{\text{prompt}}, y_{\text{preferred}}, y_{\text{rejected}})$ which can be used to fine-tune the Student model directly via DPO, a powerful and highly effective method.
- **Instruction Tuning:** The entire knowledge base of viewpoints \mathcal{V}_{KB} can be systematically reformatted into a high-quality instruction-tuning dataset. Each viewpoint is transformed into a general instruction. For example, the viewpoint *"Principle: In arithmetic, evaluate parentheses first."* can generate training examples like: `{"instruction": "Solve the following, paying close attention to the order of operations.", "input": "(4+6)*3", "output": "..."}.` This approach directly integrates the learned principles into the model's fundamental instruction-following capabilities.

The choice of distillation method is a key hyperparameter of the framework, allowing practitioners to balance between direct behavioral cloning and more nuanced preference-based or instruction-based fine-tuning to achieve optimal performance.

3.5 Interpretability by Design

A significant advantage of Socratic-RL is its inherent interpretability. The set of learned viewpoints, \mathcal{V}_{KB} , serves as a human-readable log of the system's acquired knowledge. By examining the sequence of generated viewpoints, researchers can trace the AI's learning trajectory, understand the principles it has discovered, and diagnose its failures. This "glass-box" nature contrasts sharply with the "black-box" behavior of models trained solely on outcome-based rewards. The viewpoints themselves become artifacts for analysis, representing the AI's evolving "theory" of the world.

4 Synergies and Comparative Analysis

The modular design of Socratic-RL allows it to be synergistically combined with other advanced AI techniques, enhancing its capabilities and safety.

Integration with Existing Methods. The meta-learning loop for the Teacher AI is highly compatible with Direct Preference Optimization (DPO) [20], which can be used to refine its ability to generate helpful viewpoints. Furthermore, the Teacher’s generation process can be constrained by a set of predefined rules, akin to Constitutional AI (CAI) [21], ensuring that the guidance it provides is safe, ethical, and aligned with human values.

Comparative Analysis. Table 1 provides a high-level comparison with existing LLM training paradigms, highlighting the unique combination of features in Socratic-RL.

Table 1: Socratic-RL vs. Existing LLM Training Paradigms

Paradigm	Primary Type	Feedback	Source of Feedback	Iterative Improvement of Source
Socratic-RL	Structured, "Viewpoints" (Process)	causal	Evolving "Teacher AI"	Yes (Core feature)
RLHF [16]	Scalar Reward (Outcome)	(Outcome)	Human Annotators	No
RLAIF [18]	Natural Language Critiques (Outcome)	Critiques (Outcome)	AI Critiquer Model	No (Typically static)
DPO [20]	Preference Pairs (Outcome/Process)	(Outcome/Process)	Human/AI Preferences	No
CAI [21]	Adherence to Rules/Principles	to	Pre-defined Constitution	No
Self-Refine [19]	Self-Generated Critiques (Process)	Critiques (Process)	Model Itself	Yes (Not specialized)

5 Challenges and Limitations

Despite its promise, the practical implementation of Socratic-RL faces significant hurdles that require careful consideration.

Subjectivity in Evaluation. The framework’s efficacy hinges on the Teacher AI’s ability to discern "better" from "worse" reasoning. While straightforward in objective domains like math, this becomes ill-defined in tasks with subjective criteria, such as creative writing or summarization. Defining a suitable utility function $U(v)$ for the Teacher in such domains is a substantial open research problem.

Stability of Self-Improvement. Self-referential training loops are notoriously susceptible to instability. The system could suffer from **model collapse**, where the diversity of outputs diminishes over time. Worse, it could create feedback loops that **amplify biases**. For instance, if a Teacher develops a slight stylistic preference, it may generate viewpoints that reinforce this style in the Student, which in turn provides evidence for the Teacher’s preference, leading to a rapid loss of stylistic diversity.

Computational Cost and Latency. The Socratic loop is computationally intensive. Each cycle involves an extra inference pass from the Teacher model, analysis of the trace, and potentially calculating the utility of viewpoints. This introduces significant latency and cost compared to standard fine-tuning, which may be a barrier to its application in resource-constrained or real-time scenarios.

Teacher Model Drift. The unconstrained evolution of the Teacher AI carries the risk of "epistemic drift" or even "madness," where it might develop bizarre or unhelpful theories about the world that are locally consistent but globally incorrect. Ensuring the Teacher remains grounded in reality and aligned with the intended learning goals is a critical safety and performance challenge.

6 Methodological Approach

The foundational principle of our framework is to deeply integrate core RL concepts, such as reward functions and policy gradients, directly into the optimization process of the LLM itself. This approach transforms these abstract algorithmic components into explicit, model-driven operations.

Recently, a critical aspect of RL that has gained significant attention is the concept of entropy, particularly its role in identifying the most salient parts of an experience. For instance, recent research from Alibaba has demonstrated that a minority of tokens often play a disproportionately large role during the RL process [22]. Similarly, frameworks like SEED-GPRO [23] leverage the semantic entropy of a problem to determine which parameters to update, thereby focusing the learning process on areas of higher uncertainty.

Traditional methods of defining rewards and updating parameters struggle to distill abstract reasoning or "the right way of thinking" directly from input-output pairs. Our Socratic Reinforcement Learning (Socratic-RL) framework addresses this problem by empowering the LLM to take on a more direct role within the learning algorithm.

Although this adds a layer of complexity to the RL process, it leverages a unique capability of Large Language Models: the ability to increase information density. This function is key to enabling the model to rapidly acquire new knowledge and principles from a very small number of examples.

We propose a hierarchy of knowledge representation where information density progressively increases. It begins with raw text, is then refined into token dependency relationships, and is ultimately distilled into concrete "viewpoints." This progressive concentration of knowledge is the core mechanism for achieving more effective and efficient

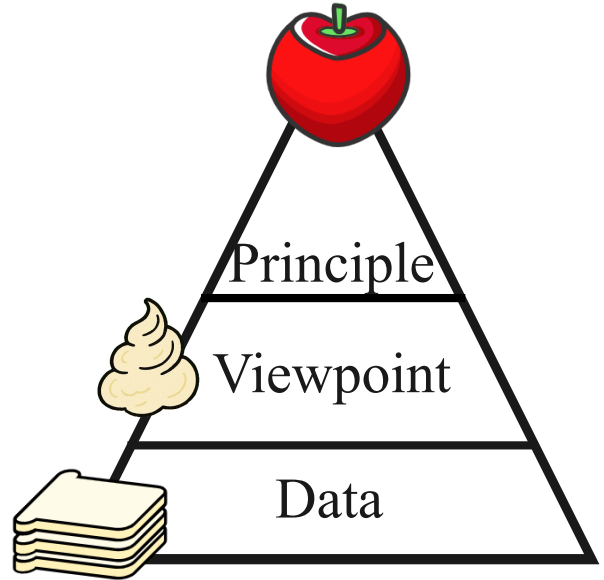


Figure 2: Knowledge Hierarchy in Socratic-RL: Principles, Viewpoints, and Data

knowledge compression within the model. For example, as shown in Figure 2, a "Principle" represents the directly inputted context. However, due to the limited input space, these are often guiding principles, such as "think step by step" or instructions in the form of an agent, which can guide general problems. Our "viewpoint" resides at an intermediate level; for a specific problem, it can provide more detailed guidance than a Principle, thereby facilitating the problem-solving process. At the same time, it is much smaller than the raw data, thus enabling the generalized resolution of similar problems.

7 Future Research Directions

The challenges outlined above define a clear roadmap for future research. We propose a phased strategy to systematically validate and mature the Socratic-RL framework.

1. **Phase 1: Proof-of-Concept with a Rule-Based Teacher.** The immediate next step is to implement the framework in a constrained, objective domain like multi-step arithmetic. The Teacher AI will initially be a rule-based system that parses the Student's Chain-of-Thought trace to detect specific, pre-defined errors (e.g., incorrect order of operations). Key research questions for this phase include: (a) Does viewpoint-guided learning show higher sample efficiency than outcome-based RL? (b) Is the knowledge distillation cycle effective at compressing procedural knowledge into the Student's parameters without catastrophic forgetting?
2. **Phase 2: Developing and Training an LLM-based Teacher.** Following the proof-of-concept, we will replace the rule-based system with an LLM-based Teacher. This phase will focus on the meta-learning aspect. We will experiment with different prompt-engineering strategies for the Teacher to elicit causal analysis and will implement and compare training methods (e.g., RL with utility rewards vs. DPO on viewpoint pairs). The central goal is to demonstrate that the Teacher's ability to teach can be measurably improved through automated feedback.
3. **Phase 3: Scaling, Stability, and Safety.** Once the core mechanics are validated, we will scale the framework to more complex domains like code generation. This phase will directly tackle the stability and safety challenges. We will investigate techniques to mitigate model collapse and bias amplification, such as using an ensemble of diverse Teachers, regularizing the Teacher's policy to prevent drastic shifts, and continuously grounding the system with a stream of fresh, external data.

8 Conclusion

Socratic Reinforcement Learning offers a conceptual and architectural blueprint for a new generation of LLMs that learn more efficiently and transparently. By shifting the training paradigm from sparse, outcome-based rewards to rich, process-oriented feedback, Socratic-RL opens the door to more sample-efficient learning. The framework's core innovations—the decoupled, evolving Teacher-Student architecture and the knowledge distillation cycle—provide a promising, albeit challenging, path toward more capable, interpretable, and continuously improving artificial intelligence. Successfully addressing the outlined challenges could represent a significant step towards creating AI systems that not only solve problems, but understand the principles behind the solutions.

References

- [1] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [3] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- [4] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [5] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.
- [6] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.
- [7] Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.
- [8] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.
- [9] Yufeng Yuan, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.
- [10] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.
- [11] Yixuan Even Xu, Yash Savani, Fei Fang, and Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.
- [12] OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [13] Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

- [14] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024.
- [15] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.
- [16] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
- [17] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.
- [18] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023.
- [19] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.
- [20] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.
- [21] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Tom Conerly, Chengan Jaeger, Tom Henighan, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Jared Kaplan, Sam McCandlish, Tom Brown, and Dario Amodei. Constitutional ai: Harmlessness from ai feedback, 2022.
- [22] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.
- [23] Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025.