Training Large Language Models with Reinforcement Learning

Marta Perun
Technical University of Vienna
e12220871@student.tuwien.ac.at

Abstract

Reinforcement learning (RL) has significantly impacted the understanding and identification of best practices for training Large Language Models (LLMs), especially with the recently released reasoning model R1 from Deepseek DeepSeek-AI et al. [2025]. It is commonly believed in the industry that Reinforcement Learning is more effective for teaching the model to generalize and enhance reasoning (Jin et al. [2025]) compared to Supervised Fine-Tuning. There are numerous methods for employing Reinforcement Learning in the training process, such as RLHF (Reinforcement Learning with Human Feedback), RLAIF (RL with AI Feedback), rule-based RL and reward model training. This paper provides a summary of the methods, as well as the scalability analysis and research gaps at the end of the paper.

1 Introduction

1.1 Motivation

Large Language Models have recently evolved into a basic component of modern AI systems, from code generation, over solving complex math and logic puzzles to writing great articles and facing creative problems, like writing poems. Alignment of LLMs with the human preferences and values, reasoning before answering in order to solve complex riddles and achieve task-specific goals is one of the big problems that the AI industry is currently focused on (Yu et al. [2025]). This has also emerged the big boom in finding new, more effective, but at the same time more efficient training techniques to get better results fast. Another problem, that I'm sure everybody currently is concerned with is the AI ethics. The need of aligning models in a way that is not harmful for humanity, different groups of people, or just individuals requires the engineer to find new methods how to train or even lead the models in a way that is considered to be safe and non-harmful. Both of these problems could not be solved just with Supervised Fine-Tuning, as it is in general not that well generalizable Chu et al. [2025], which once again underlines the importance of trying out RL techniques on training the Large Language Models.

1.2 Scope and Contributions

This paper presents a comprehensive overview of reinforcement learning techniques used to train and align LLMs with a main focus on recent developments in the research industry. In the following you can see the main last contributions that you can learn about in the following sections of the paper:

- A review of classic RL techniques, e.g. PPO and reward models
- An analysis on the new-emerging techniques based on the classic ones, including GRPO
- An exploration of rule-based RL paradigm, with a focus on constitutional AI

- A summary of practical challenges for the future and trade-offs for all the previously mentioned paradigms.
- RLAIF, RLHF

1.3 Related Work

This section will provide some overview of the field, especially the quick analysis of the papers that have been considered greatly in this work. One of the most impactful papers is the DeepSeek's R1 model DeepSeek-AI et al. [2025] that has developed the reasoning capabilities with much less data than would be needed for other techniques (e.g. Supervised Fine-Tuning). How this was achieved was to "force" the LLM to create a reasoning block before the final answer. The prompt that was used instructed the model to first reason and put it in between the "<think>", "</think>" tags, following the "<answer>" and "</answer>" that actually provides the answer to the input prompt of the user. Using rule-based Reinforcement Learning the LLM could "learn on its own" by training the policy to maximize the reward model. The rules defined were:

1. following the formatting rules (think and answer blocks provided),
2. providing a correct answer (e.g. for math problems, the correct value at the end, for programming problems similar to LeetCode the solution has to pass some tests etc.),
3. answering consistently in one language.

The last rule was introduced because a big problem that was discovered during training is that the model mixed the languages in the response, which made the response quite unreadable. With adding the additional rule to answer in one language, the performance decreased slightly. But because of the importance to align with the human values of usability of Large Language Models, it was introduced.

Applying a rule-based reinforcement learning algorithm during the training of large language models proved to be a sound choice, owing to its simplicity and computational efficiency.

Another new topic introduced in the DeepSeek's R1 paper is the GRPO (Group Relative Policy Optimization). In each modern LLM there must be a training stage to align the model's responses with the human preferences. This problem is also solved with reinforcement learning algorithms. Using SFT (Supervised Fine-Tuning) for such purpose would not make sense, as SFT much rather focuses on teaching the model what to say and not how. We would need a stage in the LLM training to optimize for preference, use reward functions that are focused on more nuanced objectives (e.g., helpfulness, harmfulness, etc.) and generalize better to unseen examples.

GRPO is an improved version of its predecessor PPO, which was one of the mostly used algorithms for Reinforcement Learning pipeline. Its improvement focuses on making the algorithm more efficient by updating the policy based within a batch rather than solely relying on absolute reward values. This group-based approach results in a more stable training, even in presence of noisy data and improves the efficiency of the training, making it an important step in scalable and reliable LLM alignment.

Another important work was done as improvements upon the newly introduced topics in the previous paper. In the paper Xie et al. [2025] the rule-based algorithm was improved by adding additional rules (e.g. having only reasoning in the "<think>"-block, and answer in the <answer> block and some others) to guarantee actual holding up onto rules and to avoid shortcuts.

REINFORCE++ algorithm was introduced as well as an upgraded and better fitted version of the GRPO algorithm. The author introduces the usage of KL-divergence in the reward model already (in comparison: in GRPO KL-divergence was used in the loss model only) and fine-tuned its estimation by ensuring non-negative values during training.

This suggests that the algorithms, despite their good work, can and should be still improved to get better results.

2 Background

In this section I will show how natural language can be seen as a Reinforcement Learning problem and will introduce the classic RL algorithms that emerged in the Large Language Model world that are currently the base for all the current more advanced solutions.

2.1 Fundamentals of Reinforcement Learning

Reinforcement Learning is a paradigm that focuses on teaching agent what is the best sequence of steps to be taken over time, to maximize the reward (in other words, to achieve some goal). It takes input from the environment and tries to predict what the next best step is, whilst always getting a reward as feedback how well it performed. Just from the definition, it can be noticed that finding a reward is an already challenging task, because in some cases (e.g. in predicting the next tokens, aka LLM world, it is sometimes also not understandable for a human, what is a better response). In order to come up with a decision the RL agent has to consider value-based methods (outcomes for the state-action pairs) as well as it has to optimize the policy-based methods that maximize the reward considering all possible scenarios.

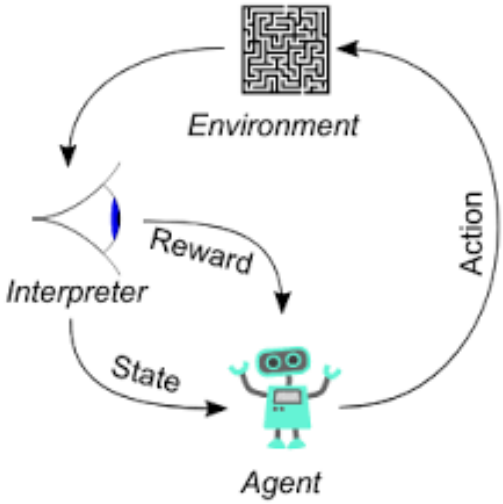

Figure 1: Reinforcement Learning environment

2.2 Language Models as RL Agents

In this chapter, you can find a quick introduction on how Natural Language generation can be viewed as a Reinforcement Learning problem.

The Language Models are responsible for predicting the next token. The environment has states (in our case, prompts) that have the goal of generating a sequence of actions (tokens) that lead to a comprehensive answer (result). The responsibility of an agent is to learn a policy that achieves the goal of creating a coherent contextual answer.

Such a formulation of a problem enables us to view fine-tuning the LLM with Reinforcement Learning not only as a model that tries to maximize the reward of choosing a correct next token, but also in general, RL could be applied to achieve some bigger-scale goals (for instance, aligning it with ethical values).

However, while calculating the cumulative reward is a challenging problem in natural language on its own, the newer strategies focused on rather comparing signals, e.g. in RLHF (Reinforcement Learning with Human Feedback) pipelines paired responses are provided to the feedback giver and then based on preferred one, the policy is optimized and reward model is trained.

This has led to another improvement, to the existence of reward-free models that fully rely on preference. The rise of the following methods has had a huge impact: Direct Policy Optimization (DPO), Group Relative Policy Optimization (GRPO), and REINFORCE++, which have shifted apart from costly and inefficient sampling and explicit reward model to a more efficient and stable training method.

2.3 Policy Optimization Techniques

Now let us take a deeper look at the policy optimization methods and the things that led to their development.

- PPO (Proximal Policy Optimization): a technique that optimizes the reward model based on feedback on each response. It became a very stabilized training technique, because it included clipping the changes, so the model does not have drastic changes, which destroy some things learned before. By taking the minimum between the unclipped and clipped objectives, PPO algorithm penalizes updates that would extremely increase or decrease action probabilities.

$$L(\theta) = \mathbb{E}_t \left[\min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} A_t,\ \text{clip} \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) A_t \right) \right]$$

  ,
  where:
  - $\pi_\theta(a_t|s_t)$ - probability of taking action a_t in state s_t recording to the new policy
  - $\pi_{\text{old}}(a_t|s_t)$ - probability recording to the old policy
  - A_t - advantage function, which represents how much better or worse the action a_t was compared to the average action at state s_t.
  - ϵ is a hyperparameter that limits how much the new policy can diverge from the old one, preventing excessively large policy updates.
- REINFORCE: a classic reinforcement learning algorithm. The idea is to update the weights to increase the probability of actions that lead to higher rewards.
- DPO (Direct Policy Optimization): This algorithm does not use a separate reward model. Instead, it directly trains the language model to prefer better responses. The training method is simple and based on inverse reinforcement learning. This makes DPO easier to use and more stable than older methods like PPO. Nika et al. [2024]
- GRPO (Group Relative Policy Optimization): is an optimized version of DP. That means that GRPO considers a couple of responses of the LLM and grades a set of those and bases on relative quality towards other responses. In such a way it can analyze and train on multiple answers at once, which makes the training process more stable and quality of the responses. The comparison between the GRPO and PPO you can see in the [Figure 1] below.

$$J(\theta) = \mathbb{E}_{q\sim P(Q),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left(r_{i,t}\hat{A}_{i,t}, \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t} \right) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right]$$

  - $q \sim P(Q)$: A question q is sampled from the question distribution $P(Q)$.
  - $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(O|q)$: A group of G output sequences $\{o_1, o_2, \ldots, o_G\}$ is sampled from the old policy given the question.
  - $\frac{1}{G}\sum_{i=1}^{G}$: The objective averages over all sampled outputs in the group.
  - $\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}$: Each sequence o_i is averaged over its token positions t.
  - $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$: The probability ratio between the current and old policy at token t in output o_i.
  - $\hat{A}_{i,t}$: The estimated advantage at time step t for output o_i, based on relative rewards within the group.
  - $\min \left(r_{i,t}\hat{A}_{i,t}, \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t} \right)$: The clipped objective used to avoid large policy updates.
  - ϵ: A hyperparameter that controls how much the policy is allowed to change between updates.
  - $\beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$: A KL-divergence penalty term that discourages the new policy from deviating too far from a reference policy, scaled by hyperparameter β.

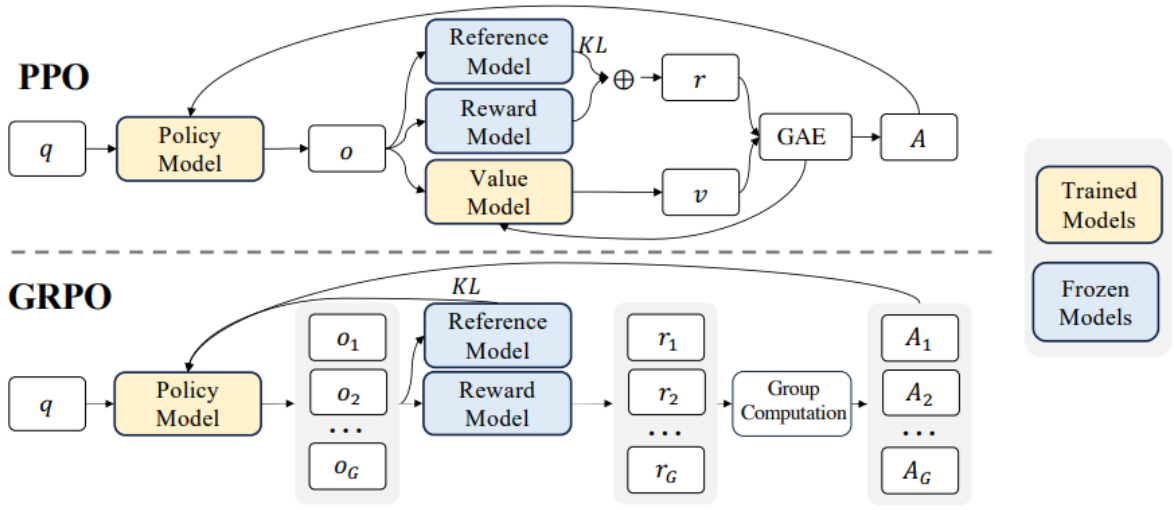

Figure 2: Comparison between PPO and GRPO. Source: Shao et al. [2024].

- REINFORCE++: an advanced algorithm that combines the 2 strong points coming from its predecessors: REINFORCE (reward function) and GRPO (group-wise comparisons). It is "an enhanced variant of the classical REINFORCE algorithm that incorporates key optimization techniques from PPO while eliminating the need for a critic network. REINFORCE++ achieves three primary objectives: (1) simplicity (2) enhanced training stability, and (3) reduced computational overhead." Hu et al. [2025]

Figure 3: Comparison of efficiency between GRPO, PPO and REINFORCE++. Source: Hu et al. [2025].

3 Human and AI-Guided Reinforcement Learning Approaches

3.1 Background

In this section, the algorithms will be introduced that play a role of fine-tuning the model's responses to be better aligned with human preferences. An introduction will be provided to 2 most commonly

used: RLHF (Reinforcement Learning with Human Feedback) and RLAIF (Reinforcement Learning with AI feedback).

3.2 Reinforcement Learning with Human Feedback

RLHF involves training the Large Language Model based on feedback received from human annotators. Their task is to judge the model's responses and evaluate how good the answer is based on some predefined factors (helpfulness, clarity etc.) This approach is mostly used together with some policy optimization techniques (e.g. PPO) to update the policy based on the data received from human annotators.

This method has gained some popularity based on the fact that it is flexible across domains, improves the output quality of the language model and alignment with human values. However, as it requires huge amount of data it is not well scalable. Another negative factor is imperfections in the data provided by the annotators, as it can, for instance, have limited consistency and potentially can still propagate biases (unconscious flaws). These aspects should therefore be taken into account when evaluating potential solutions.

3.3 Reinforcement Learning with AI Feedback

A notable development in RL is the Reinforcement Learning from AI Feedback (RLAIF). This technique builds upon the principles of Reinforcement Learning from Human Feedback (RLHF). Rather than relying only on human-generated feedback, RLAIF gets feedback provided by powerful AI models to train and align other AI systems. This approach offers a scalable alternative to traditional human-in-the-loop methods. It potentially reduces the cost and time associated with human annotation while maintaining or even improving alignment quality. As the field seeks more efficient ways to train Large Language Models and align them with desired behaviors, RLAIF provides both promising opportunities and significant questions regarding reliability and control.

4 Rule-based Reinforcement Learning

4.1 Definition

Rule-based Reinforcement Learning is a paradigm built upon the traditional RL principles and combined with some defined rules as a reward model. Traditional RL has an agent that explores the environment and provides feedback and rule-based methods introduce a set of predefined rules to follow, based on which the agent receives rewards as a signal.

An example of rule-based RL principles:

Consider an agent trained to assist in drafting official documents for legal professionals. In this context, the use of slang is undesirable. Accordingly, the agent is penalized with a negative reward (e.g., –1) if its output contains any slang expressions, and conversely, it receives a positive reward (e.g., +1) when it produces output free of slang.

This paradigm also has had great success in improving the quality of the reasoning capabilities in Large Language Models. In text generating tasks, such models often lack deeper understanding/analysis of the problem and are rather "surface" responses with no reasoning behind it. This illustrates a bigger problem, as it defines the limitations of LLMs. In such cases, rule-based methods can be used to enforce the models to think before constructing the final answer (e.g. by introducing a rule of formatting for a "think"-block - as mentioned in "1.4 - Related Works" section).

4.2 Constitutional AI: Embedding Human Values vie Rules

One of the areas that the rule-based paradigm can bring much success into, is the Constitutional AI. It is a term introduces by Anthropic AI Bai et al. [2022]. Constitutional AI is an approach of aligning human values and principles with the Large Language Models. It focuses on reducing the human feedback through the whole learning process by only delegating the overview of the "constitution" (predefined set of rules, aligned with human ethics) to the people and using self-critique and supervision to learn how to apply those rules successfully.

Introducing better ethics into AI models, especially into their reasoning, to be able to stay flexible not relying on the cost of helpfulness could potentially bring a lot of benefits, especially when seeing great success in other areas of aligning Large Language Models and their underlying principles (e.g. in aligning language models to assist legal professionals).

Rule-based frameworks guide AI behavior using clear "if-then" logic with help of:

- Predefined ethical guidelines: A rule might state, "If a user asks for medical advice, the model should check against peer-reviewed sources before answering."
- Penalty mechanisms: During training, responses that break these rules are given negative feedback (e.g., lower rewards), which encourages the model to produce more appropriate and rule-following outputs.

This approach might introduce a new perspective on how to bring ethics into AI models with potentially small cost of helpfulness of the models. However, such approach still does not resolve previous problems with constitutional AI, such as rules clash (when two rules in some way contradict each other, e.g. "maximize truthfulness" and "avoid harmful facts").

4.3 Applicability and training efficiency analysis

Rule-based frameworks provide as some additional value the computational efficiency because of the following factors:

1. *Reduced parameter complexity* - Rule-based reward models have far fewer trainable parameters than standard RLHF (Reinforcement Learning with Human Feedback) reward models. This allows for much faster fitting, since the optimization focuses on a small set of rule weights rather than millions of reward model parameters.
2. *Sample number efficiency* - Because rules are explicit and interpretable, fewer training examples are needed to achieve strong alignment. For instance, OpenAI reports that fitting RBR (rule-based rewards) weights requires less data than training a full reward model. Mu et al. [2024]
3. *Automation and scalability* - Rule-based frameworks help automatically check and enforce good training practices, which cuts down on the need for constant human supervision.
4. *Generalization* - Rule-based RL has been shown to promote cross-domain generalization and robust behavior, as models trained with verifiable, rule-based rewards can maintain or even improve performance on unseen tasks or domains.

5 Open Problems and Future Developments

5.1 Scalable Reinforcement Learning

One of the very important things to consider when designing new training pipelines for Large Language Models is scalability and computational limitations for training algorithms.

In this paragraph there is a simple comparison of the scalability of two of the most used training techniques: RL (Reinforcement Learning) and SFT (Supervised Fine-Tuning). Each one of the techniques provides some trade-offs regarding the scalability. RL offers great adaptability in dynamic environments—it works by iterative self-improvement based on the reward, while SFT adapts very quickly to specific tasks by using labeled data with a risk of overfitting and quite limited generalization.

Here are various aspects to consider when comparing the scalability of both techniques:

- **Data Efficiency**: RL scales via exploration, so there is no need for a static dataset, while SFT requires intensive data labeling.
- **Computational Demand**: RL is extremely resource-intensive (can be mitigated using partial rollouts), while applying SFT has lower cost and is more parameter-efficient (methods like LoRA can optimize the costs).
- **Performance Impact**: RL achieves state-of-the-art reasoning capabilities, while SFT provides a strong baseline through task-specific tuning.

Based on the newest research in the area: the Kimi k1.5 model provided new techniques for making RL training even more scalable Team et al. [2025]. It could be achieved by using the following techniques:

- Context length extension: the maximum number of tokens was increased to 128,000 tokens.
- Partial rollouts: this technique optimizes the computational costs of handling long sequences during training. It works by splitting a single reasoning task into smaller segments. After each segment is generated, it is saved and be reused in later training steps. This allows the model to incrementally complete long reasoning chains while updating its parameters efficiently.
- Length penalty: prevents overthinking and excessive repetition.
- Curriculum sampling: the training focuses on gradually harder problems, making the model perform better in low-success tasks.

Because of the great results of the Reinforcement Learning in NLP and LLMs areas, there is currently a lot of research striving to build faster, better and more successful algorithms.

5.2 Research gaps

While the current research is impressive, this paragraph focuses on the still-existing research gaps in the area that require some attention from the research field.

One such topic is the hybrid usage of SFT and RL, which could potentially result in even more computationally cheap training specific to the exact use case. As partial rollouts are a new topic in the field of reinforcement learning, it is important to explore which parameter optimizations are most effective in this context. This includes the research on multi-modal models. The following topics still need deeper exploration: intermediate feedback/rewards for long-sequence problems, evaluation frameworks for reinforcement learning-aligned LLMs, and mechanisms to prevent reward hacking.

The research in this area clearly holds immense potential, and every contribution is recognized.

6 Conclusion

This paper introduced the topic of using reinforcement learning algorithms for training of Large Language Models. As both of the areas gain popularity, separately and together, it's crucial to know how to leverage those technologies for even better performance of models. As Reinforcement Learning provides so many diverse algorithms, this paper provided an structured way to get into the topic, from the basics of Reinforcement Learning over some popular methods, like RLHF, RLAIF and rule-based Reinforcement Learning to the scalability aspect of those as well as identification of research gaps. The area provides great opportunities for researchers, as each singe days we can observe contributions to the field.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL https://arxiv.org/abs/2501.17161.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jian Hu, Jason Klein Liu, and Wei Shen. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models, 2025. URL https://arxiv.org/abs/2501.03262.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety, 2024. URL https://arxiv.org/abs/2411.01111.

Andi Nika, Debmalya Mandal, Parameswaran Kamalaruban, Georgios Tzannetos, Goran Radanović, and Adish Singla. Reward model learning vs. direct policy optimization: A comparative analysis of learning from human preferences, 2024. URL https://arxiv.org/abs/2403.01857.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei

Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning, 2025. URL https://arxiv.org/abs/2502.14768.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

All resources used *must* be cited properly. You can use the preset bibliography style.

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix.