100 Days After DeepSeek-R1: A Survey on Replication Studies and More Directions for Reasoning Language Models

Chong Zhang[*1,2] , **Yue Deng**[*1] , **Xiang Lin**[*1] , **Bin Wang**[*1] , **Dianwen Ng**[1], **Hai Ye**[1,3], **Xingxuan Li**[1], **Yao Xiao**[1,4], **Zhanfeng Mo**[1,5], **Qi Zhang**[2], **Lidong Bing**[1†]
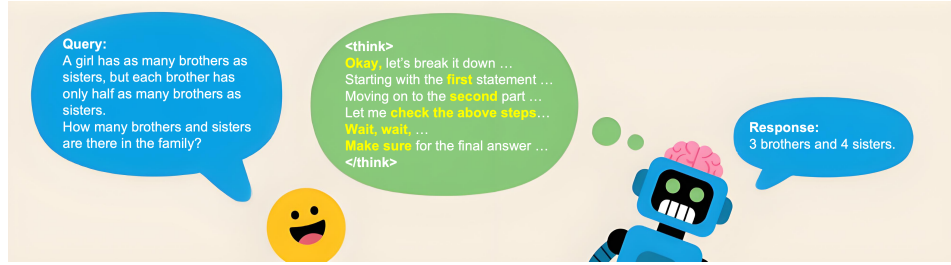
[1]MiroMind
[2]Fudan University
[3]National University of Singapore
[4]Singapore University of Technology and Design
[5]Nanyang Technological University

ABSTRACT

The recent development of reasoning language models (RLMs) represents a novel evolution in large language models. In particular, the recent release of DeepSeek-R1 has generated widespread social impact and sparked enthusiasm in the research community for exploring the explicit reasoning paradigm of language models. However, the implementation details of the released models have not been fully open-sourced by DeepSeek, including DeepSeek-R1-Zero, DeepSeek-R1, and the distilled small models. As a result, many replication studies have emerged aiming to reproduce the strong performance achieved by DeepSeek-R1, reaching comparable performance through similar training procedures and fully open-source data resources. These works have investigated feasible strategies for supervised fine-tuning (SFT) and reinforcement learning from verifiable rewards (RLVR), focusing on data preparation and method design, yielding various valuable insights. In this report, we provide a summary of recent replication studies to inspire future research. We primarily focus on SFT and RLVR as two main directions, introducing the details for data construction, method design and training procedure of current replication studies. Moreover, we conclude key findings from the implementation details and experimental results reported by these studies, anticipating to inspire future research. We also discuss additional techniques of enhancing RLMs, highlighting the potential of expanding the application scope of these models, and discussing the challenges in development. By this survey, we aim to help researchers and developers of RLMs stay updated with the latest advancements, and seek to inspire new ideas to further enhance RLMs.

[*]Equal contribution. Co-first authors are listed in random order, as are the co-authors. All authors made substantial and complementary contributions to the development of this work.

[†]Corresponding author: Lidong Bing <lidong.bing@miromind.ai>.

CONTENTS

Training Reasoning Models

- **Supervised Fine-tuning (§2)**
  - **Supervised Fine-tuning Datasets (§2.1)**: *e.g.*, OpenThoughts-114k(OpenThoughts-Team, 2025), OpenR1-Math-220k(HuggingFace, 2025), Light-R1-SFT(Wen et al., 2025a), Bespoke-Stratos-17k(Bespoke-Labs, 2025), SYNTHETIC-1(Mattern et al., 2025), s1K-1.1(Muennighoff et al., 2025), LIMO(Ye et al., 2025)
  - **Supervised Fine-tuning Models (§2.2)**: *e.g.*, OpenThinker(OpenThoughts-Team, 2025), OpenR1-Qwen-7B(HuggingFace, 2025), Light-R1(Wen et al., 2025a), Bespoke-Stratos(Bespoke-Labs, 2025), SYNTHETIC-1-SFT(Mattern et al., 2025), s1.1(Muennighoff et al., 2025), LIMO(Ye et al., 2025)
- **Reinforcement Learning from Verifiable Rewards (§3)**
  - **Data Curation (§3.1)**: *e.g.*, Open-Reasoner-Zero(Hu et al., 2025), OWM(Gandhi et al., 2025), LIMR(Li et al., 2025b), DeepScaleR(Luo et al., 2025b), Skywork-OR1(He et al., 2025a), DAPO(Yu et al., 2025), Open-R1(HuggingFace, 2025), DeepCoder(Luo et al., 2025a)
  - **Algorithm Design (§3.2.1)**: *e.g.*, PPO(Schulman et al., 2017), GRPO(Shao et al., 2024), Dr. GRPO(Liu et al., 2025c), DAPO(Yu et al., 2025), VAPO(Yue et al., 2025b), VC-PPO(Yuan et al., 2025), CPPO(Lin et al., 2025b), REIN-FORCE(Ahmadian et al., 2024), REINFORCE++(Hu, 2025), REINFORCE-Kimi(Kimi-Team et al., 2025), GPG(Chu et al., 2025b)
  - **Reward Curation (§3.2.2)**: *e.g.*, Logic-RL(Xie et al., 2025a), Oat-Zero(Liu et al., 2025b), L1(Aggarwal & Welleck, 2025), Efficient-Reasoning(Arora & Zanette, 2025), Open-Reasoner-Zero(Hu et al., 2025), Dr. GRPO(Liu et al., 2025c), DAPO(Yu et al., 2025), Demystify-Long-CoT(Yeo et al., 2025), Light-R1(Wen et al., 2025a)
  - **Sampling Strategy (§3.2.3)**: *e.g.*, Light-R1(Wen et al., 2025a), Open-Reasoner-Zero(Hu et al., 2025), SRPO(Zhang et al., 2025b), DAPO(Yu et al., 2025), Skywork-OR1(He et al., 2025a)
- **Other Techniques (§4)**
  - **Process-Reward-based RL (§4.1)**: *e.g.*, IPR(Xiong et al., 2024), PRMBench(Song et al., 2025), FineReason(Chen et al., 2025) rStar-Math(Guan et al., 2025), PRIME(Cui et al., 2025)
  - **Preference Optimization (§4.1)**: *e.g.*, EXAONE(LG-Research et al., 2025), Light-R1(Wen et al., 2025a), DPO-VP(Tu et al., 2025), RedStar(Xu et al., 2025a), DPO-R1(Zhang et al., 2025a)
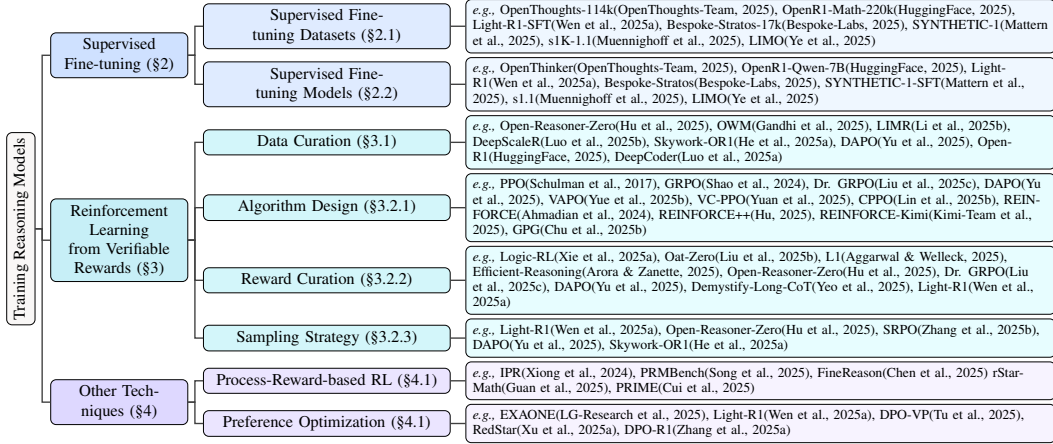
Figure 1: Taxonomy of training methods of reasoning models.

1 INTRODUCTION

Reasoning language models (RLMs) have emerged as a transformative advancement in the evolution of large language models (LLMs), such as OpenAI o-series (Jaech et al., 2024; OpenAI, 2025), DeepSeek-R1 (Guo et al., 2025), and QwQ series (Qwen-Team, 2024; 2025b). Unlike conventional LLMs that merely generate unstructured responses, these models incorporate an explicit chain-of-thought process, providing step-by-step reasoning that mimics human cognitive processes–such as invoking self-verification, reflection, and more. This shift quickly attracted attention of the LLM research community, as it meets the growing demand for better explainability in complex tasks like mathematical problem solving, code generation, and logical reasoning, as well as the pursuit of steadily increasing accuracy. The significance of RLMs lies in their potential to enhance the accuracy of language models' response with trustful rationales. Rather than only providing answers, these models also reveal their thought process, which is invaluable for educational purposes, critical decision-making, and debugging AI reasoning errors. DeepSeek-R1 (Guo et al., 2025) is a prime example of this new generation. It leverages innovative training techniques such as supervised fine-tuning (SFT) and reinforcement learning from verifiable rewards (RLVR), allowing the model to develop powerful reasoning behaviors with an affordable amount of supervised data. Especially, by using methods like RLVR and incorporating cold-start data, it achieves performance comparable to prior models (e.g. OpenAI o1 (Jaech et al., 2024)) with relatively low training costs.

However, many critical details remain undisclosed which is required to replicate the reasoning performance and the behaviors of self-verification and reflection exhibited by DeepSeek-R1. Although the DeepSeek-R1 team has publicly released their solution as training the model with Group Relative Policy Optimization (GRPO, Shao et al. (2024)) and rule-based reward systems, the optimal design of the reinforcement learning algorithm and reward system remains underexplored. Moreover, the training data and configurations of the SFT and RLVR stages are not released, leaving the impact on model performance to be further examined. In response, many replication works have attempted to explore the optimal design for RLMs from various perspectives (see Figure 1), yet a comprehensive list and comparison of these works are still lacking.

This survey aims to provide a clear review of the open-source replication works on DeepSeek-R1. According to Figure 1, the arrangement of this survey is based on methodology and generally corresponds to the training process of DeepSeek-R1, introducing current replication works on SFT, RLVR, and other technologies enhancing the reasoning capability. In introducing the conclusions made by these works, this survey attempts to summarize the common practice of replicating RLMs with comparative analysis from various perspectives, including data resources, sampling strategies, and training configurations. With the above efforts, we aim to help researchers optimize their own models by effectively referencing the prior works. The following sections are arranged as follows:

- **Supervised Fine-tuning for Reasoning Language Models.** We provide a comprehensive overview of replication works aimed to enhance the reasoning ability of language models

through supervised fine-tuning. Recognizing that the starting checkpoints and fine-tuning data resources are the key aspects for the SFT process, we conduct comparative analyses of these aspects to derive meaningful insights. We also summarize the common training practices for supervised fine-tuning.

- **Reinforcement Learning from Verifiable Rewards for Reasoning Language Models.** We present recent works that train RLMs using reinforcement learning from verifiable rewards (RLVR) by elaborating on their training data, learning algorithms, and reward system designs. Noting that various studies have adopted variants of PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024) with subtle modifications, we attempt to establish a unified theoretical framework to explain these methods, highlighting both the algorithmic changes and the underlying motivations behind each adaptation. Moreover, based on the configurations and experiment results of these works, we conclude the possibly common practice for RLVR.

- **More Directions for Reasoning Language Models.** We identify that while DeepSeek-R1 advances the training of RLMs, many supervision strategies remain unexplored. We present more directions for RLMs, including reward modeling and preference optimization and examine the strengths and weaknesses of current RLMs, such as their powerful out-of-distribution generalization and occasional overthinking. Finally, we briefly discuss extending RLMs to multimodal and multilingual applications.

2 SUPERVISED FINE-TUNING FOR REASONING LANGUAGE MODELS

DeepSeek-R1 distilled models (Guo et al., 2025), e.g., the DeepSeek-R1-Distill-Qwen series, exhibit strong reasoning capabilities despite their smaller sizes. Since then, several works (Hugging-Face, 2025; OpenThoughts-Team, 2025; Wen et al., 2025a; Zhao et al., 2025a) have attempted to reproduce this reasoning ability in smaller models by applying Supervised Fine-Tuning (SFT) on their own curated datasets. These datasets typically consist of math or coding problems and, more importantly, include one or more validated Chain-of-thoughts (CoTs) from DeepSeek-R1. This section aims to provide a comprehensive overview of how these studies approach the reproduction of distilled reasoning models.

2.1 SFT DATASETS

In this subsection, we provide a comprehensive overview of datasets used for SFT, starting with their curation processes, followed by detailed descriptions of individual datasets. We also examine key properties such as token length distributions, data contamination risks, and cross-dataset dependencies, with the goal of highlighting best practices and common patterns in constructing high-quality reasoning datasets.

2.1.1 DATA COLLECTION AND CURATION PIPELINE

A growing number of reasoning-focused datasets have been curated based on a shared set of principles aimed at improving reasoning capability through SFT. Most efforts begin by collecting questions across diverse domains, such as math, science, coding, and puzzles, either from existing benchmarks or web crawling.

After raw data collection, multiple rounds of filtering are typically employed to enhance quality. These include deduplication (e.g., via embedding similarity or n-gram), rejection sampling, and ground-truth verification. To ensure high coverage and data richness, many datasets explicitly emphasize difficulty and diversity during the selection process, often using heuristics or model pass rates to prioritize harder problems. For example, Light-R1 (Wen et al., 2025a) applies thresholding based on model correctness to form a challenging subset from a broader base. Further, most datasets rely on verified CoTs or solutions to ensure correctness and quality. Verification methods vary by domain. For instance, math problems are often validated by Math Verify (Kydlíček, 2024), coding questions through execution or unit tests, and general tasks by LLM judges. This combination of domain-aware validation and selective retention allows curators to distill high-quality reasoning traces that better support supervised fine-tuning.

While these datasets span multiple domains, the majority are mainly focused on math and coding tasks as observed in Table 1. Broader coverage across diverse reasoning tasks, such as science, logic puzzles, and open-ended questions, remains relatively limited. Notable exceptions include DeepSeek-R1 and AM (Zhao et al., 2025a), which incorporate a wider range of domains during both data collection and distillation, aiming to foster more generalizable reasoning capabilities.

2.1.2 EXISTING DATASET DETAILS

Project	Size of SFT Data	Math	Code	Other-Reasoning	Non-Reasoning
DeepSeek-R1 (Guo et al., 2025)	800k	✔	✔	✔	✔
AM (Zhao et al., 2025a)	1.4M	✔	✔	✔	✔
Synthetic-1 (Mattern et al., 2025)	894k	✔	✔	✔	✘
OpenThoughts (OpenThoughts-Team, 2025)	114k	✔	✔	✔	✘
Bespoke-Stratos (Bespoke-Labs, 2025)	16.7k	✔	✔	✔	✘
Light-R1 (Wen et al., 2025a)	76k / 3.6k	✔	✘	✘	✘
Open-R1 (HuggingFace, 2025)	220k	✔	✘	✘	✘
S1k-1.1 (Muennighoff et al., 2025)	1k	✔	✘	✔	✘
LIMO (Ye et al., 2025)	817	✔	✘	✘	✘

Table 1: Summary of recent projects including SFT data and their corresponding categories. Other-Reasoning includes science, puzzles, etc.

DeepSeek-R1. Guo et al. (2025) curates a distillation dataset of 800k training samples, comprising 600k reasoning examples and 200k non-reasoning examples, such as writing, role-playing, and other general-purpose tasks. According to the available report, parts of the non-reasoning data appear to be reused from the SFT dataset of DeepSeek-V3 (DeepSeek-AI, 2024). To create the distillation dataset, DeepSeek-R1 itself is used to generate the distillation traces. However, this interpretation is based on limited details provided, as the exact methodology has not been fully disclosed. Notably, the dataset is not publicly available.

AM. Zhao et al. (2025a) curates a large-scale reasoning dataset[1] comprising 1.4M samples across various domains. The curation mainly involves three stages: (i) Raw data collection, where they collect raw data from multiple data sources. (ii) Comprehensive data filtering, *i.e.,* deduplication via embeddings-based similarity, upsampling of difficult problems using an LLM. (iii) CoT distillation. For samples lacking reasoning traces or whose ground truths fail verification, new CoTs are generated using DeepSeek-R1. To ensure correctness of the answer, a sequence of verifiers is applied, including Math Verify and Qwen2.5-7B-Instruct (Yang et al., 2024). Code-related problems with test cases are additionally verified through execution.

Synthetic-1. Mattern et al. (2025) constructs an 894k-sample reasoning dataset[2] distilled from DeepSeek-R1, covering domains such as math, coding, and STEM. Verification is domain-specific: Math Verify is used for math questions, execution-based validation is applied for coding problems, and LLM judges assess the remaining type of problems.

OpenThoughts. OpenThoughts[3] (OpenThoughts-Team, 2025) curates a synthetic reasoning dataset with 114k examples from several sources. It covers multiple domains, including math, science, coding, and puzzles. The CoTs are generated by DeepSeek-R1 and verified. In particular, they use an LLM as a judge to verify the ground-truth answers for math and puzzle problems, and rely on code execution and unit tests to validate coding problems.

Bespoke Stratos. Bespoke-Labs (2025) curates a reasoning dataset[4] of 17k examples distilled from DeepSeek-R1, covering domains such as coding, math, science, and puzzles. They apply rejection sampling to eliminate reasoning traces with incorrect solutions. In particular, they use GPT-4o-mini (OpenAI, 2024) as a judge to filter out the traces with incorrect answers of math

---

[1] https://huggingface.co/datasets/a-m-team/AM-DeepSeek-R1-Distilled-1.4M
[2] https://huggingface.co/datasets/PrimeIntellect/SYNTHETIC-1-SFT-Data
[3] https://huggingface.co/datasets/open-thoughts/OpenThoughts-114k
[4] https://huggingface.co/datasets/bespokelabs/Bespoke-Stratos-17k

questions, increasing the proportion of retained examples from 25% to 73%, compared to a rule-based approach.

Light-R1. Light-R1 (Wen et al., 2025a) constructs a high-quality SFT dataset[5] comprising 79k samples distilled from DeepSeek-R1. They begin by collecting 1 million math problems from various sources and use DeepScaleR-1.5B-Preview (Luo et al., 2025b) to generate initial responses. Only questions with low pass rates (below α) are selected for further processing with DeepSeek-R1, yielding around 76k examples. From this set, only correct long-form CoT responses are retained, with one chosen per question to create an SFT dataset of over 70k examples, filtered for both difficulty and diversity. While training solely on this dataset was effective in reproducing the distilled model, a second stage was introduced to further enhance quality by leveraging DeepSeek-R1-Distill-Qwen-32B. This phase retains only those questions with pass rates of DeepSeek-R1-Distill-Qwen-32B below α and where DeepSeek-R1's responses were not consistently correct or incorrect. The result is a refined Stage 2 SFT dataset containing approximately 3k examples.

Open-R1. OpenR1-Math-220k[6] (HuggingFace, 2025) is a large-scale dataset for math reasoning tasks. HuggingFace (2025) collects 220k math problems from NuminaMath 1.5 (LI et al., 2024) and generate 2 to 4 CoTs for each problem using DeepSeek-R1. To ensure that each problem includes at least one correct answer, most of the CoTs are verified by Math Verify (Kydlíček, 2024), with Llama-3.3-70B-Instruct (Dubey et al., 2024) serving as a judge for 12% of the samples. Among the 220k problems, 94k are considered higher quality. According to HuggingFace (2025), the 94k subset achieves better performance in SFT, as the 131k extended problems may contain easier questions.

S1k-1.1. Muennighoff et al. (2025) curates a large-scale reasoning dataset[7] by collecting 59k questions from 16 diverse sources. Each question is paired with a reasoning trace and a solution generated by DeepSeek-R1, forming question-trace-solution triplets. After decontamination and deduplication, a three-stage filtering process produces a high-quality, diverse, and challenging subset of 1,000 samples, designed for minimal-resource training.

LIMO. While not directly focused on reproducing distilled DeepSeek-R1 models, LIMO[8] (Ye et al., 2025) still offers valuable insights into reasoning model development. LIMO first constructs tens of millions of problems from various established datasets, such as NuminaMath. They then apply a baseline difficulty filter using Qwen2.5-Math-7B-Instruct (Yang et al., 2024), removing problems that can be solved within a few attempts. Next, they collect reasoning traces from human experts and state-of-the-art models, including DeepSeek-R1, DeepSeek-R1-Distill-Qwen-32B, and Qwen2.5-32B-Instruct, and conduct a thorough analysis to identify key characteristics of high-quality reasoning chains, namely, Optimal Structural Organization, Effective Cognitive Scaffolding, and Rigorous Verification. Finally, they use a hybrid approach combining rule-based filtering with LLM-assisted curation to select high-quality solutions for each question. The resulting dataset of 817 questions demonstrates strong performance when used to fine-tune base models.

2.1.3 ANALYSIS AND DISCUSSIONS

Length Distributions. Figure 2 illustrates the token length distributions of the datasets discussed above. Although all the long CoTs of these datasets originate from the same teacher model, *i.e.,* DeepSeek-R1, their distributions exhibit observable differences. For instance, datasets such as AM and Synthetic-1 are skewed toward shorter sequences, whereas Light-R1 and Open-R1 display broader distributions with longer tails, suggesting a higher proportion of complex problems, which typically elicit longer CoTs.

Data Decontamination. Among all the works discussed above, only the technical reports of Light-R1 and LIMO explicitly mention conducting proper data decontamination against popular reasoning benchmarks, *e.g.,* AIME24/25, MATH500, and GPQA Diamond (Rein et al., 2024), during dataset

---

[5] https://huggingface.co/datasets/qihoo360/Light-R1-SFTData
[6] https://huggingface.co/datasets/open-r1/OpenR1-Math-220k
[7] https://huggingface.co/datasets/simplescaling/s1K-1.1
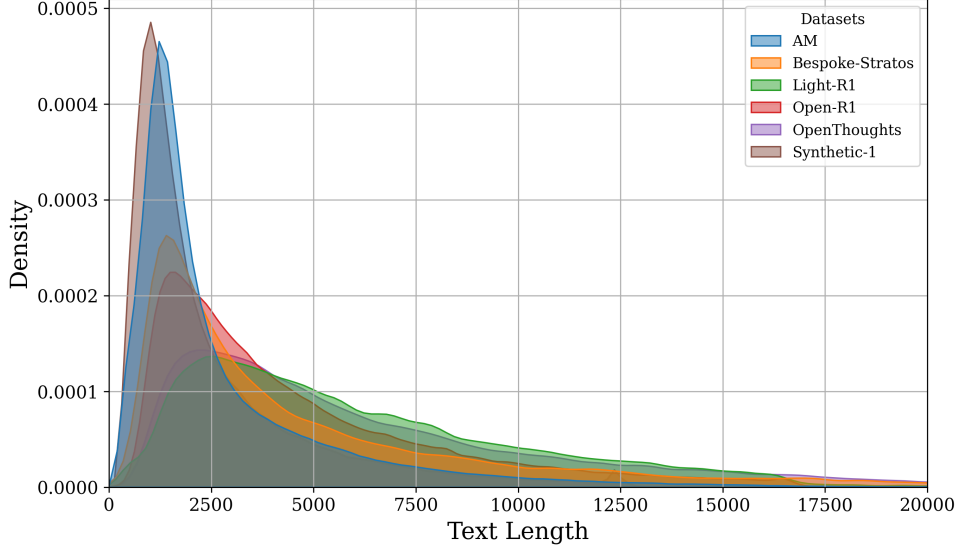[8] https://huggingface.co/datasets/GAIR/LIMO

Figure 2: Token length distributions for the aforementioned SFT datasets. The x-axis is truncated at 20,000 tokens, as examples exceeding this length are rare.

curation. Notably, Wen et al. (2025a) point out that MATH500 is partially compromised across several open-source datasets, including OpenThoughts, Open-R1, Bespoke Stratos, and others.

Cross-Referencing Dataset Sources. Recently, a growing number of math reasoning datasets have been released to support the SFT of LLMs. However, many of these datasets are not created in isolation, *i.e.,* they frequently collect or derive data from preexisting datasets, often with overlapping or reused examples. In some cases, multiple datasets originate from the same root sources, making it challenging to understand their relationships and evaluate model dataset quality fairly. For example, many datasets have obtained problems from NuminaMath.

To clarify these connections and reduce confusion, we present Figure 3, which illustrates the cross-referencing structure among popular math reasoning datasets. It highlights the web of dependencies and shared data across benchmarks, helping researchers better interpret results and avoid redundant training or evaluation setups.

2.2 TRAINING & PERFORMANCE COMPARISON

In this subsection, we first formalize the supervised fine-tuning (SFT) of reasoning language models, and then provide a detailed overview of the configuration used in the current replication studies for SFT.

Supervised Fine-tuning. Given a dataset $\mathcal{D}_{\text{SFT}} \triangleq \{(q_i, c_i)\}_{i=1}^{|\mathcal{D}|}$, where each sample (q_i, c_i) consists of a question q_i and a long CoT c_i. The long CoT can be further decomposed into a complex intermediate rationale followed by a final answer. SFT updates the parameters of the policy model π_θ by minimizing the negative log-likelihood loss:

$$\mathcal{L}_{\text{SFT}}(\theta) \triangleq -\mathbb{E}_{(q,c)\sim\mathcal{D}_{\text{SFT}}} \left[\log \pi_\theta(c \mid q)\right], \tag{1}$$

where $\pi_\theta(c \mid q)$ denotes the probability assigned by the policy to the CoT response c conditioned on the question q. This objective encourages the model to imitate the supervised demonstrations by maximizing the likelihood of the reference completions.

Comparison. In practice, SFT stage plays a crucial role in allowing the base model to learn high-quality reasoning traces from stronger models. Table 2 presents a comparative overview of SFT re-
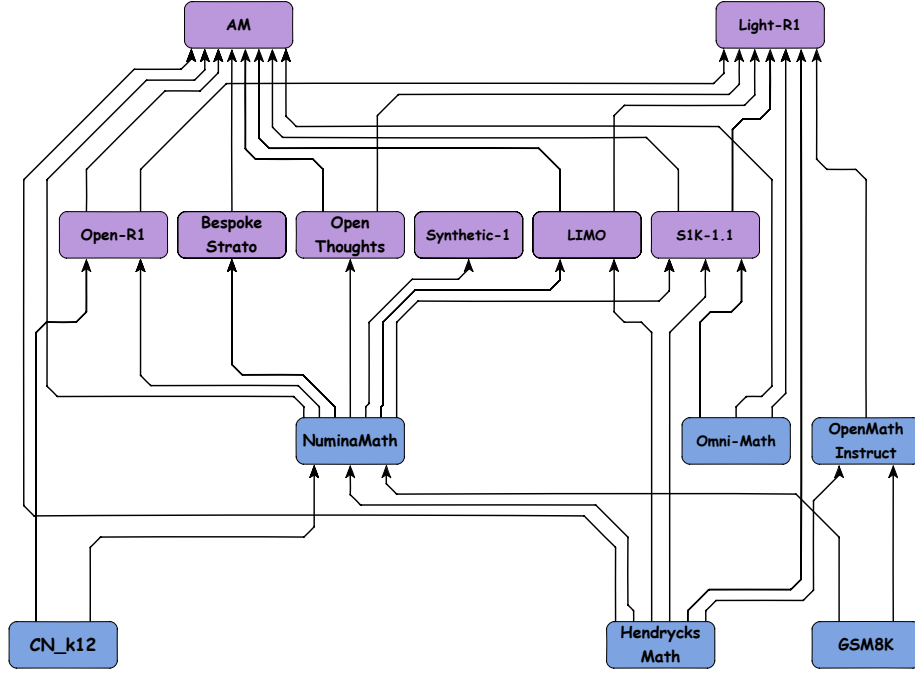
Figure 3: An illustration of cross-referenced dataset sources for popular math reasoning datasets. Arrows point from source datasets to target datasets that incorporate some of their data. The figure does not reflect dataset sizes, nor does it imply that a target dataset includes all data from its source, or only data from the source(s) indicated by the arrows. Datasets highlighted in lilac contain Chain-of-Thought traces extracted from DeepSeek-R1.

Project	Initial Checkpoint	AIME24	AIME25	MATH500
DeepSeek-R1 (Guo et al., 2025)	Qwen2.5-Math-7B / Qwen2.5-32B-Base	55.5 / 72.6	–	92.8 / 94.3
AM (Zhao et al., 2025a)	Qwen2.5-32 / 72B-Base	72.7 / 76.5	–	96.2 / 97.0
Light-R1 (Wen et al., 2025a)	Qwen2.5-32B-Instruct	73.0	64.3	–
LIMO (Ye et al., 2025)	Qwen2.5-32B-Instruct	57.1	44.5	94.8
S1k-1.1 (Muennighoff et al., 2025)	Qwen2.5-32B-Instruct	56.7	50.0	94.4
OpenThoughts (OpenThoughts-Team, 2025)	Qwen2.5-7 / 32B-Instruct	31.3 / 68.0	23.3 / 49.3	83.2 / 90.6
Bespoke-Stratos (Bespoke-Labs, 2025)	Qwen2.5-7 / 32B-Instruct	20.0 / 63.3	–	82.0 / 93.0
Open-R1 (HuggingFace, 2025)	Qwen2.5-Math-7B-Instruct	36.7	40.0	90.6
Synthetic-1 (Mattern et al., 2025)	Qwen-2.5-7B-Instruct	30.0	26.6	85.6

Table 2: Summary of recent projects including initial checkpoints and their corresponding benchmark results. Results are taken from corresponding papers. Dashes (–) indicate unavailable results. Note that the reported Open-R1 performance is from the "default" split of the dataset.

sults on common math reasoning benchmarks, AIME24/25 and MATH500 (Hendrycks et al., 2021), highlighting the impact of different dataset choices and initial checkpoints.

While many approaches emphasize scaling up the number of training samples to boost performance, LIMO and S1k-1.1 demonstrate that strong results can be achieved with significantly smaller, carefully curated datasets. In particular, only the DeepSeek-R1-distilled series and AM fine-tune from Qwen2.5-base models, while other methods rely on the stronger Qwen2.5-Instruct models. A related study by Li et al. (2025d) also indicates that instruct models exhibit higher learning efficiency than their base counterparts.

Furthermore, DeepSeek uniquely incorporates non-reasoning data for SFT. By contrast, other works focus primarily on math and coding reasoning problems, leaving the interplay between reasoning and non-reasoning data underexplored.

Training Details. Although the technical report of DeepSeek-R1 does not mention the training hyperparameters of the distillation models, we aggregate this information from replication studies to better understand common training setups. For long-context tasks such as complex reasoning, the RoPE scaling factor (θ) and maximum context length in the model configuration are often adjusted to enable extended context capabilities (Chen et al., 2023). For example, Open-R1 (HuggingFace, 2025) sets θ = 300,000 and the context length to 32,768 tokens. Commonly used learning rates include 1.0×10^{-5} and 5.0×10^{-5}, with typical batch sizes of 96 or 128. Additionally, packing is usually employed to improve training efficiency (Wang et al., 2024a).

3 REINFORCEMENT LEARNING FROM VERIFIABLE REWARDS FOR REASONING LANGUAGE MODELS

This section focuses on reinforcement learning from verifiable rewards (RLVR) for reasoning language models. First, we provide a detailed examination of techniques for training reasoning language models with RLVR, including training data preparation, reinforcement learning (RL) algorithms, and the designs of reward systems and data sampling strategies during training. Specifically, for RL algorithms and their variants, we provide an in-depth discussion of the motivation and rationale behind each of them. According to the implementation details and experiment results from the replication studies, we summarize the key insights from several aspects. We also introduce ongoing efforts to extend reasoning language models beyond closed-book examinations on scientific subjects, adapting them to a broader range of tasks.

3.1 RL DATASETS

DeepSeek-R1-Zero achieved strong performance on reasoning and knowledge tasks through a standalone RLVR process. The curated high-quality datasets employed during its RLVR process are instrumental to the success. Therefore, several replication studies have explored strategies for efficiently creating training datasets by leveraging open-source data and powerful models. In this subsection, we introduce the datasets used in RLVR. These datasets cover various tasks that are verifiable during RL training, in which we mainly focus on datasets for math and coding problem solving. We introduce the curation of each dataset, including the selection of data resources, the construction of verified questions and answers, and the detailed pre-processing procedures. Table 3 displays an overview for the statistics of these datasets.

Dataset	Organization	Size	Categories
DeepScaleR (Luo et al., 2025b)	Agentica Project	40k	Math
Skywork-OR1 (He et al., 2025a)	Skywork	129k	Math, Code
Open-Reasoner-Zero (Hu et al., 2025)	StepFun	129k	Math, Reasoning
Big Math (Albalak et al., 2025)	SynthLabs	251k	Math
DeepMath-103k (He et al., 2025b)	Tencent	103k	Math
Curated Thoughts (Hochlehnert et al., 2025b)	Bethge Lab, University of Tuebingen	222k	Math
DAPO-Math-17k (Yu et al., 2025)	ByteDance Seed	17k	Math
LIMR (Li et al., 2025b)	GAIR, Shanghai Jiao Tong University	1k	Math
Math-RLVR (Hochlehnert et al., 2025b)	Tencent AI Lab	773k	Math
SYNTHETIC-1 (Mattern et al., 2025)	Prime Intellect	144k	Code
DeepCoder (Luo et al., 2025a)	Agentica Project	24k	Code
Open-R1-CodeForces (Penedo et al., 2025)	Hugging Face	10k	Code
KodCode-V1 (Xu et al., 2025b)	Microsoft GenAI	484k	Code
Code-r1-12k (Liu & Zhang, 2025)	University of Illinois Urbana-Champaign	12k	Code
Multi-Subject-RLVR (Hochlehnert et al., 2025b)	Tencent AI Lab	638k	General
II-Thought-RL-v0 (Internet, 2025)	Intelligent-Internet	342k	General

Table 3: Verified open-source off-the-shelf datasets curated for RL training and their corresponding categories. The statistics for SYNTHETIC-1 denotes the size of its subset of algorithmic coding problems.

DeepScaleR-Preview. DeepScaleR (Luo et al., 2025b) collected around 40k unique contest-level math problems from AIME (1984-2023), AMC (prior to 2023), Omni-MATH (Gao et al., 2024)

and Still [9] datasets. They extract the answer for each problem using gemini-1.5-pro-002, removing duplicate questions, and filtering out unverifiable samples.

Skywork-OR1. Skywork-OR1 (He et al., 2025a) is trained on math and code tasks during the RL phase. The data resource of math is generally similar to DeepScaleR (Luo et al., 2025b), with including extra challenging problems from NuminaMath-1.5 (LI et al., 2024). The code training data is from LeetCode (Xia et al., 2025) and TACO (Li et al., 2023). In preprocessing, Skywork-OR1 performed an elaborated verification of data samples. Skywork-OR1 removed all instances with external URLs or potential figures to verify the validity of problems. Math samples are verified using Math-Verify (Kydlíček, 2024), while code samples are required to include a complete set of unit test cases, with its solution passing all corresponding tests. After preprocessing, deduplication is performed, resulting in a total of 105k math samples and 13.7k code samples. Skywork-OR1 also marked the difficulty level of each sample based on its pass rate when evaluated by DeepSeek-R1-Distilled models.

Open-Reasoner-Zero. Open-Reasoner-Zero (Hu et al., 2025) identifies three key aspects for data curation: quantity, diversity, and quality. Open-Reasoner-Zero collects a total of 129k training data, of which 72k are mainly cleaned from OpenR1-Math-220k (HuggingFace, 2025), and the rest are collected from various sources, including AIME (up to 2023), MATH, Numina-Math collection and Tulu3 MATH. It also leverages additional synthetic data using programmatic approaches to cover other reasoning domains. In controlling data quality, Open-Reasoner-Zero filters out samples with unverifiable formats, such as multiple-choice and proof-oriented problems. Also, it filters out non-English samples for better training stability and final model performance. Additionally, Open-Reasoner-Zero selects a challenging subset of 13k samples from the complete dataset using a intermediate model checkpoint during training, which is used to support the curriculum learning on difficulty of this work, aiming to address the shortcomings of the model and enhancing its performance on challenging scenarios.

Big-Math. Big-Math (Albalak et al., 2025) includes a massive amount of high-quality samples with open-ended problems and uniquely verifiable closed-form solutions. Each sample is categorized by its mathematics domain (eg. sequences and series). Additionally, this dataset provides a new source of 47,000 problems deriving from multiple-choice problems, namely Big-Math-Reformulated. Big-Math performs a very strict filtering and cleaning process to ensure the quality of data samples. First, a strict deduplication based on exact matching and semantic similarity is performed, together with a test set decontamination using MATH-500 and Omni-MATH test sets. Second, possibly invalid samples are removed, including problems with hyperlinks and problems that are unsolvable in 8 rollouts from Llama-3.1-405B or 64 rollouts from Llama-3.1-8B. Third, possibly unverifiable samples are removed, including problems with hyperlinks, multiple choice problems, yes/no and true/false problems, multi-part questions, questions asking for a proof, and non-English problems. Last, miscellaneous unnecessary information (eg. problem scoring) are cleaned from data samples.

DeepMath-103K. DeepMath-103K (He et al., 2025b) is collected to serve as a credible and challenging training data resource while ensuring no contamination with existing benchmarks. From a difficulty distribution estimation of current source datasets, MMIQC and WebInstructSub are selected as the resources of DeepMath-103K as these datasets are sourced more broadly from web content and contain more challenging questions. Then, a strict decontamination against common benchmarks is performed to ensure the integrity of this datasets. Moreover, each samples is rated with its difficulty by prompting GPT-4o based on the annotation guidelines provided by the Art of Problem Solving (AoPS, 2025), and is verified through a rigorous two-stage process. In question formatting and filtering, problem types inherently unsuitable for verification (e.g., proofs) were discarded, and questions phrased conversationally were automatically rewritten into a standardized format that seeks a single, specific numerical or symbolic answer. In answer verification via consistency check, three distinct solution paths are generated for each sample and only samples where all paths extract identical final answers are retained in the final dataset. The procedure ensures that every problem included in DeepMath-103K possesses a final answer that is robustly verifiable using automated rules.

---

[9]https://github.com/RUCAIBox/Slow_Thinking_with_LLMs

Other Datasets for Math Problem Solving. CuratedThoughts (Hochlehnert et al., 2025b) is curated from prevailing SFT datasets, filtering out improper samples to support stable RL training. It is collected from OpenR1-Math-220k (HuggingFace, 2025), OpenThoughts-114k and OpenThoughts-114k-Math (OpenThoughts-Team, 2025), removing multi-part questions, questions asking for a proof, questions referring to figures or charts, and questions without a valid answer. DAPO-Math-17k (Yu et al., 2025) modifies the questions from the AoPS website (AoPS, 2025) so that the expected answer would always be an integer. The simple answers making them easy to parse to minimize errors from math verifiers, providing accurate reward signals during RL training. LIMR (Li et al., 2025b) proposes the learning impact measurement to filter a small amount of samples whose learning patterns complement the model's overall performance trajectory, demonstrating that these samples tend to be more valuable for optimization.

Coding Problem Solving Datasets. A verified sample for coding problem solving should contain a number of unit tests covering the typical cases, boundary conditions, exceptional or invalid inputs, and performance extremes, as well as ensuring full coverage of all branches, conditions, and their combinations. The algorithmic coding problems subset of SYNTHETIC-1 (Mattern et al., 2025) is curated from Apps, Codecontests, Codeforces and TACO datasets. LLM-based post-processing is applied to additionally translate Python problems into Javascript, Rust and C++ problems, resulting in a total of 144k samples. DeepCoder-Preview (Luo et al., 2025a) examined popular coding data resources and filtered out easy samples and unverifiable samples with noisy questions or responses, or flawed or missing test cases. They have chosen verified samples from TACO [10], SYNTHETIC-1 and LiveCodeBench (v5, May 1, 2023 - July 31, 2024) as their train set, and LiveCodeBench (v5, August 1, 2024 - February 1, 2025) as their test set. It is ensured that all problems within this dataset are fully verifiable and have no less than 5 test cases. Open-R1-CodeForces (Penedo et al., 2025) collects more than 10k unique samples with the solutions and unit tests validated from the very first contests of CodeForces all the way to 2025. KodCode-V1 (Xu et al., 2025b) is a fully-synthetic dataset by collecting and rewriting questions from 12 distinct resources and generating the solutions, test cases, and difficulty levels by DeepSeek-R1. Code-r1-12k (Liu & Zhang, 2025) consists of 2K LeetCode samples with generally reliable test cases and 10K verified samples from TACO.

General Domain Datasets. It is an exciting idea to expand the RL training paradigm to more domains than formatted tasks such as math and coding problem solving. To this aim, Su et al. (2025) proposes Math-RLVR and Multi-Subject-RLVR which are annotated with free-form reference answers. The prediction accuracy on these datasets during training are verified by LLMs. Math-RLVR consists of 773k samples covering three educational levels: elementary, middle, and high school. Multi-Subject-RLVR consists of 638k college-level samples written by domain experts for examination purposes, extracted from ExamQA (Yu et al., 2021) which covers at least 48 first-level subjects. Similarly, II-Thought (Internet, 2025) proposes a comprehensive dataset consists of 342k samples covering math, science, code and riddle domains.

3.2 RL Components

With the release of DeepSeek-R1-Zero and DeepSeek-R1, DeepSeek showcases its success in fine-tuning LLMs for complex reasoning tasks through RL. Building on carefully curated training data, replication studies have focused on configuring key aspects of the RL framework to achieve competitive performance: the adoption of effective RL algorithms (e.g., GRPO), and the design of reward systems. Some studies have also explored advanced data sampling strategies to further boost performance. This subsection reviews representative efforts in fine-tuning reasoning language models with RL from the above aspects, highlighting their key contributions from a conclusive perspective. Table 4 provides a comparative view of the methodology for the mentioned studies.

3.2.1 Algorithms

As the most prevalent outcome-reward-based RL methods, PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) are widely used for fine-tuning LLMs. Interestingly, recent replication studies have introduced various modifications to these methods, tailoring them for specific purposes to

---

[10] https://huggingface.co/datasets/likaixin/TACO-verified

Model	Initial Checkpoint	Data Size	RL	Reward
DeepSeek-R1 (Guo et al., 2025)	DeepSeek-V3-Base	–	GRPO	Accuracy, Format
DeepSeek-R1-Zero (Guo et al., 2025)	DeepSeek-V3-Base	–	GRPO	Accuracy, Format
VAPO (Yue et al., 2025b)	Qwen2.5-32B-Base	–	VAPO	Accuracy
VC-PPO (Yuan et al., 2025)	Qwen2.5-32B-Base	–	VC-PPO	Accuracy
Open-Reasoner-Zero-32B (Hu et al., 2025)	Qwen2.5-32B-Base	129k	PPO	Accuracy
SRPO (Zhang et al., 2025b)	Qwen2.5-32B-Base	–	SRPO	Accuracy, Format
DAPO (Guo et al., 2025)	Qwen2.5-32B-Base	17k	DAPO	Accuracy, Length
Skywork-OR1-32B-Preview (He et al., 2025a)	DeepSeek-R1-Distill-Qwen-32B	105k	GRPO	Accuracy, Format
Light-R1-14B-DS (Wen et al., 2025a)	Light-R1-14B-DS-SFT	–	GRPO	Accuracy, Length
Logic-RL (Xie et al., 2025a)	Qwen2.5-7B-Instruct-1M	5k	REINFORCE++	Accuracy, Format
Oat-Zero-7B (Liu et al., 2025c)	Qwen2.5-Math-7B	–	Dr. GRPO	Accuracy
MiMo-7B-RL-Zero (Xiaomi LLM-Core Team, 2025)	MiMo-7B-Base	130k	GRPO	Accuracy
Mini-R1 (Schmid, 2025)	Qwen2.5-3B-Instruct	50k	GRPO	Accuracy, Format
TinyZero (Pan et al., 2025)	Qwen2.5-3B-Base	–	PPO	Accuracy, Format
DeepScaleR-1.5B-Preview (Luo et al., 2025b)	Deepseek-R1-Distilled-Qwen-1.5B	40k	GRPO	Accuracy, Format
GPG-1.5B (Chu et al., 2025b)	Deepseek-R1-Distilled-Qwen-1.5B	–	GPG	Accuracy, Format

Table 4: An overview of the algorithm selction and reward design of competitive open-source DeepSeek-R1 replication studies on RLVR. Models from DeepSeek-R1 series (Guo et al., 2025) are separately listed for comparison. Dashes (–) indicate unavailable numbers.

enhance training effectiveness. We review several representative RL-based LLM fine-tuning algorithms, including REINFORCE (Ahmadian et al., 2024), PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and their variants (Liu et al., 2025c; Yu et al., 2025; Yuan et al., 2025; Hu, 2025; Kimi-Team et al., 2025; Lin et al., 2025b). Moreover, we outline the modifications in these methods along with their underlying motivations, aiming to provide a clear overview of the methodological advancements in outcome-reward-based RL training methods.

LLM Policy Optimization. Recent studies have introduced a groundbreaking post-training paradigm that enhances LLMs' reasoning capabilities through RL-based training. In this framework, the LLM's answer generation process for each query is formulated as an answer sampling policy, and our objective is to optimize this LLM policy to maximize the expected reward of the generated responses. According to Guo et al. (2025); Hu et al. (2025); Kimi-Team et al. (2025), large-scale RL-based LLM policy optimization enables the base LLM to achieve a steady improvement in reasoning accuracy while also exhibiting the emergence of long-chain reasoning in its chain-of-thought.

Suppose each reasoning data pair (q, a) is i.i.d sampled from an underlying distribution $\mathcal{D}$, where each q is a query and a is the corresponding ground-truth answer. Let $\pi_\theta(\cdot|\cdot)$ be the target LLM policy parameterized by θ. The expected reward of the LLM on a sample (q, a) is $\mathbb{E}_{o \sim \pi_\theta(\cdot|q)}[r(o, a)]$, where o is an LLM-generated response to q, and $r(\cdot, \cdot)$ is a predefined reward function that quantifies whether the response o yields a. The objective of RL-based fine-tuning is to maximize the expected reward over the data distribution, i.e.,

$$\max_\theta J(\pi_\theta) \triangleq \mathbb{E}_{(q,a) \sim \mathcal{D}} \mathbb{E}_{o \sim \pi_\theta(\cdot|q)}[r(o, a)]. \tag{2}$$

A straightforward approach to maximize $J(\pi_\theta)$ is to gradually improve the LLM's parameter θ towards the policy gradient direction $\nabla_\theta J(\pi_\theta)$. However, since $\nabla_\theta \mathbb{E}_{o \sim \pi_\theta(\cdot|q)} r(o, a)$ is the gradient of an integral dependent on π_θ, $\nabla_\theta J(\pi_\theta)$ is intractable to compute via standard Monte Carlo sampling. Fortunately, the RL community has developed two powerful policy gradient estimators: REINFORCE (Williams, 1992) and Importance Sampling (Sutton & Barto, 2018):

$$\nabla_\theta \mathbb{E}_{o \sim \pi_\theta(\cdot|q)} r(o, a) = \begin{cases} \mathbb{E}_{o \sim \pi_\theta(\cdot|q)} \left[\nabla_\theta \log \pi_\theta(o|q) \cdot r(o, a) \right] & \text{(REINFORCE)}, \\ \mathbb{E}_{o \sim \pi_{\theta'}(\cdot|q)} \left[\nabla_\theta \left(\frac{\pi_\theta(o|q)}{\pi_{\theta'}(o|q)} \right) \cdot r(o, a) \right] & \text{(Importance Sampling)}, \end{cases} \tag{3}$$

where $\pi_{\theta'}$ is any parameter-frozen LLM policy. Hence, the policy gradient $\nabla_\theta J(\pi_\theta)$ can be effectively approximated using standard Monte Carlo sampling: for each data pair (q, a), we independently generate G responses to q, denoted by $\{o_i\}_{i=1}^G$, using the current LLM π_θ or the frozen LLM $\pi_{\theta'}$, and then approximate the policy gradient estimators by

$$\nabla_\theta J(\pi_\theta) = \begin{cases} \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_\theta(\cdot|q)} \left[\frac{1}{G} \sum_{i=1}^G \nabla_\theta \log \pi_\theta(o_i|q) \cdot r(o_i, a) \right] & \text{(REINFORCE)}, \\ \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta'}(\cdot|q)} \left[\frac{1}{G} \sum_{i=1}^G \nabla_\theta \left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta'}(o_i|q)} \right) \cdot r(o_i, a) \right] & \text{(Importance Sampling)}, \end{cases} \tag{4}$$

For each query q, the procedure of generating G independent responses $\{o_i\}_{j=1}^G$ from $\pi_\theta(\cdot|q)$ is called the 'rollout phase'. During this phase, the LLM policy explores enormous response samples

of varying quality. Then θ is updated to increase the likelihood $\pi_\theta(o_i|q)$ where $r(o_i, a)$ is large, thereby improving the likelihood of generating responses with high rewards. Specifically, REINFORCE is an on-policy method that requires generating new rollouts using the latest LLM policy π_θ. In contrast, the importance sampling estimator can be implemented in an off-policy manner with improved sampling efficiency, as it can reuse past rollouts generated from $\pi_{\theta'}$ by storing the corresponding probability terms $\pi_{\theta'}(o_i|q)$. A common choice is to implement $\pi_{\theta'}$ as $\pi_{\theta_{\text{old}}}$, a past snapshot of the target LLM π_θ, which is updated periodically.

In practice, the reward signals $\{r(o_i, a)\}_{i=1}^G$ are highly sparse, leading to high variance in rollout phases and policy gradient estimation. To mitigate these issues, various techniques have been developed to stabilize LLM policy gradient estimation in (4). These techniques generally fall into three categories: 1) reducing sampling variance by reward normalization or using actor-critic advantage estimation, 2) stabilizing parameter updates by clipping the importance sampling weight $\pi_\theta(o_i|q)/\pi_{\theta_{\text{old}}}(o_i|q)$, and 3) constraining policy shifts by penalizing the KL-divergence $\text{KL}(\pi_\theta|\pi_{\text{ref}})$ between the current LLM policy π_θ and a fixed reference LLM policy π_{ref}.

PPO. Since its introduction in Schulman et al. (2017), Proximal Policy Optimization (PPO) has become one of the most popular actor-critic RL algorithms for LLM policy optimization (Ouyang et al., 2022; Hu et al., 2025). In addition to the target LLM policy π_θ, which serves as the actor model, PPO introduces a critic model V_ϕ—another LLM designed to learn the value for the responses generated by the actor LLM π_θ. Specifically, the PPO objective is

$$J_{\text{PPO}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min\left(r_{i,t}(\theta)\hat{A}_{i,t}(\phi), \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}(\phi)\right)\right)\right], \quad (5)$$

where $r_{i,t}(\theta) \triangleq \pi_\theta(o_{i,t}|q, o_{i,<t})/\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})$ denotes the likelihood ratio between the current LLM policy π_θ and the past LLM policy $\pi_{\theta'}$ calculated on the t-th token prediction step; $\hat{A}_{i,t}(\phi)$ denotes the Gated Advantage Estimator (GAE) (Schulman et al., 2018) computed using the estimated value $V_\phi(o_{i,t}|q, o_{i,<t})$, which estimates the quality of each response generation state. V_ϕ is trained along with π_θ to predict the value of the response generated by π_θ. In practice (Hu et al., 2025), GAE is observed to be a more robust response quality estimator than the raw reward $r(q_i, a_i^*)$, leading to more stable LLM policy optimization.

GRPO. Group Relative Policy Optimization (GRPO) is first proposed (Guo et al., 2025) as an effective and efficient variant of PPO. Specifically, GRPO discards the critic model and GAE calculation in PPO to improve efficiency and memory consumption. To reduce the reward sampling variance, GRPO normalizes the rewards within a group of G rollout outs. In addition to clipping the likelihood ratio terms, GRPO further introduces KL-divergence penalty to ensure that π_θ would not be driven far away from the initial SFT LLM. Specifically, the GRPO objective is

$$J_{\text{GRPO}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min\left(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right) - \beta\text{KL}(\pi_\theta|\pi_{\text{ref}})_{i,t}\right)\right], \quad (6)$$

where $\hat{A}_{i,t} \triangleq (r(o_i, a) - \text{mean}(\{r(o_i, a)\}_{i=1}^G))/\text{std}(\{r(o_i, a)\}_{i=1}^G)$ denotes the group relative reward, and $\mathbf{r} \triangleq \{r(o_i, a)\}_{i=1}^G$ denotes the rewards of the response group corresponding to each sample (q, a). GRPO also incorporates the K3 KL-divergence estimator (Schulman., 2020):

$$\text{KL}(\pi_\theta|\pi_{\text{ref}})_{i,t} \triangleq \frac{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t}))} - \log\frac{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t}))} - 1. \quad (7)$$

DeepSeek-R1 (Guo et al., 2025) shows that GRPO achieves stable large-scale LLM policy optimization that incentivizes the long CoT pattern in large-scale LLMs.

REINFORCE++. REINFORCE++ (Hu, 2025) stabilizes the policy gradient update by incorporating token-wise KL-divergence penalty into the reward function. Like GRPO, REINFORCE++

normalized the penalized rewards within each rollout group. Formally, the training objective is

$$J_{\text{REINFORCE++}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon \right) \hat{A}_{i,t} \right) \right) \right], \quad (8)$$

where $\hat{A}_{i,t} \triangleq (\hat{r}_i - \text{mean}(\{\hat{R}_i\}_{i=1}^G)) / \text{std}(\{\hat{R}_i\}_{i=1}^G)$, and $\hat{R}_i$ is the penalized reward defined as

$$\hat{R}_{i,t} \triangleq r(o_i, a) - \beta \sum_{j=t}^{|o_i|} \log \left(\frac{\pi_{\theta_{\text{old}}}(o_{i,j}|q, o_{i,<j})}{\pi_{\theta_{\text{ref}}}(o_{i,j}|q, o_{i,<j})} \right). \quad (9)$$

REINFORCE on LLM.   Kimi-Team et al. (2025) shows that REINFORCE-like policy gradient can achieve stable training on 72B LLMs. Compared to the basic REINFORCE in (4), Kimi-Team et al. (2025) employs centralized rewards and adds K2 KL-divergence penalty (Schulman., 2020). This yields the following modified REINFORCE policy gradient

$$\nabla_\theta J_{\text{KIMI}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\nabla_\theta \log \pi_\theta(o_i|q)(r(o_i,a) - \bar{r}) - \frac{\beta}{2} \nabla_\theta \left(\log \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)} \right)^2 \right) \right], \quad (10)$$

where $\bar{r} \triangleq \text{mean}(\{r(o_i,a)\}_{i=1}^G)$ denotes the average reward among each rollout group.

DAPO.   Yu et al. (2025) identifies several critical shortcomings in the original GRPO algorithm, including entropy collapse, training instability, and biased loss. Entropy collapse refers to the rapid decline in policy entropy during training, where the sampled responses for certain prompts become nearly identical, reducing diversity. Additionally, the algorithm suffers from a gradient decreasing issue: when some prompts consistently achieve perfect accuracy, the resulting zero advantage leads to inefficient and unstable training. Moreover, the original sample level loss computation, where token level losses are first averaged within each sample and then aggregated across samples, introduces a length bias, as tokens in longer responses contribute less to the overall loss. To address these issues, the authors propose the Decoupled Clip and Dynamic sAmpling Policy Optimization (DAPO) algorithm:

$$J_{\text{DAPO}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)}$$
$$\left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}} \right) \hat{A}_{i,t} \right) \right] \quad (11)$$
$$\text{subject to} \quad 0 < |\{o_i \,|\, \text{is\_equivalent}(a, o_i)\}| < G,$$

where $\hat{A}_{i,t} \triangleq (r(o_i, a) - \text{mean}(\{r(o_i,a)\}_{i=1}^G)) / \text{std}(\{r(o_i,a)\}_{i=1}^G)$. Specifically, ε is decoupled into ε_{low} and $\varepsilon_{\text{high}}$, with $\varepsilon_{\text{high}}$ set to a higher value to allow more room for increasing low-probability tokens, helping to address entropy collapse issues. The dynamic sampling constraint $0 < |\{o_i \,|\, \text{is\_equivalent}(a, o_i)\}| < G$ ensures that rollouts from a given prompt contain both correct and incorrect outputs, resulting in nonzero advantages that improve training efficiency and stability. Finally, the average is computed over all tokens instead of the original sample level loss. This encourages longer sequences to make greater contributions to the overall gradient update, which is critical in long CoT RL scenarios.

Dr. GRPO.   Similarly, Liu et al. (2025c) reveals two biases in the original GRPO algorithm: a response level length bias and a question level difficulty bias. The response level length bias arises from dividing by $|o_i|$, which encourages the policy to favor shorter correct responses while preferring longer incorrect ones. The question level difficulty bias comes from normalizing the centered outcome reward by the standard deviation $\text{std}(\{R_i\}_{i=1}^G)$, which causes questions with lower variance in rewards, typically those that are either too easy or too hard, to receive greater weight

during policy updates. To address these biases, they propose GRPO Done Right (Dr. GRPO):

$$J_{\text{Dr.GRPO}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)}$$

$$\left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \Big(\min\big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\big) - \beta\text{KL}(\pi_\theta|\pi_{\text{ref}}) \Big) \right], \quad (12)$$

$$\text{where} \quad \hat{A}_{i,t} \triangleq (r(o_i,a) - \text{mean}(\{r(o_i,a)\}_{i=1}^G))/\text{std}(\{r(o_i,a)\}_{i=1}^G).$$

By eliminating both the normalization terms $1/|o_i|$ and $1/\text{std}(\{r(o_i,a)\}_{i=1}^G)$, Dr. GRPO is able to enhance token efficiency without compromising reasoning performance.

CPPO. Lin et al. (2025b) highlights a key limitation of GRPO: Although effective, it demands substantial computational resources due to the need for sampling multiple completions per query. To address this issue, they propose Completions Pruning Policy Optimization (CPPO):

$$J_{\text{CPPO}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)}$$

$$\left[\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \Big(\min\big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\big) - \beta\text{KL}(\pi_\theta|\pi_{\text{ref}}) \Big) \right], \quad (13)$$

where $\mathcal{I} \triangleq \{i \in \{1,\ldots,G\} \mid |\hat{A}_{i,t}| \geqslant \gamma\}$, γ is a predefined threshold that filters out low-impact completions, and $\hat{A}_{i,t} \triangleq (r(o_i,a) - \text{mean}(\{r(o_i,a)\}_{i=1}^G))/\text{std}(\{r(o_i,a)\}_{i=1}^G)$. This ensures that only completions (i.e. rollout samples) with sufficiently high absolute advantage contribute to the policy update. As a result, CCPO accelerates the training process by skipping the forward pass and gradient backpropagation on rollout samples with low advantages.

GPG. Chu et al. (2025b) proposes Group Policy Gradient (GPG), a REINFORCE-based method that removes complex components and directly optimizes the true objective, bypassing the use of surrogate losses:

$$J_{\text{GPG}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_\theta(\cdot|q)} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} (-\log\pi_\theta(o_{i,t}|q, o_{i,<t})\hat{A}_{i,t}) \right], \quad (14)$$

where $\hat{A}_{i,t} \triangleq (r(o_i,a) - \text{mean}(\{r(o_i,a)\}_{i=1}^G))/\text{std}(\{r(o_i,a)\}_{i=1}^G)$. This eliminates the need for the implementation of the critic model and the reference model, offering significant advantages for scalability in distributional training.

VC-PPO. Yuan et al. (2025) identifies two critical failure modes of PPO in long CoT reasoning: value initialization bias and reward signal decay. The former issue stems from initializing the value model with a reward model trained only on `<EOS>` tokens, resulting in a position-dependent advantage bias that favors shorter completions. The latter issue is caused by the trace decay rate $\lambda < 1$ used in reward propagation during GAE computation (Schulman et al., 2018), which severely weakens the reward signal for earlier tokens in long sequences. To address these issues, they propose Value-Calibrated PPO (VC-PPO), which introduces two modifications: 1) value pretraining, where the value model is pretrained under a fixed SFT policy using Monte Carlo returns (i.e., setting $\lambda = 1$) to eliminate the initial value bias; 2) decoupled-GAE, which uses different GAE trace decay λ values for policy and value updates, allowing the critic model to achieve unbiased advantage estimation by setting $\lambda = 1$, and enabling the actor model to achieve variance reduction in advantage estimation by setting $\lambda = 0.95$. This two-pronged calibration significantly improves PPO's stability and performance on long-form reasoning benchmarks like AIME24.

VAPO. Yue et al. (2025b) introduces Value-based Augmented Proximal Policy Optimization (VAPO), a value-based RL framework outperforming value-free methods in long Chain-of-Thought reasoning. VAPO adopts Value-Pretraining and Decoupled-GAE from VC-PPO (Yuan et al., 2025) to mitigate value model bias. To handle heterogeneous sequence lengths, VAPO uses token-level loss from DAPO (Yu et al., 2025) and employs the Length-Adaptive GAE trace decay rate $\lambda_{\text{policy}} \triangleq 1 - 1/(\alpha l)$, where α is a scaling hyperparameter and l denotes the length of each rollout sample. To address reward sparsity, the VAPO objective employs the Clip-Higher trick proposed in

DAPO and integrates an additional negative log-likelihood (NLL) penalty:

$$J_{\text{VAPO}}(\pi_\theta) \triangleq J_{\text{PPO-CH}}(\pi_\theta) + \mu J_{\text{NLL}}(\pi_\theta), \tag{15}$$

$$J_{\text{PPO-CH}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}$$

$$\left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \min\left(r_{i,t}(\theta)\hat{A}_{i,t}(\phi), \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}} \right) \hat{A}_{i,t}(\phi) \right) \right], \tag{16}$$

$$J_{\text{NLL}}(\pi_\theta) \triangleq \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[-\frac{1}{\sum_{i\in\mathcal{T}}|o_i|} \sum_{i\in\mathcal{T}} \sum_{t=1}^{|o_i|} \log \pi_\theta(o_{i,t}|q, o_{i,<t}) \right], \tag{17}$$

with $\hat{A}_{i,t}(\phi)$ denoting the GAE (Schulman et al., 2018) estimated by the critic model $V(\phi)$, $\mathcal{T}$ denoting the set of rollout indices that achieve correct answers and $\mu > 0$ denoting the penalty rate. In essence, the introduced NLL penalty is interpreted to perform SFT over the correct rollout samples. Furthermore, VAPO adopts the Group Sampling technique proposed in GRPO (Shao et al., 2024) to effectively generate discriminative positive and negative samples within the same prompt context. Collectively, these integrated approaches make VAPO a robust benchmark for effective and stable long-form reasoning within value-based RL frameworks.

3.2.2 REWARDS

Rewards are the cornerstone of RL training, as they define the optimization objective and guide the model's behavior. A well-designed reward provides clear and consistent signals that help the agent learn effective policies. However, reward models are often prone to reward hacking (Amodei et al., 2016; Everitt et al., 2017; Weng, 2024), prompting recent research to favor rule-based outcome reward systems. These typically fall into three categories:

Accuracy Rewards. Accuracy rewards evaluate whether a response is correct, typically assigning a score of 1 for correct answers and 0 or -1 for incorrect ones. They are widely regarded as the most fundamental type of reward and are sometimes used exclusively, reflecting a minimalist approach to reward design (Hu et al., 2025; Liu et al., 2025c).

Format Rewards. Format rewards encourage responses to follow predefined structures or reasoning formats, typically rewarding correct formatting with 1 and penalizing deviations with 0 or -1. While they are intended to promote clarity and consistency, Hu et al. (2025) found that models trained solely with accuracy rewards, when guided by well-designed prompts, can still quickly learn and reinforce the desired formatting. This suggests that explicit format rewards may be unnecessary in some cases. Moreover, such rewards may inadvertently incentivize reward hacking behaviors (Hu et al., 2025; Xie et al., 2025a).

Length Rewards. These rewards influence the verbosity of the model's output. Some approaches reward generating responses of a desired length (Aggarwal & Welleck, 2025), while others incentivize brevity without sacrificing accuracy (Arora & Zanette, 2025). Yu et al. (2025) implement a linear length penalty when responses exceed a predefined maximum length, whereas Yeo et al. (2025) propose a cosine-based reward that encourages longer reasoning processes for incorrect answers and more concise ones for correct responses, which is also adopted by Wen et al. (2025a).

3.2.3 SAMPLING STRATEGIES

Intuitively, properly selecting samples during training is crucial for effective RL training. On the one hand, curriculum learning methods that gradually increase task difficulty during training improve the utility of difficult samples. Several studies have adopted these strategies to boost the training process. Open-Reasoner-Zero (Hu et al., 2025) leveraged a two-step curriculum learning process to efficiently use difficult samples. SRPO (Zhang et al., 2025b) adopts a two-stage training paradigm to bridge the gap between math and coding tasks: the first stage emphasizes mathematical data to develop step-by-step reasoning, followed by coding data to build on this foundation, enabling progressive improvement in both domains. On the other hand, the proper use of rejection sampling techniques could improve the general sample efficiency and stabilize training. Light-R1 (Wen et al., 2025a) implements a broader two-sided weight clipping mechanism for importance sampling. This

mechanism limit the influence of extreme values to stabilize the training process. DAPO (Yu et al., 2025) and Skywork-OR1 (He et al., 2025a) adopt dynamic sampling which filters out zero advantage sample groups to increase sample efficiency and training stability, according to their claims that these groups do not contribute to the policy loss, but may contribute to the KL loss or entropy loss, leading to a more unstable training process. Besides, Skywork-OR1 (He et al., 2025a) and SRPO (Zhang et al., 2025b) introduce epoch-level history resampling strategies that drops samples with all correctly predicted samples in last epoch to focus training on harder cases, enhancing learning efficiency. MiMo (Xiaomi LLM-Core Team, 2025) further argues that such a strategy introduces instability in policy updates, and develops an easy data resampling strategy, managing to improve sampling efficiency without risking policy collapse. This strategy maintains an easy data pool during training where problems with perfect pass rates are stored, and samples data from this pool with a 10% probability when performing rollouts.

3.3 ANALYSIS AND DISCUSSIONS

Model	Initial Checkpoint	AIME24	AIME25	MATH500
DeepSeek-R1 (Guo et al., 2025)	DeepSeek-V3-Base	79.8*	–	97.3*
DeepSeek-R1-Zero (Guo et al., 2025)	DeepSeek-V3-Base	71.0*	–	95.9*
OpenAI o4 mini (OpenAI, 2025)	–	93.4*	92.7*	–
Seed-Thinking-v1.5 (ByteDance-Seed, 2025)	–	86.7*	74.0*	–
Qwen3-235B (ByteDance-Seed, 2025)	–	85.7*	81.5*	–
VAPO (Yue et al., 2025b)	Qwen2.5-32B-Base	60.4	–	–
SRPO (Zhang et al., 2025b)	Qwen2.5-32B-Base	50.0	–	–
DAPO (Guo et al., 2025)	Qwen2.5-32B-Base	50.0	–	–
VC-PPO (Yuan et al., 2025)	Qwen2.5-32B-Base	48.8	–	–
Open-Reasoner-Zero-32B (Hu et al., 2025)	Qwen2.5-32B-Base	48.1	36.0	92.2
DeepSeek-R1-Zero-Qwen-32B (Guo et al., 2025)	Qwen2.5-32B-Base	47.0*	–	91.6*
Skywork-OR1-32B-Preview (He et al., 2025a)	DeepSeek-R1-Distill-Qwen-32B	79.7	69.0	–
DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025)	Qwen2.5-32B-Base	72.6†	–	94.3†
Light-R1-14B-DS (Wen et al., 2025a)	Light-R1-14B-DS-SFT	74.0	60.2	
DeepSeek-R1-Distill-Qwen-14B (Guo et al., 2025)	Qwen2.5-Math-14B	69.7†	–	93.9†
Skywork-OR1-Math-7B (He et al., 2025a)	DeepSeek-R1-Distill-Qwen-7B	69.8	52.3	–
Skywork-OR1-7B-Preview (He et al., 2025a)	DeepSeek-R1-Distill-Qwen-7B	63.6	45.8	–
Light-R1-7B-DS (Wen et al., 2025a)	DeepSeek-R1-Distill-Qwen-7B	59.1	44.3	–
DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025)	Qwen2.5-Math-7B	55.5†	–	92.8†
MiMo-7B-RL-Zero (Xiaomi LLM-Core Team, 2025)	MiMo-7B-Base	56.4	46.3	93.6
Oat-Zero-7B (Liu et al., 2025c)	Qwen2.5-Math-7B	43.3	–	80.0
DeepScaleR-1.5B-Preview (Luo et al., 2025b)	Deepseek-R1-Distilled-Qwen-1.5B	43.1	–	87.8
GPG-1.5B (Chu et al., 2025b)	Deepseek-R1-Distilled-Qwen-1.5B	33.3	–	87.6
DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025)	Qwen2.5-Math-1.5B	28.9†	–	83.9†

Table 5: Performance on math reasoning tasks of competitive open-source DeepSeek-R1 replication studies on RLVR, where models trained from base models and other models are separately listed for better comparison. Performance of popular proprietary RLVR models (marked with *) and R1-distilled models (marked with †) are also listed for better comparison. Dashes (–) indicate unavailable results.

As displayed in Table 5, although DeepSeek-R1 (Guo et al., 2025) reported that smaller models (e.g., not larger than 32B) may fail to match the performance of distillation models through RL training, the community has actively sought solutions to this limitation. In this subsection, we present insights from several projects that attempt to replicate RLVR performance on LLMs ranging from 1.5B to 32B.

3.3.1 RECIPES OF TRAINING DATA

Quantity and Diversity. Quantity and diversity are emphasized as key aspects in training reasoning language models suitable for multiple tasks. Skywork-OR1 (He et al., 2025a), Seed-Thinking-v1.5 (ByteDance-Seed, 2025), MiMo (Xiaomi LLM-Core Team, 2025) and Qwen3 series (Qwen-Team, 2025a) have all proposed that they leverage massive RL data from various domains during training. Open-Reasoner-Zero (Hu et al., 2025) leverages data synthesis and self-distillation to expand the training dataset.

Difficulty. Several works introduce their data preparation pipeline to construct challenging datasets for RL training, providing inspirations on difficulty rating. Light-R1 (Wen et al., 2025a)

and Skywork-OR1 (He et al., 2025a) conduct offline data selection that leverages a trained checkpoint to sample and verify responses for the query of each training sample, keeping only the samples with a moderate pass rate, and filtering out the samples with overly high or low pass rates, indicating that the corresponding queries are either too easy or too hard. DeepScaleR (Luo et al., 2025b) revealed that samples with an overly high pass rate are too easy for model training, while samples with a zero pass rate are often unverifiable or contain errors, therefore, both should be filtered out. Open-Reasoner-Zero (Hu et al., 2025) further adopts this filtering strategy to construct training data from synthetic and distilled data that is more noisy. KodCode (Xu et al., 2025b) performs difficulty rating with an off-the-shelf LLM rather than a trained checkpoint of their own model. LIMR (Li et al., 2025b) proposes the learning impact measurement to filter samples whose learning patterns complement the model's overall performance trajectory, proving that these samples tend to be more valuable for training.

Data Cleaning.   As a fundamental step during data preparation, data cleaning is crucial to construct less noisy datasets for effective training. Especially, for RL training on reasoning tasks, several works emphasize the necessity to filter out unsolvable questions and unverifiable answers. DAPO (Yu et al., 2025) modifies the questions from the original samples, ensuring that the expected answers are always integers. The advantages of such simple answers is that they are easy to parse, minimizing errors generated by formula parsers, and providing accurate reward signals during RLVR. BigMath (Albalak et al., 2025) conducts a very strict cleaning process to remove questions with hyperlinks, referring to figures or charts, containing multiple sub-questions, and non-English questions. It also removes questions being unsuitable for verification, including multiple-choice and mathematical proving questions. These strategies are also adopted by various works including Open-Reasoner-Zero (Hu et al., 2025) and SRPO (Zhang et al., 2025b), in which MiMo (Xiaomi LLM-Core Team, 2025) performs data cleaning via an off-the-shelf LLM.

De-duplication and Decontamination.   De-duplication and decontamination are also important in constructing RL training datasets. When collecting data from multiple sources, Light-R1 (Wen et al., 2025a) emphasized the necessity of decontamination to ensure fair evaluation, and DeepScaleR (Luo et al., 2025b) and Skywork-OR1 (He et al., 2025a) performed elaborated de-duplication for efficient training.

Curriculum Learning Based on Data Difficulty.   The training process of Open-Reasoner-Zero (Hu et al., 2025) adopts curriculum learning: the 32B model was initially trained for 1100 steps with data sampled from the full dataset, followed by the selection of a challenging 13k subset based on the model's success rate, which was then used to fine-tune the model for improved performance on the most difficult reasoning problems.

Overall.   We observe a consistent trend: datasets used for RL training are carefully designed to include data where models are likely to make mistakes (i.e., models neither consistently succeed nor completely fail). Such uncertainty creates opportunities for learning. This calibrated challenge level encourages models to engage in deeper reasoning and reflection, often resulting in longer and more informative responses.

3.3.2   RL ALGORITHM DESIGN

REINFORCE, PPO, and GRPO.   As discussed in Section 3.2, existing efforts primarily utilize algorithms such as REINFORCE (Williams, 1992), PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and their variants. For instance, Open-Reasoner-Zero (Hu et al., 2025) adopts vanilla PPO instead of GRPO. Their empirical studies reveal that vanilla PPO offers a notably stable and robust training process when using GAE parameters of $\lambda = 1.0$ and $\gamma = 1.0$. Similarly, Logic-RL (Xie et al., 2025a) reports that PPO ($\lambda = 1.0$ and $\gamma = 1.0$) significantly outperforms GRPO and REINFORCE++ (Hu, 2025) in both accuracy and reward. Moreover, REINFORCE++ consistently surpasses GRPO across nearly all evaluation metrics, with GRPO demonstrating the weakest overall performance among the three RL algorithms. However, Logic-RL only trains on its K&K text logic dataset, which may not generalize to math and coding domains. Nonetheless, GRPO and its variants have also yielded promising results. Light-R1 (Wen et al., 2025a) reports substantial performance improvements using GRPO on DeepSeek-R1-Distill-Qwen-14B.

Variants. DAPO (Yu et al., 2025) utilizes GRPO with clip-higher, dynamic sampling, and token-level policy gradient loss, and achieves similarly strong results on AIME 2024 using Qwen2.5-32B-Base. In addition, MiMo-7B-RL-Zero (Xiaomi LLM-Core Team, 2025) adopts GRPO with clip-higher and dynamic sampling to achieve effective training starting from a 7B base model. Dr. GRPO (Liu et al., 2025c) demonstrates effectiveness on a 7B base model by introducing simple modifications to the original GRPO–namely removing the normalization terms for response length and advantage.

Overall. The community has made engineering efforts to adapt the algorithms for more stable training. However, these algorithms are not significantly different from the theoretical perspective.

3.3.3 MODEL SIZE AND TYPE

Model sizes. Existing studies have demonstrated the effectiveness of RL across a wide range of model sizes, from 1.5B to 32B parameters. Specifically, DeepScaleR (Luo et al., 2025b) surpasses o1-preview on AIME24 using DeepSeek-R1-Distill-Qwen-1.5B by scaling RL training. Open-Reasoner-Zero (Hu et al., 2025), Logic-RL Xie et al. (2025a), and Dr. GRPO (Liu et al., 2025c) have all shown that RL is effective on 7B base models. Light-R1 (Wen et al., 2025a) further demonstrates that GRPO yields strong performance on DeepSeek-R1-Distill-Qwen-14B. In addition, both Open-Reasoner-Zero (Hu et al., 2025) and DAPO (Yu et al., 2025) provide evidence that GRPO remains effective when applied to 32B base models.

Model types. Studies have shown the effectiveness of RL in different model types, including both base and long-CoT models (i.e., R1-distilled models and their fine-tuned variants). Light-R1 (Wen et al., 2025a) first enhances R1-distilled models through SFT, then demonstrates further performance improvements via RL optimization. Skywork-OR1 (He et al., 2025a) achieves significant gains by applying RL directly to R1-distilled models.

3.3.4 CONTEXT LENGTH

Maximum Response Length. Improvements in reasoning capabilities are often associated with longer responses, as reflection and rethinking may occur during the reasoning process. As such, maximum response length is another important factor in RL training. If the allowed response length is too short, longer rollouts may be cut off, resulting in zero reward even when the reasoning trajectories are valid. To address this, Light-R1 (Wen et al., 2025a) sets the maximum response length to 24k, with training response length converging to approximately 9k. DAPO (Yu et al., 2025) uses a 16k maximum response length, where training response length converges to around 5k.

Curriculum Learning Based on Maximum Response Length. DeepScaleR (Luo et al., 2025b) adopts a progressive approach, gradually increasing the maximum response length from 8k to 16k and then to 24k, with performance improving consistently at each step. Similarly, Skywork-OR1 (He et al., 2025a) employs such multi-stage training with progressively extended maximum response lengths, reaching up to 32k.

Truncated Rollouts. Skywork-OR1 (He et al., 2025a) conducts an ablation study to examine whether truncated rollouts should be masked (i.e., excluded from advantage calculation to avoid penalizing these rollouts). However, experimental results show that applying this masking strategy does not yield improved scaling behavior in later training stages, typically when the context length reaches 32k. Consequently, Skywork-OR1 opts not to apply masking for truncated rollouts during training.

3.3.5 REWARD MODELING

Accuracy Reward. One of the key factors contributing to the success of RL in LLMs is the use of straightforward accuracy rewards. The minimal reward function design reduces the risk of reward hacking by leaving little room for unintended optimization. However, rule-based reward system could fail in corner cases. To address this limitation, Seed-Thinking-v1.5 (ByteDance-Seed, 2025) proposes a more generalizable approach that leverages LLMs to evaluate a wide range of scenarios. Their framework includes a Seed-Verifier for straightforward answer verification and a

Seed-Thinking-Verifier designed for cases requiring in-depth analytical reasoning. MiMo (Xiaomi LLM-Core Team, 2025) proposes Test Difficulty Driven Reward as a sample-difficulty-aware mechanism for better accuracy rewarding.

Other Rewards. Apart from accuracy reward, several works have explored the incorporation of additional rewards or penalties. For example, Open-Reasoner-Zero (Hu et al., 2025), Logic-RL (Xie et al., 2025a), and DeepScaleR (Luo et al., 2025b) integrate formatting considerations into their reward modeling. Notably, Open-Reasoner-Zero reports that the format reward rapidly saturates, typically reaching its maximum within approximately 60 steps. However, there are no rigorous ablation studies to prove the effectiveness of the format reward. DAPO (Yu et al., 2025) assigns a punitive reward to truncated samples to reduce the reward noise of the training process, which successfully stabilizes the generation entropy and results in better performance.

3.3.6 KL Loss

The KL loss is commonly utilized to constrain the divergence between the online policy and the frozen reference policy. However, ablation studies in Open-Reasoner-Zero (Hu et al., 2025) suggest that KL regularization may not be essential for large-scale RL. In fact, it can significantly restrict the increase in response length. Similarly, DAPO (Yu et al., 2025), Dr. GRPO (Liu et al., 2025c), SRPO (Zhang et al., 2025b) and MiMo (Xiaomi LLM-Core Team, 2025) omit the KL loss during training and still achieve strong performance on various model sizes. On the other hand, Light-R1 (Wen et al., 2025a) and Logic-RL (Xie et al., 2025a) retain the KL loss and also report substantial improvements. Skywork-OR1 (He et al., 2025a) conducts an ablation study on DeepSeek-R1-Distill-Qwen-7B and observes that incorporating KL loss causes the actor to stay too closely aligned with the reference model. The KL divergence quickly drops toward zero, limiting policy exploration. As a result, performance on AIME24 plateaus, with limited improvement over training. Based on these findings, Skywork-OR1 chooses not to apply KL loss during training.

3.4 RLVR ON OTHER TASKS

The complex reasoning ability of DeepSeek-R1 has been significantly enhanced through RLVR, enabling its success in various reasoning-intensive tasks, such as complex context understanding and problem solving. Fundamentally, RLVR allows an LLM agent to learn and perform any task with feasible answer verification, stimulating its complex reasoning ability without requiring human-guided process supervision. Building on this inspiration, several works have explored the effectiveness of the complex reasoning paradigm with RLVR on various tasks.

Logical Reasoning. TinyZero (Pan et al., 2025) and Mini-R1 (Schmid, 2025) attempts to reproduce the "aha moment" of Deepseek R1 on the countdown game with simple rule-based outcome reward. The Countdown game is a numerical puzzle in which players use a set of randomly drawn numbers and basic arithmetic operations (+, -, ×, ÷) to reach or approximate a given target number as closely as possible. In their rule-based reward system, there contains (1) format reward that ensures the model's response follows think-answer format, (2) validity reward that guarantees that the answer uses each number exactly once, and (3) accuracy reward that requires the final answer to correctly compute the target number. Similarly, Dalal (2025b;c) examine the reward design on solving Sudoku puzzles. Apart from the necessary format compliance reward, these works carefully crafted the validity and accuracy rewards of the puzzle. For validity, solutions must be presented in a readable grid format, adhering to criteria such as the number of rows and columns, and the proper use of box-drawing characters. Additionally, solutions are required to fully preserve the original clues given by the inputs, and comply with the game's rules which prohibit repeated digits in any row, column, or 3×3 box. For accuracy, solutions are evaluated based on the ratio of empty cells correctly filled by the model, with completely correct solutions receiving an extra large reward. Logic-RL(Xie et al., 2025a) and ZebraLogic (Lin et al., 2025a) explore the capability of reasoning language models on deductive reasoning puzzles, where the methods can only be supervised by the correctness of outcome, and unsolvable samples may exist. These works have found that the design of format rewards should be elaborated to avoid hacking and encourage reflection, and extra reward on completely correct answer is necessary, which should also apply to those tasks with simple answer format, such as math/code problem solving and QA tasks.

Application-oriented Tasks. Reasoning language models are expected to learn to tackle real-world application-oriented tasks through thinking, planning and reflecting. To this end, SWE-RL (Wei et al., 2025) introduces a RL-based approach for GitHub issue fixing. Given the incorrect code context and the corresponding issue information, the LLM is required to reasoning about the issue and fix the issue by generating a corrected program. SWE-RL designs a rule-based reward function that computes the sequence similarity between the generated solution and the ground truth repaired program (extracted from the oracle patch merged by the pull request) as the reward. Additionally, solutions with wrong format should receive a large negative penalty. Similar to SWE-RL, MT-R1-Zero (Feng et al., 2025b) proposes a a rule-metric mixed reward mechanism for training on machine translation. RAG-RL (Huang et al., 2025a) equips LLM with retrieval-augmented generation (RAG) capabilities for multi-hop QA using a rule-based reward system comprising of three components: answer rewards, citation rewards, and formatting rewards. Answer rewards evaluate the exact match between model predictions and ground truth results to incentivize correct final answers. Citation rewards count the recall of relevant citations cited in the final answer to encourage effective citations. Formatting rewards utilize a binary function to enforce the desired output format, ensuring to present proper XML tags and required headings while preventing excessive text and raw Unicode. RLSF (Jha et al., 2024) attempts to address chemistry tasks with rule-based evaluation generated by RDKit (2025) as the rewards, which is a token-level vector based on the presence or absence of any syntactical errors. These chemical tasks includes (1) Forward synthesis, which involves predicting the product of a chemical reaction based on given reactants and reagents; (2) Retrosynthesis, which involves determining the reactants required to create a specific product; and (3) Molecular generation, which involves generating a molecule that meets specific requested chemical and biological properties in natural language.

Exploration Beyond Supervision. Through the reinforcement learning process, exciting observations reveal that LLMs have demonstrated remarkably promising and unexpected capabilities. Several works have discovered the emergence of new abilities in LLMs through RL on complex reasoning tasks, without the guidance of any supervision. RL-Poet (Doria, 2025) tunes Pleias-350M with 200K verses, transforming the small language model into a poet using only format rewards rather than outcome-based rewards. The rule-based reward system of RL-Poet consists of three components: (1) Non-repetition reward that penalizes repetition; (2) Verse reward that enforces a structured verse format, requiring the mean line length to be within 30% of the length of prompt, and at least 80% of lines to start with uppercase; and (3) Quatrain reward that ensures the output to be formatted with four-line verse blocks, adhering to the standard quatrain structure. The trained model could generate literary poems across diverse topics and moods, demonstrating its creative writing capabilities. More excitingly, Dalal (2025a) explore the potential of RL for knowledge discovery, by extending its utility to discover a more efficient sorting algorithm. During the RL process, the LLM is required to improve the efficiency of a baseline sorting algorithm on a series of competitive test cases. These test cases are meticulously designed to emphasize pattern diversity, size scaling, data type variation, and difficulty distribution. The model is optimized using rule-based rewards. In addition to the necessary format and validity rewards, the critical performance reward evaluates the logarithmic execution time on given test cases. Experiments show that although the baseline sorting algorithm (i.e., Timsort) is already good enough, the model have discovered several outstanding algorithms, in which the Hybrid Partitioning with QuickSelect achieved a 47.92x speedup over the baseline Timsort implementation on a dataset of 42,385 elements with a Gaussian distribution. These results highlight the potential of complex reasoning language models to surpass the limitations of supervised data resources, and even humans, by adopting RL training strategies.

4   MORE DIRECTIONS

While recent efforts have made significant progress in replicating and extending the capabilities of DeepSeek-R1, several open questions and challenges remain in the development of robust reasoning language models. In this section, we highlight emerging directions that appear in recent literature or hold potential to shape the next generation of reasoning models. Each subsection explores a complementary aspect, from alternative training methods and alignment strategies to broader concerns around generalizability, robustness, safety, and inclusivity.

4.1 Alternative Approaches for Reasoning Enhancement

While reinforcement learning from verifiable rewards (RLVR) has driven notable progress in reasoning language models, its current form remains limited, particularly in capturing intermediate reasoning steps and aligning with human expectations. To address these gaps, recent research has explored alternative approaches that complement or extend traditional RLVR techniques. In this section, we discuss two emerging directions: (i) more expressive and step-aware reward modeling methods, and (ii) preference optimization strategies that reduce computational overhead while improving training stability.

Reward Modeling Techniques. The effectiveness of reinforcement learning in training reasoning language models is largely dependent on the quality and alignment of the reward model, particularly its ability to reflect human preferences or factual correctness rooted in scientific principles. Traditionally, reward model quality is assessed by its accuracy in assigning feedback. However, as argued by Razin et al. (2025), accuracy alone is insufficient. An accurate reward model does not necessarily make for an effective teacher. Empirical results from reinforcement learning with human feedback (RLHF) setups demonstrate that low reward variance can significantly hinder learning, regardless of a reward model's accuracy. Specifically, it leads to a flattened optimization landscape, resulting in slow convergence and worse performance than less accurate models with higher reward variance. Additionally, reward models can suffer from compatibility issues in behavior across different language models. A reward model that provides more informative gradients and high reward variance for one model may produce low reward variance for another, significantly impeding effective learning.

To improve the optimization landscape characterized by low reward variance, enhancing the reward model can be achieved through several strategies. Process-level Reward Modeling (PRM) (Xiong et al., 2024; Li et al., 2024b; Song et al., 2025; Chen et al., 2025) transcends simplistic outcome-based score annotations by providing feedback at each intermediate step within a reasoning process, rather than solely assessing the final outcome. This granular supervision enables models to navigate complex, multi-step tasks more effectively, ensuring that each action aligns with the desired reasoning trajectory. By focusing on step-level evaluations, PRM introduces a more stochastic reward pattern, enhancing the model's adaptability and robustness in dynamic environments. Notably, rStar-Math (Guan et al., 2025) enhances PRM by incorporating a Process Preference Model (PPM). The PPM is trained to assess intermediate reasoning steps by providing step-level preference data, rather than relying solely on final outcomes. This approach allows the model to distinguish between more and less promising reasoning paths during Monte Carlo Tree Search. Additionally, they implement a self-evolution strategy within rStar-Math, where both the policy small language model and the PPM are iteratively refined from scratch. Through multiple rounds of evolution, these models progressively enhance their reasoning capabilities, leading to significant performance improvements. Besides, PRIME (Cui et al., 2025) introduces a scalable reinforcement learning framework to enhance reasoning capabilities in language models. PRIME employs an implicit PRM that is trained solely on outcome labels, thereby eliminating the need for expensive, manually annotated step-level labels. This approach enables online updates using policy rollouts and outcome labels, ensuring scalability and adaptability during reinforcement learning training. By integrating implicit process rewards with traditional outcome rewards, PRIME effectively computes advantages during policy updates, enhancing training efficiency and addressing challenges related to credit assignment in complex reasoning tasks. Particularly, this method reduces development overhead and mitigates issues like reward hacking and overoptimization by avoiding explicit process-level annotations and facilitating online updates of the PRM.

Preference Optimization. Although the training method of DeepSeek-R1 significantly enhances the model's reasoning ability, it requires substantial computational resources for online RL training. In contrast to online approaches like PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024), Direct Preference Optimization (DPO) demands much less computational resources (Rafailov et al., 2023). By simply constructing chosen and rejected pairs, models can be trained directly, making DPO a more efficient alternative to PPO and GRPO.

Several works adopt DPO to improve the reasoning performance of language models. EXAONE Deep (LG-Research et al., 2025) presents a series of reasoning language models which utilize 1.6M

data samples for supervised fine-tuning (SFT), 20K instances for preference optimization with the method from (Xiao et al., 2025), and 10K for online RL training. As a result, EXAONE Deep 2.4B and 7.8B outperform models in comparable sizes, and EXAONE Deep 32B shows competitive performance against leading models. However, the training data is not publicly open-sourced, and there is no detail on how performance can be improved after preference optimization. Light-R1 (Wen et al., 2025a) employs a method of curriculum post-training with three training stages–two SFT stages and a DPO stage at the last. The first SFT stage uses 76k training samples, and the second one uses 3k highly difficult samples. To construct the preference pairs for the last DPO stage, it uses the rollouts from DeepSeek-R1 with verified correct answers as chosen samples, and the rollouts from the checkpoint after the second stage with verified incorrect answers as the corresponding reject samples. Iterative DPO (Tu et al., 2025) finds that DPO can rapidly improve the model's reasoning capabilities by using various methods of constructing chosen and rejected pairs. Through multiple rounds of DPO training, they show that this method can rival the performance of online RL approaches, such as Simple-RL-Zero (Zeng et al., 2025). RedStar (Xu et al., 2025a) also studies DPO training to enhance the model's reasoning capabilities. It constructs positive and negative samples using a rule-based reward: a reward of 1 is assigned if verification succeeds, and 0 otherwise. The study compares DPO with PPO and REINFORCE++ (Hu, 2025) and shows that DPO training is more effective than PPO in improving the model's reasoning abilities. DPO-R1 (Zhang et al., 2025a) explores the feasibility of DPO and RAFT (Dong et al., 2023), indicating that DPO substantially enhances model performance while maintaining high training efficiency, and incorporating a SFT warm-up phase before DPO further boosts performance. Nonetheless, DPO still lags slightly behind PPO in overall effectiveness in their experiments.

4.2 GENERALIZABILITY

Achieving robust generalization is often considered as a critical challenge in the deep learning era. However, current studies of reasoning language models in pre-training, supervised fine-tuning, and reinforcement learning have demonstrated that these models are well generalized to handle out-of-distribution tasks when learning to improve their reasoning ability.

Continual Pre-training. Continual pre-training on mathematical reasoning tasks has been shown to substantially enhance both specialized and general reasoning abilities in language models. GRPO (Shao et al., 2024) studied the continual pre-training in mathematical reasoning and demonstrated that models like DeepSeekMath-Base 7B exhibit enhanced performance not only in mathematical problem-solving but also on benchmarks such as the Massive Multitask Language Understanding (MMLU, Xuan et al. (2025)) and Big-Bench Hard (BBH, Suzgun et al. (2022)). For instance, DeepSeekMath-Base 7B achieved a 54.9% score on MMLU and 59.5% on BBH, surpassing its precursor, DeepSeek-Coder-Base-v1.5, which scored 49.1% and 55.2% respectively. This suggests that mathematical training can positively influence a model's general reasoning capabilities.

Supervised Fine-tuning. REFT (Trung et al., 2024) and Light-R1 (Wen et al., 2025a) validate that supervised fine-tuning (SFT) plays a critical role in enhancing the generalization of language models by providing structured, high-quality reasoning examples that serve as strong inductive priors. It bootstraps initial reasoning capabilities, enabling models to internalize latent abstractions and problem-solving strategies that transfer across tasks. By exposing the model to diverse solution paths, SFT establishes a stable base policy for further reinforcement learning that significantly improves the efficiency and effectiveness of subsequent reward-based optimization, reducing reward hacking and guiding exploration toward more reliable, outcome-driven reasoning behaviors. The power of SFT on generalization is further emphasized by LIMO (Ye et al., 2025), which demonstrates that carefully curated, high-quality training examples play a pivotal role in enabling broader generalization. The study highlights the importance of strategic data selection in fostering versatile reasoning capabilities, as well as robustness to out-of-distribution (OOD) variations, such as Chinese math problems that were not present in the training set.

Reinforcement Learning. Current outcome-reward-based reinforcement learning (RL) for reasoning language models has demonstrated strong potential for out-of-domain generalization. Through an RL process, reasoning language models demonstrate strong generalization capability across tasks, languages, and modalities, being far beyond what is possible with imitative learning

alone. Llama3-SWE-RL (Huang et al., 2025a) demonstrates improved results on five out-of-domain tasks, including function coding, library use, code reasoning, mathematics, and general language understanding, despite being trained solely on the code repair task. In contrast, a supervised fine-tuning baseline led to an average performance degradation. RL-Poet (Doria, 2025) demonstrates the ability to generate literary poems in multiple languages with correct poetic rules, despite being trained almost exclusively on English data. Compared to the prevailing imitative learning paradigm, these results highlight the potential of achieving artificial general intelligence through general reinforcement learning. In other respects, Tang et al. (2025) introduces Any-Generation Reward Optimization (AGRO) that enhances the generalizability of reasoning language models by integrating learning from both on-policy and off-policy experiences. Specifically, AGRO leverages both current (on-policy) data, collected from the model's existing policy, and historical (off-policy) data, experiences gathered from previous policies, to enable models to learn from a broader spectrum of scenarios. This comprehensive exposure mitigates the risk of overfitting and enhances the model's adaptability to novel situations. Consequently, AGRO-trained models are better equipped to generalize across diverse tasks and environments, a critical attribute for deploying reasoning language models in real-world applications where variability is inherent.

In comparing the roles the supervised fine-tuning (SFT) and outcome-reward-based reinforcement learning (RL) play in the context of generalization, Chu et al. (2025a) demonstrates that RL significantly enhances a model's ability to generalize across both textual and visual domains. In contrast, SFT often encourages memorization of the training data, which can impair performance on out-of-distribution tasks. Interestingly, while RL drives generalization, SFT remains crucial for stabilizing the model's output format—an essential property that facilitates effective downstream RL optimization, highlighting the complementary nature of SFT and RL in shaping models that can acquire and transfer knowledge across diverse, multimodal tasks. However, recent studies have raised concerns regarding the limitations of RL when applied to reasoning language models. Yue et al. (2025a) points out that RL training in reasoning language models may narrow the scope of reasoning capabilities while enhancing sampling efficiency, and that RL-trained models generally underperform compared to base models in pass@k metrics at larger k values. Similarly, Hochlehnert et al. (2025a) observes that the generalization ability of RL methods on smaller language models is significantly limited, possibly due to the restricted prior knowledge available for RL training to exploit.

In summary, these findings underscore both the promise and challenges of applying RL to reasoning and generalization in reasoning language models. While general RL approaches demonstrate encouraging out-of-domain performance gains and broader adaptability, careful attention must be given to potential trade-offs.

4.3 Safety

Ensuring the safety and robustness of large language models (LLMs) against vulnerabilities and attacks is a critical research area that has been widely explored in previous literature (Kaddour et al., 2023; Zhao et al., 2023; Das et al., 2025). However, reasoning language models introduce new safety challenges arising from their training algorithms, adversarial attacks during inference, and vulnerabilities related to their deployment environments (Zhou et al., 2025; Jiang et al., 2025). In this section, we review recent advancements addressing these emerging concerns and highlight promising approaches for enhancing detection and defense mechanisms.

Self-Evolution and Reward Hacking. Reasoning language models have demonstrated significant potential, paving the way toward superintelligent models capable of continuous self-improvement (Leike & Sutskever, 2023; Li et al., 2024a; Tao et al., 2024). However, their self-evolution process introduces safety concerns and risks producing uncontrollable outcomes misaligned with human values and preferences (Taubenfeld et al., 2024). A promising direction for improving these models involves using reinforcement learning algorithms with reward signals (Yu et al., 2025; Guo et al., 2025; Jaech et al., 2024). However, this approach inevitably introduces the issue of reward hacking, a longstanding challenge within the reinforcement learning research community (Amodei et al., 2016; Everitt et al., 2017; 2021; Weng, 2024). Reward hacking occurs when a model exploits flaws or ambiguities in the reward function, primarily because the reinforcement learning environment is imperfect and struggles to provide complete and accurate reward signals.

Jailbreaking on Reasoning Language Models. Jailbreak attacks and defenses play crucial roles in maintaining the robustness and security of large language models (LLMs) (Zhou et al., 2024; Yi et al., 2024). The same philosophy applies to reasoning language models (Kuo et al., 2025). Recent work by Sabbaghi et al. (2025) introduces an adversarial reasoning method for constructing effective jailbreak trajectories, achieving an attack success rate of 56% on OpenAI-o1 (Jaech et al., 2024) and 100% on Deepseek-R1 (Guo et al., 2025). Yao et al. (2025) discusses inherent flaws in reasoning language models and proposes a novel jailbreaking attack targeting reasoning language models. Arrieta et al. (2025) also states that Deepseek-R1 produces more unsafe responses than OpenAI models. These results highlight the importance of employing safety-focused supervised fine-tuning and reinforcement learning to safeguard reasoning language models against adversarial attacks. However, previous studies indicate that incorporating safety alignment can inadvertently compromise the reasoning capabilities of these models (Huang et al., 2025b). Moreover, Zhao et al. (2025b) and (Jiang et al., 2025) observes substantial decreases in both helpfulness and harmlessness in reasoning language models compared to baseline models.

Overthinking. Reasoning language models allow for extended reasoning chains during inference, but this capability can sometimes cause issues like overthinking (Sui et al., 2025; Chen et al., 2024a). Commercial model services typically charge more for output tokens, including reasoning tokens, than input tokens. Thus, attacks like OverThink (Kumar et al., 2025) can trigger excessive reasoning, raising operational and environmental costs. Additionally, overthinking suggests that the model is heavily reliant on internal simulations of potential actions and outcomes. Consequently, studies from Cuadron et al. (2025) and Feng et al. (2025a) have emphasized that reasoning language models may exhibit reduced performance in agentic scenarios when environmental feedback is neglected.

Effective safety measures for reasoning language models typically combine prevention, detection, and mitigation strategies. First, the methodologies to mitigate reward hacking include better algorithm design (Amodei et al., 2016; Uesato et al., 2020; Pan et al., 2022) and training strategies Denison et al. (2024); Li et al. (2025c). Guan et al. (2024) introduces reasoning alignment over safety policies to enhance a model's robustness against jailbreak attacks. Jiang et al. (2025) introduces decoding strategies aimed at improving the safety of reasoning language models, with some performance trade-offs, and provides post-training datasets for better alignment. Additionally, several studies have equipped safeguard models with reasoning capabilities to more effectively detect potential threats (Liu et al., 2025a; Wen et al., 2025b).

4.4 MULTIMODAL AND MULTILINGUAL

Multimodal reasoning language models are primarily developed via two predominant approaches: post-alignment (Zhang et al., 2023; Chu et al., 2024; Chen et al., 2024b; Grattafiori et al., 2024) and mixed-modality pretraining (Team et al., 2023; 2024; Nguyen et al., 2025). However, both approaches generally yield weaker reasoning capabilities compared to single-modality models (Wu et al., 2023; Liang et al., 2024; Wang et al., 2024b). Recent studies have sought to improve test-time scaling for multimodal reasoning language models across various modalities, including visual (Liu et al., 2024; Wang et al., 2025b; Du et al., 2025; Sun et al., 2025), audio (Du et al., 2024; Ma et al., 2025; Xie et al., 2025b; Li et al., 2025a), and others, such as 3D data, tabular information, and sensor inputs (Wang et al., 2024c; Dai et al., 2025). Furthermore, research by Du et al. (2025) demonstrates that reasoning capabilities developed in single-modality reasoning language models can effectively transfer to multimodal contexts. However, applying advanced RL and PRM to multimodal large reasoning language models remains a challenging yet promising research direction (Wu et al., 2025).

The challenges associated with multilingual reasoning language models differ primarily due to the limited availability of resources in certain languages, resulting in weaker performance from the base model (Nguyen et al., 2023; Qin et al., 2024). Research in this area remains limited, with two central issues emerging: (1) evaluating the extent to which reasoning abilities trained predominantly in English can generalize effectively to other languages, and (2) determining whether multilingual contexts necessitate specialized model capabilities to effectively facilitate insight or trigger "aha" moments. Xuan et al. (2025) observes that reasoning-enhanced models do not uniformly improve multilingual capabilities, emphasizing the importance of targeted multilingual reasoning enhancements in reasoning language models. Researchers have also proposed multilingual SFT and RL algorithms to enhance consistency across different languages (Lai & Nissim, 2024; Chai et al., 2024;

Wang et al., 2025a). We anticipate that future research will focus on more efficient training strategies for multilingual reasoning language models, with particular emphasis on improving performance in low-resource languages.

5 CONCLUSIONS

In this survey, we present a comprehensive overview of the replication efforts inspired by DeepSeek-R1, with a particular emphasis on the methodologies and insights underpinning supervised fine-tuning and reinforcement learning approaches. We explore how open-source projects have curated instruction-tuning datasets, implemented outcome-reward-based reinforcement learning strategies, and designed reward systems aimed at enhancing models' reasoning capabilities. Beyond synthesizing trends from current initiatives, we also offer our perspective on promising future directions for the field. These include the expansion of reasoning skills beyond mathematical and coding tasks, the advancement of model safety and interpretability, and the refinement of reward mechanisms to foster more sophisticated reasoning behaviors. We hope this survey not only captures the recent progress but also provides a solid foundation for ongoing research and marks a step forward toward achieving artificial general intelligence.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662. URL `https://aclanthology.org/2024.acl-long.662/`.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models, 2025. URL `https://arxiv.org/abs/2502.17387`.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

AoPS. Aops wiki:competition ratings. `https://artofproblemsolving.com/wiki/index.php/AoPS_Wiki:Competition_ratings`, 2025. Accessed: May 1, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently, 2025. URL `https://arxiv.org/abs/2502.04463`.

Aitor Arrieta, Miriam Ugarte, Pablo Valle, José Antonio Parejo, and Sergio Segura. o3-mini vs deepseek-r1: Which one is safer? *arXiv preprint arXiv:2501.18438*, 2025.

Bespoke-Labs. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation. https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation, 2025. Accessed: 2025-01-22.

ByteDance-Seed. Seed-thinking-v1.5: Advancing superb reasoning models with reinforcement learning. `https://github.com/ByteDance-Seed/Seed-Thinking-v1.5`, 2025. Accessed: 2025-04-10.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xiannian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, et al. xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning. *arXiv preprint arXiv:2401.07037*, 2024.

Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, Chaoqun Liu, Lidong Bing, Deli Zhao, Anh Tuan Luu, and Yu Rong. Finereason: Evaluating and improving llms' deliberate reasoning through reflective puzzle solving. *arXiv preprint arXiv:2502.20238*, 2025.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023. URL https://arxiv.org/abs/2306.15595.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024a.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024b.

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. *arXiv preprint arXiv:2501.17161*, 2025a.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025b.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Yue Dai, Soyeon Caren Han, and Wei Liu. Multimodal graph constrastive learning and prompt for chartqa. *arXiv preprint arXiv:2501.04303*, 2025.

Hrishbh Dalal. Ai as algorithm designer: Teaching llms to improve sorting through trial and error in grpo. *Personal Website*, March 2025a. URL https://hrishbh.com/ai-as-algorithm-designer-teaching-llms-to-improve-sorting-through-trial-and-error-in-grpo/.

Hrishbh Dalal. Teaching language models to invent or optimize efficient sudoku algorithms through reinforcement learning, 3 2025b. URL https://hrishbh.com/teaching-language-models-to-invent-or-optimize-efficient-sudoku-algorithms-through-reinforcement-learning/.

Hrishbh Dalal. Teaching language models to solve sudoku through reinforcement learning, 3 2025c. URL https://hrishbhdalal.com/projects/teaching-language-models-sudoku. Accessed on March 19, 2025.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, et al. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*, 2024.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Alexander Doria. Rl, reasoning & writing - grpo on base model. `https://colab.research.google.com/drive/1Ty0ovsrpw8i-zJvDhlSAtBIVw3EZfHK5`, 2025.

Yexing Du, Ziyang Ma, Yifan Yang, Keqi Deng, Xie Chen, Bo Yang, Yang Xiang, Ming Liu, and Bing Qin. Cot-st: Enhancing llm-based speech translation with multimodal chain-of-thought. *arXiv preprint arXiv:2409.19510*, 2024.

Yifan Du, Zikang Liu, Yifan Li, Wayne Xin Zhao, Yuqi Huo, Bingning Wang, Weipeng Chen, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Virgo: A preliminary exploration on reproducing o1-like mllm. *arXiv preprint arXiv:2501.01904*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*, 2017.

Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*, 198 (Suppl 27):6435–6467, 2021.

Xiachong Feng, Longxu Dou, and Lingpeng Kong. Reasoning does not necessarily improve role-playing ability. *arXiv preprint arXiv:2502.16940*, 2025a.

Zhaopeng Feng, Shaosheng Cao, Jiahan Ren, Jiayuan Su, Ruizhe Chen, Yan Zhang, Zhe Xu, Yao Hu, Jian Wu, and Zuozhu Liu. Mt-r1-zero: Advancing llm-based machine translation via r1-zero-like reinforcement learning, 2025b. URL `https://arxiv.org/abs/2504.10160`.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024. URL `https://arxiv.org/abs/2410.07985`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Melody Y Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, et al. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*, 2024.

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. *arXiv preprint arXiv:2501.04519*, 2025.

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.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reaonser series. `https://capricious-hydrogen-41c.notion.site/Skywork-Open-Reaonser-Series-1d0bc9ae823a80459b46c149e4f51680`, 2025a. Notion Blog.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning, 2025b. URL `https://arxiv.org/abs/2504.11456`.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL `https://arxiv.org/abs/2103.03874`.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025a.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Ameya Prabhu, and Matthias Bethge. Curatedthoughts: Data curation for rl training datasets, 2025b. URL `https://huggingface.co/datasets/bethgelab/CuratedThoughts`.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, and Heung-Yeung Shum Xiangyu Zhang. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. `https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero`, 2025.

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Julia Hockenmaier, and Tong Zhang. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *arXiv preprint arXiv:2503.12759*, 2025a.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, Zachary Yahn, Yichang Xu, and Ling Liu. Safety tax: Safety alignment makes your large reasoning models less reasonable. *arXiv preprint arXiv:2503.00555*, 2025b.

HuggingFace. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

Intelligent Internet. Ii-thought : A large-scale, high-quality reasoning dataset. `https://ii.inc/web/blog/post/ii-thought`, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Piyush Jha, Prithwish Jana, Pranavkrishna Suresh, Arnav Arora, and Vijay Ganesh. Rlsf: Reinforcement learning via symbolic feedback. *arXiv preprint arXiv:2405.16661*, 2024.

Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. Safechain: Safety of language models with long chain-of-thought reasoning capabilities. *arXiv preprint arXiv:2502.12025*, 2025.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

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, 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, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. Overthink: Slowdown attacks on reasoning llms. *arXiv e-prints*, pp. arXiv–2502, 2025.

Martin Kuo, Jianyi Zhang, Aolin Ding, Qinsi Wang, Louis DiValentin, Yujia Bao, Wei Wei, Da-Cheng Juan, Hai Li, and Yiran Chen. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking. *arXiv preprint arXiv:2502.12893*, 2025.

Hynek Kydlíček. Math-verify: Math verification library, 2024. URL https://github.com/huggingface/math-verify. If you use this software, please cite it using the metadata from this file.

Huiyuan Lai and Malvina Nissim. mcot: Multilingual instruction tuning for reasoning consistency in language models. *arXiv preprint arXiv:2406.02301*, 2024.

Jan Leike and Ilya Sutskever. Introducing superalignment. https://openai.com/blog/introducing-superalignment, 2023. Accessed: 2024-04-01.

LG-Research, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Kijeong Jeon, et al. Exaone deep: Reasoning enhanced language models. *arXiv preprint arXiv:2503.12524*, 2025.

Gang Li, Jizhong Liu, Heinrich Dinkel, Yadong Niu, Junbo Zhang, and Jian Luan. Reinforcement learning outperforms supervised fine-tuning: A case study on audio question answering. *arXiv preprint arXiv:2503.11197*, 2025a.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*, 2023.

Siheng Li, Cheng Yang, Zesen Cheng, Lemao Liu, Mo Yu, Yujiu Yang, and Wai Lam. Large language models can self-improve in long-context reasoning. *arXiv preprint arXiv:2411.08147*, 2024a.

Xingxuan Li, Weiwen Xu, Ruochen Zhao, Fangkai Jiao, Shafiq Joty, and Lidong Bing. Can we further elicit reasoning in llms? critic-guided planning with retrieval-augmentation for solving challenging tasks. *arXiv preprint arXiv:2410.01428*, 2024b.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025b.

Xuying Li, Zhuo Li, Yuji Kosuga, and Victor Bian. Output length effect on deepseek-r1's safety in forced thinking. *arXiv preprint arXiv:2503.01923*, 2025c.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025d.

Zijing Liang, Yanjie Xu, Yifan Hong, Penghui Shang, Qi Wang, Qiang Fu, and Ke Liu. A survey of multimodel large language models. In *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, pp. 405–409, 2024.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. Zebralogic: On the scaling limits of llms for logical reasoning, 2025a. URL https://arxiv.org/abs/2502.01100.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models, 2025b.

Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. https://github.com/ganler/code-r1, 2025.

Wei Liu, Junlong Li, Xiwen Zhang, Fan Zhou, Yu Cheng, and Junxian He. Diving into self-evolving training for multimodal reasoning. *arXiv preprint arXiv:2412.17451*, 2024.

Yue Liu, Hongcheng Gao, Shengfang Zhai, Jun Xia, Tianyi Wu, Zhiwei Xue, Yulin Chen, Kenji Kawaguchi, Jiaheng Zhang, and Bryan Hooi. Guardreasoner: Towards reasoning-based llm safeguards. *arXiv preprint arXiv:2501.18492*, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. There may not be aha moment in r1-zero-like training — a pilot study. https://oatllm.notion.site/oat-zero, 2025b. Notion Blog.

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*, 2025c.

Michael Luo, Sijun Tan, Roy Huang, Xiaoxiang Shi, Rachel Xin, Colin Cai, Ameen Patel, Alpay Ariyak, Qingyang Wu, Ce Zhang, Li Erran Li, and Ion Stoica Raluca Ada Popa. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025a. Notion Blog.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025b. Notion Blog.

Ziyang Ma, Zhuo Chen, Yuping Wang, Eng Siong Chng, and Xie Chen. Audio-cot: Exploring chain-of-thought reasoning in large audio language model. *arXiv preprint arXiv:2501.07246*, 2025.

Justus Mattern, Sami Jaghouar, Manveer Basra, Jannik Straube, Matthew Di Ferrante, Felix Gabriel, Jack Min Ong, Vincent Weisser, and Johannes Hagemann. Synthetic-1: Two million collaboratively generated reasoning traces from deepseek-r1, 2025. URL https://www.primeintellect.ai/blog/synthetic-1-release.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*, 2023.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, et al. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52, 2025.

OpenAI. Gpt-4o system card, August 2024. URL `https://openai.com/index/gpt-4o-system-card/`. Accessed: 2025-05-01.

OpenAI. Introducing openai o3 and o4-mini, 2025. URL `https://openai.com/index/introducing-o3-and-o4-mini/`.

OpenThoughts-Team. Open Thoughts. `https://open-thoughts.ai`, January 2025.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022. URL `https://api.semanticscholar.org/CorpusID:246426909`.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. `https://github.com/Jiayi-Pan/TinyZero`, 2025. Accessed: 2025-01-24.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. Codeforces. `https://huggingface.co/datasets/open-r1/codeforces`, 2025.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*, 2024.

Qwen-Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL `https://qwenlm.github.io/blog/qwq-32b-preview/`.

Qwen-Team. Qwen3: Think deeper, act faster, 2025a. URL `https://qwenlm.github.io/blog/qwen3/`.

Qwen-Team. Qwq-32b: Embracing the power of reinforcement learning, 2025b. URL `https://qwenlm.github.io/blog/qwq-32b/`.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective. *arXiv preprint arXiv:2503.15477*, 2025.

RDKit. Rdkit: Open-source cheminformatics software, 2025. URL `https://www.rdkit.org`.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=Ti67584b98`.

Mahdi Sabbaghi, Paul Kassianik, George Pappas, Yaron Singer, Amin Karbasi, and Hamed Hassani. Adversarial reasoning at jailbreaking time. *arXiv preprint arXiv:2502.01633*, 2025.

Philipp Schmid. Mini-r1: Reproduce deepseek r1 „aha moment" a rl tutorial. `https://huggingface.co/blog/open-r1/mini-r1-contdown-game`, 2025.

J. Schulman. Approximating kl divergence, 2020. URL `http://joschu.net/blog/kl-approx.html`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

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.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*, 2025.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains, 2025. URL https://arxiv.org/abs/2503.23829.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen, Zhong, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models, 2025. URL https://arxiv.org/abs/2503.16419.

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. Mm-verify: Enhancing multimodal reasoning with chain-of-thought verification. *arXiv preprint arXiv:2502.13383*, 2025.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Yunhao Tang, Taco Cohen, David W Zhang, Michal Valko, and Rémi Munos. Rl-finetuning llms from on-and off-policy data with a single algorithm. *arXiv preprint arXiv:2503.19612*, 2025.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

Amir Taubenfeld, Yaniv Dover, Roi Reichart, and Ariel Goldstein. Systematic biases in llm simulations of debates. *arXiv preprint arXiv:2402.04049*, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7601–7614, 2024.

Songjun Tu, Jiahao Lin, Xiangyu Tian, Qichao Zhang, Linjing Li, Yuqian Fu, Nan Xu, Wei He, Xiangyuan Lan, Dongmei Jiang, et al. Enhancing llm reasoning with iterative dpo: A comprehensive empirical investigation. *arXiv preprint arXiv:2503.12854*, 2025.

Jonathan Uesato, Ramana Kumar, Victoria Krakovna, Tom Everitt, Richard Ngo, and Shane Legg. Avoiding tampering incentives in deep rl via decoupled approval. *arXiv preprint arXiv:2011.08827*, 2020.

Shuhe Wang, Guoyin Wang, Yizhong Wang, Jiwei Li, Eduard Hovy, and Chen Guo. Packing analysis: Packing is more appropriate for large models or datasets in supervised fine-tuning, 2024a. URL `https://arxiv.org/abs/2410.08081`.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. Demystifying multilingual chain-of-thought in process reward modeling. *arXiv preprint arXiv:2502.12663*, 2025a.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, William Wang, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025b.

Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*, 2024b.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*, 2024c.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*, 2025a.

Xiaofei Wen, Wenxuan Zhou, Wenjie Jacky Mo, and Muhao Chen. Thinkguard: Deliberative slow thinking leads to cautious guardrails. *arXiv preprint arXiv:2502.13458*, 2025b.

Lilian Weng. Reward hacking in reinforcement learning. `https://lilianweng.github.io/posts/2024-11-28-reward-hacking/`, Nov 2024. Accessed: 2025-03-27.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL `https://doi.org/10.1007/BF00992696`.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 2247–2256. IEEE, 2023.

Jinyang Wu, Mingkuan Feng, Shuai Zhang, Ruihan Jin, Feihu Che, Zengqi Wen, and Jianhua Tao. Boosting multimodal reasoning with mcts-automated structured thinking. *arXiv preprint arXiv:2502.02339*, 2025.

Yunhui Xia, Wei Shen, Yan Wang, Jason Klein Liu, Huifeng Sun, Siyue Wu, Jian Hu, and Xiaolong Xu. Leetcodedataset: A temporal dataset for robust evaluation and efficient training of code llms, 2025. URL `https://arxiv.org/abs/2504.14655`.

Teng Xiao, Yige Yuan, Zhengyu Chen, Mingxiao Li, Shangsong Liang, Zhaochun Ren, and Vasant G Honavar. Simper: A minimalist approach to preference alignment without hyperparameters. *arXiv preprint arXiv:2502.00883*, 2025.

Xiaomi LLM-Core Team. Mimo: Unlocking the reasoning potential of language model – from pretraining to posttraining, 2025. URL `https://github.com/XiaomiMiMo/MiMo`.

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, 2025a. URL `https://arxiv.org/abs/2502.14768`.

Zhifei Xie, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. Audio-reasoner: Improving reasoning capability in large audio language models. *arXiv preprint arXiv:2503.02318*, 2025b.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Watch every step! llm agent learning via iterative step-level process refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1556–1572, 2024.

Haotian Xu, Xing Wu, Weinong Wang, Zhongzhi Li, Da Zheng, Boyuan Chen, Yi Hu, Shijia Kang, Jiaming Ji, Yingying Zhang, et al. Redstar: Does scaling long-cot data unlock better slow-reasoning systems? *arXiv preprint arXiv:2501.11284*, 2025a.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. Kodcode: A diverse, challenging, and verifiable synthetic dataset for coding, 2025b. URL `https://arxiv.org/abs/2503.02951`.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Yun Xing, Junjue Wang, Huitao Li, Xin Li, Kunyu Yu, et al. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. *arXiv preprint arXiv:2503.10497*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Yang Yao, Xuan Tong, Ruofan Wang, Yixu Wang, Lujundong Li, Liang Liu, Yan Teng, and Yingchun Wang. A mousetrap: Fooling large reasoning models for jailbreak with chain of iterative chaos. *arXiv preprint arXiv:2502.15806*, 2025.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL `https://arxiv.org/abs/2502.03387`.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*, 2024.

Dian Yu, Kai Sun, Dong Yu, and Claire Cardie. Self-teaching machines to read and comprehend with large-scale multi-subject question-answering data. *arXiv preprint arXiv:2102.01226*, 2021.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025a. URL `https://arxiv.org/abs/2504.13837`.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025b. URL `https://arxiv.org/abs/2504.05118`.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Hanning Zhang, Jiarui Yao, Chenlu Ye, Wei Xiong, and Tong Zhang. Online-dpo-r1: Unlocking effective reasoning without the ppo overhead. https://efficient-unicorn-451.notion.site/Online-DPO-R1-Unlocking-Effective-Reasoning-Without-the-PPO-Overhead-1908b9a70e7b80c3bc83f4cf04b2f175?pvs=4, 2025a. Notion Blog.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqi Kuang, Shouyu Yin, Chaohang Wen, Haotian Zhang, Bin Chen, and Bing Yu. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm, 2025b. URL https://arxiv.org/abs/2504.14286.

Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 1.4 million open-source distilled reasoning dataset to empower large language model training, 2025a. URL https://arxiv.org/abs/2503.19633.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Weixiang Zhao, Xingyu Sui, Jiahe Guo, Yulin Hu, Yang Deng, Yanyan Zhao, Bing Qin, Wanxiang Che, Tat-Seng Chua, and Ting Liu. Trade-offs in large reasoning models: An empirical analysis of deliberative and adaptive reasoning over foundational capabilities. *arXiv preprint arXiv:2503.17979*, 2025b.

Kaiwen Zhou, Chengzhi Liu, Xuandong Zhao, Shreedhar Jangam, Jayanth Srinivasa, Gaowen Liu, Dawn Song, and Xin Eric Wang. The hidden risks of large reasoning models: A safety assessment of r1. *arXiv preprint arXiv:2502.12659*, 2025.

Weikang Zhou, Xiao Wang, Limao Xiong, Han Xia, Yingshuang Gu, Mingxu Chai, Fukang Zhu, Caishuang Huang, Shihan Dou, Zhiheng Xi, et al. Easyjailbreak: A unified framework for jailbreaking large language models. *arXiv preprint arXiv:2403.12171*, 2024.