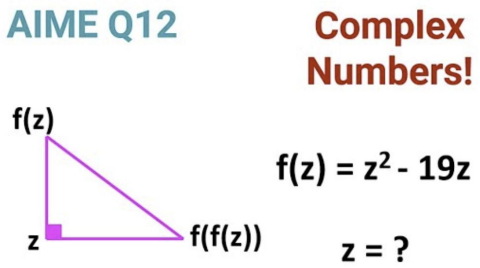
DeepSeek-R1

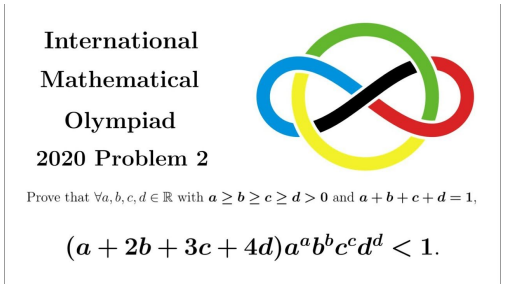
Reinforcement Learning with Verifiable Rewards

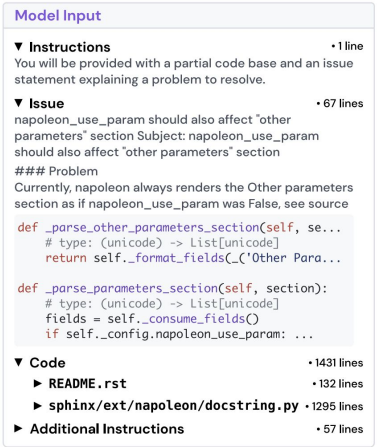
Swapnil Mallick, Howard Yin, Jenny Xu, Sayem Zaman

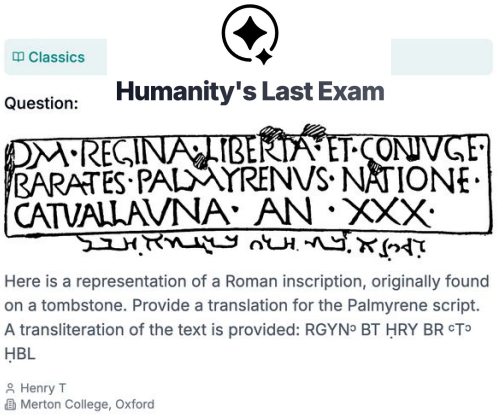
Reasoning is hard for AI

Reasoning is the ability to do ***complex cognitive tasks***









Source: Center for AI Safety et al., Nature (2026)

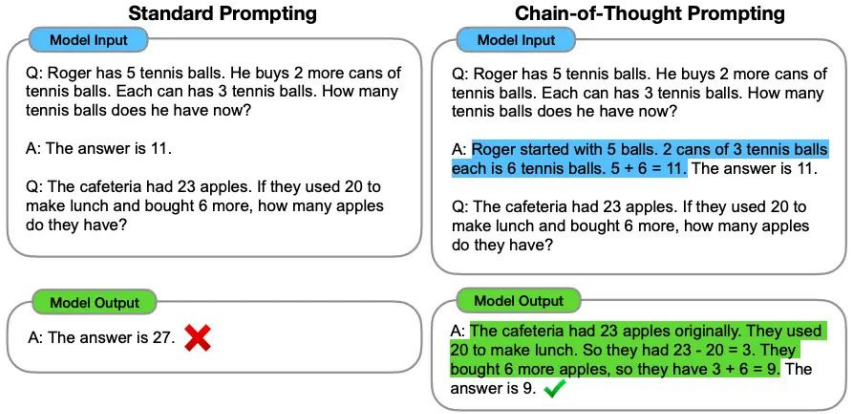
Why does Chain of Thought (CoT) improve reasoning?

CoT: Instead of using standard question and direct answer pairs you:

1. **Few-shot CoT:** Provide examples that demonstrate the reasoning process

2. **Zero-shot CoT:** Just append “Let’s think step by step”

LLMs are *auto-regressive*: Any token they generate become part of the context for the next token. Generating intermediate “steps”/tokens allows them to decompose problem, store intermediate results, etc etc (break problem into sub-problem)



(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let’s think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Source: Wei et al. (2022)

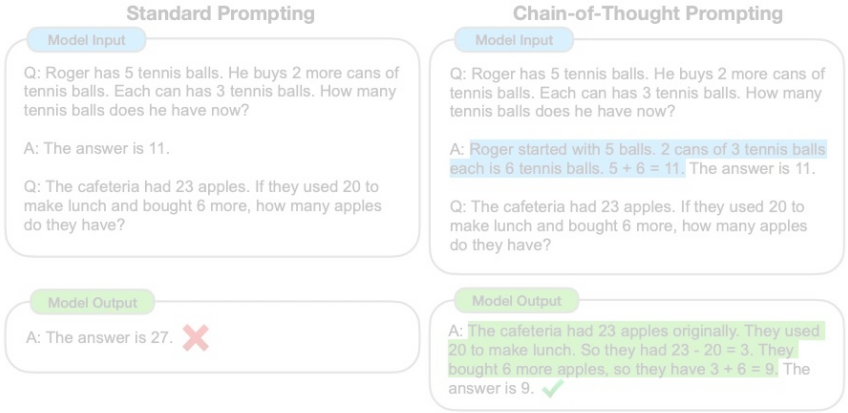
Why does Chain of Thought (CoT) improve reasoning?

CoT: Instead of using standard question and direct answer pairs you:

1. **Few-shot CoT:** Provide examples that demonstrate the reasoning process

2. **Zero-shot CoT:** Just append “Let’s think step by step”

LLMs are *auto-regressive*: Any token they generate become part of the context for the next token. Generating intermediate “steps”/tokens allows them to decompose problem, store intermediate results, etc etc (break problem into sub-problem)



(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let’s think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

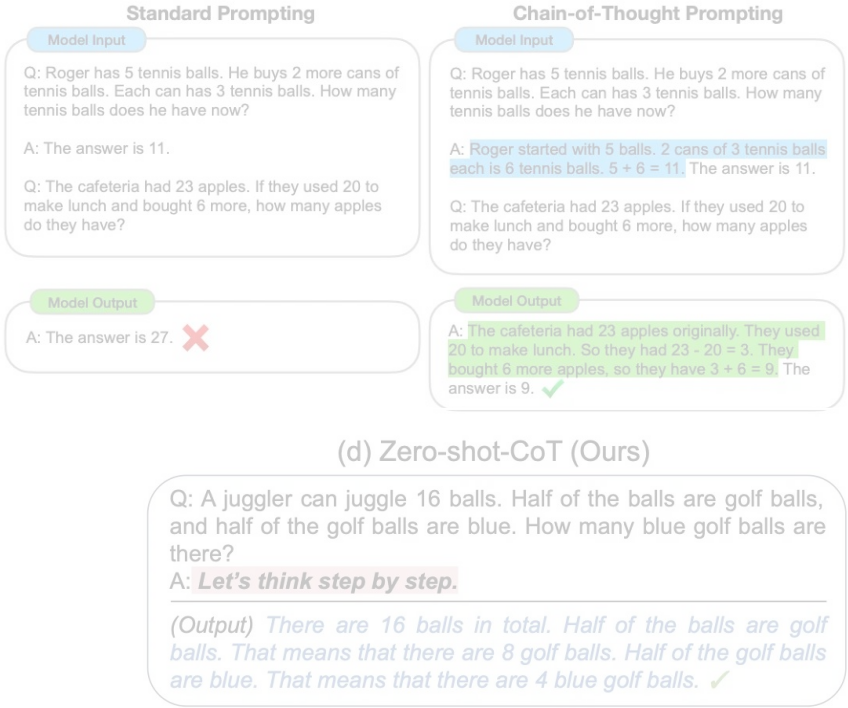
Source: Wei et al. (2022)

Why does Chain of Thought (CoT) improve reasoning?

CoT: Instead of using standard question and direct answer pairs you:

1. **Few-shot CoT:** Provide examples that demonstrate the reasoning process
2. **Zero-shot CoT:** Just append “Let’s think step by step”

LLMs are *auto-regressive*: Any token they generate become part of the context for the next token. Generating intermediate “steps”/tokens allows them to decompose problem, store intermediate results, etc etc (break problem into sub-problem)



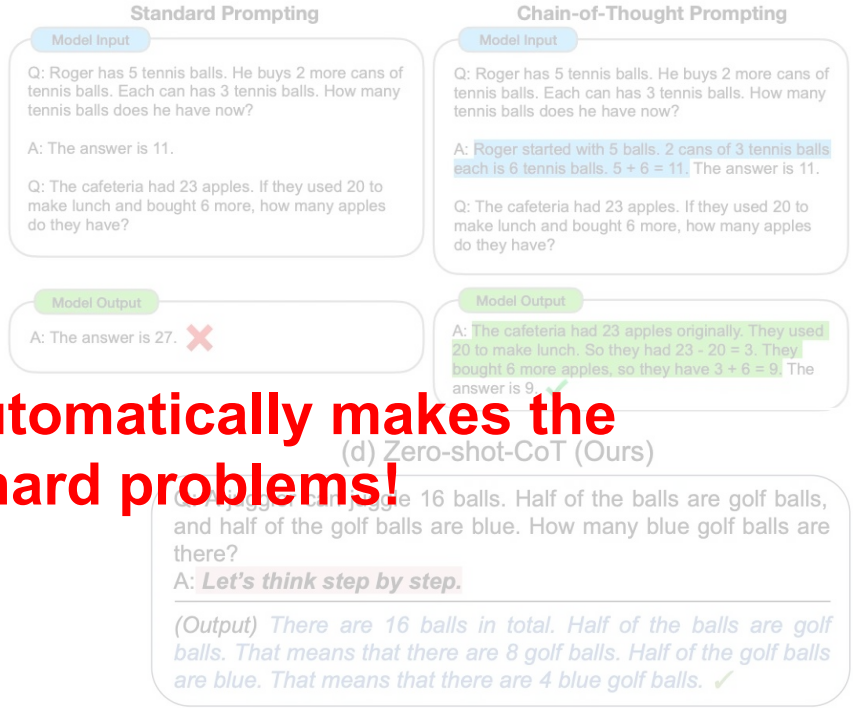
Source: Wei et al. (2022)

Why does Chain of Thought (CoT) improve reasoning?

CoT: Instead of using standard question and direct answer pairs you:

1. **Few-shot CoT:** Provide examples that demonstrate the reasoning process
2. **Zero-shot CoT:** Just append "Let's think step by step"

LLMs are *auto-regressive*: Any token they generate becomes part of the input for the next token. Generating intermediate "steps"/tokens allows them to decompose problem, store intermediate results, etc etc (break problem into sub-problem)



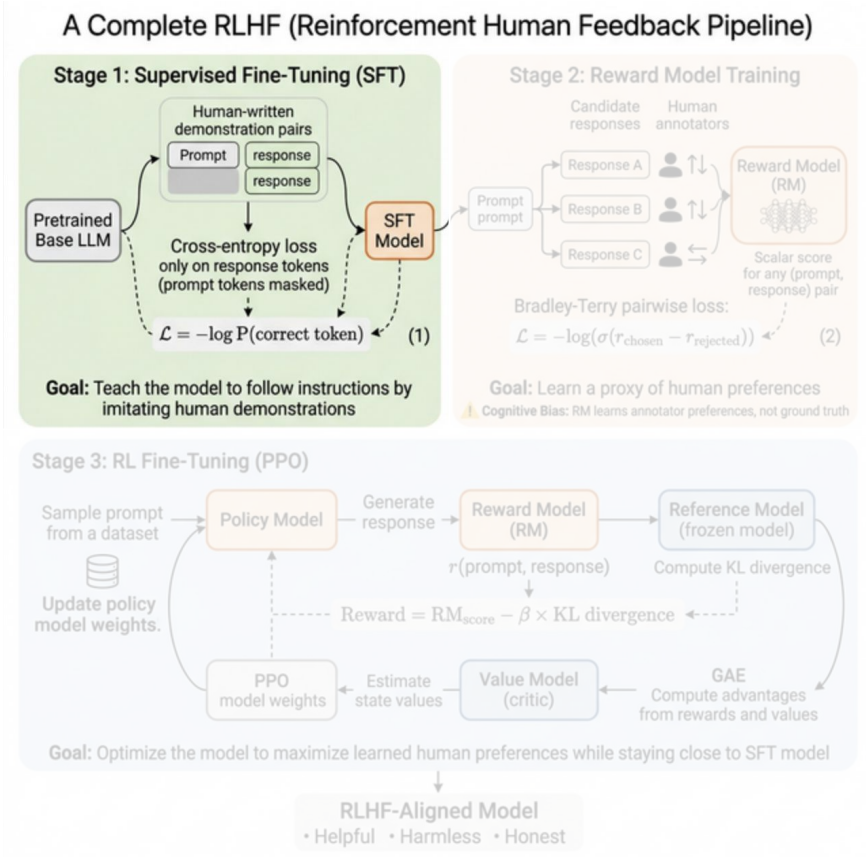
Introducing Verifiable Reward automatically makes the model use CoT for "perceived" hard problems!

Source: Wei et al. (2022)

Reinforcement Learning via Human Feedback (RLHF)

Stage 1: SFT Training Process to Imitate human behavior

1. Take a **(prompt, response)** pair from a dataset.
2. Feed the FULL sequence ino the model
3. Compute the loss ONLY on the response tokens
4. Back-propagate and update weights
5. Repeat for thousands of examples



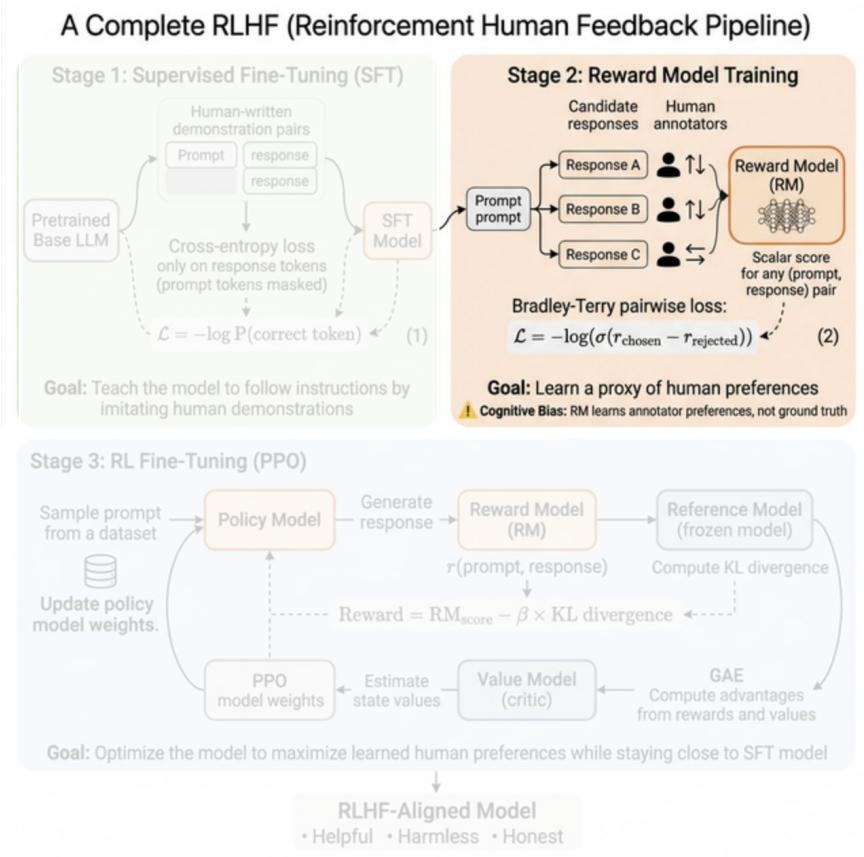
Source: Gemini Nano Banana Pro 2

Reinforcement Learning via Human Feedback (RLHF)

Stage 2: Reward Model Training

1. Take the SFT model and generate multiple candidate response for the same prompt
2. Train a separate Reward Model (RM) which is often another Neural Network to predict a scalar score that reflects those human preferences
3. Then train model with pairwise ranking loss
4. Reward Model assigns higher scores to response that humans preferred

This introduces cognitive bias!

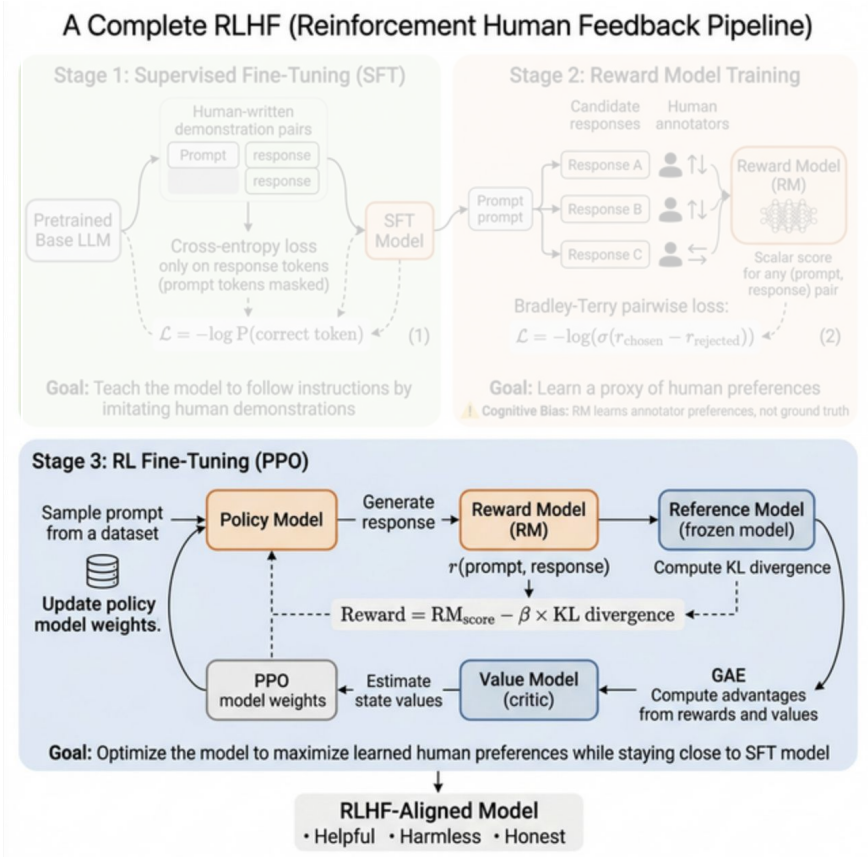


Source: Gemini Nano Banana Pro 2

Reinforcement Learning via Human Feedback (RLHF)

Stage 3: Use the learned Reward Model to fine-tune the SFT model using PPO

1. Sample a prompt from the dataset
2. The **policy model (LLM)** generates a response
3. The **Reward Model** scores the response
4. A KL penalty is added to prevent model from drifting too far from SFT Model (Reward Hacking)
5. PPO updates policy to maximize the reward



Source: Gemini Nano Banana Pro 2

Issues with the RLHF pipeline

1. Limitations of ***SFT***

    a. SFT can’t generalize preferences - if training data doesn’t cover a scenario, model doesn’t know what “good” means

    b. Treats all tokens equally - Loss doesn’t distinguish between critical tokens and filler tokens

    c. Can’t learn from comparisons - SFT only sees “good” examples, why is one better?

Issues with the (standard) RLHF pipeline

1. Limitations of ***SFT***
2. Reward Model (RM) implicitly learns ***cognitive bias***
    a. RM learns what the annotators *like.*
    b. If the annotators like confident-sounding answers even if those re wrong, then RM rewards those patterns.

Issues with the (standard) RLHF pipeline

1. Limitations of ***SFT***
2. Reward Model (RM) implicitly learns ***cognitive bias***
3. Issues with ***PPO***

Issues with the (standard) RLHF pipeline

1. Limitations of ***SFT***
2. Reward Model (RM) implicitly learns ***cognitive bias***
3. Issues with ***PPO***
4. RLHF constraints models to replicate human thought process

"*Furthermore, by constraining models to replicate human thought processes, their performance is inherently capped by the provided exemplars, which* ***prevents the exploration of superior, non-human-like reasoning pathways.***"

Reinforcement Learning with Verifiable Rewards

Instead of doing SFT → RLHF, at a high level, DeepSeek R1 does the reverse:

1. Rules based (reward-shaped) RL: Produces a model that reasons well but has messy outputs

2. Then SFT on the RL model's own *best outputs*

The key insight is: For reasoning tasks, **you don't need human preference** labels at all.

Use verifiable rewards: If the math is correct, reward = 1, if wrong, reward = 0.

Reinforcement Learning with Verifiable Rewards

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to
Response: <think> To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both $\cdots$ $\left(\sqrt{a - \sqrt{a+x}}\right)^2 = x^2 \implies a - \sqrt{a+x} = x^2.$ Rearrange to isolate the inner square root term: $(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$ $\ldots$ Wait, wait. Wait. That's an aha moment I can flag here. Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$ We started with the equation: $\sqrt{a - \sqrt{a+x}} = x$ First, let's square both sides: $a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$ Next, I could square both sides again, treating the equation: $\cdots$ $\ldots$

Table 3 | An interesting "aha moment" of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

Source: DeepSeek-R1 (2026)

Group Relative Policy Optimization (GRPO)

DeepSeek-R1-Zero represents a significant **milestone** in demonstrating that LLMs can autonomously develop **advanced reasoning capabilities** through **pure reinforcement learning** (RL), without relying on human-annotated supervised fine-tuning (SFT)

To facilitate **large-scale RL** efficiency, we adopt Group Relative Policy Optimization (**GRPO**) (Shao et al., 2024).

The **GPRO** Objective Function

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right)$$

The **Advantage** Formula

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}.$$

Unbiased Estimator of the **KL Divergence**

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1$$

Group Relative Policy Optimization (GRPO)

GRPO was originally proposed to **simplify the training process** and **reduce the resource consumption** of Proximal Policy Optimization (PPO), which is widely used in the RL stage of LLMs

GRPO foregoes the **value model**, instead estimating the **advantages** from **group** scores.

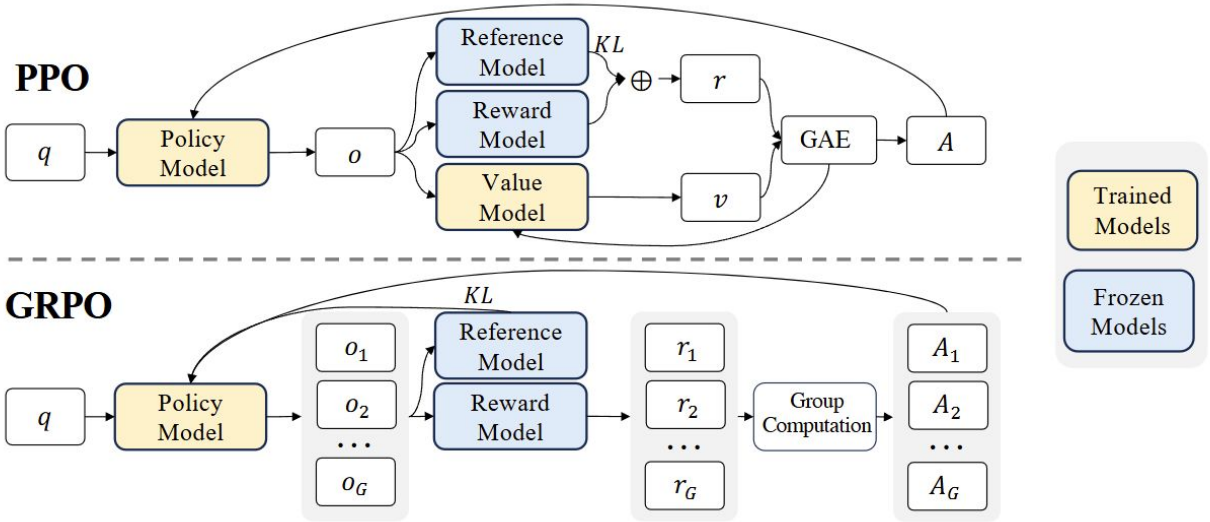


Figure 3. Demonstration of PPO and our GRPO.

Group Relative Policy Optimization (GRPO)

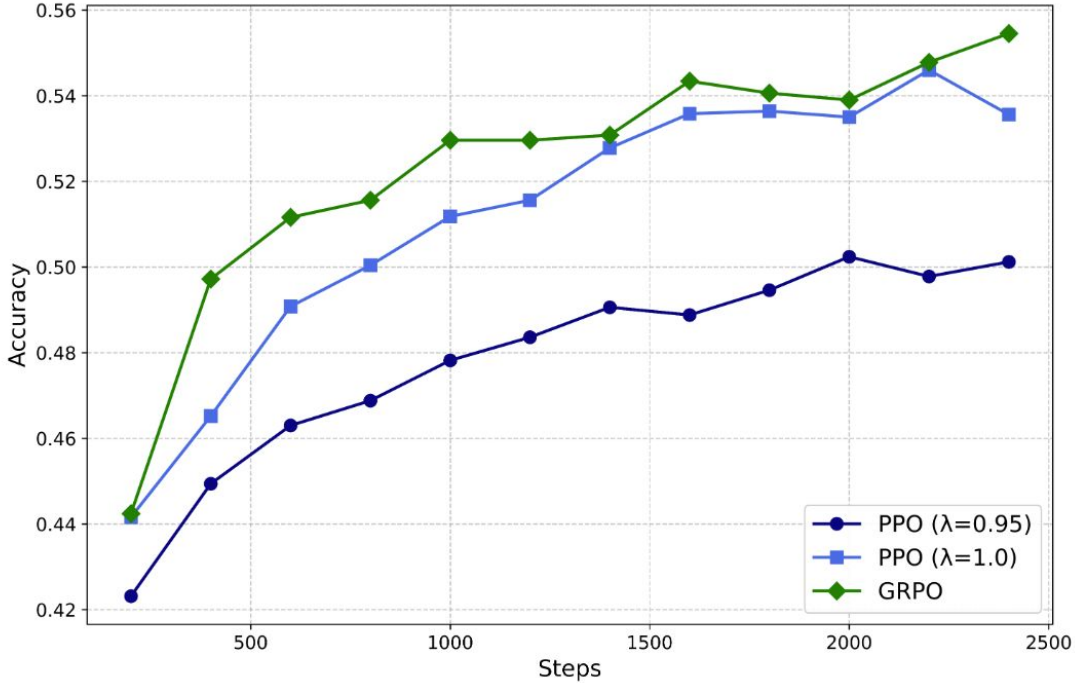


Figure 4. Performance of PPO and GRPO on the MATH task using DeepSeekCoder-V2-Lite

Considering the **memory** and **computational overhead** associated with training an additional **value model**, GRPO presents a more **practical alternative**, especially when training **large-scale models** with **constrained resources**

PPO requires additional **hyperparameter tuning**, particularly of the λ coefficient in **GAE** and is **highly sensitive** to this parameter

PPO demands **additional cost** for **hyperparameter optimizatio**

Reward Design for DeepSeek-R1-Zero

We apply the **RL** technique on the DeepSeek-V3 base to train DeepSeek-R1-Zero, with a straightforward template, to first produce a **reasoning process** followed by the final **answer**, during training.

For DeepSeek-R1-Zero, we employ **rule-based** rewards to deliver feedback for data in mathematical, coding, and logical reasoning domains.

- **Accuracy** rewards evaluate whether the response is **correct**.
- **Format** rewards complement the accuracy reward model by enforcing **specific formatting requirements**. This ensures that the model's thought process is **explicitly delineated**, enhancing **interpretability** and facilitating **subsequent analysis**.

$$Reward_{\text{rule}} = Reward_{\text{acc}} + Reward_{\text{format}}$$

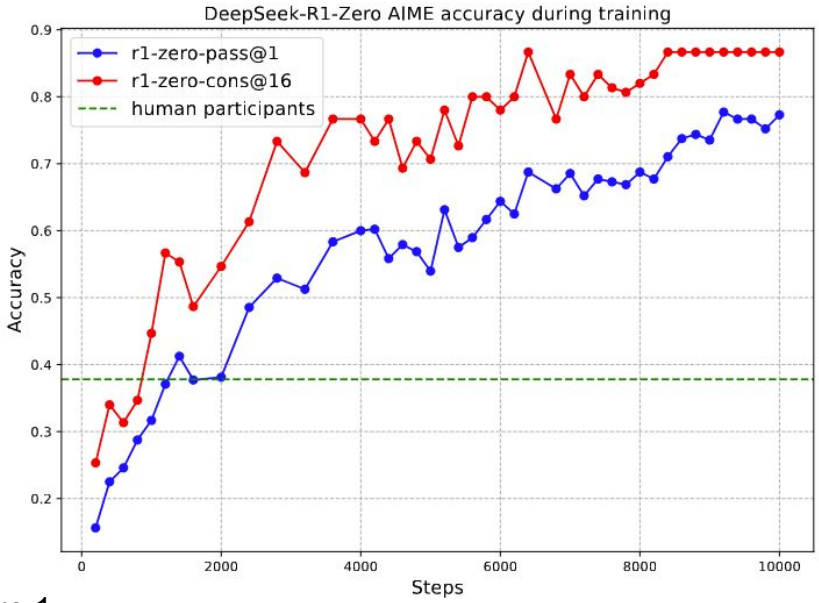
Neural reward models—whether outcome-based or process-based—to reasoning tasks are abstained

Neural reward models are susceptible to **reward hacking** during large-scale reinforcement learning

Necessitates substantial **computational resources**

Introduces **additional complexity** into the training pipeline, thereby **complicating** the overall **optimization process**

Self-evolution of DeepSeek-R1-Zero: Accuracy & Reasoning



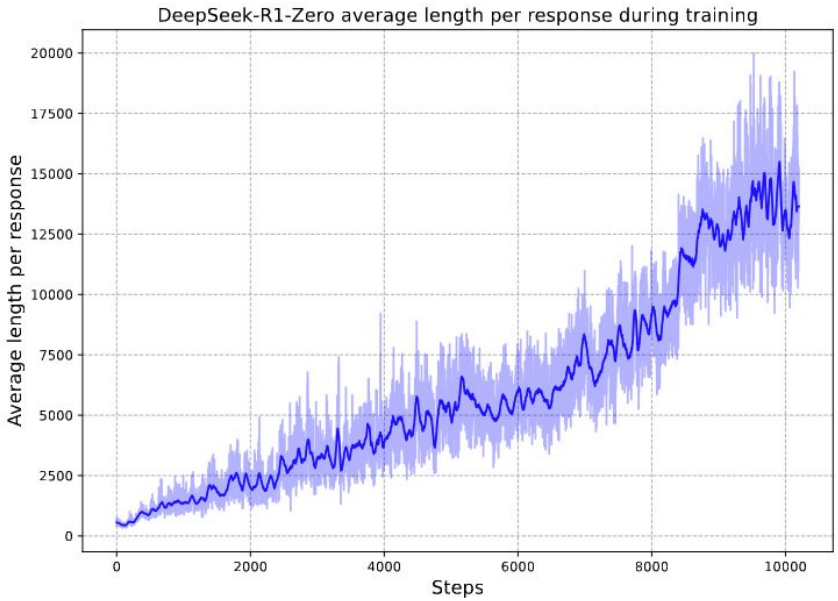
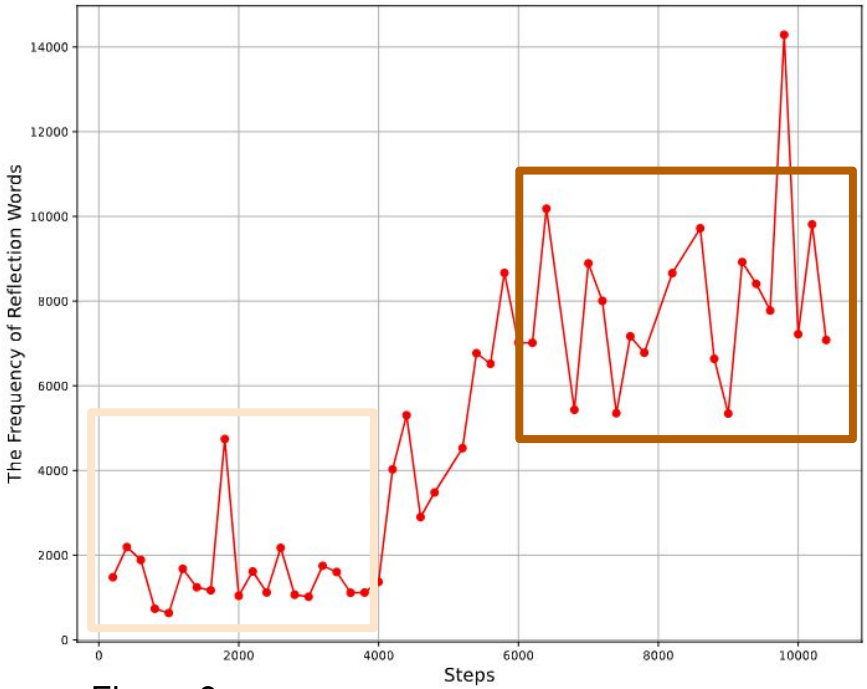


Figure 1
(a) AIME **accuracy** of DeepSeek-R1-Zero during training. The **baseline** is average score of human participants in the competition.
(b) The average response **length** of DeepSeek-R1-Zero on the training set during the **RL** process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

Note: a training step refers to a **single policy update operation**.

Evolution of DeepSeek-R1-Zero: Reasoning Behaviors



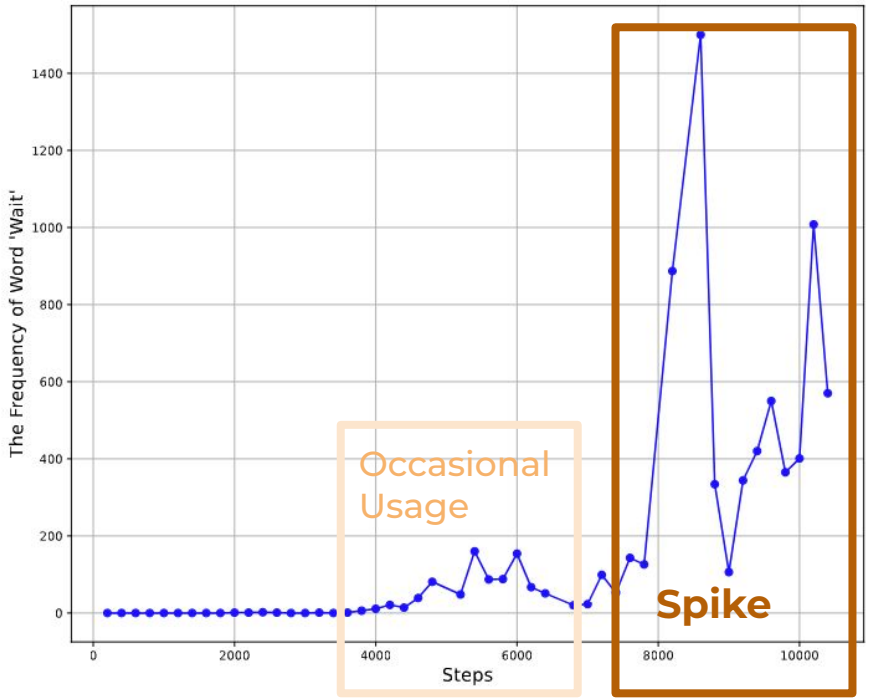


Figure 9
(a) Frequency of **representative reflective words** during the training process;
(b) Specific **occurrence** patterns of the word **“wait”** throughout the training process.

Evolution of DeepSeek-R1-Zero: Reasoning Capability

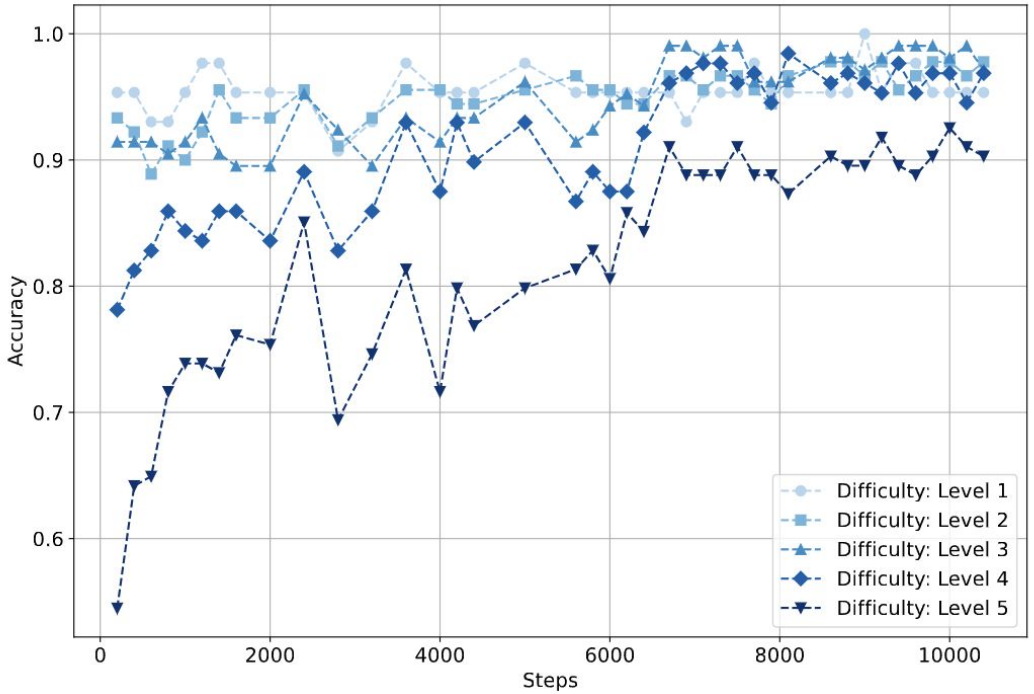


Figure 8: Performance of DeepSeek-R1-Zero on problems with varying **difficulty** levels in the **MATH** dataset

Easy problems (**levels 1-3**)
- **quickly** reach high accuracy (0.90-0.95)
- remain **stable**

Hard problems (**level 4**)
- remarkable improvement from near 0.78 to 0.95

The **hardest** problems (**level 5**)
- The most dramatic improvement from near 0.55 to 0.90

But…Limitations of Pure RL in DeepSeek-R1-Zero

Poor Readability and Language Mixing

Because it evolved without human-aligned formatting constraints, DeepSeek-R1-Zero struggles with **readability** and frequently **mixes languages,** such as combining **English** and **Chinese** within a **single chain-of-thought response**

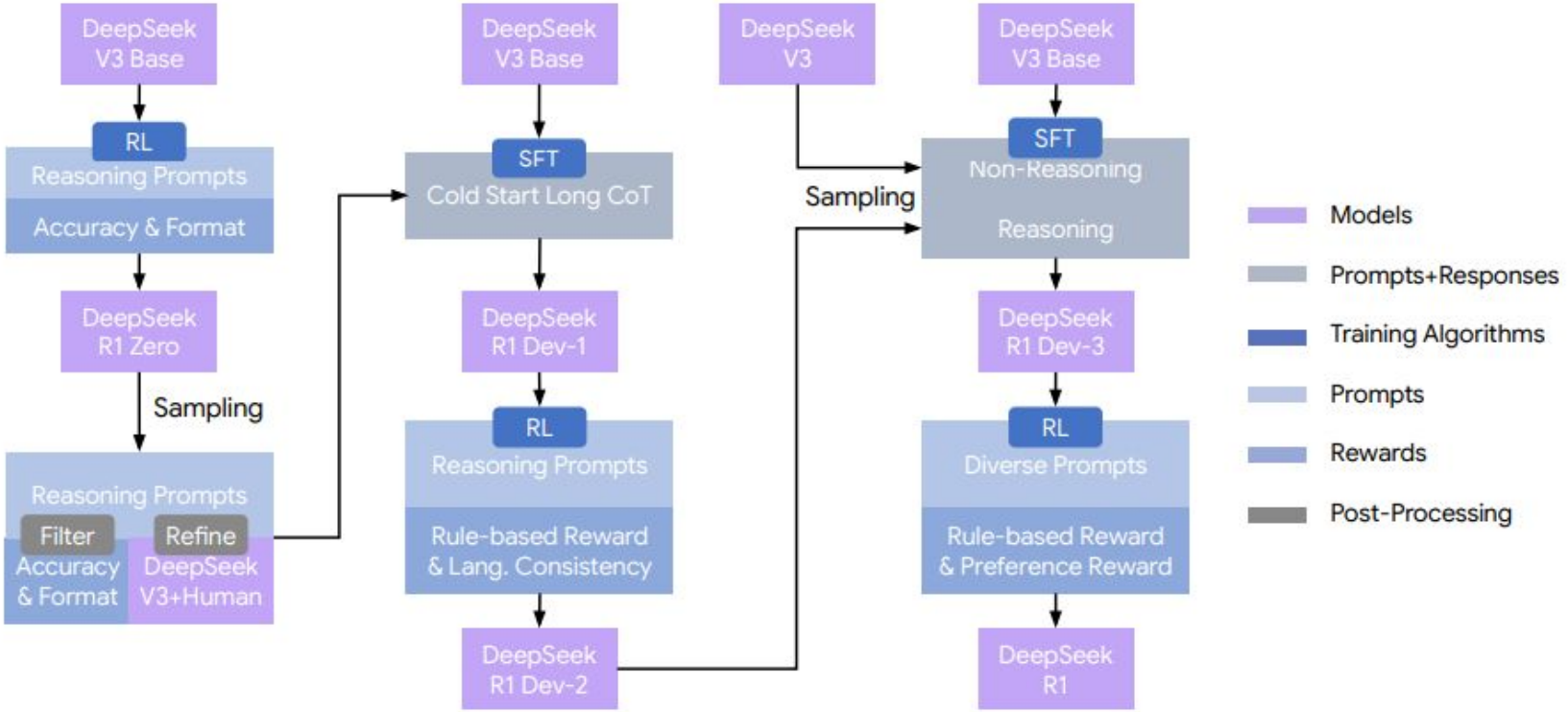
Lack of Robust Reward Model

Pure RL depends entirely on reliable reward signals. While **rule-based verifiers** work well for **math** and **code**, it is incredibly difficult to construct **robust** reward models for **subjective tasks** like creative writing.

Narrow Utility

Because the training is narrowly focused on **verifiable reasoning tasks**, DeepSeek-R1-Zero exhibits **limited** performance in **broader**, **general-domain** areas like **writing** and **open-domain** questions

DeepSeekR1 pipeline

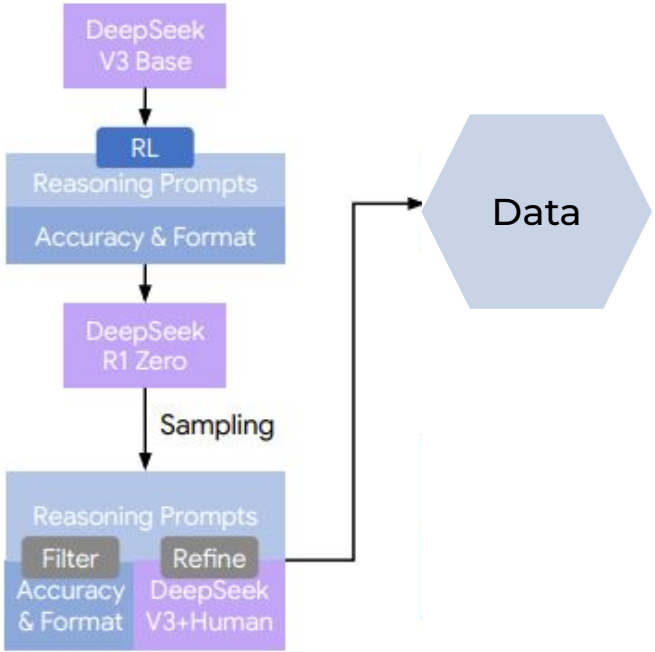


Stage 1 Stage 2 Stage 3

deepseekR1 https://arxiv.org/pdf/2501.12948

DeepSeekR1 pipeline

Deepseek R1 - zero

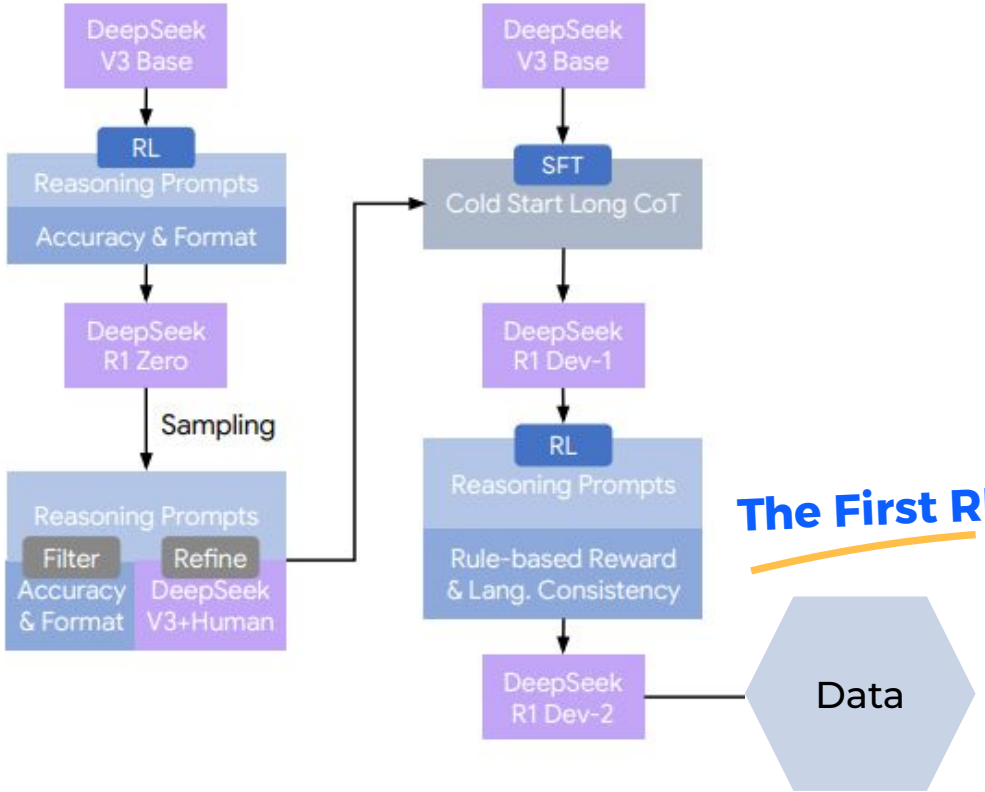


cold start data: from R1-zero's CoT.

=> readable. good interactive experience

deepseekR1 https://arxiv.org/pdf/2501.12948

DeepSeekR1 pipeline



The First RL

Core step:

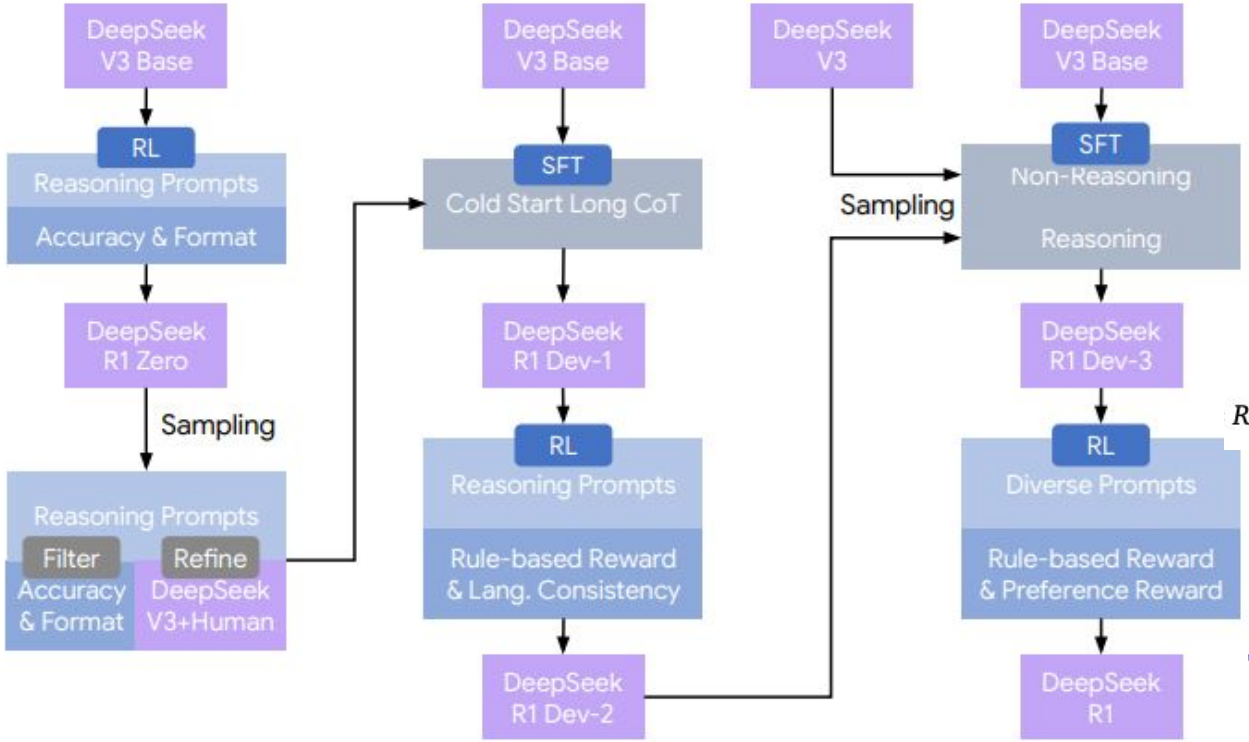
1. Dont use R1-zero anymore
2. Supervised Fine tuning V3
3. Get R1 Dev-1
4. First RL
5. Train with reasoning task
6. Rule based reward + lang consistency reward

$$Reward_{language} = \frac{Num(Words_{target})}{Num(Words)}$$

7. Get the Dev-2

deepseekR1 https://arxiv.org/pdf/2501.12948

DeepSeekR1 pipeline



Core step:

1. Dont use R1 Dev
2. Rejection sampling
3. Supervised Fine tuning V3
4. Training on reasoning + non-reasoning data
5. Get R1 Dev3
6. Start RL
   Using reasoning + general prompt
7. $Reward =$

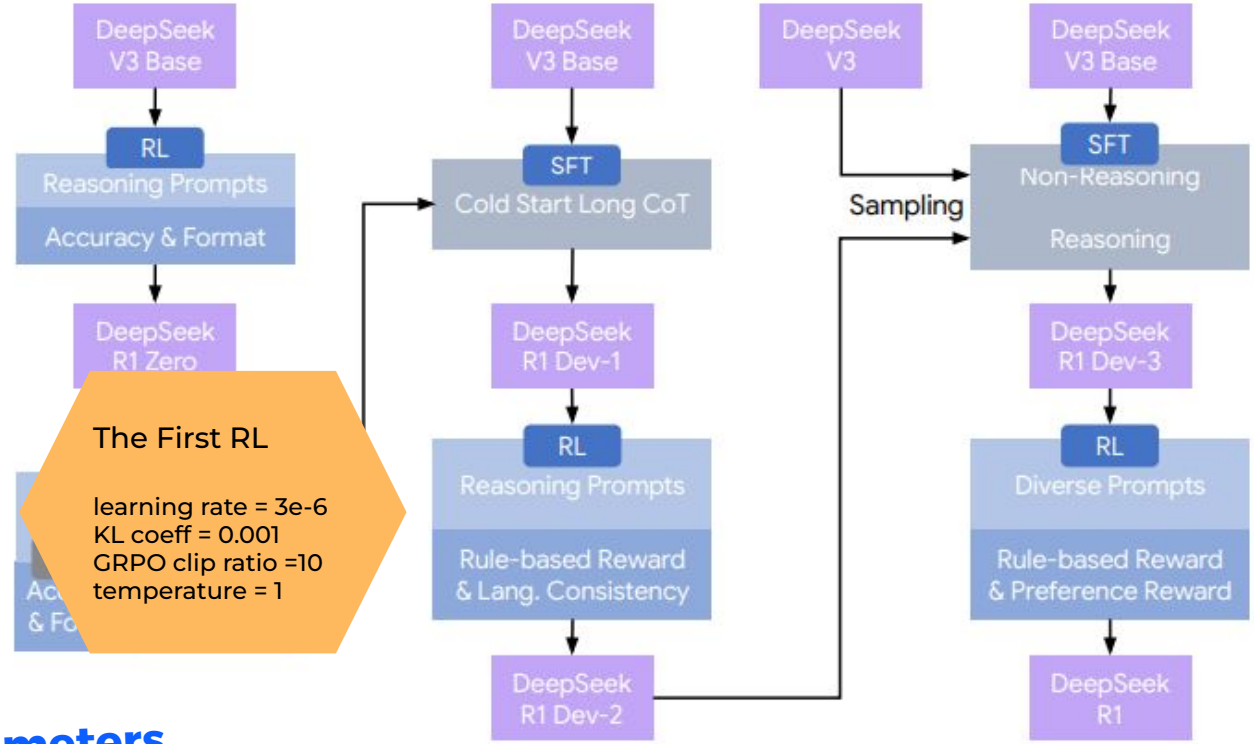
$$Reward_{\text{reasoning}} + Reward_{\text{general}} + Reward_{\text{language}}$$

8. Get SeepSeek R1
   (a) language consistency
   (b) handle general task too!

The Second RL

deepseekR1 https://arxiv.org/pdf/2501.12948

DeepSeekR1 pipeline



Hyper parameters

deepseekR1 https://arxiv.org/pdf/2501.12948

Training parameters details

The First RL

01

Sample 32 reasoning questions

02

Generate responses
16 outputs per question → 512 trajectories

03

Compute rewards
- correctness reward
- format reward
- language consistency

04

GRPO optimization
lr = 3e-6
KL = 0.001
clip ε = 10
temperature = 1

The Second RL

01

Sample mixed prompts
- reasoning tasks
- general instruction

02

Generate responses
temperature = 0.7

03

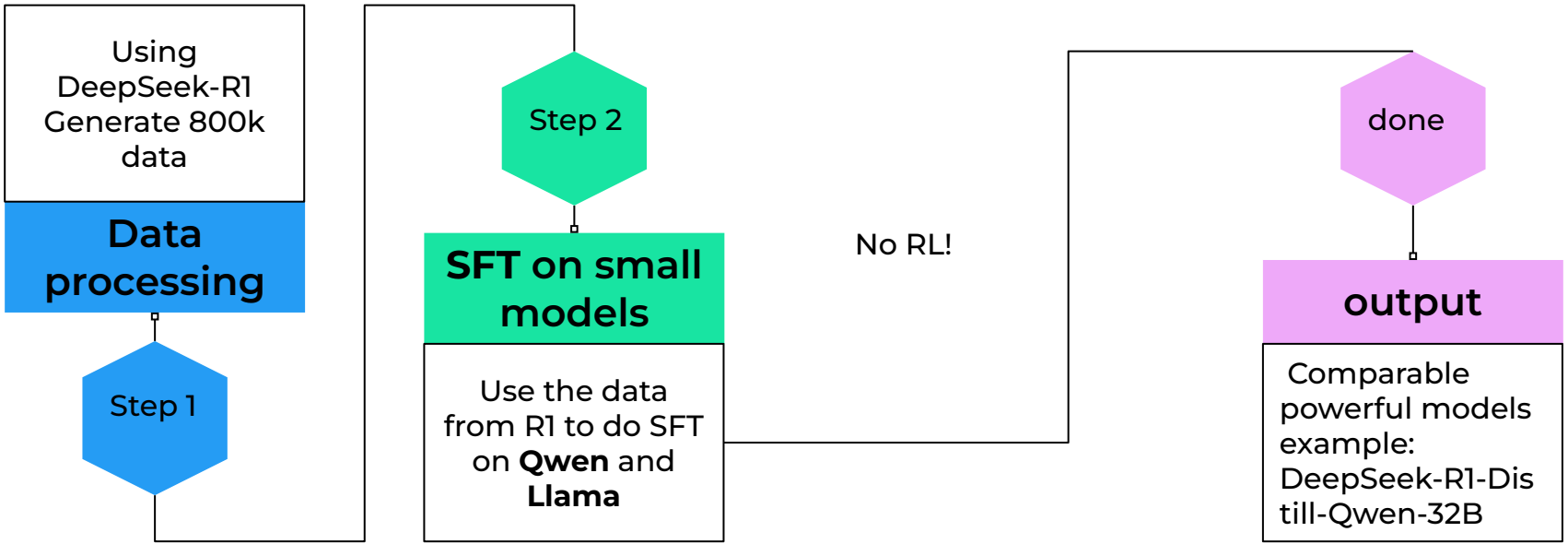
Compute rewards
- reasoning rule reward
- helpfulness reward model
- safety reward model

04

GRPO optimization
Same parameters

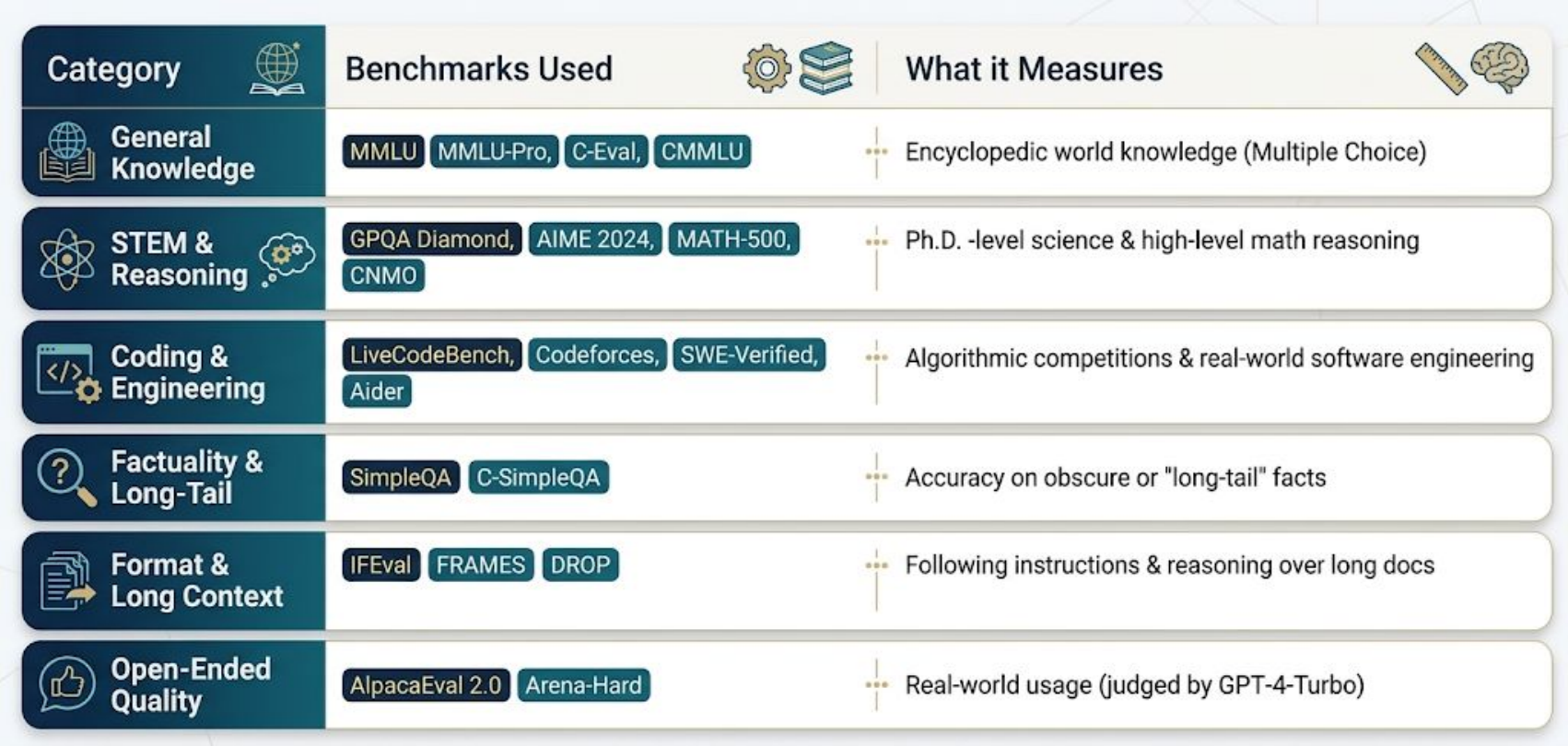
Total steps = 1700
Preference rewards on last 400 steps only

Distillation



Evaluation of DeepSeek-R1

Benchmarks:



Source: Gemini Nano Banana Pro 2

Evaluation of DeepSeek-R1

Benchmark Decontamination:

- Decontamination is the process of removing training data that overlaps with the benchmark test questions and solutions.
- The purpose of decontamination is to prevent the model from scoring well simply by memorizing benchmark data.
- It helps ensure that evaluation results reflect the model’s true reasoning and problem-solving capability.

Evaluation of DeepSeek-R1

Benchmark Decontamination:

- The base model, DeepSeek V3, has a knowledge cut-off of July 2024 which predates some evaluation benchmarks like CMNO 2024.
- The authors filtered out text sequences from sources like web pages and GitHub files that contained matching **10-gram sequences** from the evaluation questions and answers
- Eliminated about **6 million potentially contaminated pre-training texts** in mathematics domain alone
- For **post-training** SFT and RL training prompts, they applied the same 10-gram filtering protocol
- **Limitation:** This method catches exact or near-exact matches but may miss out on detecting paraphrases.

Evaluation of DeepSeek-R1

Baselines:

- DeepSeek-R1 is evaluated against several strong baselines:
  - DeepSeek-V3
  - Claude-Sonnet-3.5-1022
  - GPT-4o-0513
  - OpenAI-o1-mini
  - OpenAI-o1-1217
- Maximum generation length set to 32,768 tokens to support long-form reasoning
- Note: For OpenAI-o1-1217, the authors used the officially reported results for that model instead of running it themselves.

Evaluation of DeepSeek-R1

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (EM)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	92.3
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	96.6	96.3
	Codeforces (Rating)	717	759	1134	1820	2061	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	79.8
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	97.3
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

Comparison between DeepSeek-R1 and other representative models.

- **General Knowledge and STEM:**
  - DeepSeekR1 outperforms most of the baselines except for OpenAI-o1-1217 in MMLU
  - An improved accuracy in STEM-related questions achieved through large-scale reinforcement learning is the main reason behind the gains in these areas.

Evaluation of DeepSeek-R1

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (EM)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	92.3
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	96.6	96.3
	Codeforces (Rating)	717	759	1134	1820	2061	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	79.8
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	97.3
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

Comparison between DeepSeek-R1 and other representative models.

- **Coding and Engineering:**
  - On coding algorithm tasks, DeepSeek-R1 demonstrates performance on par with OpenAI-o1-1217, surpassing the other models by a large margin
  - On engineering-oriented coding tasks, OpenAI-o1-1217 outperforms DeepSeek-R1 on Aider but achieves comparable performance on SWE Verified
  - This is attributed to very limited amount of related RL training data

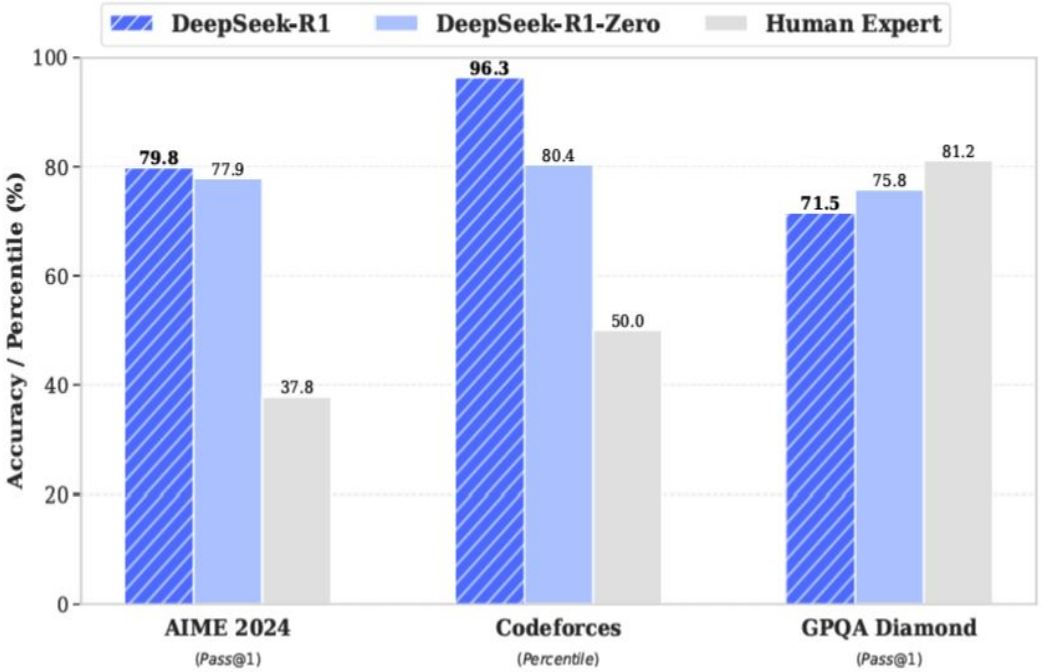
Evaluation of DeepSeek-R1

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (EM)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	92.3
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	96.6	96.3
	Codeforces (Rating)	717	759	1134	1820	2061	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	79.8
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	97.3
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

Comparison between DeepSeek-R1 and other representative models.

- **Mathematics:**
  - Deepseek-R1 performs on par with OpenAI o1-1217
  - It surpasses the other baseline models by a large margin
- **Writing and Open-domain Question Answering:**
  - DeepSeek-R1 achieves remarkable performance on AlpacaEval2.0 and ArenaHard
- **Instruction Following:**
  - DeepSeek-R1 fails to surpass the baseline models

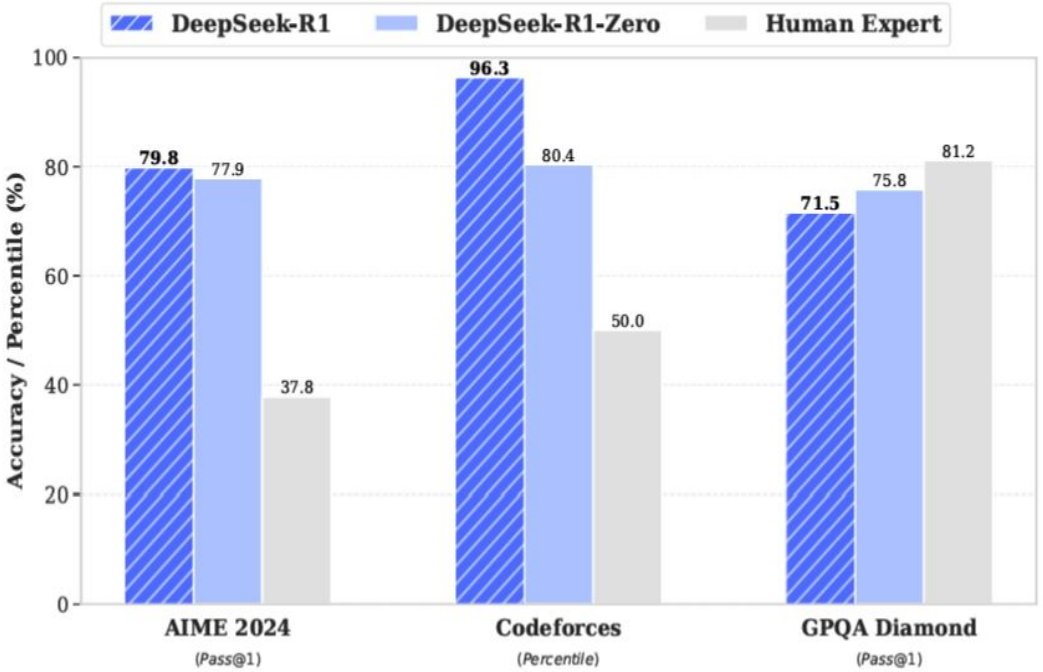
Evaluation of DeepSeek-R1



The benchmark performance of DeepSeek-R1 and DeepSeek-R1-Zero is compared with human scores across different datasets

- **Mathematics (AIME 2024):**
  - DeepSeek-R1 achieves a **pass@1 score of 79.8%**
  - Surpasses the average score achieved by human experts in high-school level competition

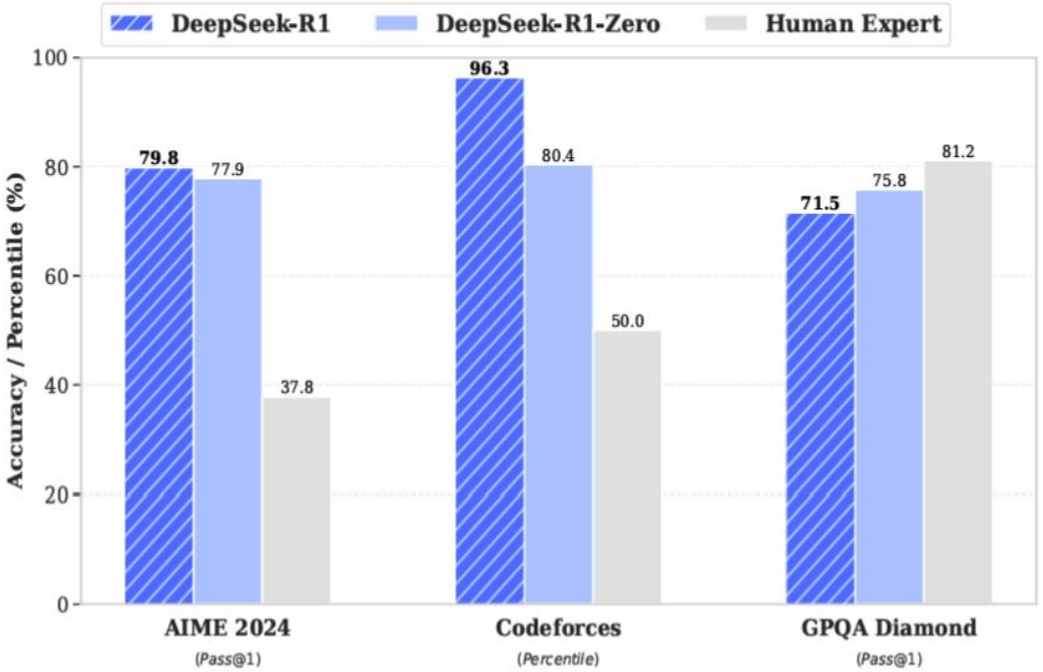
Evaluation of DeepSeek-R1



The benchmark performance of DeepSeek-R1 and DeepSeek-R1-Zero is compared with human scores across different datasets

- **Competitive Coding (Codeforces):**
  - DeepSeek-R1 outperforms **96.3%** of human participants
  - This indicates DeepSeek-R1's advanced problem-solving capabilities in algorithmic tasks, outperforming the vast majority of human competitors on the platform.

Evaluation of DeepSeek-R1



The benchmark performance of DeepSeek-R1 and DeepSeek-R1-Zero is compared with human scores across different datasets

- **Scientific Reasoning (GPQA Diamond):**
  - DeepSeek-R1 achieves a pass@1 score of **71.5%**
  - Though it is impressive for Ph.D level tasks in Physics, Chemistry and Biology, it remains lower than human expert performance of 81.2
  - This may be due to human experts having web access for these questions unlike the model

Evaluation of DeepSeek-R1

Performance Comparison with DeepSeek-V3

	Benchmark (Metric)	V3-Base	V3	R1-Zero	R1
English	MMLU (EM)	87.1	88.5	88.8	**90.8**
	MMLU-Redux (EM)	86.2	89.1	85.6	**92.9**
	MMLU-Pro (EM)	64.4	75.9	68.9	**84.0**
	DROP (3-shot F1)	89.0	91.6	89.1	**92.2**
	IF-Eval (Prompt Strict)	58.6	**86.1**	46.6	83.3
	GPQA Diamond (Pass@1)	-	59.1	**75.8**	71.5
	SimpleQA (Correct)	20.1	24.9	30.3	30.1
	FRAMES (Acc.)	-	73.3	82.3	82.5
	AlpacaEval2.0 (LC-winrate)	-	70.0	24.7	**87.6**
	ArenaHard (GPT-4-1106)	-	85.5	53.6	**92.3**
Code	LiveCodeBench (Pass@1-COT)	-	36.2	50.0	**65.9**
	Codeforces (Percentile)	-	58.7	80.4	**96.3**
	Codeforces (Rating)	-	1134	1444	**2029**
	SWE Verified (Resolved)	-	42.0	43.2	**49.2**
	Aider-Polyglot (Acc.)	-	49.6	12.2	**53.3**
Math	AIME 2024 (Pass@1)	-	39.2	77.9	**79.8**
	MATH-500 (Pass@1)	-	90.2	95.9	**97.3**
	CNMO 2024 (Pass@1)	-	43.2	**88.1**	78.8
Chinese	CLUEWSC (EM)	82.7	90.9	93.1	92.8
	C-Eval (EM)	90.1	86.5	**92.8**	91.8
	C-SimpleQA (Correct)	-	**68.0**	66.4	63.7

Comparative Analysis of DeepSeek-V3 and DeepSeek-R1

- DeepSeek-R1 and DeepSeek-V3 are built on the same base model: DeepSeek-V3-Base
- The key question is which capabilities improve under different post-training strategies
- Overall, DeepSeek-R1 shows strong reasoning abilities whereas DeepSeek V3 shows better instruction following capabilities

Evaluation of DeepSeek-R1

	Benchmark (Metric)	R1-Zero	R1-Dev1	R1-Dev2	R1-Dev3	R1
English	MMLU (EM)	88.8	89.1	**91.2**	91.0	90.8
	MMLU-Redux (EM)	85.6	90.0	93.0	93.1	92.9
	MMLU-Pro (EM)	68.9	74.1	83.8	83.1	**84.0**
	DROP (3-shot F1)	89.1	89.8	91.1	88.7	**92.2**
	IF-Eval (Prompt Strict)	46.6	71.7	72.0	78.1	**83.3**
	GPQA Diamond (Pass@1)	**75.8**	66.1	70.7	71.2	71.5
	SimpleQA (Correct)	30.3	17.8	28.2	24.9	30.1
	FRAMES (Acc.)	82.3	78.5	81.8	81.9	**82.5**
	AlpacaEval2.0 (LC-winrate)	24.7	50.1	55.8	62.1	**87.6**
	ArenaHard (GPT-4-1106)	53.6	77.0	73.2	75.6	**92.3**
Code	LiveCodeBench (Pass@1-COT)	50.0	57.5	63.5	64.6	**65.9**
	Codeforces (Percentile)	80.4	84.5	90.5	92.1	**96.3**
	Codeforces (Rating)	1444	1534	1687	1746	**2029**
	SWE Verified (Resolved)	43.2	39.6	44.6	45.6	**49.2**
	Aider-Polyglot (Acc.)	12.2	6.7	25.6	44.8	**53.3**
Math	AIME 2024 (Pass@1)	77.9	59.0	74.0	78.1	**79.8**
	MATH-500 (Pass@1)	95.9	94.2	95.9	95.4	**97.3**
	CNMO 2024 (Pass@1)	**88.1**	58.0	73.9	77.3	78.8
Chinese	CLUEWSC (EM)	93.1	92.8	92.6	91.6	92.8
	C-Eval (EM)	**92.8**	85.7	91.9	86.4	91.8
	C-SimpleQA (Correct)	66.4	58.8	64.2	66.9	63.7

Experimental Results at each stage of DeepSeek-R1.

Performance Trends Across Stages

- R1-Zero begins with a decent reasoning ability
- Dev1 improves instruction following but reasoning ability drops in coding and math benchmarks
- Dev2 strengthens reasoning in math and coding benchmarks
- Dev3 incorporates reasoning and non-reasoning data to bring stability
- R1 improves overall quality with RL

Evaluation of DeepSeek-R1 Distillation Model

- The authors use DeepSeek-R1 as teacher model to train relatively smaller models like Qwen and LlaMA to reduce this cost.
- The distilled models are trained on a dataset containing 80,000 samples generated by DeepSeek-R1.
- The goal is to transfer strong reasoning ability to the smaller and cheaper models
- The student models use Supervised Fine-Tuning (SFT) only - no RL stage is used in this setup
- Benchmarks used for evaluating distilled models:

  i) AIME 2024 ; ii) MATH-500 ; iii) GPQA Diamond ; iv) LiveCodeBench ;

  v) CodeForces

Evaluation of DeepSeek-R1 Distillation Model

Model	AIME 2024		MATH	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	**72.6**	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	**86.7**	**94.5**	**65.2**	**57.5**	**1633**

Comparison of DeepSeek-R1 distilled models and other comparable models on reasoning-related benchmarks.

- Two well-established LLM's are used as baselines:
  - GPT-4o-0513
  - Claude-3.5-Sonnet-1022
- In most cases, distilled models surpass the baseline models

- Smallest model DeepSeek-R1-Distill-Qwen-1.5B outperforms the baselines on math benchmarks and CodeForces
- Smaller models that are distilled from strong reasoning teacher models can still be very capable

Drawbacks of DeepSeek-R1

- **Structure output and tool use:**
  - The model is still fails to produce well-structured outputs compared with some existing models.
  - It cannot use external tools like search engines that can significantly improve answer quality and formatting.
- **Token Efficiency:**
  - It tends to *overthink* and too many tokens on simple tasks
- **Language Mixing:**
  - Since the model is optimized in English and Chinese, there can be mixed-language responses if a question is asked in another language

Key Findings

- Importance of Base Checkpoints:
  - The authors started with smaller models
  - These models failed to show meaningful reasoning improvement on AIME which they used as a key validation benchmark
  - As the responses became lengthier, smaller models became repetitive and could not use long CoT
  - Shifting to larger models led to improvements from RL training
  - RL-based improvements depend highly on the strength of the base model

Key Findings

- Importance of Verifiers:
  - Success of RL largely depends on reliable reward signals.
  - Rule-based reward models and LLM-based evaluators work well for tasks that have clear and well-defined answers (e.g. math questions)
  - Designing reliable reward models is difficult for subjective tasks (e.g. writing)
  - RL is most effective when reward signal is trustworthy

Key Findings

- Iterative SFT and RL pipeline:
  - SFT is crucial for tasks where reward signals are hard to define
  - RL is helps the model to explore better reasoning trajectories
  - Only RL can lead to reward hacking in long-form tasks
  - Only SFT can prevent the model to fully acquire reasoning ability that can be gained via exploration
  - Best results come from a multi-stage training pipeline that consists of both SFT and RL

Possible Improvements

- Provide stronger analysis of reward hacking with examples and benchmarks where reward hacking was seen
- Incorporate multilingual benchmarks (e.g. M-IFEval) for analysis to show the model’s limitations and see if how well reasoning transfers across languages
- Augment external tools because many reasoning failures can be fixed easily using these tools rather than using pure reasoning

Thank you!

NOT FOR PRESENTATION: Possible Questions

1. Why use GRPO over PPO?
   Save resource on mem and computation
   PPO using per step KL penalty -> limit generating long CoT
2. Why does RLHF learn an implicit reward model?
   RL too powerful, reward hacking
3. why final reward can give long CoT and high reasoning.
   coz the if guess at beginning, will have low accuracy ~ 0
   if think longer, the accuracy will increase, the model will learning that
   longer reasoning -> response can help it check itself
   if its thinking the wrong answer rate will decrease.
   very similar to alpha go.
   win +1
   loss -1
   no step by step reward.
4. How aha moment happen?
   maybe it failed at this point or similar point before, it knows should think for longer time.
5. for r1 first RL, why set grpo clip ratio to 10, make the real interval = (0, 11]
   => if new policy too good, reject, but accept no matter how bad it is.
   encourage exploration.
   use the KL term to play the real regulator role.
6. how they distill the small model?
   SFT on small llms
7. Is the CoT the longer the better?
   No, overthinking.