

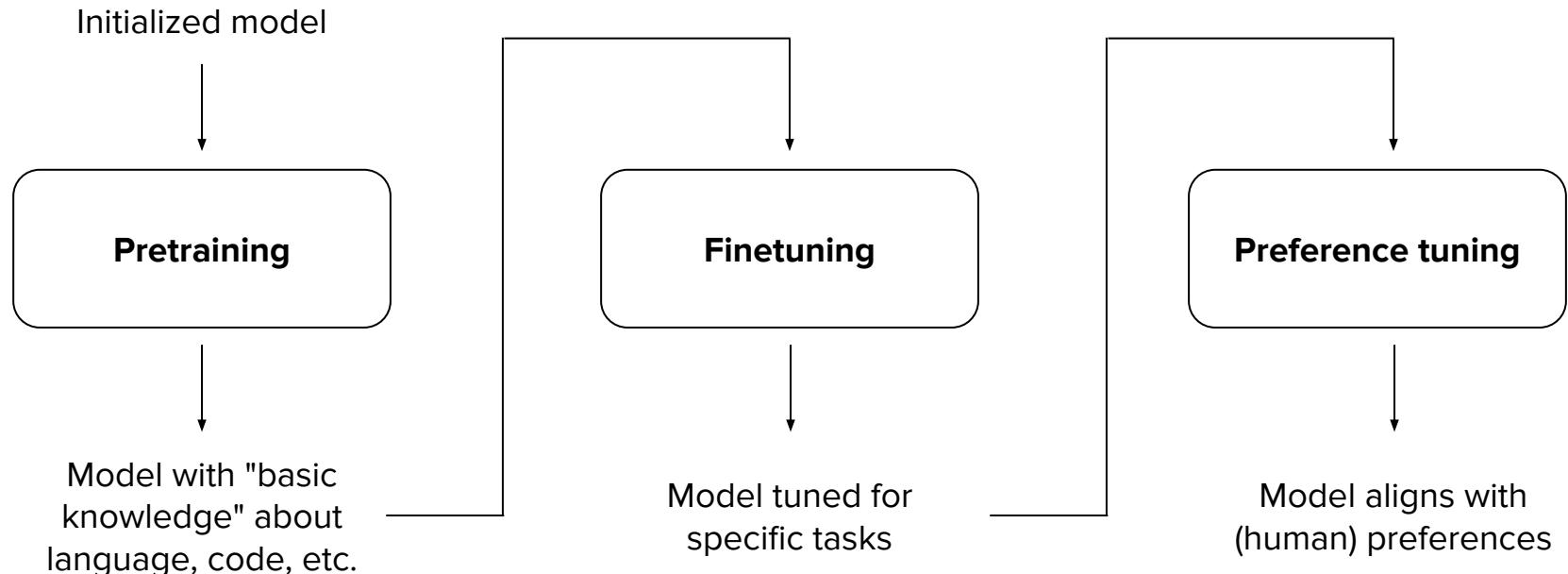
CME 295: Transformers & Large Language Models



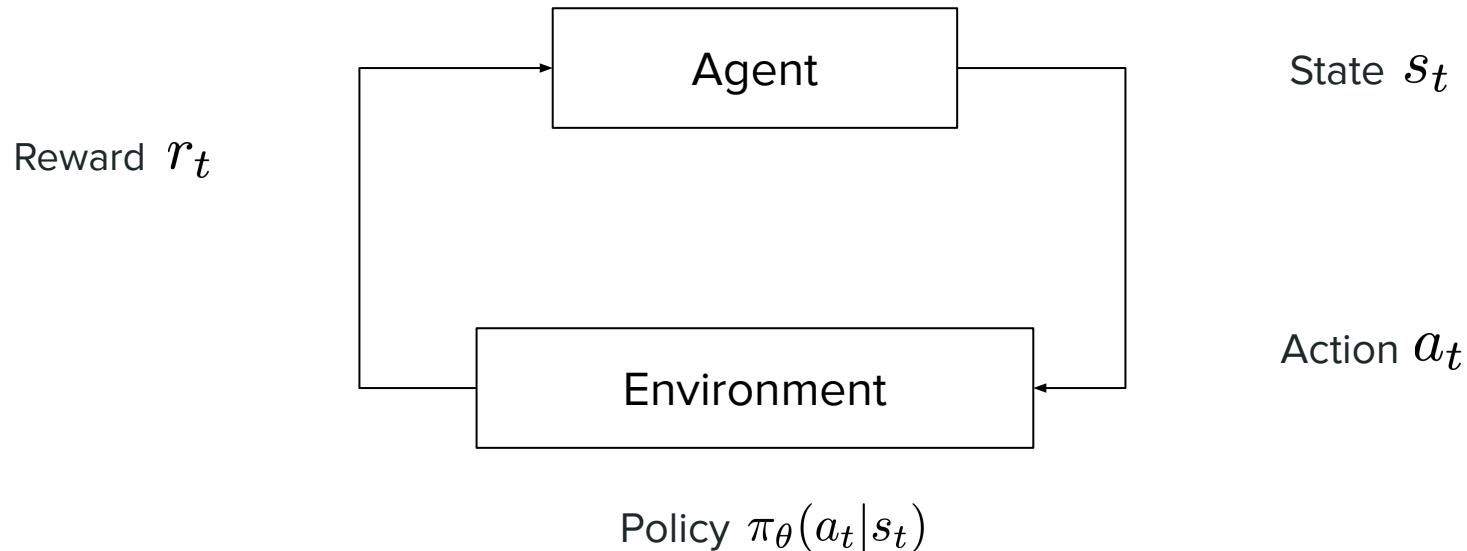
Afshin Amidi & Shervine Amidi



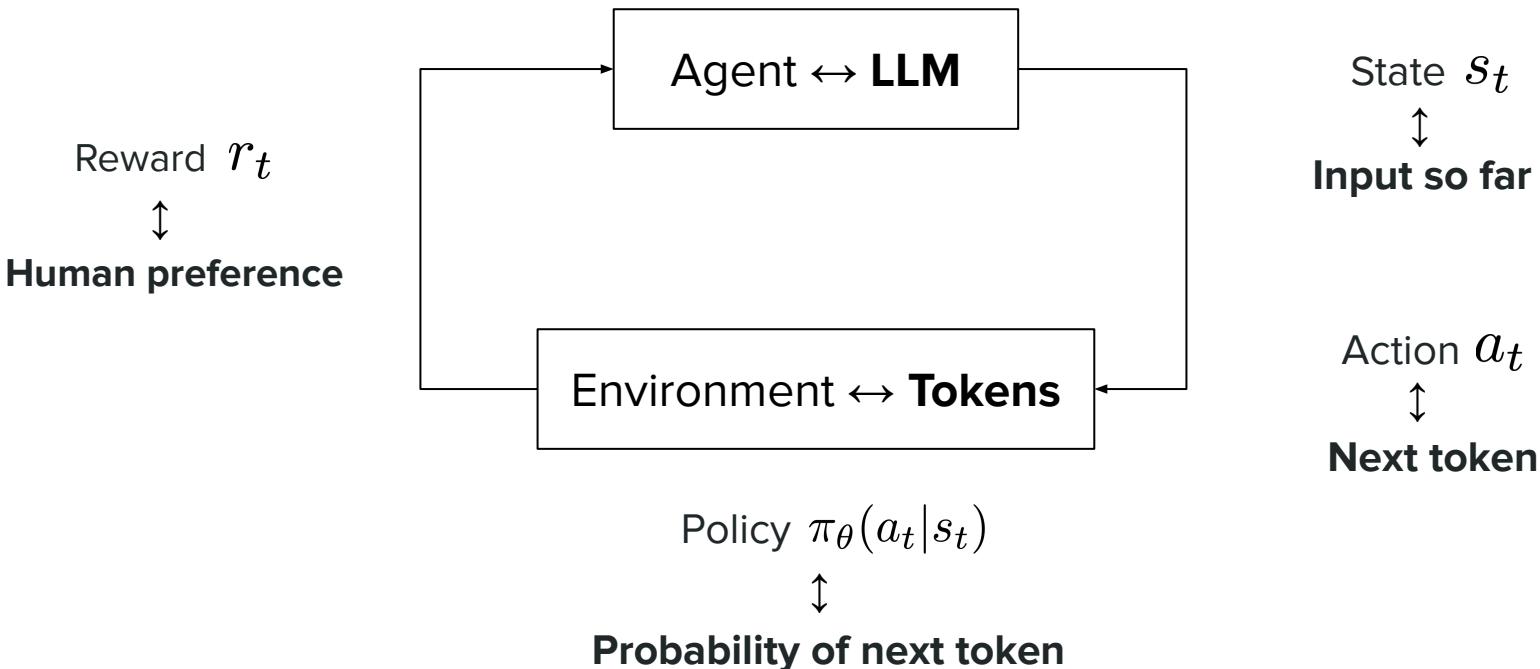
Recap of last episode...



Recap of last episode...



Recap of last episode...



Recap of last episode...

$$\mathcal{L}(\theta) = \boxed{\text{Maximize advantages}} + \boxed{\text{Don't deviate too much from old/base model}}$$

Recap of last episode...

$$\mathcal{L}(\theta) = \boxed{\text{Maximize advantages}} +$$

Don't deviate too much
from old/base model

PPO-clip

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right]$$

Recap of last episode...

$$\mathcal{L}(\theta) = \boxed{\text{Maximize advantages}} + \boxed{\text{Don't deviate too much from old/base model}}$$

PPO-clip $L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right]$

PPO-KL penalty $L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t - \beta \text{KL}[\pi_{\theta_{\text{old}}}(\cdot | s_t), \pi_\theta(\cdot | s_t)] \right]$

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Weaknesses.

- Limited reasoning
- // like coding posts, mathematical post

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Weaknesses.

- Limited reasoning
- Knowledge is static // upto date data trained.

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Weaknesses.

- Limited reasoning
- Knowledge is static
- Cannot perform actions *// like placing an order,
// doing something actions.*

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Weaknesses.

- Limited reasoning
- Knowledge is static
- Cannot perform actions
- Hard to evaluate //

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Weaknesses.

- Limited reasoning
- Knowledge is static
- Cannot perform actions
- Hard to evaluate



Focus of lectures 7 & 8

Practical consideration & caveats of vanilla LLMs

Strengths.

- Great at imitation or idea generation
- Amazing at generating or debugging code

Weaknesses.

- **Limited reasoning**
- Knowledge is static
- Cannot perform actions
- Hard to evaluate



Focus of today



Transformers & Large Language Models

Reasoning models

Scaling with RL

GRPO

Applications

- ① What are the reasoning model available?
- ② How they have trained

Terminology

Tentative definition

Reasoning = Ability to **solve a problem**

Complex problem

(free form tasks i.e. coding; mathematical education)

Terminology

Tentative definition

Reasoning = Ability to **solve** a **problem**

Not reasoning

"What is the course code of Stanford's
Transformers & LLMs class?" *(not reasoning)*

Reasoning

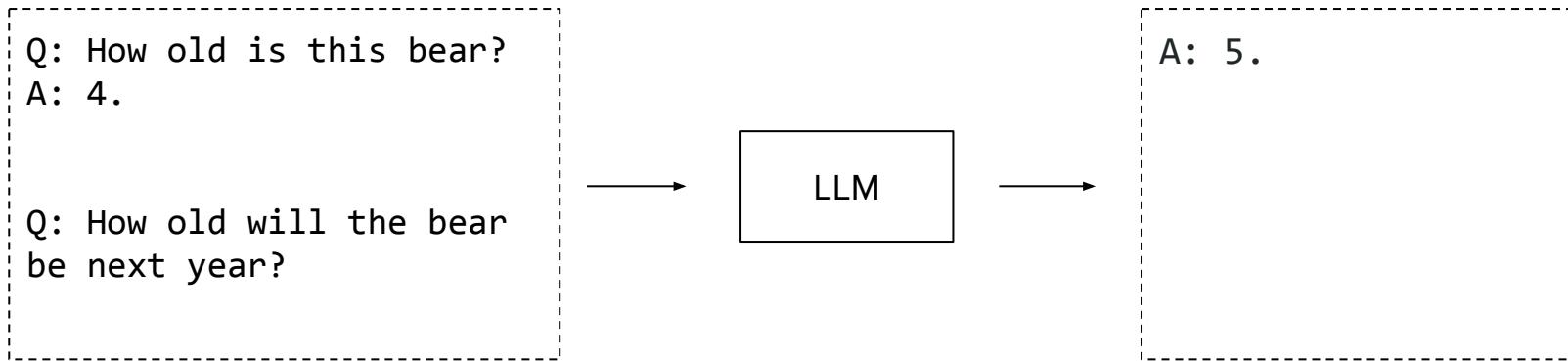
"The bear was born in 2020. How old is
this bear now?" *(math question)*

Core idea behind improving reasoning

Strategy. Teach model to explain its reasoning before answering (Chain of Thought)

Core idea behind improving reasoning

Strategy. Teach model to explain its reasoning before answering (**Chain of Thought**)



Core idea behind improving reasoning

Strategy. Teach model to explain its reasoning before answering (**Chain of Thought**)

Q: How old is this bear?
A: The bear was born in
2020. It is therefore 4.

Q: How old will the bear
be next year?



A: It will be one
year older than
its age this year,
which was 4.
Hence, it will be
5.



in-context learning with
mathematical solution.

Core idea behind improving reasoning

Strategy. Teach model to explain its reasoning before answering (**Chain of Thought**)



Idea for reasoning models: Do CoT but at a much larger scale.

Chain of thought

- LLM decomposes the problem and solve them independently to give final answer
- more complex happens

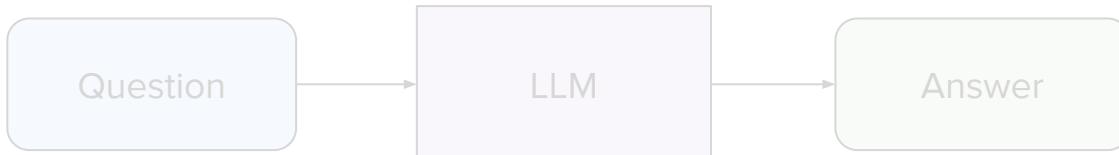
Enhancing model reasoning abilities

Until now

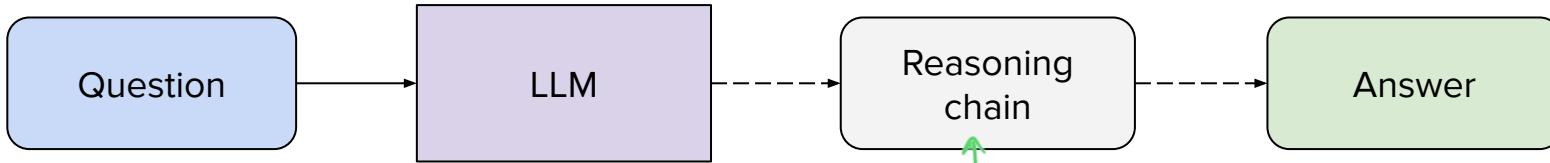


Enhancing model reasoning abilities

Until now



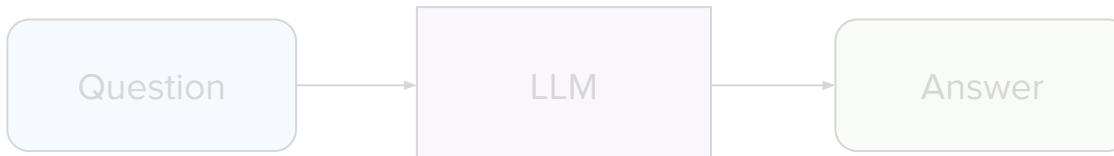
New paradigm



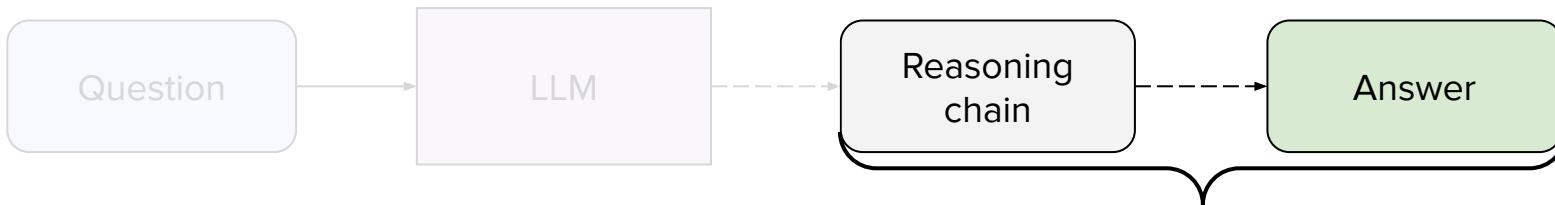
don't directly generate answer
but take time to think
i.e. (goes with chain of thought before answering)

Enhancing model reasoning abilities

Until now



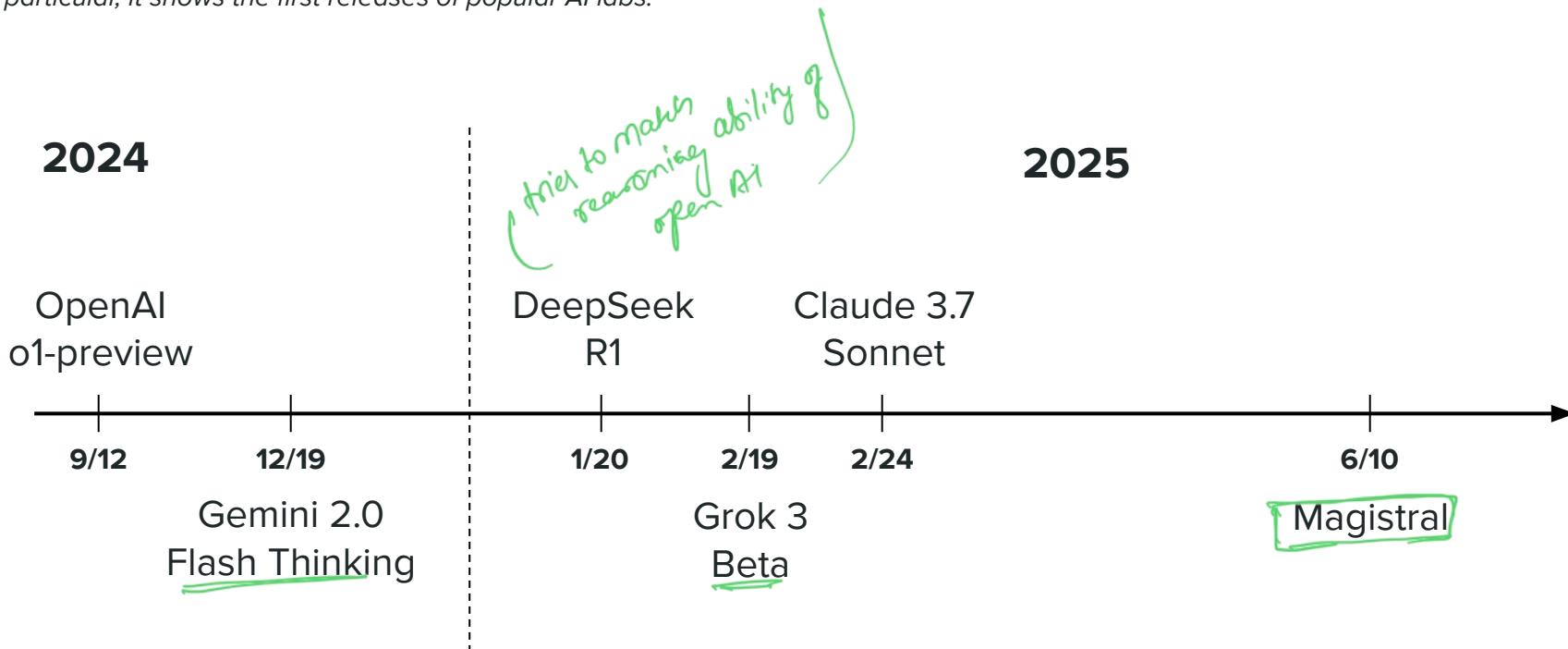
New paradigm



Output = Reasoning + Answer

Reasoning model release is trendy

Disclaimer: This timeline is not at scale for better readability. It is also there for illustrative purposes and should not be considered exhaustive. In particular, it shows the first releases of popular AI labs.



How to spot a reasoning model

ChatGPT 5 Thinking

Thinking here
is asking but talking
here for chain of thought

Share

...

Activity · 5s

X

My bear was born in 2020. How old is he now?

Thought for 5s >

As of today (November 3, 2025), he's either **5 years old** if his 2025 birthday has already passed, or **4 years old** if his birthday is later in November or December. If you share the exact date, I can give the precise age.

□ ⚡ 🗃 ⌂ ⌂ ...

↓

Thinking
of
for
chain
of
thought

Ask anything

+ 🔍 Extended thinking

0 ⓘ

ChatGPT can make mistakes. Check important info.

Thinking

- Considering age calculation

The question is simple: the bear was born in 2020, and we need to calculate age as of November 3, 2025. If the bear's birthday has occurred before or on that day, they would be 5. If after November 3, they're still 4. So, either

- Thought for 5s

Done

How to spot a reasoning model

ChatGPT 5 Thinking

Share

...

Activity · 5s

X

My bear was born in 2020. How old is he now?

Thought for 5s >

As of today (November 3, 2025), he's either **5 years old** if his 2025 birthday has already passed, or **4 years old** if his birthday is later in November or December. If you share the exact date, I can give the precise age.

□ ⚡ 🎁 ⌂ ⏪ ...



Ask anything



🕒 Extended thinking ▾



ChatGPT can make mistakes. Check important info.

Thinking

- Considering age calculation

The question is simple: the bear was born in 2020, and we need to calculate age as of November 3, 2025. If the bear's birthday has occurred before or on that day, they would be 5. If after November 3, they're still 4. So, either

⌚ Thought for 5s

Done

How to spot a reasoning model

ChatGPT 5 Thinking ▾

Share

...

Activity · 5s

X

My bear was born in 2020. How old is he now?

Thought for 5s >

As of today (November 3, 2025), he's either **5 years old** if his 2025 birthday has already passed, or **4 years old** if his birthday is later in November or December. If you share the exact date, I can give the precise age.

Ask anything

↓

+  Extended thinking ▾

0 

ChatGPT can make mistakes. Check important info.

"Thought summary"

Thinking

- Considering age calculation

The question is simple: the bear was born in 2020, and we need to calculate age as of November 3, 2025. If the bear's birthday has occurred before or on that day, they would be 5. If after November 3, they're still 4. So, either

- ⌚ Thought for 5s
Done

Complete chain of thought usually hidden

How to spot a reasoning model

Pricing

Text tokens

Prices per 1M tokens.

MODEL	INPUT	CACHED INPUT	OUTPUT
gpt-5	\$1.25	\$0.125	\$10.00
gpt-5-mini	\$0.25	\$0.025	\$2.00
gpt-5-nano	\$0.05	\$0.005	\$0.40
gpt-5-chat-latest	\$1.25	\$0.125	\$10.00
gpt-5-codex	\$1.25	\$0.125	\$10.00
⋮			

While reasoning tokens are not visible via the API, they still occupy space in the model's context window and are billed as output tokens.

OpenAI

Copy page

Anthropic

- You're charged for the full thinking tokens generated by the original request, not the summary tokens.
- The billed output token count will **not match** the count of tokens you see in the response.
- The first few lines of thinking output are more verbose, providing detailed reasoning that's particularly helpful for prompt engineering purposes.

Google

Gemini 2.5 Pro

gemini-2.5-pro

Our state-of-the-art multipurpose model, which excels at coding and complex reasoning tasks.

Try it in Google AI Studio

Standard	Batch	Free Tier	Paid Tier, per 1M tokens in USD
Input price	Free of charge	\$1.25, prompts <= 200k tokens \$2.50, prompts > 200k tokens	
Output price (including thinking tokens)	Free of charge	\$10.00, prompts <= 200k tokens \$15.00, prompts > 200k	
Context caching price	Not available	\$0.125, prompts <= 200k tokens \$0.25, prompts > 200k \$4.50 / 1,000,000 tokens per hour (storage price)	
Grounding with Google Search	Not available	1,500 RPD (free), then \$35 / 1,000 grounded prompts	
Grounding with Google Maps	Not available	10,000 RPD (free), then \$25 / 1,000 grounded prompts	
Used to improve our products	Yes	No	

How to spot a reasoning model

Pricing

Text tokens

Prices per 1M tokens.

Batch Flex Standard Priority

MODEL	INPUT	CACHED INPUT	OUTPUT
gpt-5	\$1.25	\$0.125	\$10.00
gpt-5-mini	\$0.25	\$0.025	\$2.00
gpt-5-nano	\$0.05	\$0.005	\$0.40
gpt-5-chat-latest	\$1.25	\$0.125	\$10.00
gpt-5-codex	\$1.25	\$0.125	\$10.00
⋮			

While reasoning tokens are not visible via the API, they still occupy space in the model's context window and are billed as output tokens.

OpenAI

Copy page

Anthropic



You're charged for the full thinking tokens generated by the original request, not the summary tokens.

- The billed output token count will **not match** the count of tokens you see in the response.
- The first few lines of thinking output are more verbose, providing detailed reasoning that's particularly helpful for prompt engineering purposes.

Google

Gemini 2.5 Pro

gemini-2.5-pro

Our state-of-the-art multipurpose model, which excels at coding and complex reasoning tasks.

Try it in Google AI Studio

Standard	Batch	Free Tier	Paid Tier, per 1M tokens in USD
Input price	Free of charge	\$1.25, prompts <= 200k tokens \$2.50, prompts > 200k tokens	
Output price (including thinking tokens)	Free of charge	\$10.00, prompts <= 200k tokens \$15.00, prompts > 200k	
Context caching price	Not available	\$0.125, prompts <= 200k tokens \$0.25, prompts > 200k \$4.50 / 1,000,000 tokens per hour (storage price)	
Grounding with Google Search	Not available	1,500 RPD (free), then \$35 / 1,000 grounded prompts	
Grounding with Google Maps	Not available	10,000 RPD (free), then \$25 / 1,000 grounded prompts	
Used to improve our products	Yes	No	

Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

You have n teddy bears in a line. Each bear has a size.

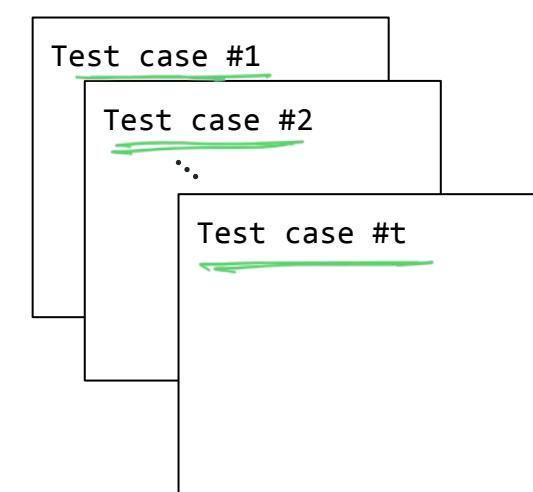
Find the biggest bear that is smaller than the largest bear.

```
def second_biggest_bear(bears):
    largest = max(bears)
    return max(
        b for b in bears
        if b < largest
    )
```

Problem

Solution

Verification



Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

You have n teddy bears in a line. Each bear has a size.

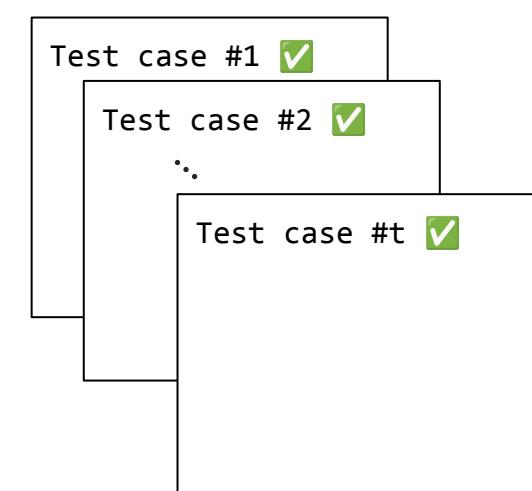
Find the biggest bear that is smaller than the largest bear.

```
def second_biggest_bear(bears):
    largest = max(bears)
    return max(
        b for b in bears
        if b < largest
    )
```

Problem

Solution

Verification



Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

task_id	prompt	canonical_solution	test
string · lengths 11·12	string · lengths 6.1% 240·365 24.4%	string · lengths 186·271 18.9%	string · lengths 455·624 18.9%
HumanEval/0	<pre>from typing import List def has_close_elements(numbers: List[float], threshold: float) -> bool: """ Check if in given list of numbers, are any two numbers closer to each other than given threshold. >>> has_close_elements([1.0, 2.0, 3.0], 0.5) False >>> has_close_elements([1.0, 2.8, 3.0, 4.0, 5.0, 2.0], 0.3) True """</pre>	<pre>for idx, elem in enumerate(numbers): for idx2, elem2 in enumerate(numbers): if idx != idx2: distance = abs(elem - elem2) if distance < threshold: return True return False</pre>	<pre>METADATA = { 'author': 'jt', 'dataset': 'test' } def check(candidate): assert candidate([1.0, 2.0, 3.9, 4.0, 5.0, 2.2], 0.3) == True assert candidate([1.0, 2.0, 3.9, 4.0, 5.0, 2.2], 0.05) == False assert candidate([1.0, 2.0, 5.9, 4.0, 5.0], 0.95) == True assert candidate([1.0, 2.0, 5.9, 4.0, 5.0], 0.8) == False assert candidate([1.0, 2.0, 3.0, 4.0, 5.0, 2.0], 0.1) == True assert candidate([1.1, 2.2, 3.1, 4.1, 5.1], 1.0) == True assert candidate([1.1, 2.2, 3.1, 4.1, 5.1], 0.5) == False</pre>

Examples: HumanEval, CodeForces, SWE-bench

Used for dataset and test case for training COT

Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

Math. Solve a challenging math problem (e.g. olympiads)

Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

Math. Solve a challenging math problem (e.g. olympiads)

The bear was born in 2020.
How old is the bear now?

It is 2025 now. Subtract the birth year from the current year: $2025 - 2020 = 5$.

Answer: 5

5

Problem

Reasoning

Ground truth

Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

Math. Solve a challenging math problem (e.g. olympiads)

The bear was born in 2020.
How old is the bear now?

It is 2025 now. Subtract the birth year from the current year: $2025 - 2020 = 5$.

Answer:

Verification 

Problem

Reasoning

Ground truth

Reasoning-based benchmarks

Coding. Solve a coding problem, fix a bug.

Math. Solve a challenging math problem (e.g. olympiads)

id	problem	solution	answer
int64	string · lengths	string · lengths	string · lengths
60	Every morning Aya goes for a \$9\$-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of \$s\$ kilometers per hour, the walk takes her 4 hours, including \$t\$ minutes spent in the coffee shop. When she walks \$s+2\$ kilometers per hour, the walk takes her 2 hours and 24 minutes, including \$t\$ minutes spent in the coffee shop. Suppose Aya walks at \$s+\frac{1}{2}\$ kilometers per hour. Find the number of minutes the walk takes her, including the \$t\$ minutes spent in the coffee shop.	$\frac{9}{s} + t = 4$ in hours and $\frac{9}{s+2} + t = 2.4$ in hours. Subtracting the second equation from the first, we get, $\frac{9}{s} - \frac{9}{s+2} = 1.6$. Multiplying by $(s)(s+2)$, we get $9s+18-9s=18=1.6s^2+3.2s$. Multiplying by $5/2$ on both sides, we get $0 = 4s^2 + 8s - 45$. Factoring gives us $(2s-5)(2s+9) = 0$, of which the solution we want is $s=2.5$. Substituting this back to the first equation, we can find that $t = 0.4$ hours. Lastly, $s + \frac{1}{2} = 3$ kilometers per hour, so $\frac{9}{3} + 0.4 = 3.4$ hours, or $\boxed{204}$ minutes -Failure.net The amount of hours spent while walking on the first travel is $\frac{240-t}{6}$. Thus, we have the equation $(240-t)(s) = 540$, and by the same logic, the second equation yields $(144-t)(s+2) = 540$. We have $240s-st = 540$, and $288+144s-2t-st = 540$. We subtract the two equations to get $96s+2t-288 = 0$, so we have $48s+t = 144$, so $t = 144-48s$ and now we have $\frac{240-t}{6} = 540$. The numerator of	204

Examples: AIME, GSM8K

math exams for olympiads

Reasoning-based benchmark metrics

Pass@k = "Probability that at least 1 of k attempts succeeds"

// Kind of Best-of-N methods

generate k answers and find the probability for first output generated and check with ground truth (mention based on dataset)

Reasoning-based benchmark metrics

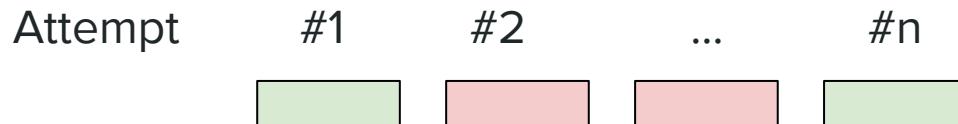
Pass@k = "Probability that at least 1 of k attempts succeeds"



Reasoning-based benchmark metrics

Pass@k = "Probability that at least 1 of k attempts succeeds"

generate n outcomes
but only $\leq c$ succeeds

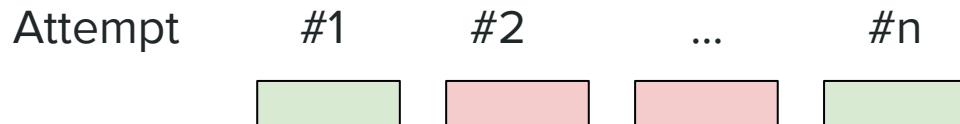


$$n = c + (n - c)$$

Successful Unsuccessful

Reasoning-based benchmark metrics

Pass@k = "Probability that at least 1 of k attempts succeeds"



$$n = c + (n - c)$$

Successful Unsuccessful

$$\begin{aligned} \text{Pass}@k &= ? \\ &= P(\text{at least 1 attempt out of } k \text{ is correct}) \\ &= 1 - P(\text{all } k \text{ attempts incorrect}) \\ &= 1 - \frac{n-c}{n} \times \frac{(n-c-1)}{n-1} \times \dots \therefore \frac{\text{total } n-c \text{ / unsuccessful}}{\text{total } n} \\ &= 1 - \frac{n-c}{n} \times \frac{(n-c-1)}{n-1} \times \dots \therefore \frac{\text{total } n-c-k}{\text{total } n} \end{aligned}$$

Reasoning-based benchmark metrics

Pass@k = "Probability that at least 1 of k attempts succeeds"

Attempt #1 #2 ... #n



$$n = c + (n - c)$$

which is a Taylor Series
based on Taylor series
Sampling without replacement

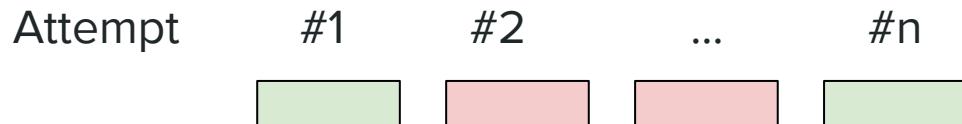
$$\text{Pass}@k = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}$$

Successful Unsuccessful
at least 1 of
 $\binom{k}{n}$ is correct out of
 n -sample

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Reasoning-based benchmark metrics

Pass@k = "Probability that at least 1 of k attempts succeeds"



$$n = c + (n - c)$$

↓ ↓
Successful Unsuccessful

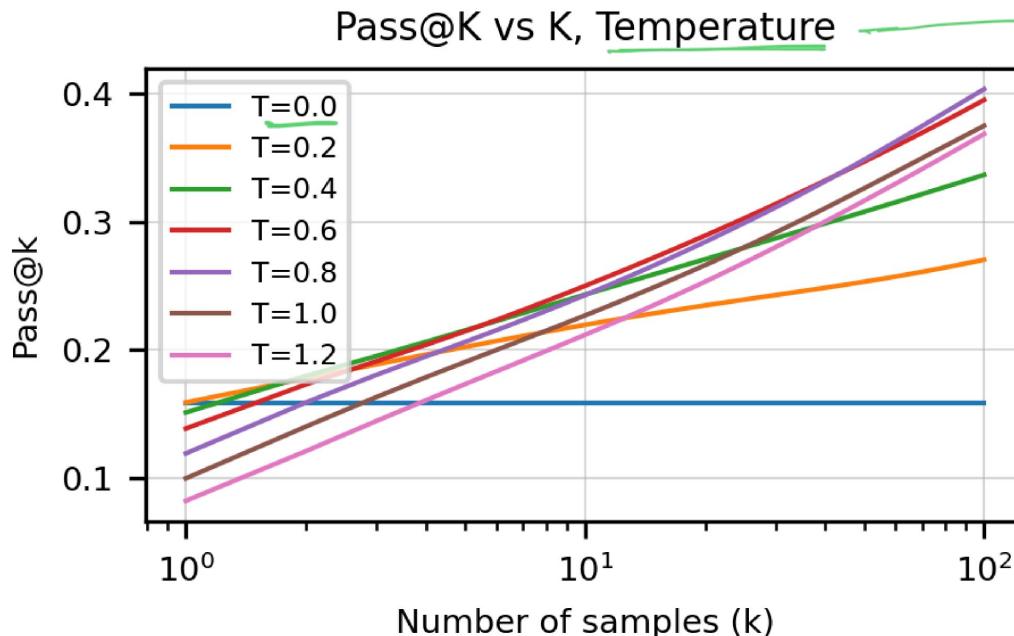
Special
case

$$\text{Pass}@1 = \frac{c}{n}$$

↑
single attempt or finding successful attempts.

Reasoning-based benchmark metrics

Pass@k = "Probability that at least 1 of k attempts succeeds"



↓ temperature : solution will be good but not diverse
↑ temperature : diverse o/P ; harm performance
∴ need to find optimal temperature which would be having better performance with diverse o/P
∴ It calculates based on Empirical

Reasoning-based benchmark metrics

- **Pass@k.** For use cases where checking is easy / can afford higher latency
- **Pass@1.** For use cases where we care about a single generation

Reasoning-based benchmark metrics

- **Pass@k.** For use cases where checking is easy / can afford higher latency
- **Pass@1.** For use cases where we care about a single generation
- **Cons@k.** "Consensus at k", equivalent to comparing the answer from majority voting with the ground truth *↓ based on majority votes*



Transformers & Large Language Models

Reasoning models

Scaling with RL

GRPO

Applications

Develop "test-time scaling"

Idea. Incentivize model to reason before answering.



writing reasoning chain is challenging and needic
so, SFT helps but need quality data & it
has be written by hand
&, it's difficult to have human-written reasoning

Develop "test-time scaling"

Idea. Incentivize model to reason before answering.

Considerations.

- ① • Reasoning chain is hard to write from scratch (**SFT data by hand impractical**)

Develop "test-time scaling"

Idea. Incentivize model to reason before answering.

Considerations.

- Reasoning chain is hard to write from scratch (**SFT data by hand impractical**)
- Don't want to limit the model to human-written reasoning



Develop "test-time scaling"

Idea. Incentivize model to reason before answering.

Considerations.

- Reasoning chain is hard to write from scratch (**SFT data by hand impractical**)
- Don't want to limit the model to human-written reasoning
- Natural verifiable reward ("did it solve the problem?" → "yes" or "no")

3

↓
Like for coding - test cases
math problem - ground truth.

Develop "test-time scaling"

Idea. Incentivize model to reason before answering.

Considerations.

- Reasoning chain is hard to write from scratch (**SFT data by hand impractical**)
- Don't want to limit the model to human-written reasoning
- Natural verifiable reward ("did it solve the problem?" → "yes" or "no")

Let's try RL!



to fetch chain of thoughts i.e "a reasoning chain"

Reward 1: verify that CoT is there

```
<think>  
    :  
</think>  
ANSWER
```

Template

Reward 1: verify that CoT is there

<think>

⋮

</think>

ANSWER

Template

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 ⋯

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

⋯

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 ⋯

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: ⋯

⋯

Sample response

Reward 1: verify that CoT is there

<think>

⋮

</think>

ANSWER

Template

↓ reasoning chain.

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>

①

To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both ⋯

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

⋯

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 ⋯

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

③

First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

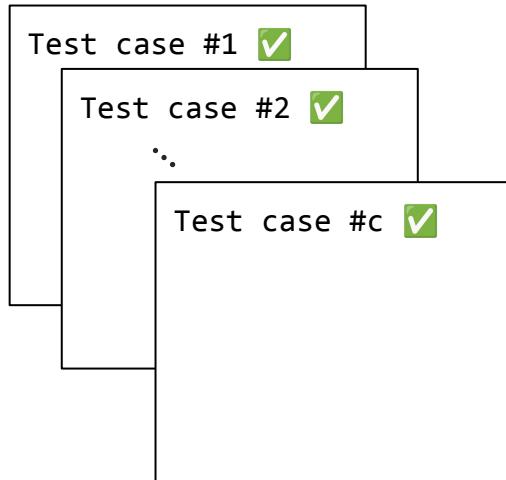
④

Next, I could square both sides again, treating the equation: ⋯

⋯

Sample response

Reward 2: verify that solution is correct

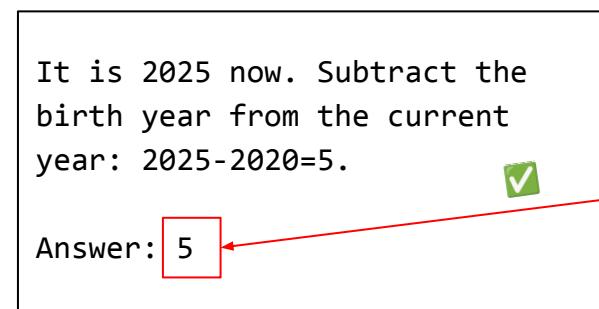


Code verification

Reward 2: verify that solution is correct



Code verification



Math verification

Run RL on "verifiable" rewards

- ① **formatting**
(think delimiters?)

Run RL on "verifiable" rewards

① **formatting**
(think delimiters?)

② **accuracy**
(correct solution?)

Run RL on "verifiable" rewards

Rewards

- ||
- ① **formatting**
(think delimiters?)
- +
- ② **accuracy**
(correct solution?)
- // validation
either ground Truth for math
or test case for coding)*
- delimiters which generally
being annotated for training
i.e. let's start, next, start, hence, poon
etc...*

Run RL on "verifiable" rewards

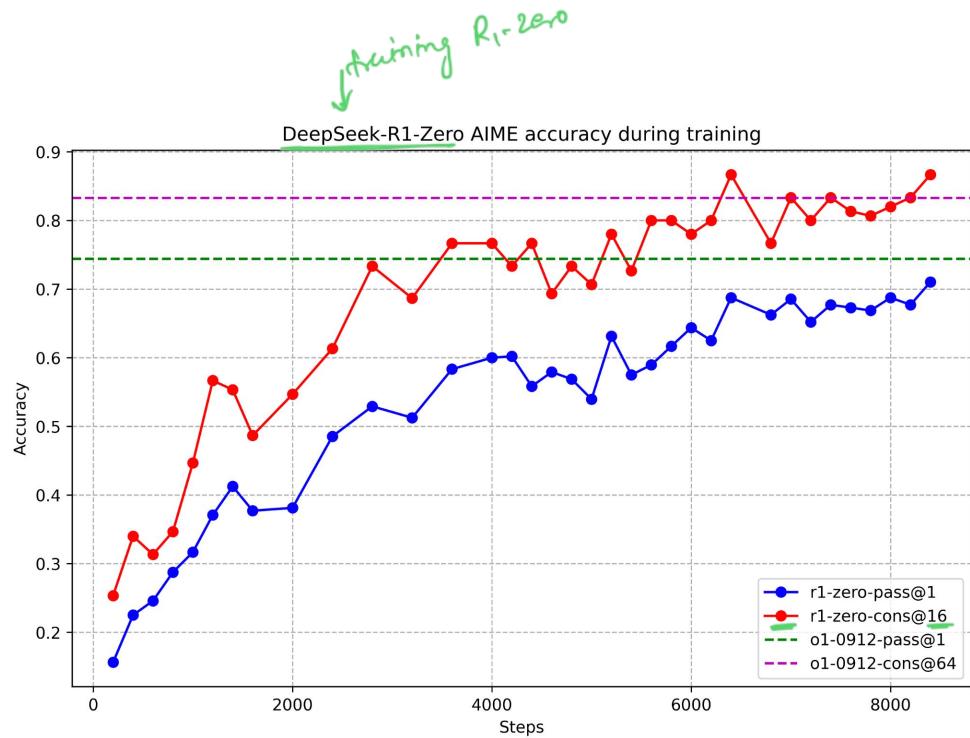
Rewards

||

1 formatting
(think delimiters?)

+

2 accuracy
(correct solution?)



Control thinking at inference time

Problem. Not all prompts are equal

Some prompts doesn't need to think much
or some does much.

Control thinking at inference time

Problem. Not all prompts are equal

Ideas to control "thinking".

- ✓ Dynamic budget

Control thinking at inference time

Problem. Not all prompts are equal

Ideas to control "thinking".

- Dynamic budget
- Context awareness → *to think, model should have content awareness i.e. how much need to think.*

Control thinking at inference time

Problem. Not all prompts are equal

Ideas to control "thinking".

- Dynamic budget
- Context awareness
- Budget forcing *→ Limited budget to think.
ee time is up for thinking'*

Control thinking at inference time

Problem. Not all prompts are equal

Ideas to control "thinking".

- Dynamic budget
- Context awareness
- Budget forcing
- ✓ "Continuous" thoughts

model can also think in
other direction like \rightarrow hidden state
or more deviate from content \Rightarrow



Transformers & Large Language Models

Reasoning models

Scaling with RL

GRPO

Applications

↑
Incentivize RL model for using
satiating much for accurate
ensure it.

group Relative Policy
Optimized

Common RL algorithm for reasoning

GRPO = Group Relative Policy Optimization

Common RL algorithm for reasoning

GRPO = **G**roup **R**elative **P**olicy **O**ptimization

$$\mathcal{L}(\theta) = \boxed{\text{Maximize advantages}} + \boxed{\text{Don't deviate too much from old / base model}}$$

*↓ from Previous Iteration of
base model*

Common RL algorithm for reasoning

GRPO = **G**roup **R**elative **P**olicy **O**ptimization

$$\mathcal{L}(\theta) = \boxed{\text{Maximize advantages}} + \boxed{\text{Don't deviate too much from old / base model}}$$

$$\text{Advantage} \sim \text{Reward} - \text{Avg}(\text{reward of group})$$

"tells how good the completion is comparing each reward to average of other one in group"

Completion of reward

taking average reward for each group.
i.e. for a math problem, generating multiple completions and their completions will have revised by see how much better current reward in comparison to average of group

Common RL algorithm for reasoning

GRPO = Group Relative Policy Optimization

$$\mathcal{L}(\theta) = \boxed{\text{Maximize advantages}} + \boxed{\text{Don't deviate too much from old / base model}}$$

Advantage ~ Reward - **Avg(reward of group)**

Big difference compared to PPO!

The diagram illustrates the GRPO loss function $\mathcal{L}(\theta)$ as a sum of two components. The first component, 'Maximize advantages', is enclosed in a dashed box. The second component, 'Don't deviate too much from old / base model', is also enclosed in a dashed box. Below the first component, the 'Advantage' formula is shown: 'Advantage' is approximately equal to 'Reward' minus 'Avg(reward of group)'. A vertical arrow points from this formula down to the text 'Big difference compared to PPO!', which is underlined in green.

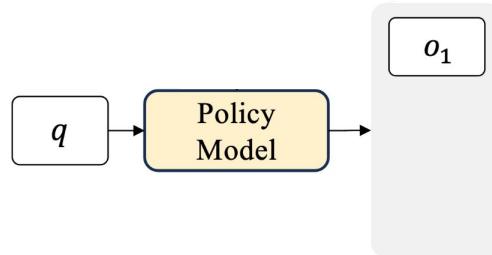
Comparison between GRPO and PPO

GRPO

q

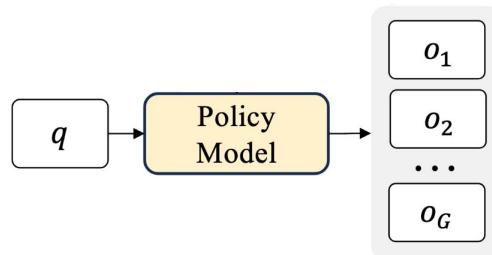
Comparison between GRPO and PPO

GRPO



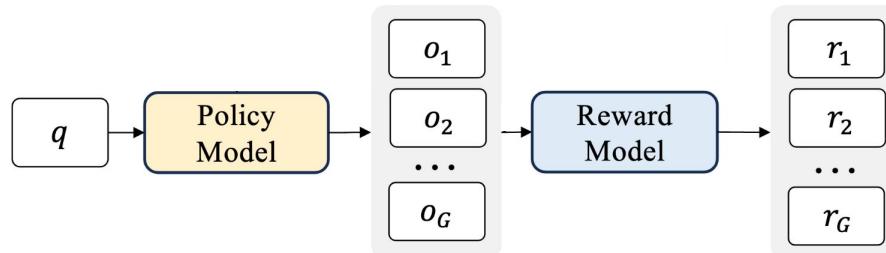
Comparison between GRPO and PPO

GRPO



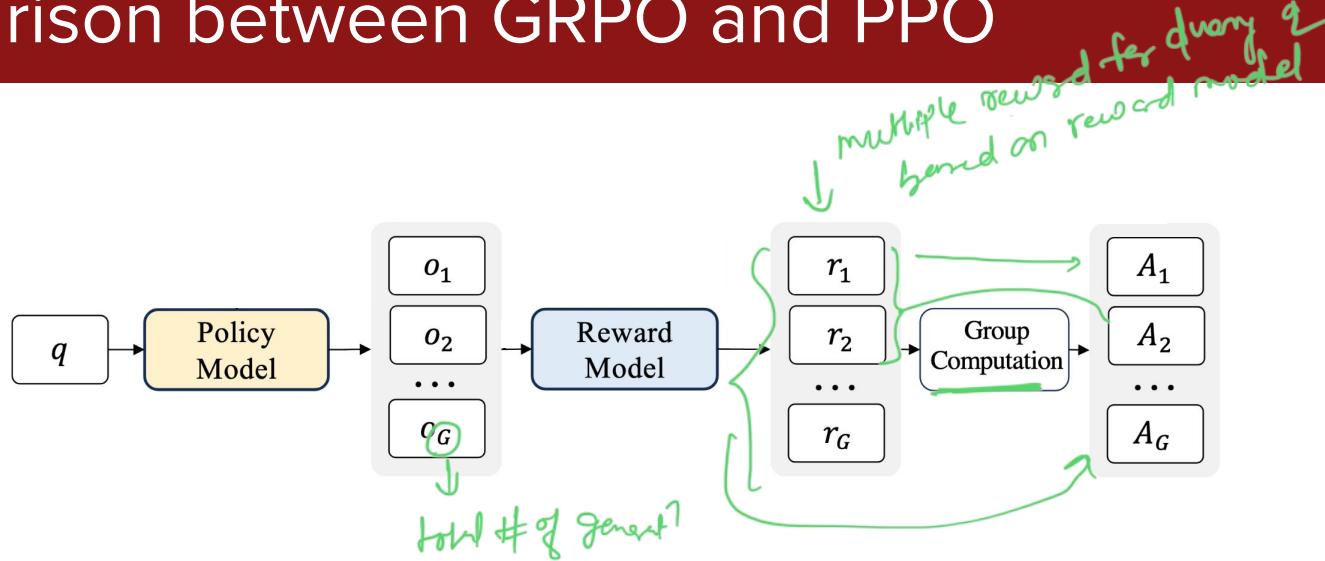
Comparison between GRPO and PPO

GRPO



Comparison between GRPO and PPO

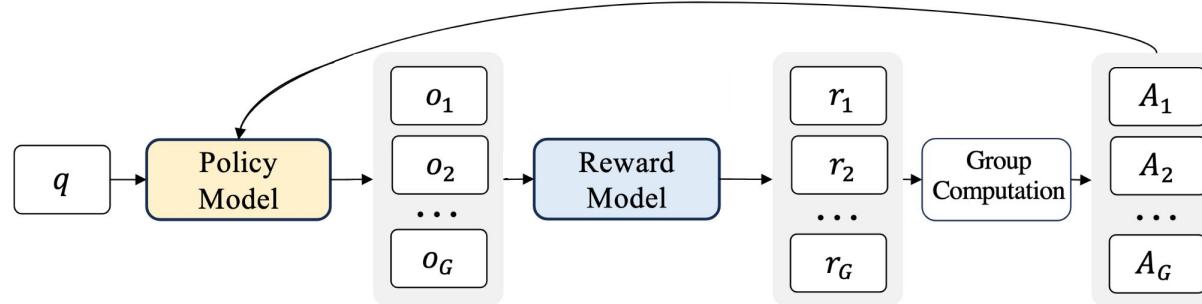
GRPO



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

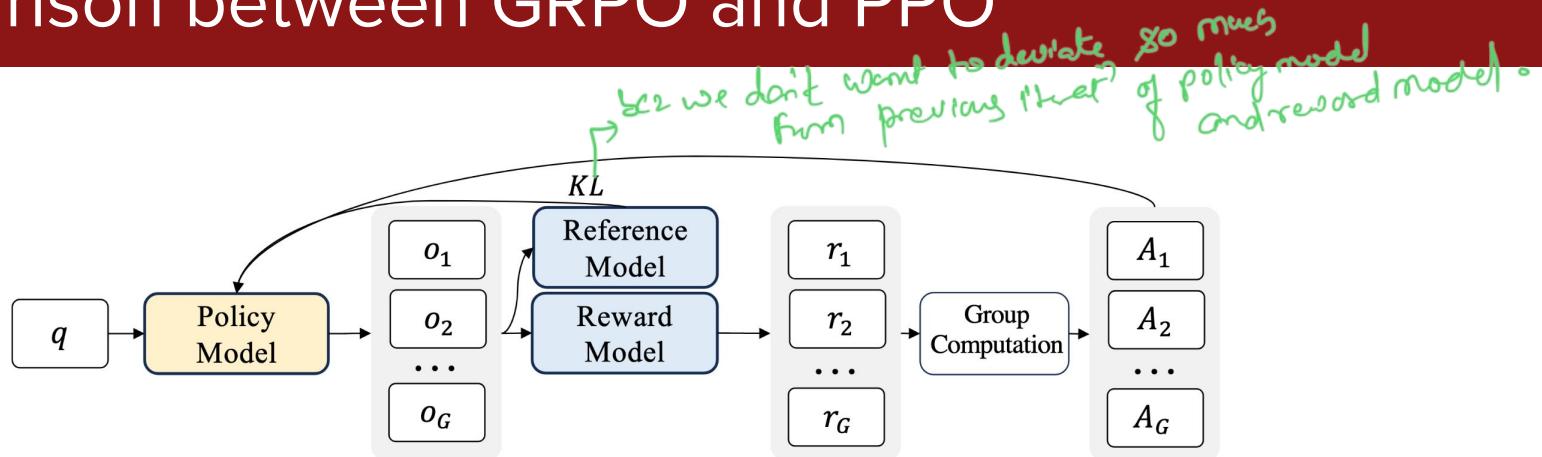
Comparison between GRPO and PPO

GRPO



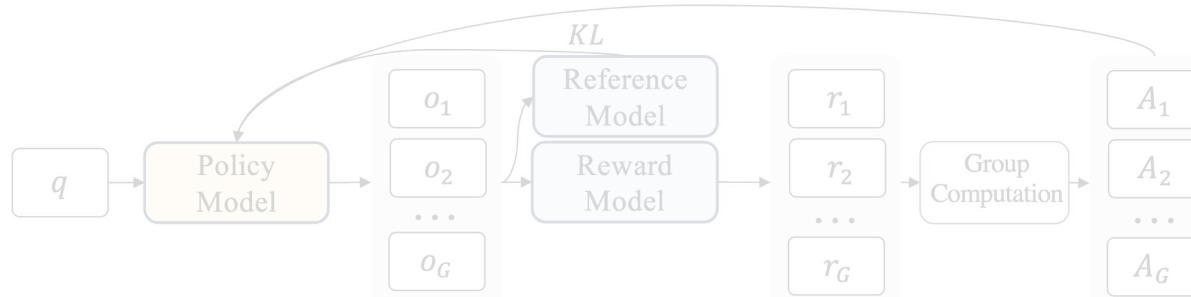
Comparison between GRPO and PPO

GRPO

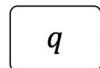


Comparison between GRPO and PPO

GRPO

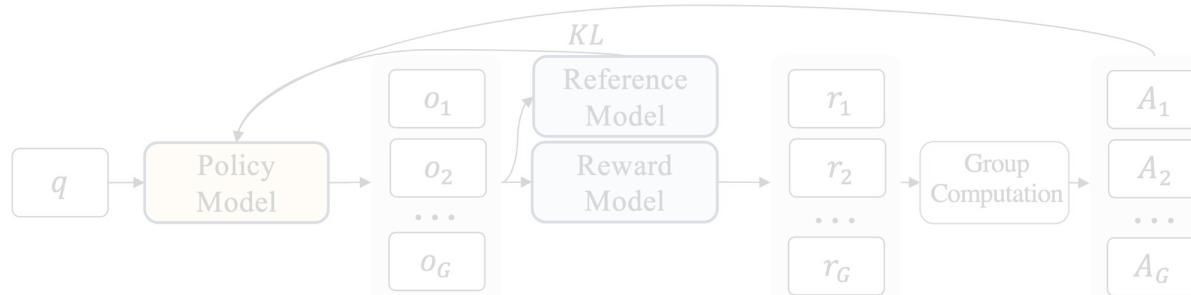


PPO

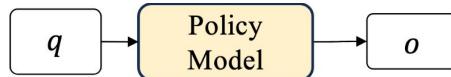


Comparison between GRPO and PPO

GRPO

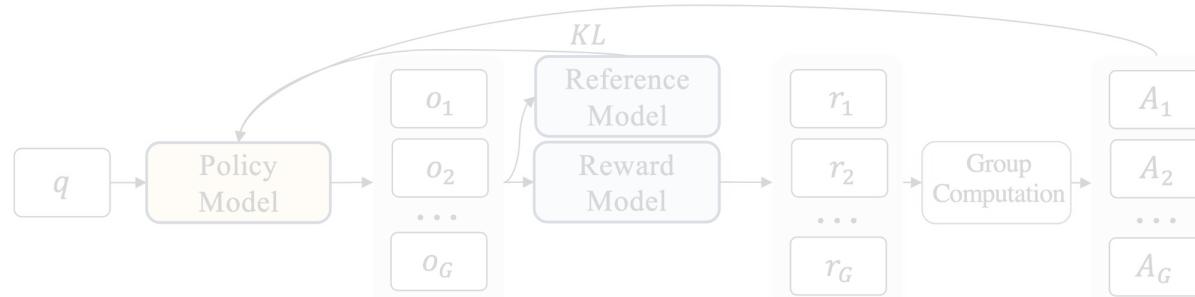


PPO

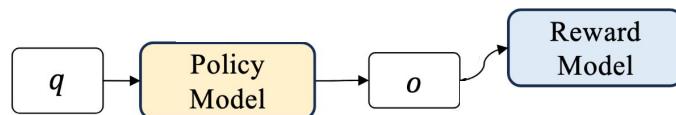


Comparison between GRPO and PPO

GRPO

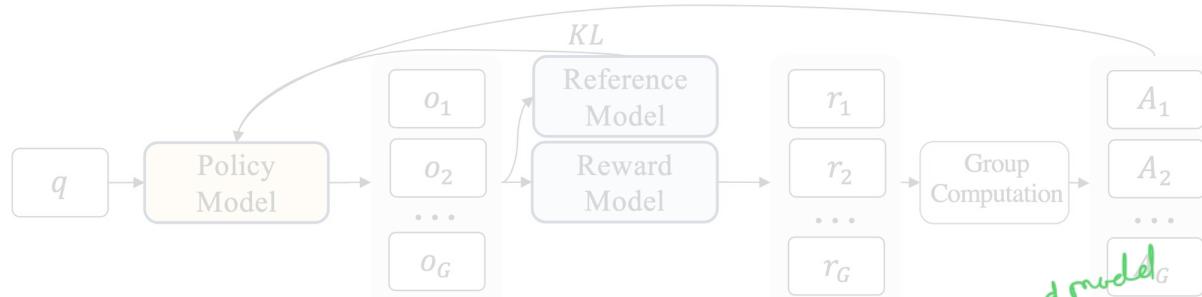


PPO



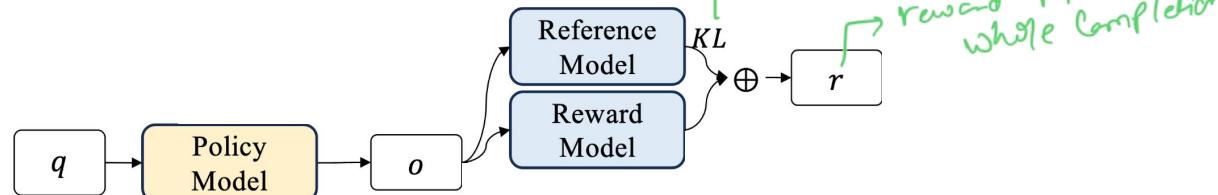
Comparison between GRPO and PPO

GRPO



Comparing the reward model
with all

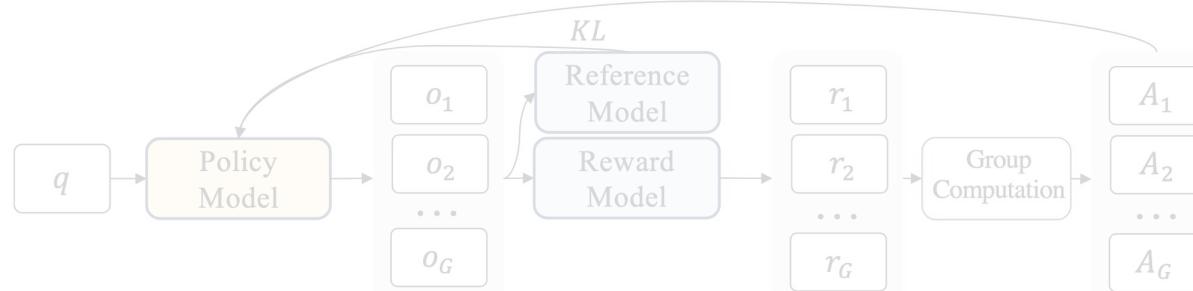
PPO



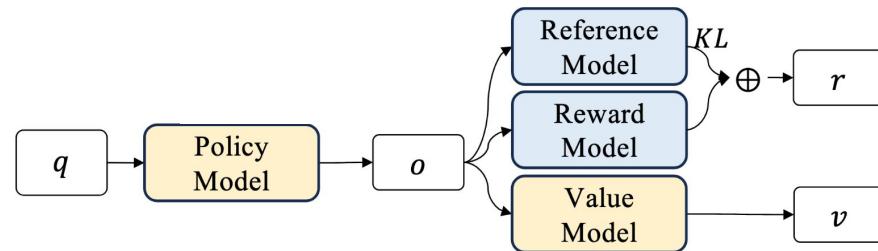
reward for
whole completion.

Comparison between GRPO and PPO

GRPO

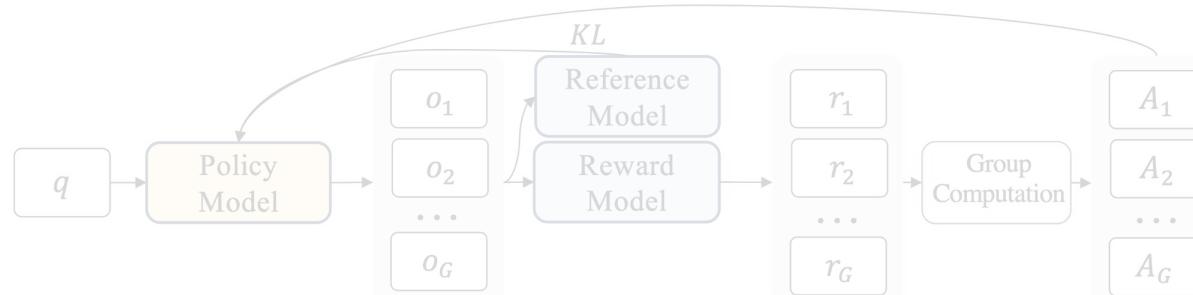


PPO

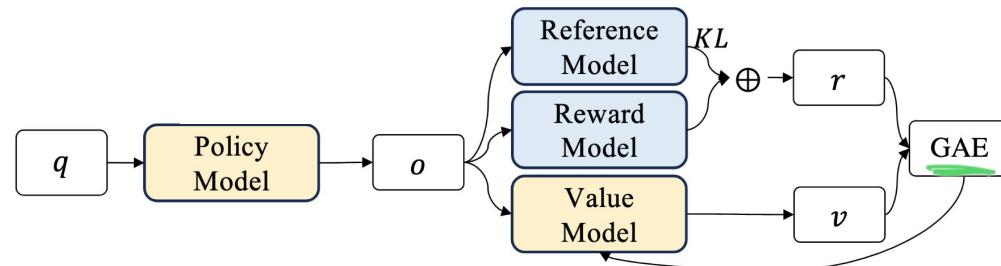


Comparison between GRPO and PPO

GRPO

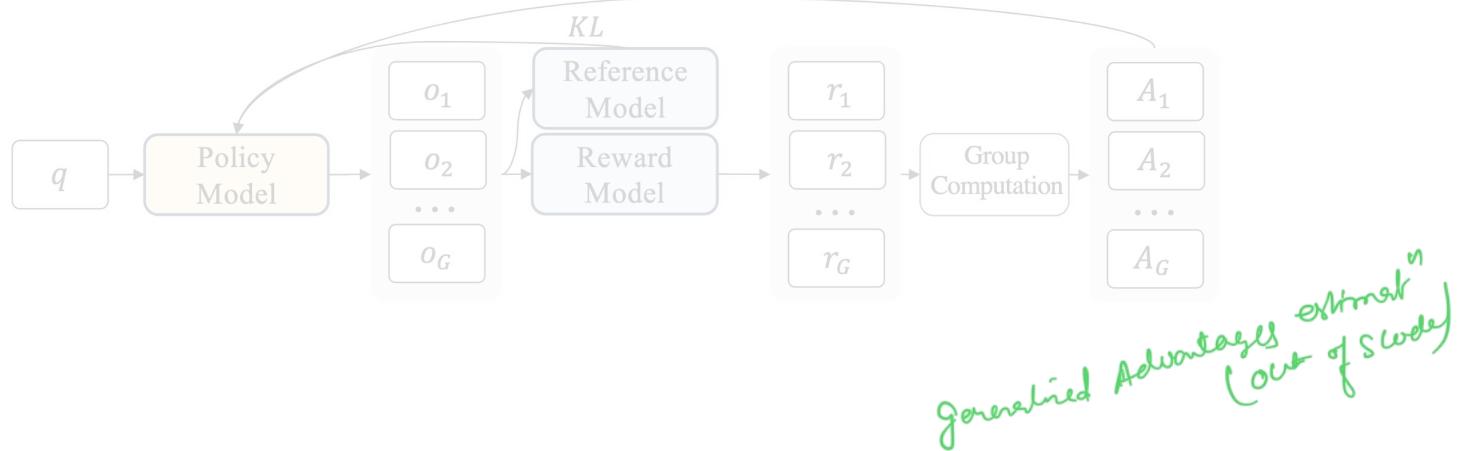


PPO

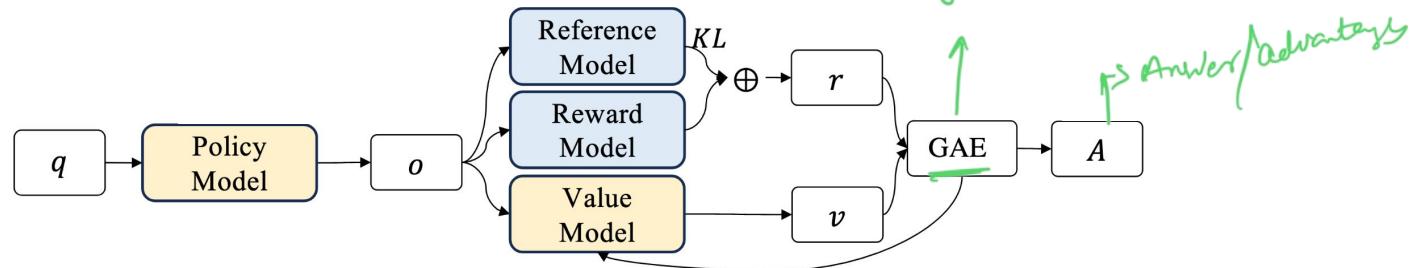


Comparison between GRPO and PPO

GRPO

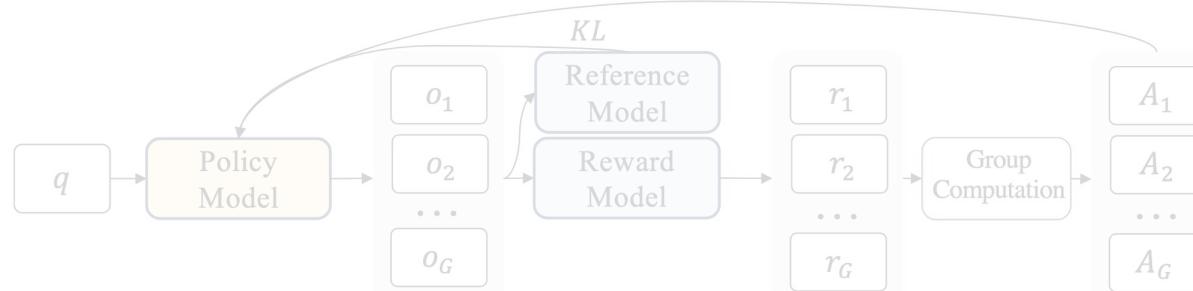


PPO

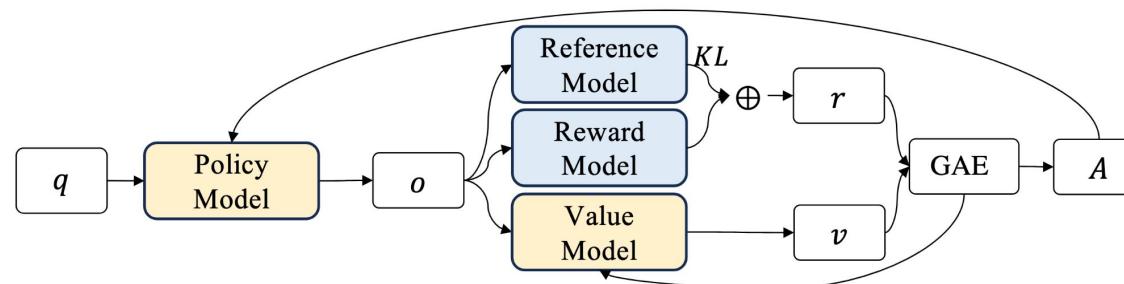


Comparison between GRPO and PPO

GRPO

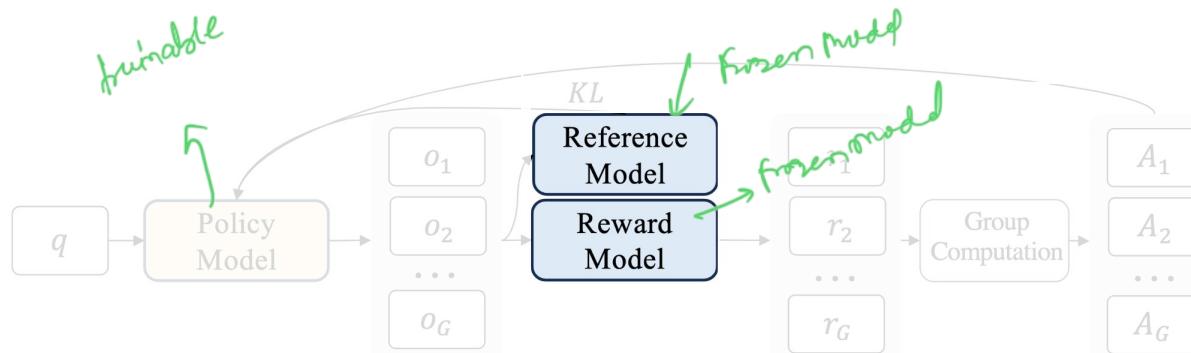


PPO

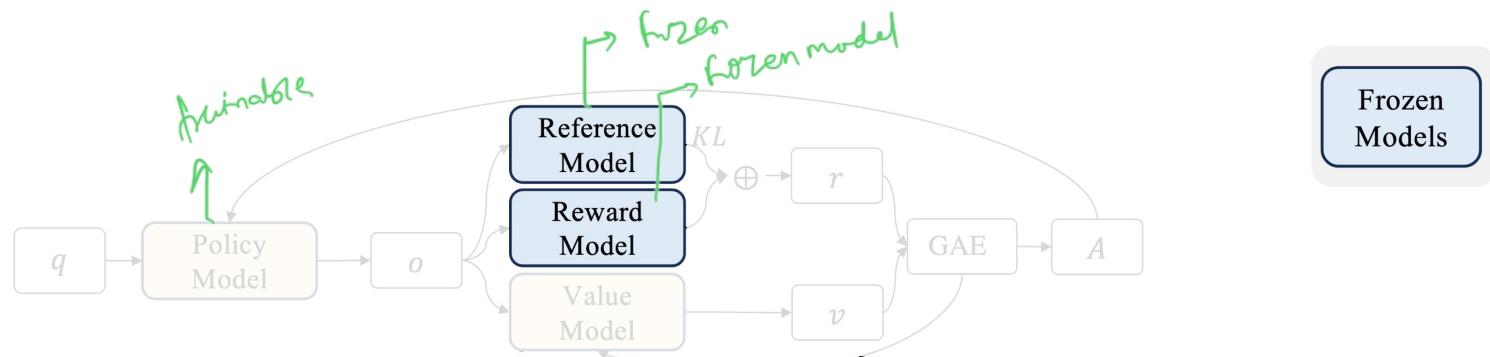


Comparison between GRPO and PPO

GRPO



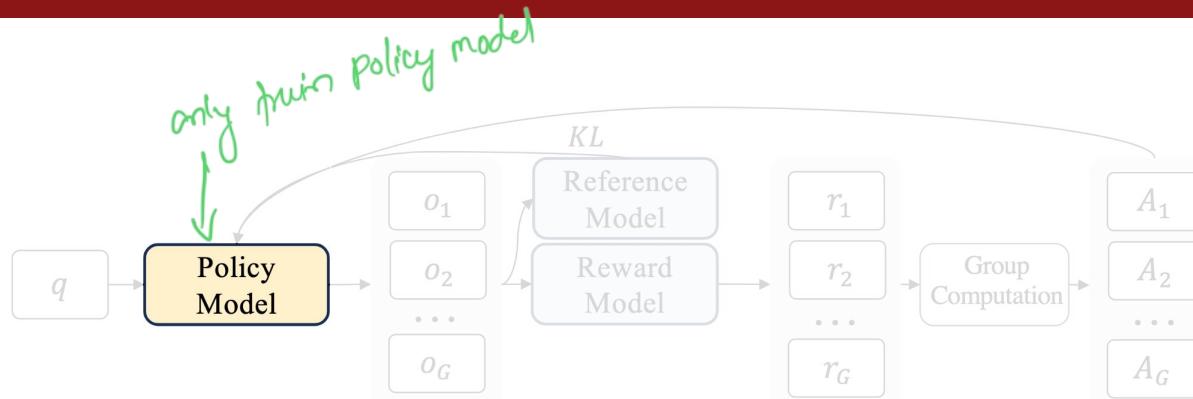
PPO



Frozen
Models

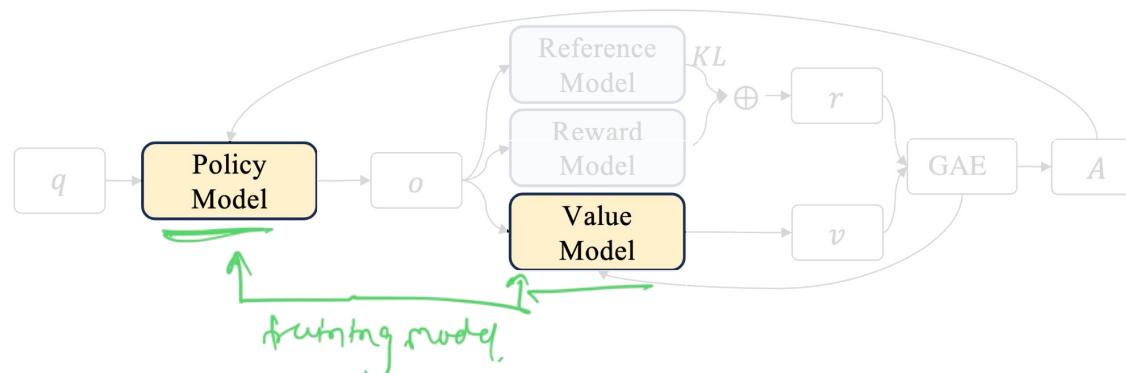
Comparison between GRPO and PPO

GRPO



only train policy model

PPO



Comparison between GRPO and PPO

GRPO

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

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

PPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right]$$

Comparison between GRPO and PPO

Similarities. Ratio

GRPO

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

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

PPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right]$$

Comparison between GRPO and PPO

Similarities. Ratio, clipping

GRPO

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

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}$$

PPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \epsilon, 1 + \epsilon \right) A_t \right]$$

Comparison between GRPO and PPO

Differences. KL penalty

GRPO

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

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

PPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} \textcolor{green}{A_t}, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) \textcolor{green}{A_t} \right]$$

Comparison between GRPO and PPO

Differences. KL penalty, advantage estimation

GRPO — used for reasoning based problem training

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

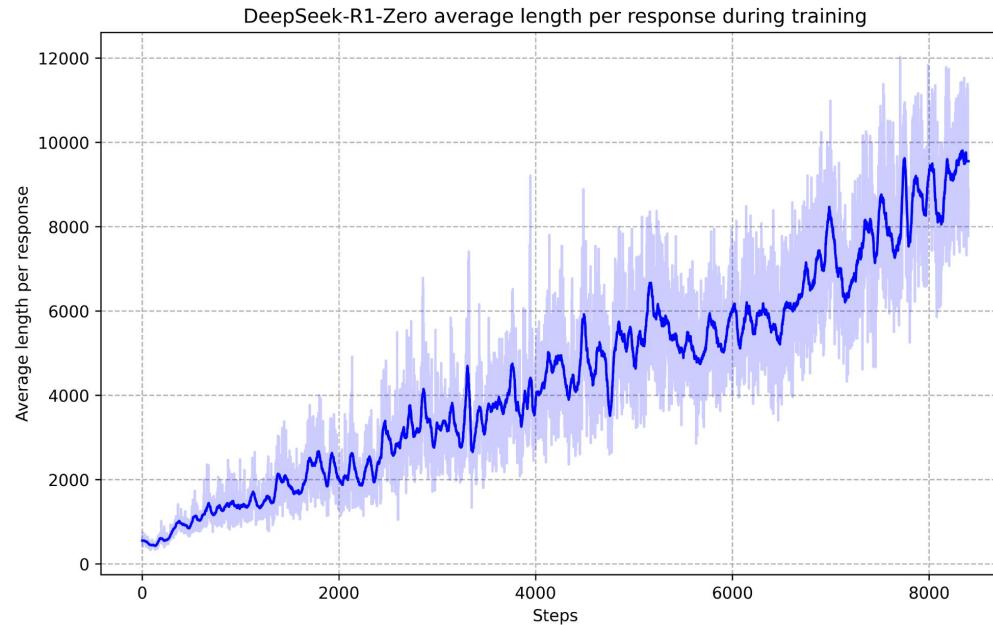
$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1-\varepsilon, 1+\varepsilon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

PPO → Heavily used for Preference training

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right]$$

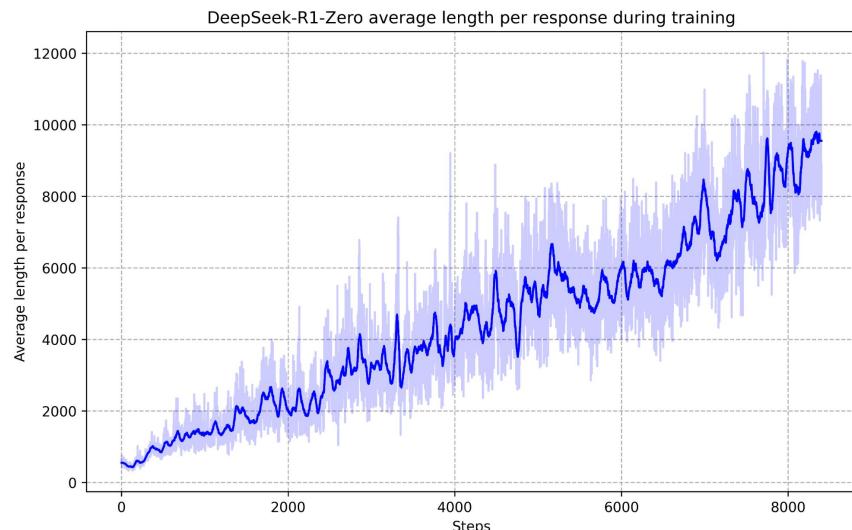
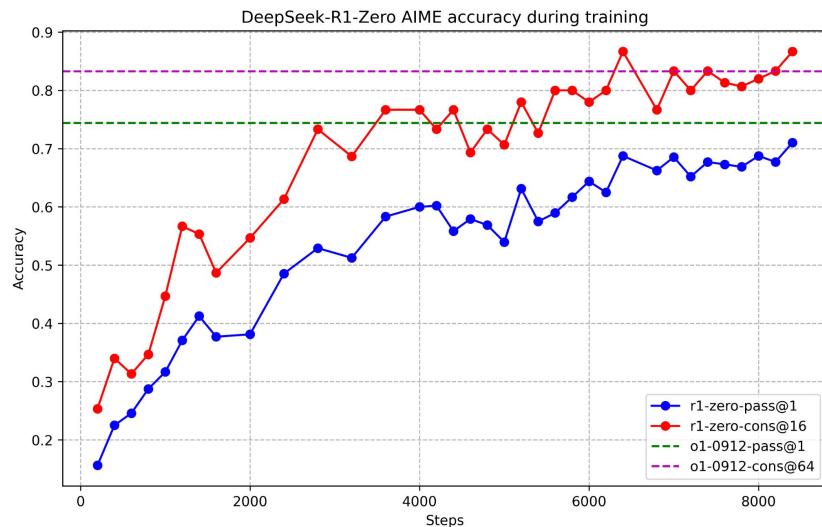
"Increasing output length" phenomenon

Observation. Response length keeps on increasing with RL training



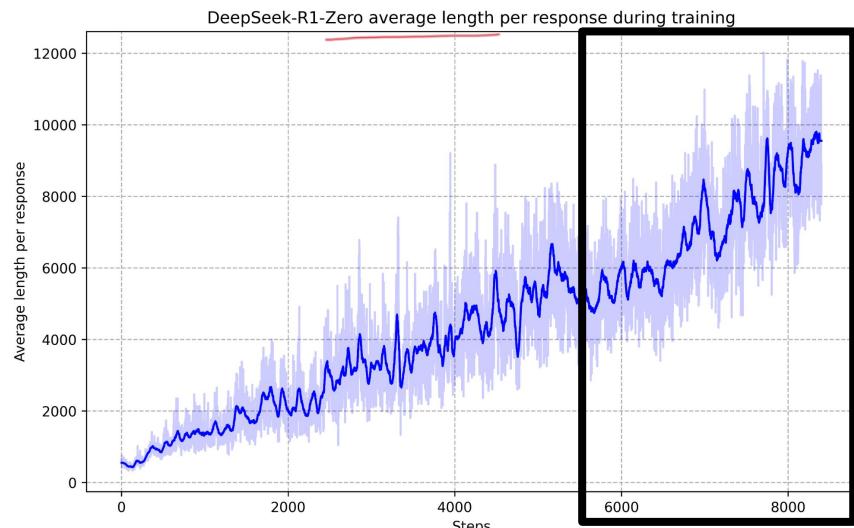
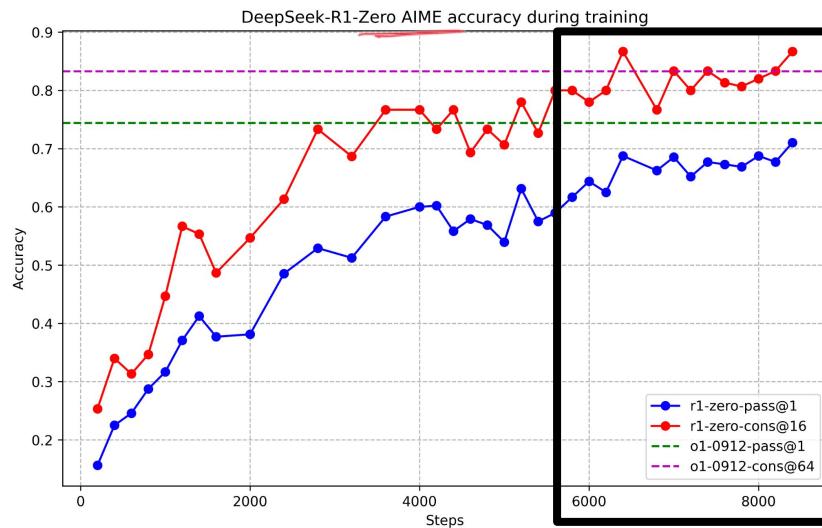
"Increasing output length" phenomenon

Observation. Response length keeps on increasing with RL training



"Increasing output length" phenomenon

Observation. Response length keeps on increasing with RL training



"Increasing output length" phenomenon

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

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

w.r.t
goal
token length

if $o_i \rightarrow$ high
then $|o_i|$ is small
i.e. shorter sentence will have
higher weight
and longer sentence will have
lower weight

"Increasing output length" phenomenon

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

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

"Increasing output length" phenomenon

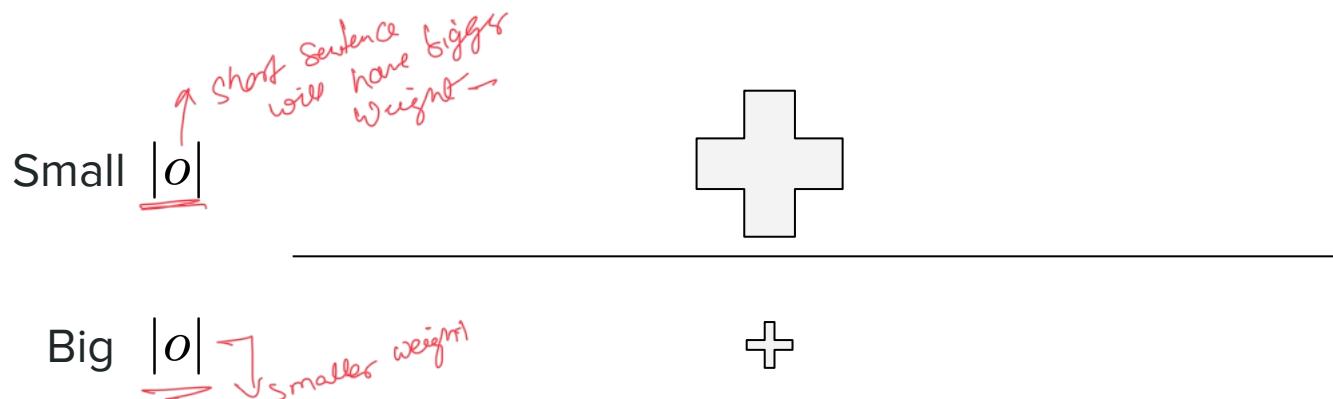
$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \boxed{\frac{1}{|o_i|}} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}$$

"Increasing output length" phenomenon

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

$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \boxed{\frac{1}{|o_i|}} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}$$

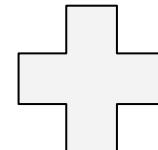


"Increasing output length" phenomenon

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

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

Short output



Long output



"Increasing output length" phenomenon

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

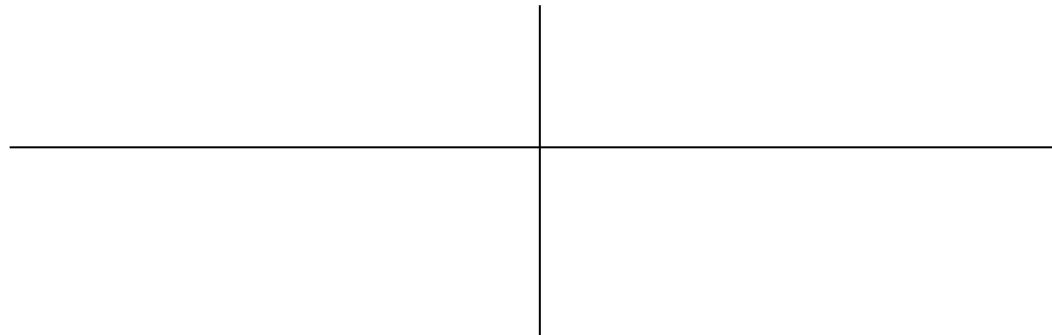
$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \left[\min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t}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

$$A > 0$$

$$A < 0$$

Short output

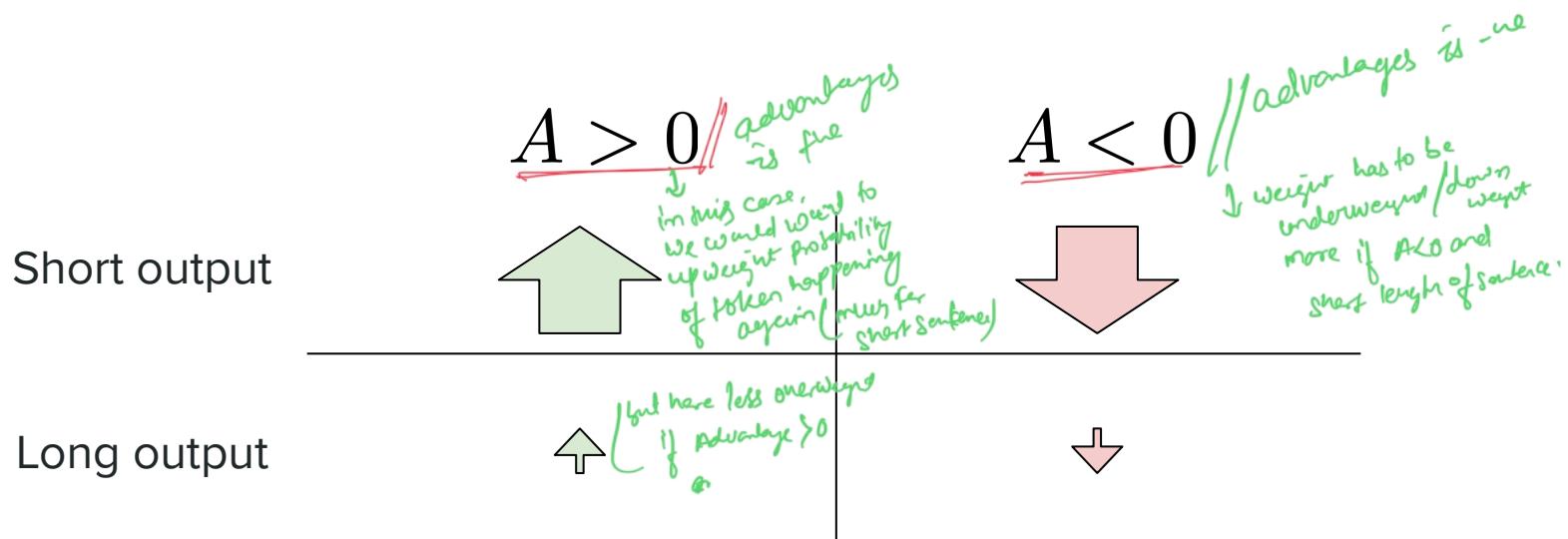
Long output



"Increasing output length" phenomenon

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

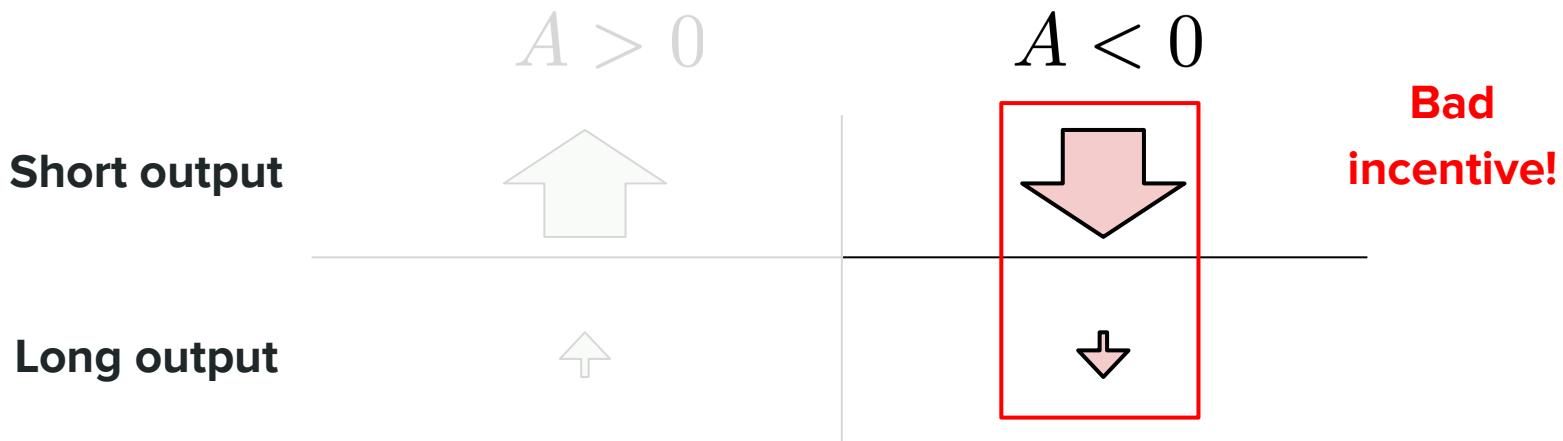
$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \boxed{\frac{1}{|o_i|}} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1-\varepsilon, 1+\varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}$$



"Increasing output length" phenomenon

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

$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \boxed{\frac{1}{|o_i|}} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t}, \text{clip} \left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1 - \varepsilon, 1 + \varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL} [\pi_\theta || \pi_{ref}] \right\}$$



Mitigating "increasing length" phenomenon

Problem.

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{t=1}^{|\mathbf{o}_i|}$$

do something for this for
taking advantage of short sentence / longer sentence
with advantages.

Mitigating "increasing length" phenomenon

Problem. $\frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{t=1}^{|\mathbf{o}_i|}$

Remedy. Equalize token-level contributions

Mitigating "increasing length" phenomenon

Problem. $\frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{t=1}^{|\mathbf{o}_i|}$

Remedy. Equalize token-level contributions

- DAPO $\frac{1}{\sum_{i=1}^G |\mathbf{o}_i|} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$

Mitigating "increasing length" phenomenon

Problem.

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{t=1}^{|\mathbf{o}_i|}$$

Remedy. Equalize token-level contributions

- DAPO $\frac{1}{\sum_{i=1}^G |\mathbf{o}_i|} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$

- Dr. GRPO $\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$

Mitigating "increasing length" phenomenon

Problem.

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{t=1}^{|\mathbf{o}_i|}$$

↓ removing it

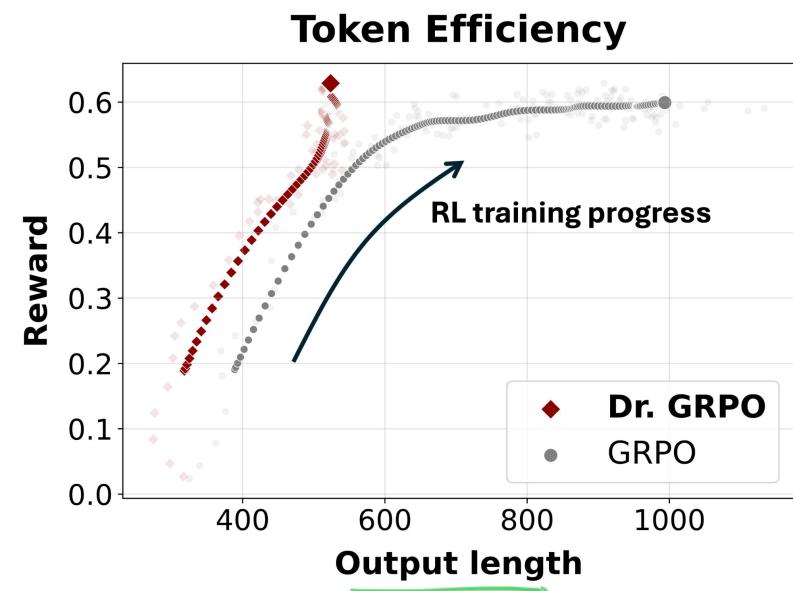
Remedy. Equalize token-level contributions

- DAPO

$$\frac{1}{\sum_{i=1}^G |\mathbf{o}_i|} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$$

- Dr. GRPO

$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$$



Mitigating "increasing length" phenomenon

Problem.

$$\frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}_i|} \sum_{t=1}^{|\mathbf{o}_i|}$$

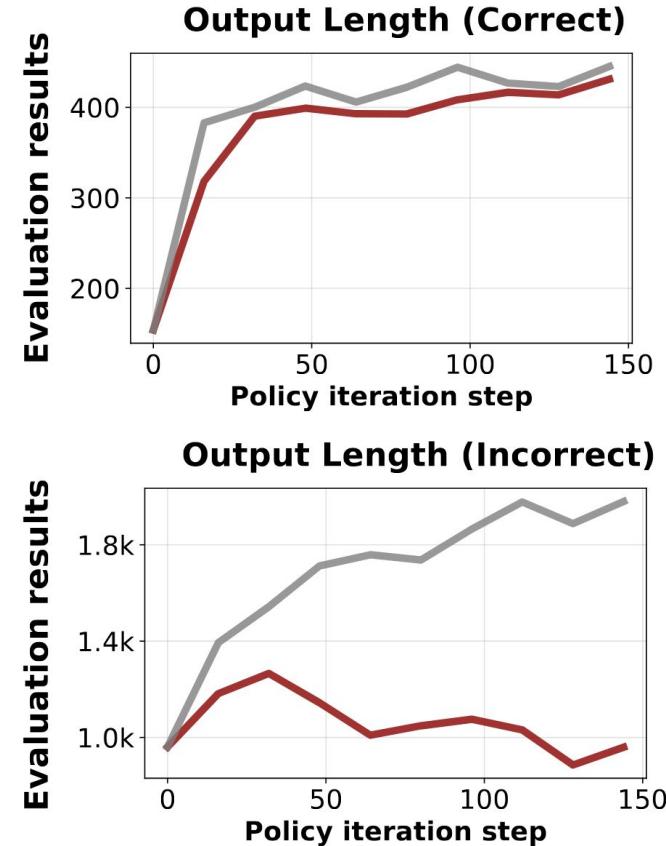
Remedy. Equalize token-level contributions

- DAPO

$$\frac{1}{\sum_{i=1}^G |\mathbf{o}_i|} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$$

- Dr. GRPO

$$\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|\mathbf{o}_i|}$$



Exploration of other adjustments

- Bias linked to level of difficulty //modificⁿ 1

$$\hat{A}_{i,t} = \frac{R(\mathbf{q}, \mathbf{o}_i) - \text{mean}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}{\text{std}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})} \leftarrow \text{removing it in modificⁿ 1}$$

Exploration of other adjustments

- Bias linked to **level of difficulty**

$$\hat{A}_{i,t} = \frac{R(\mathbf{q}, \mathbf{o}_i) - \text{mean}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}{\text{std}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}$$

- Encourage **diversity**

E-influences for policy iteration

$$\text{clip}\left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon\right) \longrightarrow \text{clip}\left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}}\right)$$

$$1 - \varepsilon \leq \frac{\pi}{\pi_{\text{old}}} \leq 1 + \varepsilon$$

$$\pi \leq (1 + \varepsilon) \pi_{\text{old}}$$

Deriving ε in lower and higher bounds.

Exploration of other adjustments

- Bias linked to **level of difficulty**

$$\hat{A}_{i,t} = \frac{R(\mathbf{q}, \mathbf{o}_i) - \text{mean}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}{\text{std}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}$$

- Encourage **diversity**

$$\text{clip}\left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon\right) \longrightarrow \text{clip}\left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}}\right)$$

Exploration of other adjustments

- Bias linked to **level of difficulty**

$$\hat{A}_{i,t} = \frac{R(\mathbf{q}, \mathbf{o}_i) - \text{mean}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}{\text{std}(\{R(\mathbf{q}, \mathbf{o}_1), \dots, R(\mathbf{q}, \mathbf{o}_G)\})}$$

- Encourage **diversity**

$$\text{clip}\left(r_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon\right) \longrightarrow \text{clip}\left(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}}\right)$$

...among others!



Transformers & Large Language Models

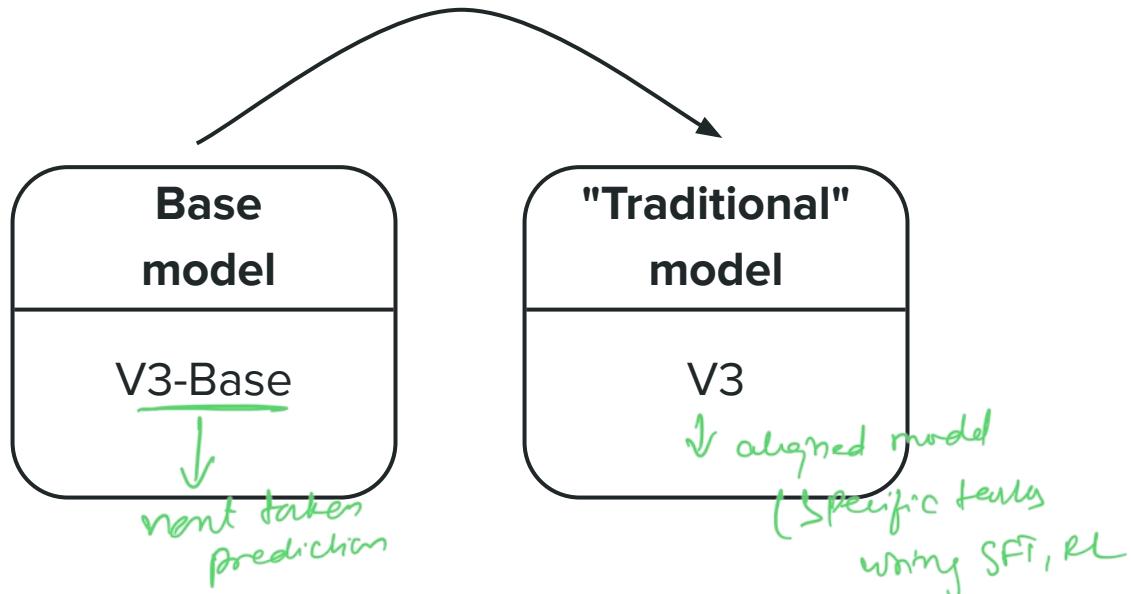
Reasoning models

Scaling with RL

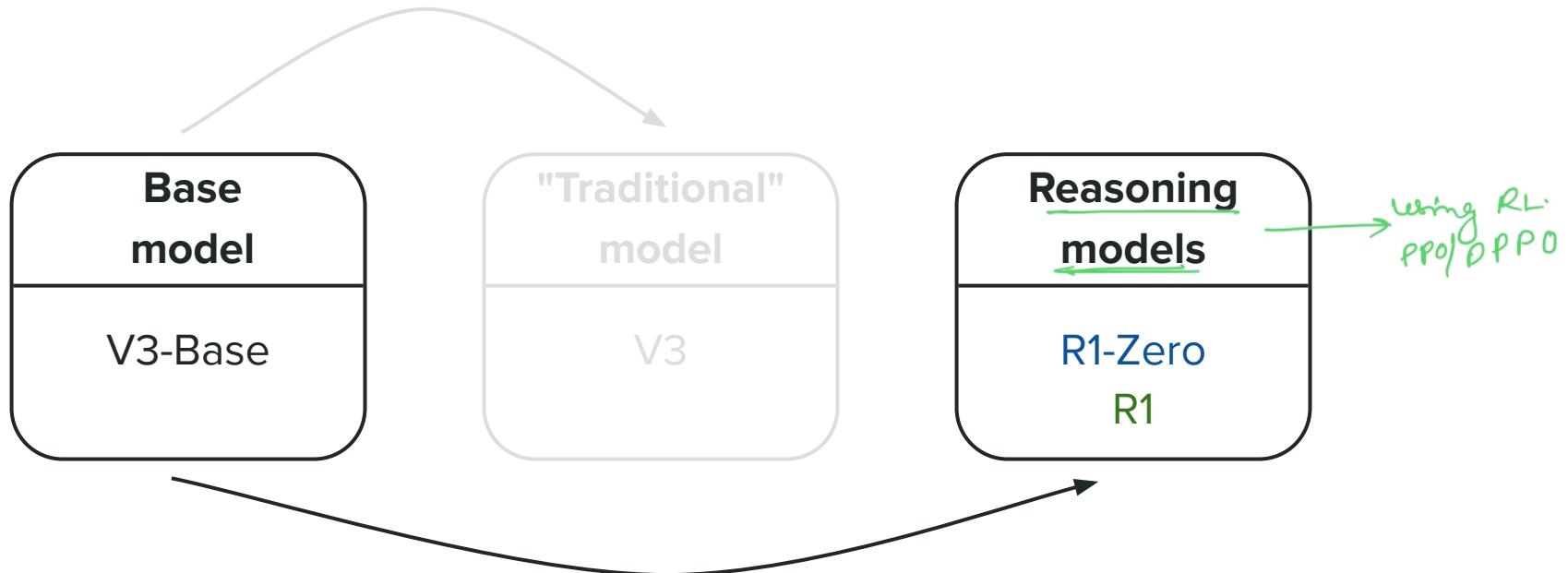
GRPO

Applications

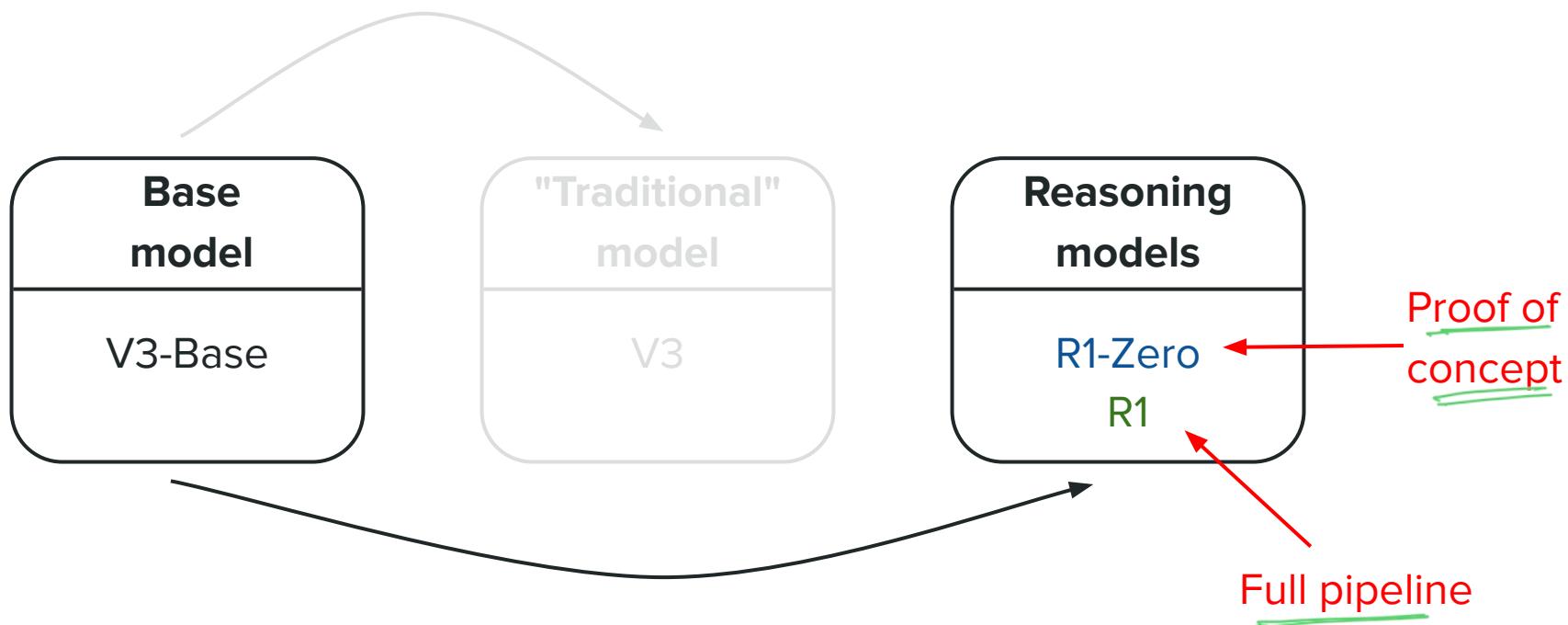
Stitching it all together



Stitching it all together



Stitching it all together



DeepSeek R1-Zero's training recipe

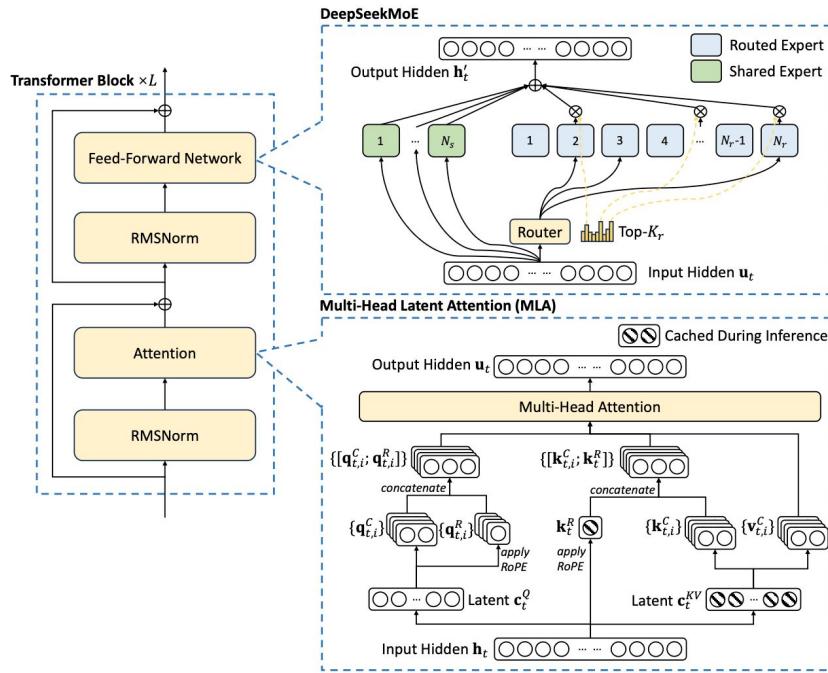
DeepSeek R1-Zero's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**

V3-Base
↑
Training on "next token prediction"
[pure model, No SFT, RL
or finetuned
for aligned
task.]

DeepSeek R1-Zero's training recipe

1 Pretrain model with "traditional" techniques: **V3-Base**



MoE [based on "mixture of Experts"]
[multi-latent attention] ↓ next token prediction

~671B total, ~37B active

[Total tokens are being fetched in Vormela model]

[These many tokens can be used for prefetching/inference]

DeepSeek R1-Zero's training recipe

① Pretrain model with "traditional" techniques: **V3-Base**

② GRPO with reasoning data: **R1-Zero**

↓
group Relative Policy options
Converging data based on
human-level ground truths
validate the function
(for coding: test cases)
math = ground truth

DeepSeek R1-Zero's training recipe

1 Pretrain model with "traditional" techniques: **V3-Base**

2 GRPO with reasoning data: **R1-Zero**

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>.

User: <this placeholder is replaced by a reasoning query>

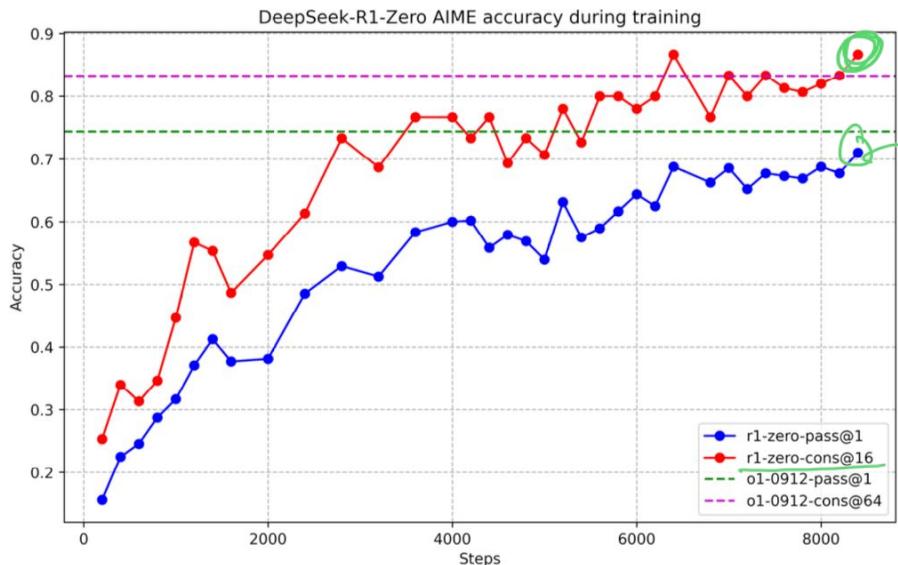
Assistant:

Template

DeepSeek R1-Zero's training recipe

1 Pretrain model with "traditional" techniques: **V3-Base**

2 GRPO with reasoning data: **R1-Zero**



Consensus of $K \rightarrow u = 16$
i.e. Warming based
on majority
of votes.

↓
PA8:

DeepSeek R1-Zero's training recipe

1 Pretrain model with "traditional" techniques: **V3-Base**

2 GRPO with reasoning data: **R1-Zero**

Benefits	Challenges
Reasoning abilities without any SFT ✓	<p>Chains of reasoning have formatting and readability issues</p> 

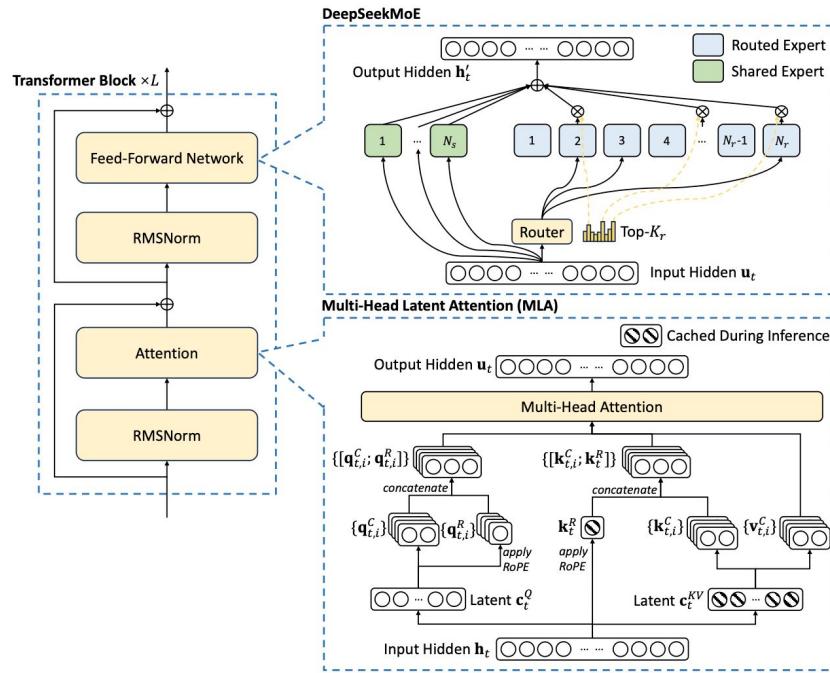
DeepSeek R1's training recipe

Figure from "DeepSeek-V3 Technical Report", DeepSeek-AI, 2024.

"DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", DeepSeek-AI, 2025.

DeepSeek R1's training recipe

1 Pretrain model with "traditional" techniques: V3-Base



MoE

~671B total, ~37B active

Figure from "DeepSeek-V3 Technical Report", DeepSeek-AI, 2024.

"DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", DeepSeek-AI, 2025.

DeepSeek R1's training recipe

1 Pretrain model with "traditional" techniques: **V3-Base**

2 "Small-scale" SFT with reasoning data
but remember no RL

DeepSeek R1's training recipe

1 Pretrain model with "traditional" techniques: **V3-Base**

2 "Small-scale" SFT with reasoning data



Data source: long CoTs generated with R1-Zero and rewritten by humans

↓
human readable
chain of thoughts

DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data

DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data

~same RL process as with R1-Zero

Reward = Formatting + accuracy + language consistency

Some language input/output

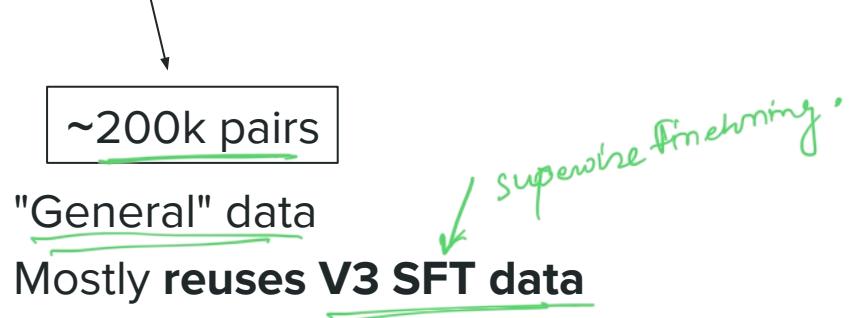
DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data
- 4 "Large-scale" SFT with reasoning and non-reasoning data

↓
larger reasoning
better but remember why
"Adjointage matters here"
"with # of tokens"

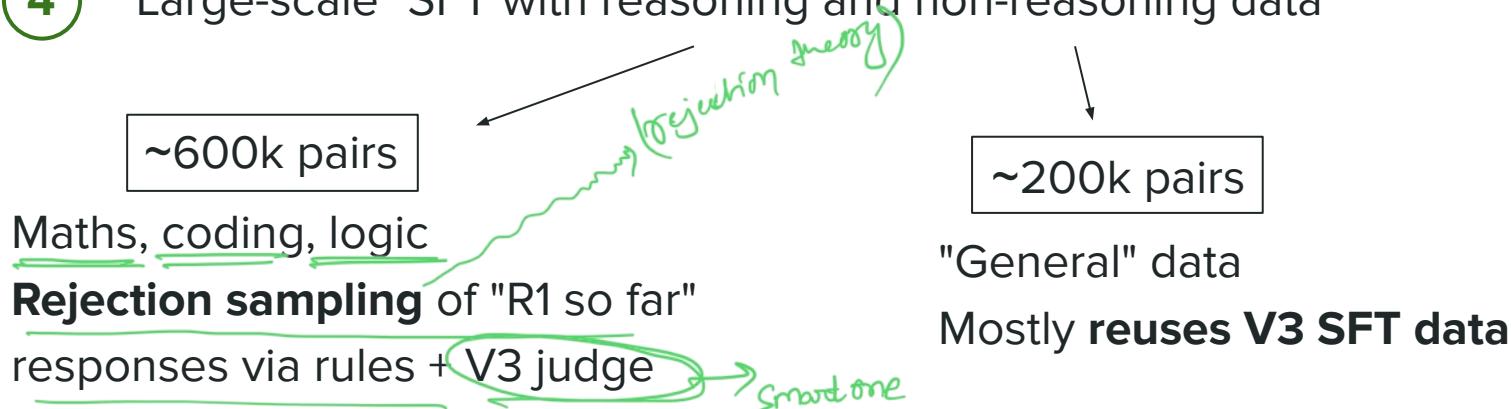
DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data
- 4 "Large-scale" SFT with reasoning and non-reasoning data



DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data
- 4 "Large-scale" SFT with reasoning and non-reasoning data



DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data
- 4 "Large-scale" SFT with reasoning and non-reasoning data
- 5 GRPO with reasoning and non-reasoning data: **R1**

Then started with kind of RL
training technique -
"group returning Policy optimization"

DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data
- 4 "Large-scale" SFT with reasoning and non-reasoning data
- 5 GRPO with reasoning and non-reasoning data: **R1**

Maths, coding, logic

Reward = Formatting + accuracy

↑ Accuracy
for checking
ground truth
if p. o.f.

DeepSeek R1's training recipe

- 1 Pretrain model with "traditional" techniques: **V3-Base**
- 2 "Small-scale" SFT with reasoning data
- 3 GRPO with reasoning data
- 4 "Large-scale" SFT with reasoning and non-reasoning data
- 5 GRPO with reasoning and non-reasoning data: **R1**

Maths, coding, logic

Reward = Formatting + accuracy

"General" data
Mostly **reuses V3 RL data**

Reward = helpfulness + harmlessness

helpful information
fewer feedback needed
which can harm i.e.
detect with harm

DeepSeek R1's results

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
MMLU (Pass@1)	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
English	IF-Eval (Prompt Strict)	86.5	84.3	86.1	84.8	-
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	75.7
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	47.0
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-
	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	96.6
	Codeforces (Rating)	717	759	1134	1820	2061
	SWE Verified (Resolved)	50.8	38.8	42.0	41.6	48.9
Code	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	61.7
	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4
	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	-
	C-Eval (EM)	76.7	76.0	86.5	68.9	-
	C-SimpleQA (Correct)	55.4	58.7	68.0	40.3	-
						63.7

DeepSeek R1's results

Want

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
MMLU (Pass@1)	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
English 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

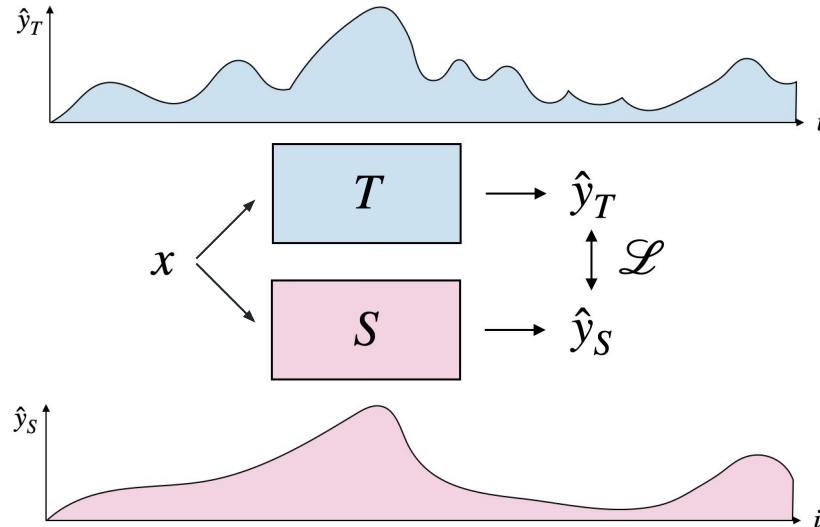
DeepSeek R1's results

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
MMLU (Pass@1)	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
English	IF-Eval (Prompt Strict)	86.5	84.3	86.1	84.8	-
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	75.7
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	47.0
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-
	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	96.6
	Codeforces (Rating)	717	759	1134	1820	2061
	SWE Verified (Resolved)	50.8	38.8	42.0	41.6	48.9
Code	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	61.7
	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-
	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-
Chinese	C-Eval (EM)	76.7	76.0	86.5	68.9	-
	C-SimpleQA (Correct)	55.4	58.7	68.0	40.3	-
						63.7

And what about distillation?

And what about distillation?

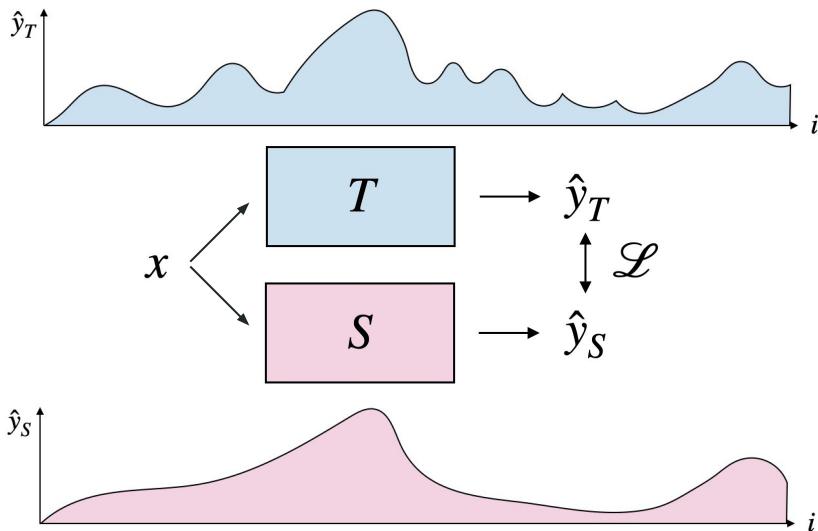
Distillation seen in lecture 2



Goal: match next token distribution

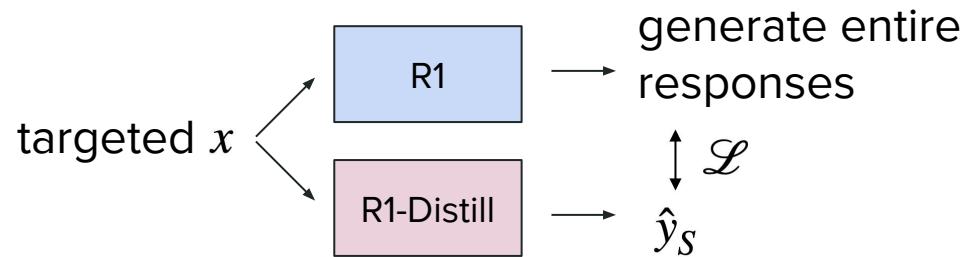
And what about distillation?

Distillation seen in lecture 2



Goal: match next token distribution

Distillation used here



Goal: SFT-learn reasoning traces

And what about distillation?

Results.

And what about distillation?

Results.

Competitive

Model	AIME 2024		MATH-500	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
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
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

And what about distillation?

Results.

Competitive

"Good" use of compute

Model	AIME 2024		MATH-500	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
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
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

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCodeBench
	pass@1	cons@64	pass@1	pass@1	pass@1
OwO-32B-Preview	50.0	60.0	90.6	54.5	41.9
DeepSeek-R1-Zero-Qwen-32B	47.0	60.0	91.6	55.0	40.2
DeepSeek-R1-Distill-Qwen-32B	72.6	83.3	94.3	62.1	57.2

Thank you for your attention!
