DeepSeek-R1, GRPO, and Agentic Reinforcement Learning

Why this lecture starts with DeepSeek-R1

- Reasoning traces reveal how outcome rewards change model behavior
  - PPO explains clipping, critics, advantages, and policy updates
    - GRPO replaces the critic with same-prompt group comparisons
- Agentic RL adds environment state, temporal credit, and systems constraints
- The lecture ends with evaluation, reliability, and reward-channel integrity

How do reasoning models work?

- Explicitly outputs its reasoning steps as it attempts to answer/solve their query

Isn’t this just **C**hain **o**f **T**hought?

Isn’t this just **C**hain **o**f **T**hought?

Key Elements of a **CoT** Prompt:

- **Explicitly** ask for step-by-step reasoning
- Use phrases like "Let’s think step by step" or "Break down the problem"
- Include placeholder steps (e.g., "First... Second...")
- Request verification of the answer to encourage critical thinking

Isn’t this just **C**hain **o**f **T**hought?

Yes / No

…and why?

Isn’t this just Chain of Thought?

CoT

Encourages the model through prompting to generate a sequence of thoughts ***but*** might not be refined, and might produce a bad chain of reasoning.

Reasoning

Isn’t this just Chain of Thought?

CoT

Encourages the model through prompting to generate a sequence of thoughts ***but*** might not be refined, and might produce a bad chain of reasoning.

Reasoning

Trained to produce multiple paths of reasoning, evaluating them, and selecting the most promised one.

Isn’t this just Chain of Thought?

CoT

Encourages the model through prompting to generate a sequence of thoughts ***but*** might not be refined, and might produce a bad chain of reasoning.

Reasoning

Trained to produce multiple paths of reasoning, evaluating them, and selecting the most promised one.

Why is this different?

Isn’t this just Chain of Thought?

CoT

A clever prompting technique to elicit reasoning steps from instruct capable language models that aren’t trained to reason

Reasoning

Is *explicitly* trained through reinforcement learning to produce **correct** reasoning steps

Isn’t this just Chain of Thought?

CoT

A clever prompting technique to elicit reasoning steps from instruct capable language models that aren’t trained to reason

Prompted to produce things that *sound* logical

Reasoning

Is *explicitly* trained through reinforcement learning to produce **correct** reasoning steps

Trained to produce *logical* steps

DeepSeek-R1-Zero

DeepSeek-R1-Zero

- **DeepSeek-V3-Base** - a previously created MoE model made with SFT

DeepSeek-R1-Zero

- **DeepSeek-V3-Base** - a previously created MoE model made with SFT
- Initial test (hence **Zero**) to see if a reasoning LLM can be taught *purely* through reinforcement learning (**GRPO**)

DeepSeek-R1-Zero

- **DeepSeek-V3-Base** - a previously created MoE model made with SFT
- Initial test (hence **Zero**) to see if a reasoning LLM can be taught *purely* through reinforcement learning (**GRPO**)
  - Shows that the typical SFT-then-RL approach wasn't needed

DeepSeek-R1-Zero

- **DeepSeek-V3-Base** - a previously created MoE model made with SFT
- Initial test (hence **Zero**) to see if a reasoning LLM can be taught *purely* through reinforcement learning (**GRPO**)
  - Shows that the typical SFT-then-RL approach wasn't needed
  - Emergent reasoning behaviors:
    - Self-Verification
    - Reflection
    - Longer **C**hain **o**f **T**houghts

DeepSeek-R1-Zero

- **Deep**
- Initial
  throu
  - S
  - E

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: $\cdots$

$\ldots$

DeepSeek-R1-Zero

- **Deep**
- Initial
  throu
  - S
  - E

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 $\cdots$

$\left(\sqrt{a - \sqrt{a+x}}\right)^2$ ^{2} $\overline{x} = x^2$.

Rearrange to is ıare

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

Interrupts its chain of thought

Self-identifies turning point in its

$\cdots$

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: $\cdots$

$\cdots$

A reasoning trace becomes a trajectory when actions change state

- **A completion ends in text; an agent episode changes an external environment**
  - The learner needs actions, observations, state evidence, rewards, and termination reasons
  - Only policy-owned tokens should enter the policy loss
  - Environment turns define useful credit boundaries
  - The trace must record what the policy knew before each action
    - Reliable learning begins with a faithful trajectory contract

Let’s talk Reinforcement Learning

- Researchers designed a reward for encouraging reasoning

Let’s talk Reinforcement Learning

- Researchers designed a reward for encouraging reasoning

Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using* **<think>** *and* **<answer>** *tags correctly*

Let’s talk Reinforcement Learning

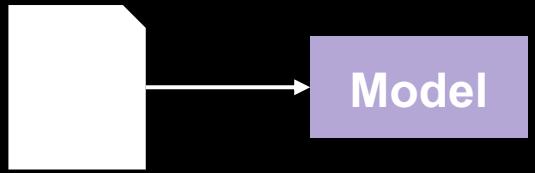
- Researchers designed a reward for encouraging reasoning
- Training flow:

Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using* **<think>** *and* **<answer>** *tags correctly*

Let’s talk Reinforcement Learning

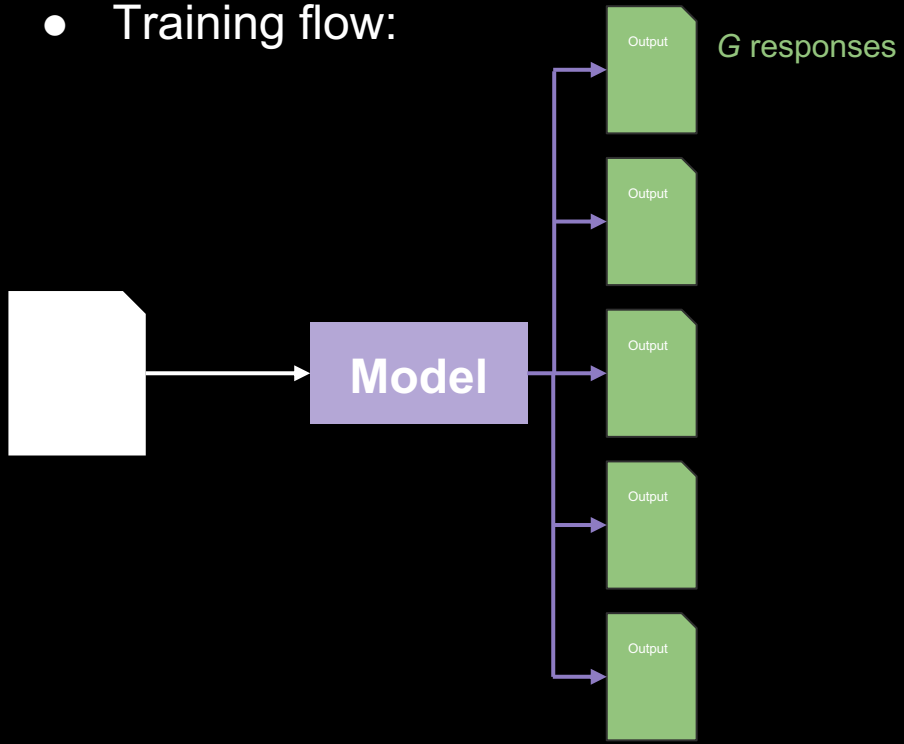
- Researchers designed a reward for encouraging reasoning
- Training flow:

Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using **<think>** and **<answer>** tags correctly*



Let’s talk Reinforcement Learning

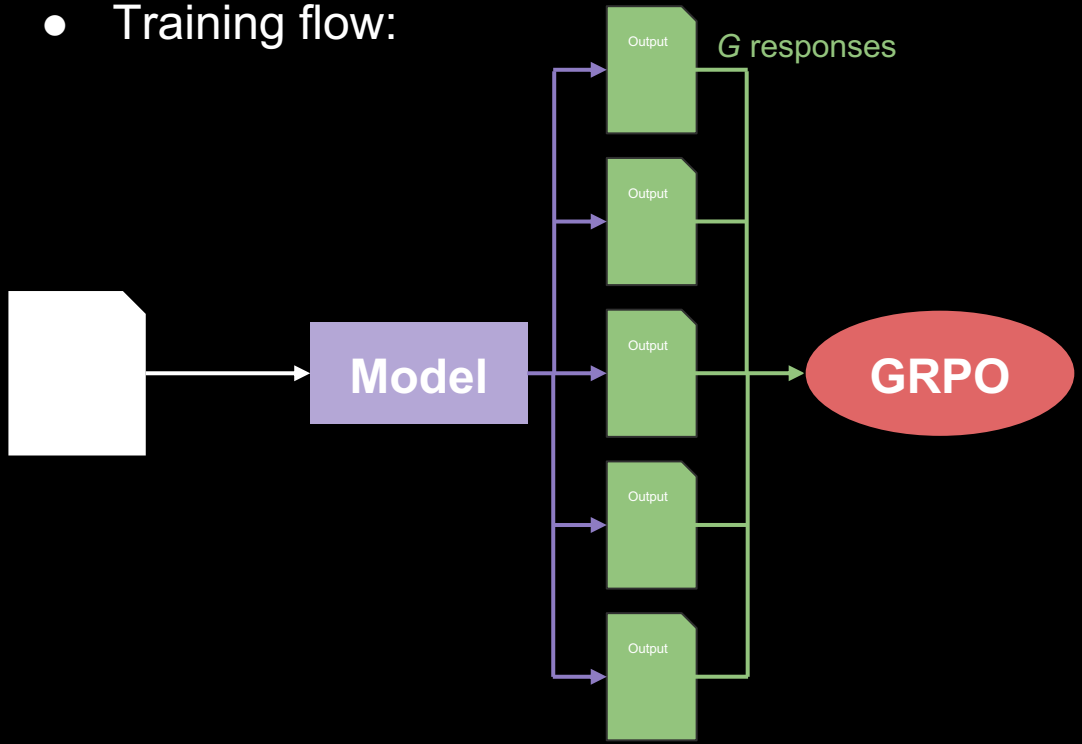
- Researchers designed a reward for encouraging reasoning
- Training flow:



Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using **<think>** and **<answer>** tags correctly*

Let’s talk Reinforcement Learning

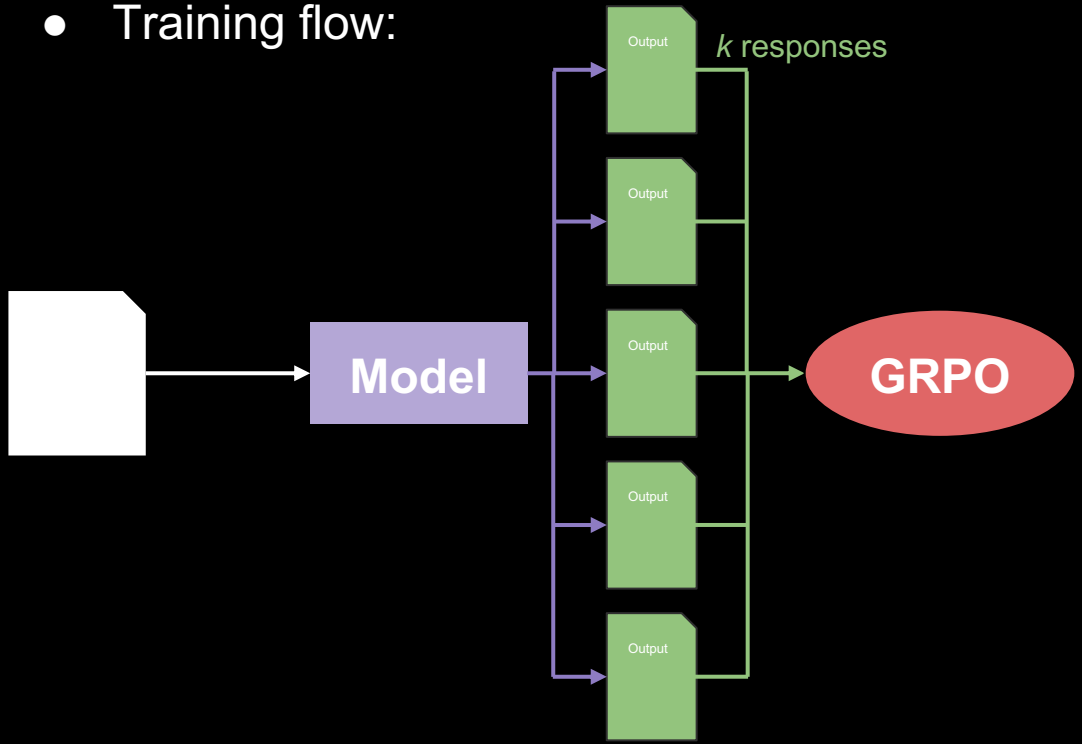
- Researchers designed a reward for encouraging reasoning
- Training flow:



Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using **<think>** and **<answer>** tags correctly*

Let’s talk Reinforcement Learning

- Researchers designed a reward for encouraging reasoning
- Training flow:



Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using **<think>** and **<answer>** tags correctly*

What the heck is **GRPO**?

- **G**roup **R**elative **P**olicy **O**ptimization - an evolution from **PPO**

What the heck is **GRPO**?

- **G**roup **R**elative **P**olicy **O**ptimization - an evolution from **PPO**

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

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

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

Proximal Policy Optimization (PPO), briefly

What is PPO?

- Proximal Policy Optimization is a technique published by OpenAI in 2017
  - Probability distribution output (Often Gaussian for continuous, Categorical for discrete, etc)
    - Mean (μ) and standard deviation/variance (σ or σ2, respectively) are the layer's outputs
- It is an **Actor**-**Critic** algorithm - the **Actor** learns our policy π, the **Critic** learns how good chosen actions were (**V**)
- Utilizes Advantage (**A**) to train the **Actor**

What is PPO?

- Proximal Policy Optimization is a technique published by OpenAI in 2017
  - Probability distribution output (Often Gaussian for continuous, Categorical for discrete, etc)
    - Mean (**μ**) and standard deviation/variance (**σ** or **σ2**, respectively) are the layer's outputs
- It is an **Actor**-**Critic** algorithm - the **Actor** learns our policy **π**, the **Critic** learns how good chosen actions were (**V**)
- Utilizes Advantage (**A**) to train the **Actor**

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

What is PPO?

- Proximal Policy Optimization is a technique published by OpenAI in 2017
  - Probability distribution output (Often Gaussian for continuous, Categorical for discrete, etc)
    - Mean (μ) and standard deviation/variance (σ or σ2, respectively) are the layer's outputs
- It is an **Actor**-**Critic** algorithm - the **Actor** learns our policy π, the **Critic** learns how good chosen actions were (**V**)
- Utilizes Advantage (**A**) to train the **Actor**

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

Discounted Rewards

Critic Evaluation

What is PPO?

- Proximal Policy Optimization is a technique published by OpenAI in 2017
  - Probability distribution output (Often Gaussian for continuous, Categorical for discrete, etc)
    - Mean (**μ**) and standard deviation/variance (**σ** or **σ2**, respectively) are the layer's outputs
- It is an **Actor**-**Critic** algorithm - the **Actor** learns our policy **π**, the **Critic** learns how good chosen actions were (**V**)
- Utilizes Advantage (**A**) to train the **Actor**

How good was this action?

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How good was our state in general?

What is PPO?

- Proximal Policy Optimization is a technique published by OpenAI in 2017
  - Probability distribution output (Often Gaussian for continuous, Categorical for discrete, etc)
    - Mean (**μ**) and standard deviation/variance (**σ** or **σ2**, respectively) are the layer's outputs
- It is an **Actor**-**Critic** algorithm - the **Actor** learns our policy **π**, the **Critic** learns how good chosen actions were (**V**)
- Utilizes Advantage (**A**) to train the **Actor**

How good was this action?

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

How good was our state in general?

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

$$r = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

$$r = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$

Probability the **current** policy would choose this action.

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

$$r = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$

Probability the **current** policy would choose this action.

Probability the **previous** policy would choose this action.

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

$$r = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$

Probability the **current** policy would choose this action.

Probability the **previous** policy would choose this action.

$$loss = min(r, clamp(1-\epsilon, 1+\epsilon, r)) * A$$

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

$$r = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$

Probability the **current** policy would choose this action.

Probability the **previous** policy would choose this action.

Typically ~0.2

$$loss = min(r, clamp(1-\epsilon, 1+\epsilon, r)) * A$$

What is PPO?

- Clipped objective function to avoid large policy changes in the actor

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

How much better than average did we perform for this action?

$$r = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$$

Probability the **current** policy would choose this action.

Probability the **previous** policy would choose this action.

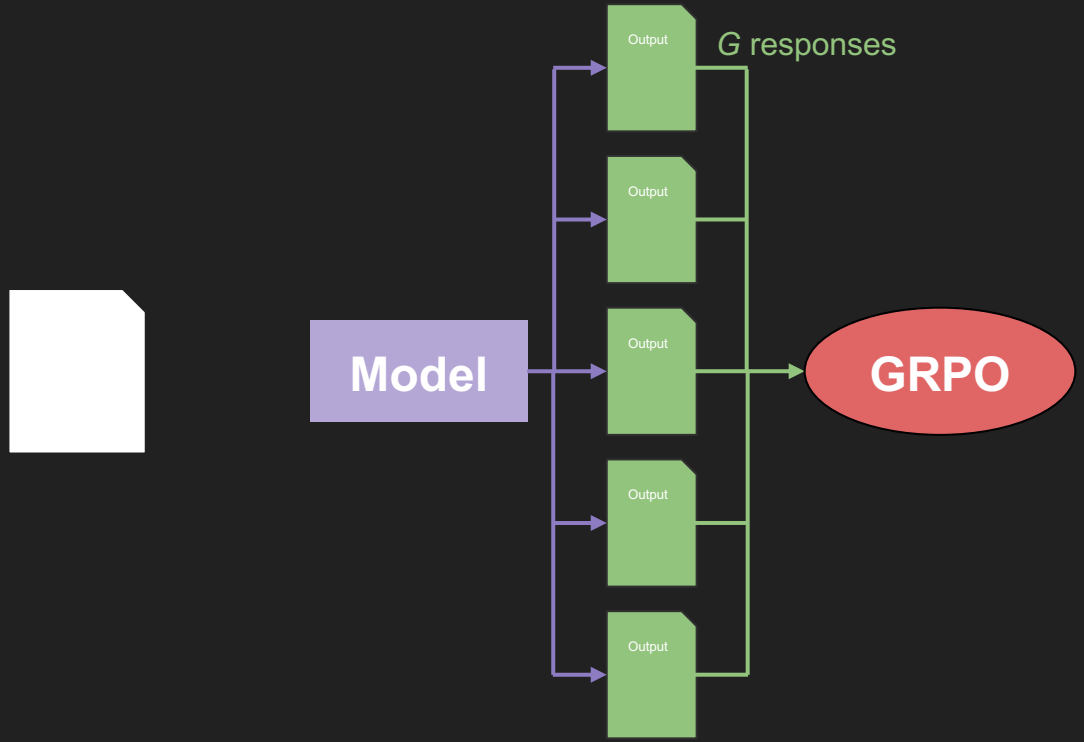
Typically ~0.2

$$loss = min(r, clamp(1-\epsilon, 1+\epsilon, r)) * A$$

This prevents large changes to the policy for any one action, creating a pessimistic lower bound

Let’s talk Reinforcement Learning

Focus	What it measured
Accuracy	*Correctly providing answers in reasoning tasks*
Format	*Using **<think>** and **<answer>** tags correctly*



Group Relative Policy Optimization (GRPO), briefly

What the heck is **GRPO**?

- **G**roup **R**elative **P**olicy **O**ptimization - an evolution from **PPO**

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

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

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

The math behind GRPO

Objective Function
(what we are optimizing)

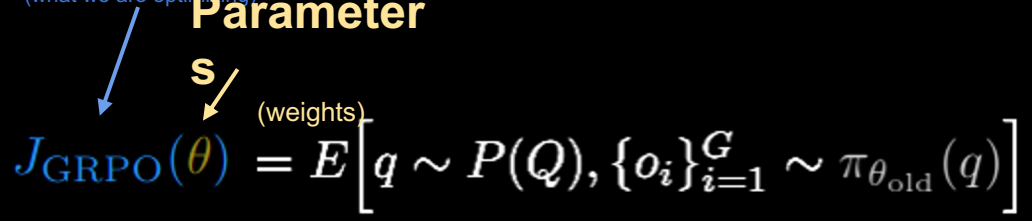
$$J_{\text{GRPO}}(\theta) = E\left[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(q)\right]$$

Objective functions represents a quantity to be optimized in a problem

The math behind GRPO

Objective Function
(what we are optimizing)

Parameter
s
(weights)



Objective functions represents a quantity to be optimized in a problem

The math behind GRPO

Objective Function
(what we are optimizing)

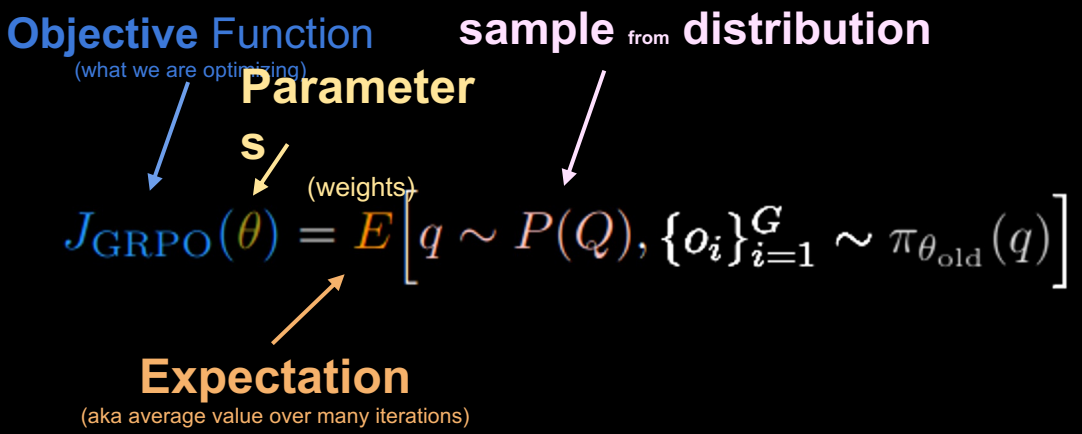
Parameter
s
(weights)

$$J_{\text{GRPO}}(\theta) = E\left[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(q)\right]$$

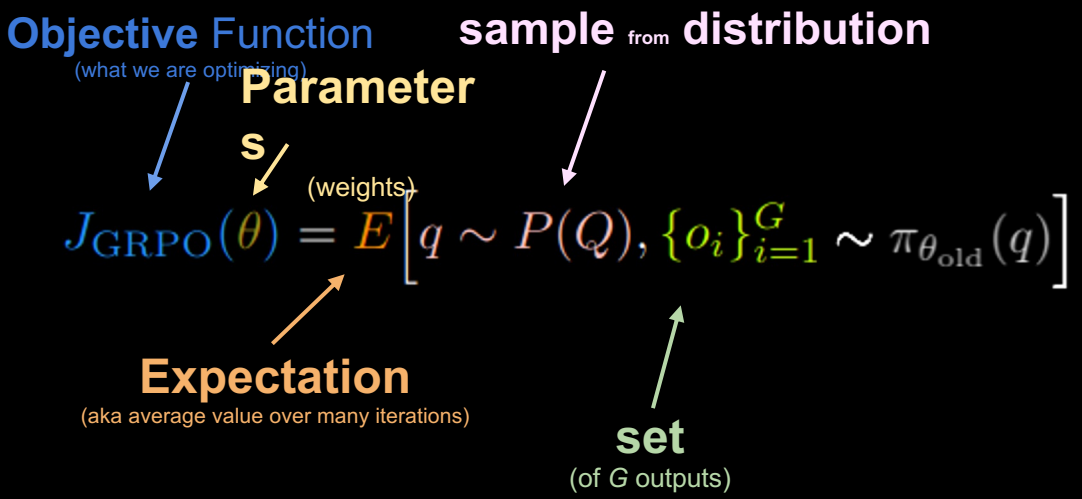
Expectation
(aka average value over many iterations)

An ***expectation*** is the average over many samples or repetitions.

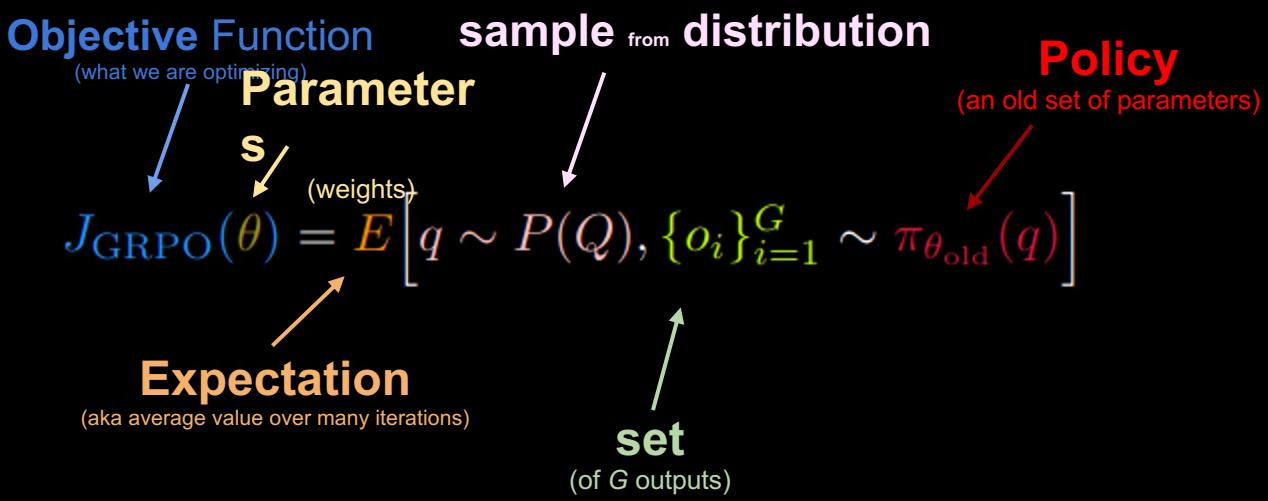
The math behind GRPO



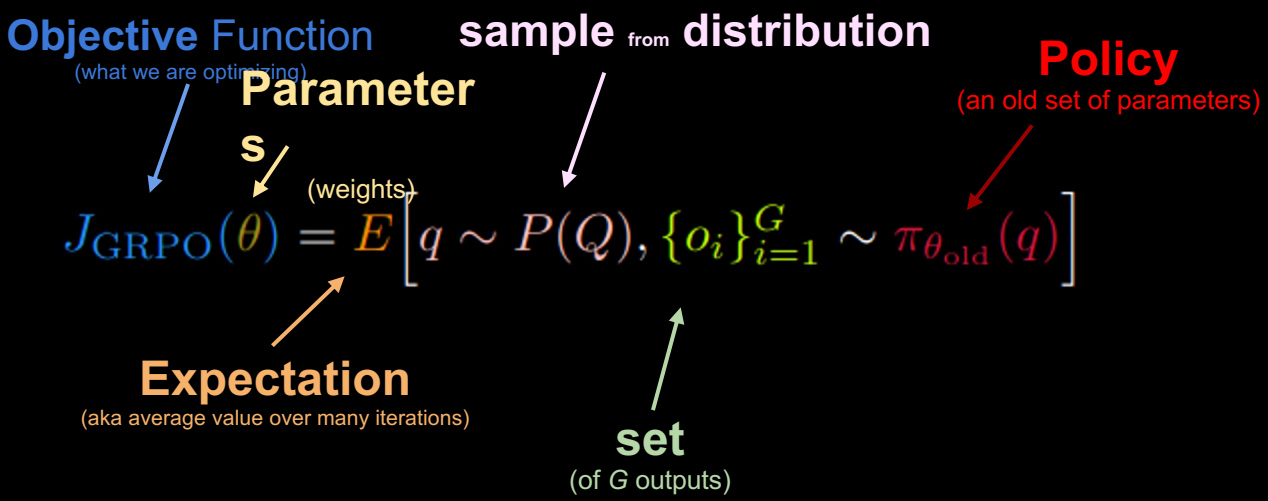
The math behind GRPO



The math behind GRPO



The math behind GRPO



This says:
Sample a query from a collection of prompts, creating a set of *G* outputs using the old policy, then take the average of its scores over many iterations to iteratively improve our weights.

The math behind **GRPO**

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

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

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

The math behind **GRPO**

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

The math behind GRPO

Take the **average**

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

The math behind GRPO

Take the **average**

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

Probability **ratio**

The math behind **GRPO**

$$A(s,a) = \frac{Q_\pi(s,a)}{V(s)}$$

Take the **average**

Advantage

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

Probability **ratio**

The math behind **GRPO**

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

Take the **average**

Advantage

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

Probability **ratio**

The math behind **GRPO**

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

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

Take the **average**

Advantage

Probability **ratio**

Here, **advantage** is looking at *relative* improvement by comparing each response to the group's average, driving model updates towards better relative responses

Also, by not having to train a critic model, we are more stable *and* more efficient!

The math behind **GRPO**

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

Take the **average**

Advantage

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

Probability **ratio**

The math behind **GRPO**

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

Take the **average**

Advantage

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

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

The math behind **GRPO**

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

Take the **average**

Advantage

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

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

The math behind **GRPO**

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

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

Take the **average**

Advantage

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

KL Divergence

The math behind **GRPO**

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

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

Take the **average**

Advantage

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

KL Divergence

KL Divergence (**K**ullback–**L**eibler), also known as relative entropy, measures how different two probability distributions are.

The math behind **GRPO**

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

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

Take the **average**

Advantage

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

KL Divergence

$$D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1$$

The math behind **GRPO**

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

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

Take the **average**

Advantage

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

KL Divergence

$$D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1$$

Prior policy

New policy

The math behind **GRPO**

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

Take the **average**

Advantage

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

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

KL Divergence

Prior policy

$$D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1$$

New policy

KL divergence is seeing how different the new policy is from the old policy - we are *penalizing* large changes!

The math behind **GRPO**

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

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

Take the **average**

Advantage

Probability **ratio**

How aggressive are our updates? Forms a ***trust region***

KL Divergence

This says:

Update its **strategy** based on how much better an action performed compared to the average (**advantage**). However, it only makes small, careful adjustments (**trust region**) by clipping the size of the update and discouraging big changes (**KL** divergence), then averages these adjustments over multiple experiences to find the best overall improvement.

The math behind **GRPO**

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

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

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

The math behind **GRPO**

$$J_{\text{GRPO}}(\theta) = E\left[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(q)\right]$$

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

$$D_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1$$

From single-turn GRPO to agent trajectories

Multi-turn GRPO must mask policy-owned tokens

- **Tool outputs, user messages, and observations are context for the policy**
  - Apply the objective only where the ownership mask m_i,t equals one
  - Store behavior-policy log probabilities with every generated token
  - Store the owner mask and environment turn with the same trace
  - A missing boundary trains on data the policy did not produce
    - Ownership is part of the objective, not a logging convenience

Equal-reward groups carry no relative learning signal

- **GRPO compares samples generated for the same prompt**
  - If every sample succeeds, normalized advantages approach zero
  - If every sample fails, the same collapse occurs
  - Mixed-outcome groups provide the relative evidence GRPO needs
  - Sampling and curriculum should track the moving frontier
    - Group composition controls whether the update contains signal

Agent rewards should verify goals and invariants

- **Success requires the intended goal predicates**
  - Safety and integrity constraints must remain true
  - Permission and consent checks belong in the reward gate
  - Evaluator integrity belongs in the same protected boundary
  - Terminal-state checks remain defensible when dense credit is uncertain
    - Reward design specifies both achievement and admissible behavior

DeepSeek-R1-**Zero** Performance problems

DeepSeek-R1-**Zero** Performance problems

- **Readability** was poor, especially in its thinking
  - Lack of markdown formatting or structure in thinking tokens
  - Few steps noted
  - Jumps in logic
  - Redundant and **unconventional** text
    - Not human friendly/aligned

DeepSeek-R1-Zero Performance problems

- **Readability** was poor, especially in its thinking
  - Lack of markdown formatting or structure in thinking tokens
  - Few steps noted
  - Jumps in logic
  - Redundant and **unconventional** text
    - Not human friendly/aligned
- **Mixed language** responses/thinking
  - No RL penalty, so it mixed languages as long as accuracy improved

DeepSeek-R1-Zero Performance problems

- **Readability** was poor, especially in its thinking
  - Lack of markdown formatting or structure in thinking tokens
  - Few steps noted
  - Jumps in logic
  - Redundant and **unconventional** text
    - Not human friendly/aligned
- **Mixed language** responses/thinking
  - No RL penalty, so it mixed languages as long as accuracy improved
- **Reward hacking**
  - RL resulted in focusing on the reward, not a useful model

DeepSeek-R1-Zero Performance

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
OpenAI-o1-0912	74.4	83.3	94.8	77.3	63.4	1843
DeepSeek-R1-Zero	71.0	86.7	95.9	73.3	50.0	1444

Benchmark scores support only the claims their harness measures

- **Report the environment version, tools, policy, budget, and evaluator**
  - Measure repeated-run success and constraint violations
  - Measure recovery, efficiency, and failure severity
  - A frozen benchmark score does not establish deployment reliability
  - Match observable state and failure modes to the research claim
    - Evaluation quality is bounded by the harness contract

DeepSeek-R1

- DeepSeek-Zero gave us a good blueprint, but can we fix its limitations?

PIPELINE:

DeepSeek-**R1**

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
    - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples

DeepSeek-**R1**

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
    - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples

Why?

DeepSeek-**R1**

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
    - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples
        - Improve Readability - demonstrate format
        - Get rid of language mixing

DeepSeek-**R1**

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples
2. 🔬🧮 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)

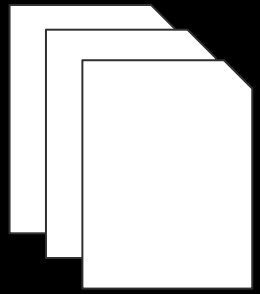
DeepSeek-R1

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

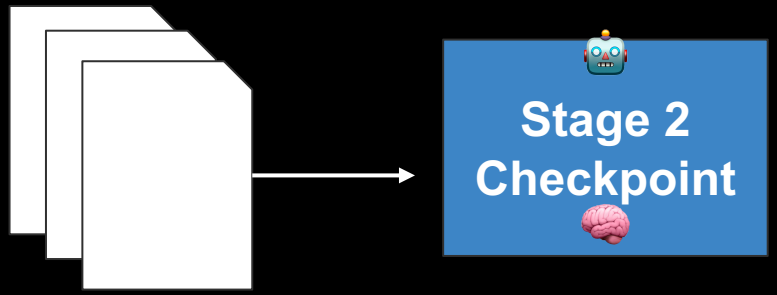
PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples
2. 🔬🧮 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)
3. 👩‍⚕️🚫 Refined Reasoning Rejection Sampling SFT

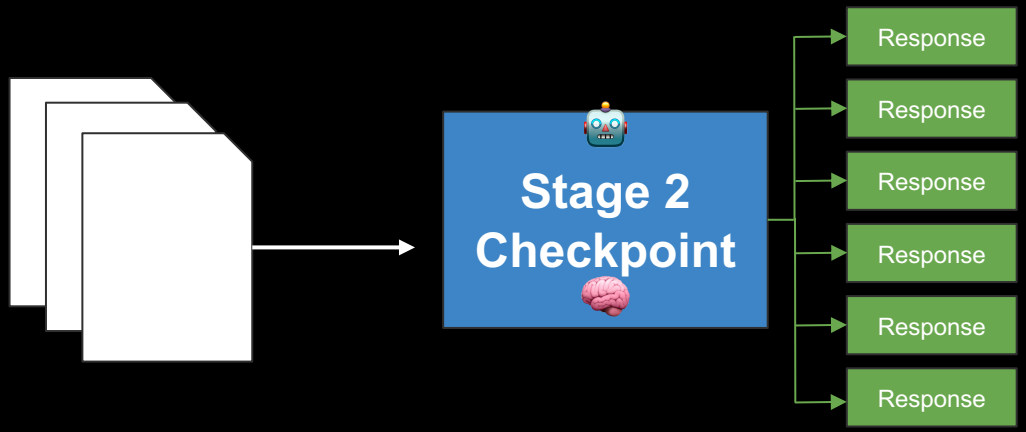
DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



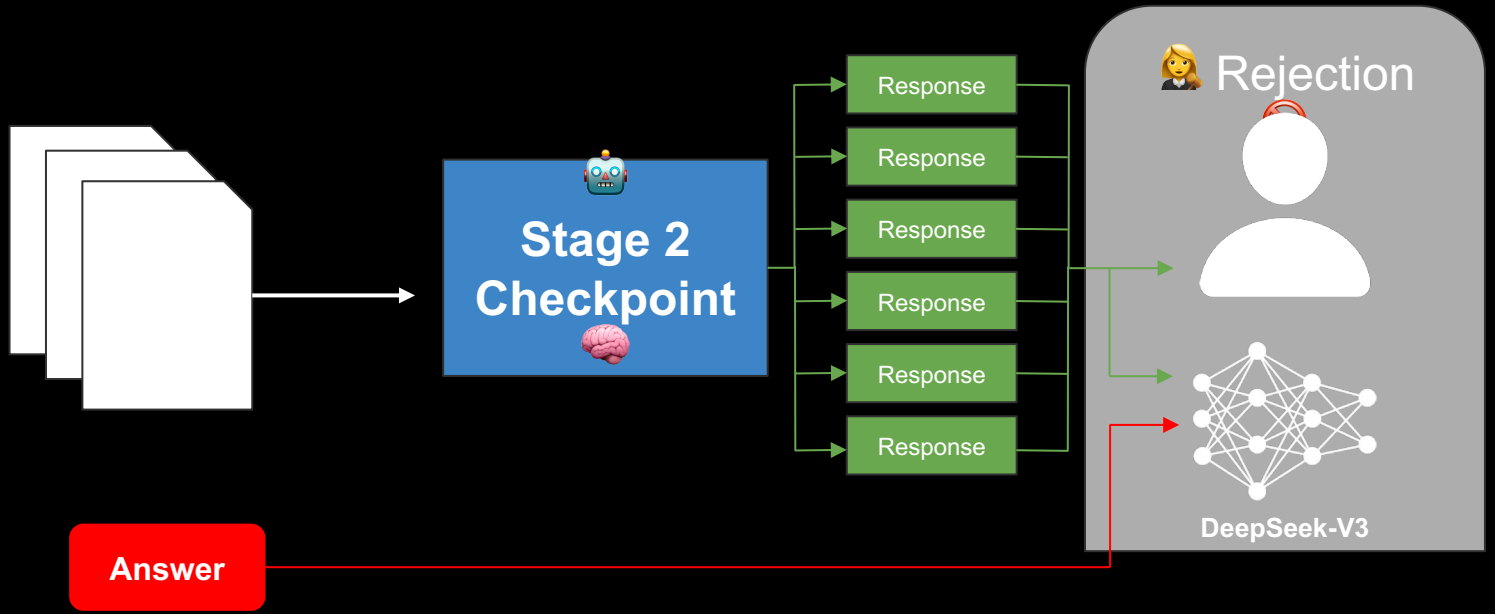
DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



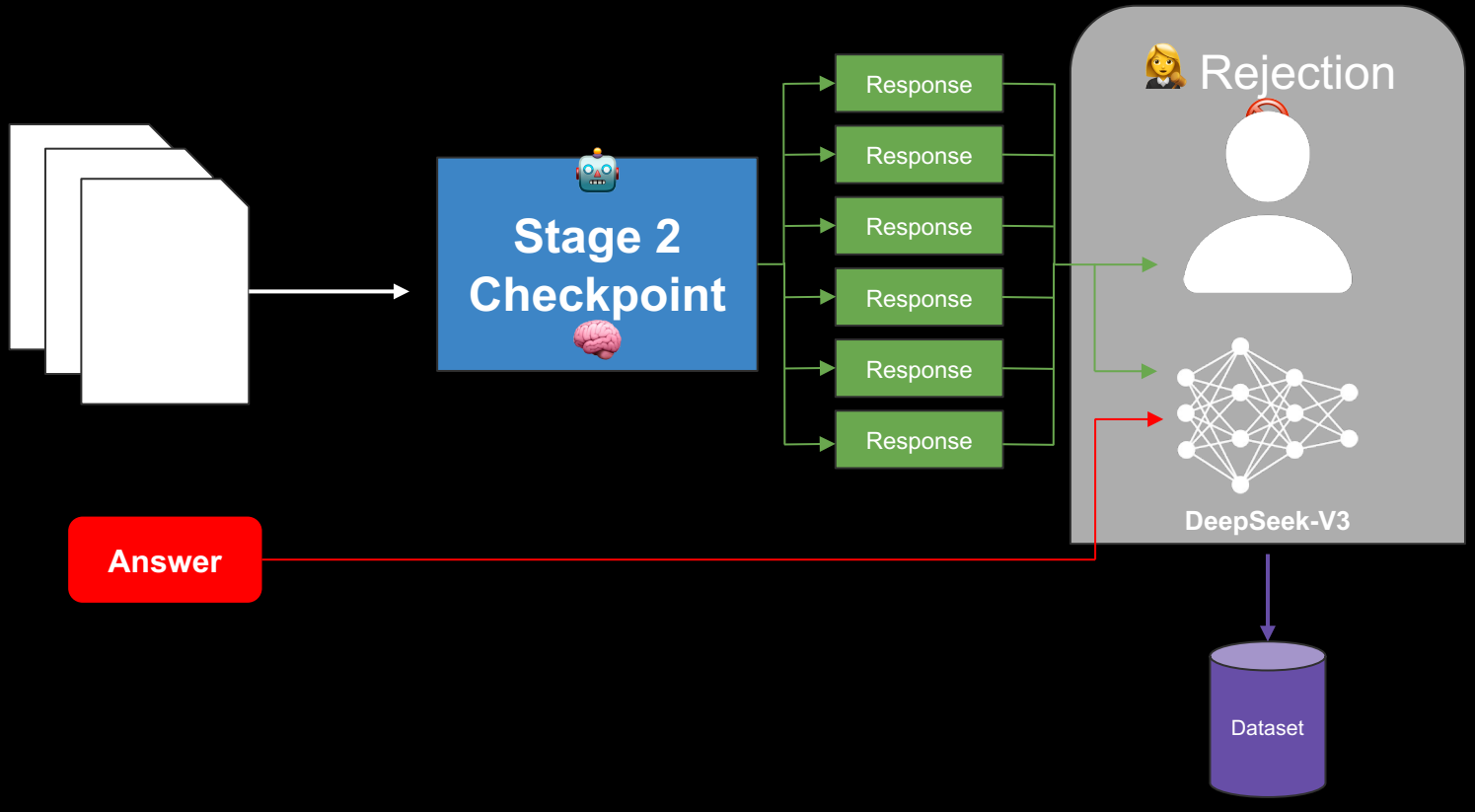
DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



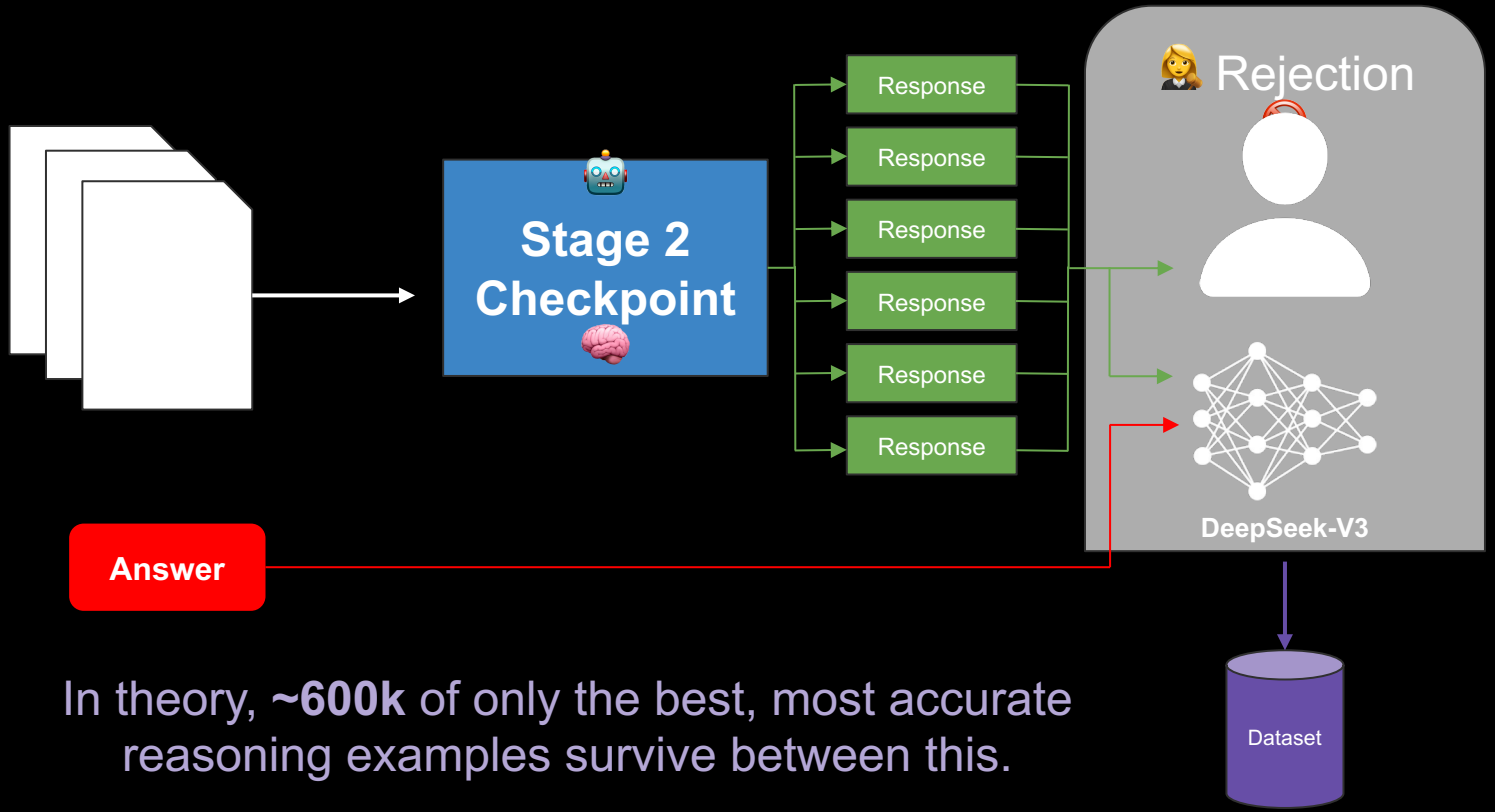
DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT

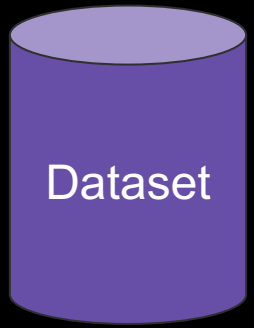


DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



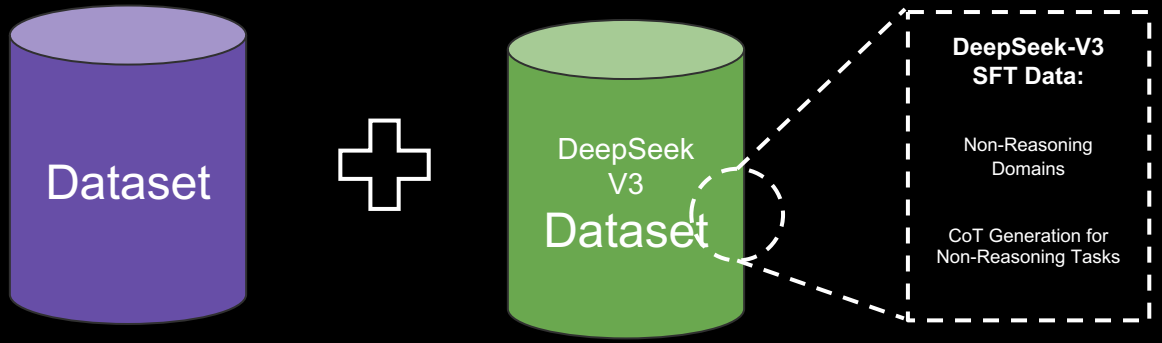
In theory, **~600k** of only the best, most accurate reasoning examples survive between this.

DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT

DeepSeek-R1 Stage 3 - Refined Reasoning Rejection Sampling SFT



DeepSeek-R1

- DeepSeek-Zero gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples
2. 🔬🧮 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)
3. 👩‍🔬🚫 Refined Reasoning Rejection Sampling SFT
   - Take ***DeepSeek-V3-Base*** (*not* Stage 2 Checkpoint) and combined dataset - train for **2** epochs

DeepSeek-R1

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-**Base*** high quality human-readable **CoT** examples
2. 🔬📱 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)
3. 👩‍🎨🚫 Refined Reasoning Rejection Sampling SFT
   - Take ***DeepSeek-V3-Base*** (*not* Stage 2 Checkpoint) and combined dataset - train for **2** epochs

Why?

DeepSeek-R1

- DeepSeek-Zero gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples
2. 🔬🧮 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)
3. 👩‍🎨🚫 Refined Reasoning Rejection Sampling SFT
   - Take ***DeepSeek-V3-Base*** (*not* Stage 2 Checkpoint) and combined dataset - train for **2** epochs

Speculation: Probably to avoid **overfitting** the rejection SFT dataset and to better incorporate **broader language** capabilities

DeepSeek-**R1**

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-**Base*** high quality human-readable **CoT** examples
2. 🔬🧮 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)
3. 👩‍🎨🚫 Refined Reasoning Rejection Sampling SFT
   - Take ***DeepSeek-V3-Base*** (*not* Stage 2 Checkpoint) and combined dataset - train for **2** epochs

This is essentially **knowledge distillation**.

DeepSeek-R1

- DeepSeek-**Zero** gave us a good blueprint, but can we fix its limitations?

PIPELINE:

1. ❄️🥶 Cold Start
   - Fine tune *DeepSeek-V3-Base* high quality human-readable **CoT** examples
2. 🔬🧮 Reasoning-Oriented RL
   - RL w/ a focus on reasoning-intensive tasks (coding, math, science, logic)
3. 👩‍🔬🚫 Refined Reasoning Rejection Sampling SFT
   - Take ***DeepSeek-V3-Base*** (*not* Stage 2 Checkpoint) and combined dataset - train for **2** epochs
4. 🦺👷 Alignment RL - Helpfulness, Harmlessness, & Reasoning in All Scenarios

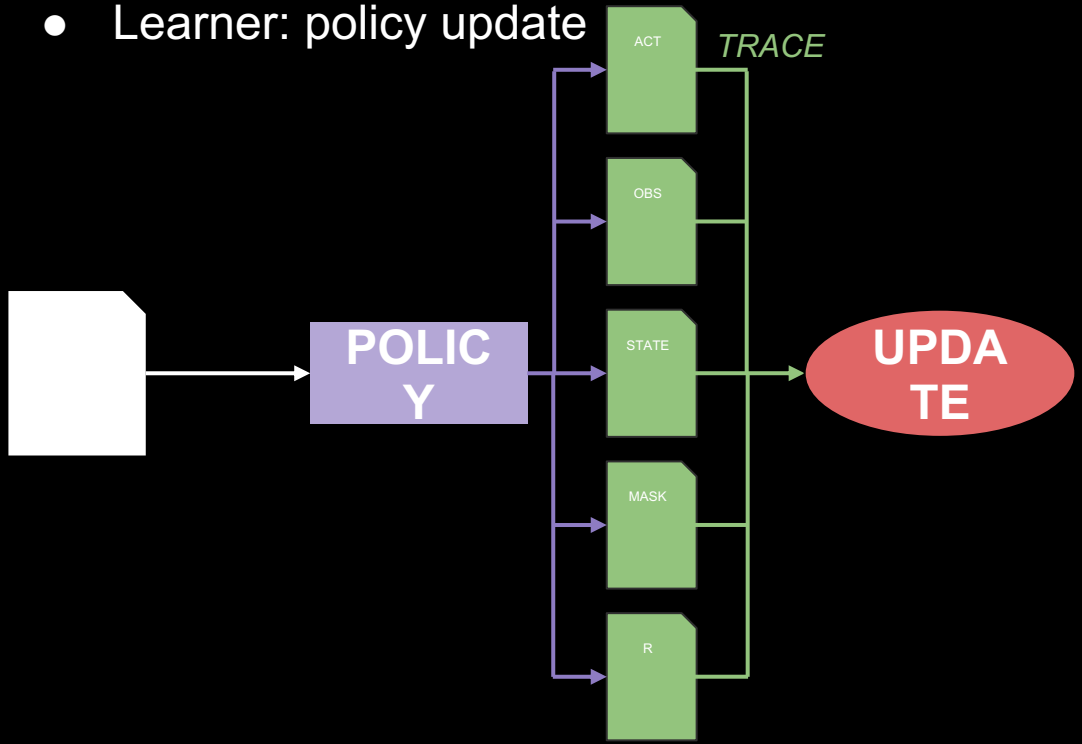
Agentic RL requires a training system

Rollout design is part of the algorithm

- **Task sampler: difficulty, failure modes, and domains**
  - Initial state: snapshot, seed, and perturbation
  - Policy: checkpoint, temperature, and tool set
  - Interaction: turn granularity, timeout, and budget
  - Selection: keep, retry, filter, and replay—with reasons
    - These choices define the data distribution seen by the learner

Trace contracts connect RL

- Harness: interaction
- Learner: policy update



Owner	Contract
Harness	*Tools, context, control flow*
Learner	*Masks, advantages, updates*

Asynchrony buys throughput and creates policy lag

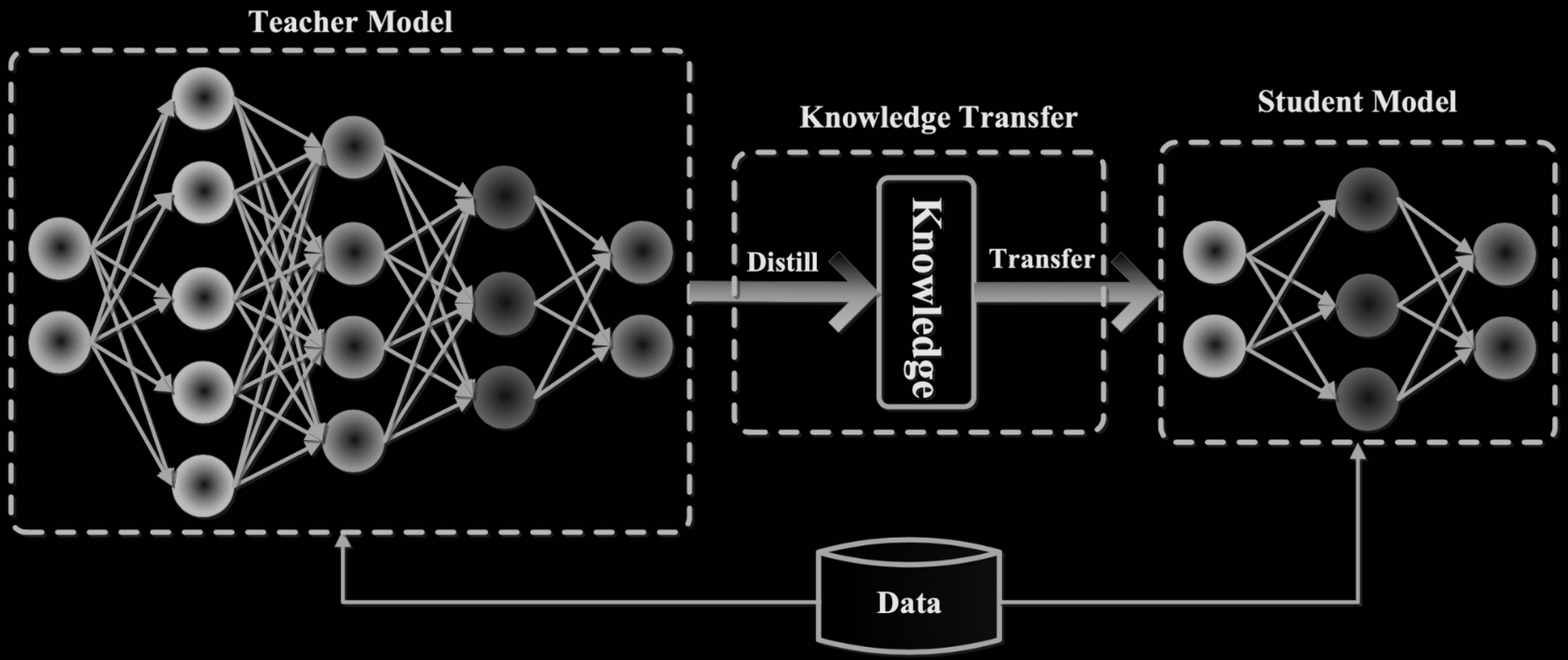
- **Actors generate with behavior policy μ while the learner updates πθ**
  - Queue age and importance-ratio tails reveal stale traces
  - Clipping controls variance but cannot make old data on-policy
  - Choose a staleness rule before training begins
  - Accept, reweight, clip, or drop traces by explicit criteria
    - Throughput gains are valid only with measured off-policy drift

Monitor policy lag with data

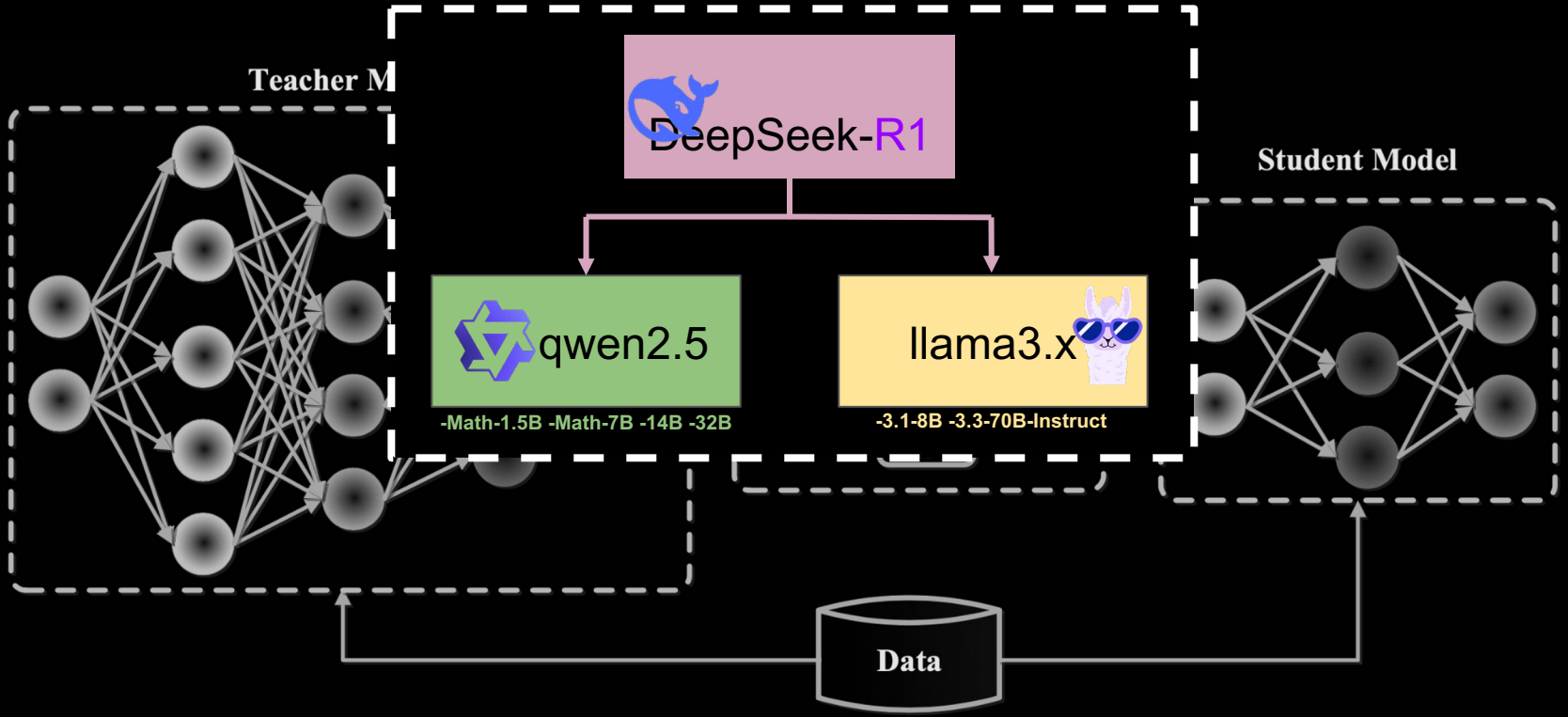
- **Policy age: updates between rollout and learner**
  - Ratio tails: p95, p99, and maximum importance weights
  - Effective sample size and behavior-to-learner KL
  - Reward, violation, and drop rate by trace age
  - Gradient norm, loss spikes, and clipping fraction
    - A queue metric alone cannot diagnose off-policy learning

Knowledge Distillation

Knowledge Distillation



Knowledge Distillation



Knowledge Distillation

3.2. Distilled Model Evaluation

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

Table 5 | Comparison of DeepSeek-R1 distilled models and other comparable models on reasoning-related benchmarks.

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes
  - A normal reward might be 0 or 1 for correct or not, but **PRM** would reward individual steps being correct. For example - solving an algebra problem would give *partial credit* to correct steps towards solving it, even if wrong.

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes
  - A normal reward might be 0 or 1 for correct or not, but **PRM** would reward individual steps being correct. For example - solving an algebra problem would give *partial credit* to correct steps towards solving it, even if wrong.
  - Tries to encourage good behavior throughout the <*think*> tags rather than just the <*answer*> tags.

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes
  - A normal reward might be 0 or 1 for correct or not, but **PRM** would reward individual steps being correct. For example - solving an algebra problem would give *partial credit* to correct steps towards solving it, even if wrong.
  - Tries to encourage good behavior throughout the <*think*> tags rather than just the <*answer*> tags.
- Abandoned because:

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes
  - A normal reward might be 0 or 1 for correct or not, but **PRM** would reward individual steps being correct. For example - solving an algebra problem would give *partial credit* to correct steps towards solving it, even if wrong.
  - Tries to encourage good behavior throughout the <*think*> tags rather than just the <*answer*> tags.
- Abandoned because:
  - Challenging to define a fine-grain step in reasoning

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes
  - A normal reward might be 0 or 1 for correct or not, but **PRM** would reward individual steps being correct. For example - solving an algebra problem would give *partial credit* to correct steps towards solving it, even if wrong.
  - Tries to encourage good behavior throughout the <*think*> tags rather than just the <*answer*> tags.
- Abandoned because:
  - Challenging to define a fine-grain step in reasoning
  - Determining if an intermediate step is correct is *hard*

Failed Experiments

- **PRM** - **P**erplexity **R**eward **M**odel - a reward function that is designed to evaluate and reward step-by-step reasoning processes
  - A normal reward might be 0 or 1 for correct or not, but **PRM** would reward individual steps being correct. For example - solving an algebra problem would give *partial credit* to correct steps towards solving it, even if wrong.
  - Tries to encourage good behavior throughout the <*think*> tags rather than just the <*answer*> tags.
- Abandoned because:
  - Challenging to define a fine-grain step in reasoning
  - Determining if an intermediate step is correct is *hard*
  - Reward hacking

Failed Experiments

- **MCTS** (**M**onte **C**arlo **T**ree **S**earch)

Failed Experiments

- **MCTS** (**M**onte **C**arlo **T**ree **S**earch) - a tree search algorithm to guide decision making

Failed Experiments

- **MCTS** (**M**onte **C**arlo **T**ree **S**earch) - a tree search algorithm to guide decision making
- Utilized at test-time to enhance the model exploring solution space

Failed Experiments

- **MCTS** (**M**onte **C**arlo **T**ree **S**earch) - a tree search algorithm to guide decision making
- Utilized at test-time to enhance the model exploring solution space
- Abandoned because:

Failed Experiments

- **MCTS** (**M**onte **C**arlo **T**ree **S**earch) - a tree search algorithm to guide decision making
- Utilized at test-time to enhance the model exploring solution space
- Abandoned because:
  - Language models have a huge search space as it’s an unstructured space - setting search depth limits risked getting stuck in local optima

Failed Experiments

- **MCTS** (**M**onte **C**arlo **T**ree **S**earch) - a tree search algorithm to guide decision making
- Utilized at test-time to enhance the model exploring solution space
- Abandoned because:
  - Language models have a huge search space as it's an unstructured space - setting search depth limits risked getting stuck in local optima
  - Couldn't effectively create a value model to guide token-level search in complex reasoning tasks

Reliability needs repeated evidence

- **Report mean success with confidence intervals**
  - Measure pass^k across repeated runs
  - Track constraint violations and safe recovery
  - Track efficiency and failure severity
  - Audit a frozen policy on held-out environment versions
    - A single successful trajectory is a demonstration, not reliability evidence

The reward channel belongs outside the agent’s control

- **The policy may write working state and request tool actions**
  - Tests, graders, permissions, and evaluator prompts stay protected
  - Reward services stay outside the writable workspace
  - The agent must not edit evidence that determines its reward
  - Audit logs and state receipts should be append-only
    - Trust boundaries are part of the learning objective’s validity

Train over trajectories; deploy only on repeated evidence