DeepSeek R1: Reasoning via RL[1]

$6M Is All You Need?

J. Setpal

January 30, 2025

MACHINE LEARNING @ PURDUE

[1] Guo, et. al. [Jan 2025]

Welcome to Reading Group!

Ask many, many questions! **Please don't hold questions until the end.**

Welcome to Reading Group!

Ask many, many questions! **Please don't hold questions until the end.**

Discussion is awesome, restating the obvious is good practice and boosts clarity for everyone.

Welcome to Reading Group!

Ask many, many questions! **Please don't hold questions until the end.**

Discussion is awesome, restating the obvious is good practice and boosts clarity for everyone.

I probably get some things wrong, and definitely can't answer every question. Idea is to work through it together.

Outline

1. RL Review
2. Training Details
3. Performance Evals

Outline

1. RL Review
2. Training Details
3. Performance Evals

PPO

We start with the standard clipped surrogate objective introduced in PPO:

$$\mathcal{J}_{PPO}(\theta) := \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{|o|}\sum_{i=1}^{|o|} \min\left(\frac{\pi_{\theta_{old}}(o_i|q, o_{<i})}{\pi_\theta(o_i|q, o_{<i})}A_i, \text{clip}\left(\frac{\pi_{\theta_{old}}(o_i|q, o_{<i})}{\pi_\theta(o_i|q, o_{<i})}, 1 \pm \varepsilon\right) A_i\right) \tag{1}$$

PPO

We start with the standard clipped surrogate objective introduced in PPO:

$$\mathcal{J}_{PPO}(\theta) := \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{|o|}\sum_{i=1}^{|o|} \min\left(\frac{\pi_{\theta_{old}}(o_i|q, o_{<i})}{\pi_\theta(o_i|q, o_{<i})}A_i, \text{clip}\left(\frac{\pi_{\theta_{old}}(o_i|q, o_{<i})}{\pi_\theta(o_i|q, o_{<i})}, 1 \pm \varepsilon\right) A_i\right) \tag{1}$$

One issue with this is A_i, which is computed using the **Generalized Advantage Estimator**[2] –

1. Estimates long-range trajectory rewards.

[2] Schulman et. al [ICLR 2016]

PPO

We start with the standard clipped surrogate objective introduced in PPO:

$$\mathcal{J}_{PPO}(\theta) := \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{|o|}\sum_{i=1}^{|o|} \min\left(\frac{\pi_{\theta_{old}}(o_i|q, o_{<i})}{\pi_\theta(o_i|q, o_{<i})}A_i, \text{clip}\left(\frac{\pi_{\theta_{old}}(o_i|q, o_{<i})}{\pi_\theta(o_i|q, o_{<i})}, 1 \pm \varepsilon\right) A_i\right) \tag{1}$$

One issue with this is A_i, which is computed using the **Generalized Advantage Estimator**[2] –

1. Estimates long-range trajectory rewards.
2. However, requires a neural reward model which is expensive to train and can be unstable.

[2] Schulman et. al [ICLR 2016]

RLHF Synopsis (1/2)

We'll review the RLHF pipeline per Zeiger et al. It has 3-primary phases:

1. **Supervised Fine-Tuning (SFT)**: A pre-trained LLM (π_{PT}) is fine-tuned on high-quality, domain-specific datasets to obtain π_{SFT}.

RLHF Synopsis (1/2)

We'll review the RLHF pipeline per Zeiger et al. It has 3-primary phases:

1. **Supervised Fine-Tuning (SFT)**: A pre-trained LLM (π_{PT}) is fine-tuned on high-quality, domain-specific datasets to obtain π_{SFT}.
2. **Reward Modelling:** Next, we obtain a reward model $r_\phi(x, y)$ that models user preferences.

RLHF Synopsis (1/2)

We'll review the RLHF pipeline per Zeiger et al. It has 3-primary phases:

1. **Supervised Fine-Tuning (SFT)**: A pre-trained LLM (π_{PT}) is fine-tuned on high-quality, domain-specific datasets to obtain π_{SFT}.
2. **Reward Modelling:** Next, we obtain a reward model $r_\phi(x, y)$ that models user preferences. We start by sampling from π_{SFT}:

$$\mathcal{D} := \{(x_j, y_1, y_2)\}_{i=1,j=1}^{N,K} \sim \pi_{SFT}(y|x), \{x_i\}_{i=1}^{K} \tag{2}$$

$$y_w \succ y_l | x \sim r^*(x, y) \; \forall \; (y_1, y_2) \in \mathcal{D} \tag{3}$$

where $r^*(x, y)$ is the unknown optimal policy.

RLHF Synopsis (1/2)

We'll review the RLHF pipeline per Zeiger et al. It has 3-primary phases:

1. **Supervised Fine-Tuning (SFT)**: A pre-trained LLM (π_{PT}) is fine-tuned on high-quality, domain-specific datasets to obtain π_{SFT}.
2. **Reward Modelling:** Next, we obtain a reward model $r_\phi(x, y)$ that models user preferences. We start by sampling from π_{SFT}:

$$\mathcal{D} := \{(x_j, y_1, y_2)\}_{i=1,j=1}^{N,K} \sim \pi_{SFT}(y|x), \{x_i\}_{i=1}^{K} \quad (2)$$

$$y_w \succ y_l | x \sim r^*(x, y) \; \forall \; (y_1, y_2) \in \mathcal{D} \quad (3)$$

where $r^*(x, y)$ is the unknown optimal policy. Per Bradley-Terry:

$$p^*(y_1 \succ y_2 | x) = \sigma_{softmax[y_1]}(r^*(x, y)); \; y \in \{y_1, y_2\} \quad (4)$$

is the preference distribution optimized over negative log-likelihood on a parameterized model $r_\phi(x, y)$.

RLHF Synopsis (1/2)

We'll review the RLHF pipeline per Zeiger et al. It has 3-primary phases:

1. **Supervised Fine-Tuning (SFT)**: A pre-trained LLM (π_{PT}) is fine-tuned on high-quality, domain-specific datasets to obtain π_{SFT}.
2. **Reward Modelling:** Next, we obtain a reward model $r_\phi(x, y)$ that models user preferences. We start by sampling from π_{SFT}:

$$\mathcal{D} := \{(x_j, y_1, y_2)\}_{i=1,j=1}^{N,K} \sim \pi_{SFT}(y|x), \{x_i\}_{i=1}^{K} \tag{2}$$

$$y_w \succ y_l | x \sim r^*(x, y) \; \forall \; (y_1, y_2) \in \mathcal{D} \tag{3}$$

   where $r^*(x, y)$ is the unknown optimal policy. Per Bradley-Terry:

$$p^*(y_1 \succ y_2 | x) = \sigma_{softmax[y_1]}(r^*(x, y)); \; y \in \{y_1, y_2\} \tag{4}$$

   is the preference distribution optimized over negative log-likelihood on a parameterized model $r_\phi(x, y)$. Some notes:

   a. Rewards are normalized over x to motivate lower variance.

RLHF Synopsis (1/2)

We'll review the RLHF pipeline per Zeiger et al. It has 3-primary phases:

1. **Supervised Fine-Tuning (SFT)**: A pre-trained LLM (π_{PT}) is fine-tuned on high-quality, domain-specific datasets to obtain π_{SFT}.
2. **Reward Modelling:** Next, we obtain a reward model $r_\phi(x, y)$ that models user preferences. We start by sampling from π_{SFT}:

$$\mathcal{D} := \{(x_j, y_1, y_2)\}_{i=1,j=1}^{N,K} \sim \pi_{SFT}(y|x), \{x_i\}_{i=1}^{K} \quad (2)$$

$$y_w \succ y_l | x \sim r^*(x, y) \; \forall \; (y_1, y_2) \in \mathcal{D} \quad (3)$$

where $r^*(x, y)$ is the unknown optimal policy. Per Bradley-Terry:

$$p^*(y_1 \succ y_2|x) = \sigma_{softmax[y_1]}(r^*(x, y)); \; y \in \{y_1, y_2\} \quad (4)$$

is the preference distribution optimized over negative log-likelihood on a parameterized model $r_\phi(x, y)$. Some notes:

   a. Rewards are normalized over x to motivate lower variance.
   b. r_ϕ is π_{SFT} with the final linear layer returning the scalar reward.

RLHF Synopsis (2/2)

3. **RL Fine-Tuning:** Finally, we use r_ϕ to fine-tune π_{SFT}, with the following objectives:
   a. r_ϕ should be maximized. **Assumption:** $r^* \approx r_\phi$.

RLHF Synopsis (2/2)

3. **RL Fine-Tuning:** Finally, we use r_ϕ to fine-tune π_{SFT}, with the following objectives:
   a. r_ϕ should be maximized. **Assumption:** $r^* \approx r_\phi$.
   b. We *do not* want mode-collapse (random tokens that maximize reward).

RLHF Synopsis (2/2)

3. **RL Fine-Tuning:** Finally, we use r_ϕ to fine-tune π_{SFT}, with the following objectives:
   a. r_ϕ should be maximized. **Assumption:** $r^* \approx r_\phi$.
   b. We *do not* want mode-collapse (random tokens that maximize reward). **Solution:** KL Divergence.

RLHF Synopsis (2/2)

3. **RL Fine-Tuning:** Finally, we use r_ϕ to fine-tune π_{SFT}, with the following objectives:
   a. r_ϕ should be maximized. **Assumption:** $r^* \approx r_\phi$.
   b. We *do not* want mode-collapse (random tokens that maximize reward). **Solution:** KL Divergence.

Mathematically, RLHF posits the following optimization problem:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)}(r_\phi(x, y)) - \beta \mathbb{D}_{KL}[\pi_\theta(y|x) \mid\mid \pi_{SFT}(y|x)] \quad (5)$$

RLHF Synopsis (2/2)

3. **RL Fine-Tuning:** Finally, we use r_ϕ to fine-tune π_{SFT}, with the following objectives:
   a. r_ϕ should be maximized. **Assumption:** $r^* \approx r_\phi$.
   b. We *do not* want mode-collapse (random tokens that maximize reward). **Solution:** KL Divergence.

   Mathematically, RLHF posits the following optimization problem:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)}(r_\phi(x, y)) - \beta \mathbb{D}_{KL}[\pi_\theta(y|x) \mid\mid \pi_{SFT}(y|x)] \quad (5)$$

   This is equivalent to the reward function:

$$r(x, y) = r_\phi(x, y) - \beta(\log \pi_\theta(y|x)) - \log(\pi_{SFT}(y|x)) \quad (6)$$

   Which is maximized using **Proximal Policy Optimization**.

GRPO (1/2)

For each question q, GRPO samples outputs $\{o_1, \ldots, o_G\}$ with the training objective being to maximize (1):

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

$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_{\theta_{old}}(o_i|q)}{\pi_{\theta}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_{\theta_{old}}(o_i|q)}{\pi_{\theta}(o_i|q)}, 1 \pm \varepsilon\right)A_i\right) + \beta D_{KL}(\pi_\theta||\pi_{\text{ref}})\right) \tag{7}$$

$$D_{KL}(\pi_\theta||\pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1 \tag{8}$$

$$A_i = \frac{r_i - \bar{r}}{\sigma_{std}(r)};\ \ r \in \{r_1, \ldots, r_G\} \tag{9}$$

GRPO (2/2)

We have defined everything except the reward function.

GRPO (2/2)

We have defined everything except the reward function. Unfortunately the paper also is vague about this

GRPO ($^2/_2$)

We have defined everything except the reward function. Unfortunately the paper also is vague about this; it specifies a rule-based system:

1. **Accuracy rewards:** For objective tasks, correctness $\propto$ reward .

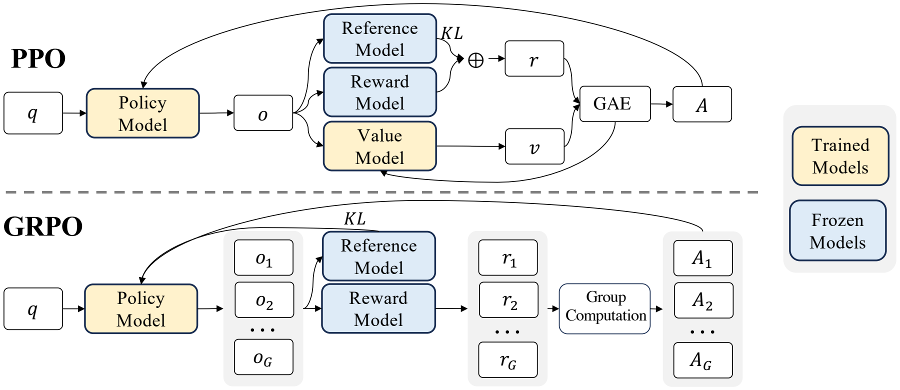
GRPO (2/2)

We have defined everything except the reward function. Unfortunately the paper also is vague about this; it specifies a rule-based system:

1. **Accuracy rewards:** For objective tasks, correctness $\propto$ reward .
2. **Format rewards:** For following the required format, it gets a reward.

We have defined everything except the reward function. Unfortunately the paper also is vague about this; it specifies a rule-based system:

1. **Accuracy rewards:** For objective tasks, correctness $\propto$ reward .
2. **Format rewards:** For following the required format, it gets a reward.



Outline

1. RL Review
2. Training Details
3. Performance Evals

Approach Overview

The training process can largely can be broken down into three stages:

1. **RL on a base model** – gets us DeepSeek-R1-Zero.
2. **RL+SFT on a checkpoint** – gets us DeepSeek-R1.
3. **Distillation** – gets us DeepSeek-R1-Distill-`<model-name>`.

We'll discuss each of these in detail next.

Less is More Philosophy

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

[3] https://arxiv.org/abs/2305.11206

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

Less (controlled) randomness, unbiased initialization in unstable regimes can avoid local optima and training failures.

[3] https://arxiv.org/abs/2305.11206

Less is More Philosophy

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

Less (controlled) randomness, unbiased initialization in unstable regimes can avoid local optima and training failures.

This is strong exemplified in the approaches used for R1's training:

[3] `https://arxiv.org/abs/2305.11206`

Less is More Philosophy

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

Less (controlled) randomness, unbiased initialization in unstable regimes can avoid local optima and training failures.

This is strong exemplified in the approaches used for R1's training:

1. *No need* for neural rewards, don't want reward hacking.

[3] `https://arxiv.org/abs/2305.11206`

Less is More Philosophy

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

Less (controlled) randomness, unbiased initialization in unstable regimes can avoid local optima and training failures.

This is strong exemplified in the approaches used for R1's training:

1. *No need* for neural rewards, don't want reward hacking.
2. *No need* for SFT, let GRPO figure it out from scratch.

[3] `https://arxiv.org/abs/2305.11206`

Less is More Philosophy

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

Less (controlled) randomness, unbiased initialization in unstable regimes can avoid local optima and training failures.

This is strong exemplified in the approaches used for R1's training:

1. *No need* for neural rewards, don't want reward hacking.
2. *No need* for SFT, let GRPO figure it out from scratch.
3. *No need* for complex prompting, don't bias RL towards approaches.

[3] `https://arxiv.org/abs/2305.11206`

Less is More Philosophy

A core finding from Less is More for Aligment (LIMA)[3] – a small set of *synethetic* examples encourages better generalization.

Less (controlled) randomness, unbiased initialization in unstable regimes can avoid local optima and training failures.

This is strong exemplified in the approaches used for R1's training:

1. *No need* for neural rewards, don't want reward hacking.
2. *No need* for SFT, let GRPO figure it out from scratch.
3. *No need* for complex prompting, don't bias RL towards approaches.

This, combined with **carefully selected data** are key to the incredible benchmark performance.

[3] `https://arxiv.org/abs/2305.11206`

RL on a Base Model

Start by taking a base model (no SFT, but models language well).

RL on a Base Model

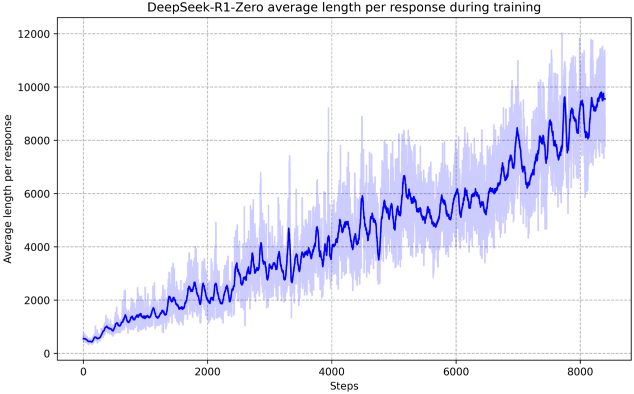
Start by taking a base model (no SFT, but models language well).

Train with RL until convergence on task.

RL on a Base Model

Start by taking a base model (no SFT, but models language well).

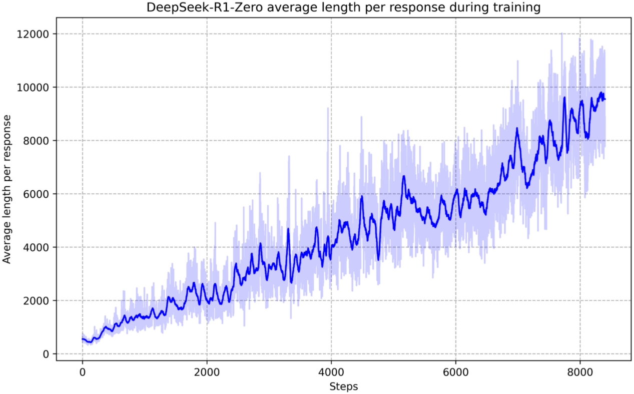
Train with RL until convergence on task. **Interesting consequence:**



RL on a Base Model

Start by taking a base model (no SFT, but models language well).

Train with RL until convergence on task. **Interesting consequence:**



Thinking more, and thinking anthropomorhically as emergent properties.

Challenges with Direct RL

There are a couple of negative consequences:

1. **Poor Readibility:** No rewards motivating readibility.

Challenges with Direct RL

There are a couple of negative consequences:

1. **Poor Readibility:** No rewards motivating readibility.
2. **Language Mixing:** No rewards motivating consistency of language – although this one should have been fairly easy to implement?

Challenges with Direct RL

There are a couple of negative consequences:

1. **Poor Readibility:** No rewards motivating readibility.
2. **Language Mixing:** No rewards motivating consistency of language – although this one should have been fairly easy to implement?

Aside

They did figure this out for R1, but it degraded performance slightly.

Challenges with Direct RL

There are a couple of negative consequences:

1. **Poor Readibility:** No rewards motivating readibility.
2. **Language Mixing:** No rewards motivating consistency of language – although this one should have been fairly easy to implement?

Aside

They did figure this out for R1, but it degraded performance slightly.

Solution? **Cold Start**. Use SFT *briefly*, to encourage approaching problems consistently.

Cold Start

So, what is cold start?
Idea: Collect small amount of long CoT data; to fine-tune the base model as the initial actor.

Cold Start

So, what is cold start?
Idea: Collect small amount of long CoT data; to fine-tune the base model as the initial actor.

From here, we expect the local optima to find a solution that leverages readibility but *prevents* language mixing.

Cold Start

So, what is cold start?
Idea: Collect small amount of long CoT data; to fine-tune the base model as the initial actor.

From here, we expect the local optima to find a solution that leverages readibility but *prevents* language mixing.

How to actually collect the data?

Cold Start

So, what is cold start?
Idea: Collect small amount of long CoT data; to fine-tune the base model as the initial actor.

From here, we expect the local optima to find a solution that leverages readibility but *prevents* language mixing.

How to actually collect the data? They used a couple of strategies:

1. Few shot prompting using a long CoT as examples.

Cold Start

So, what is cold start?
Idea: Collect small amount of long CoT data; to fine-tune the base model as the initial actor.

From here, we expect the local optima to find a solution that leverages readibility but *prevents* language mixing.

How to actually collect the data? They used a couple of strategies:

1. Few shot prompting using a long CoT as examples.
2. Prompting for detailed explanations, with reflection & verification.

Cold Start

So, what is cold start?
Idea: Collect small amount of long CoT data; to fine-tune the base model as the initial actor.

From here, we expect the local optima to find a solution that leverages readibility but *prevents* language mixing.

How to actually collect the data? They used a couple of strategies:

1. Few shot prompting using a long CoT as examples.
2. Prompting for detailed explanations, with reflection & verification.
3. Manually refining DeepSeek-R1-Zero outputs to remove language mixing, fixing readibility and general response refinment.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.
2. Now, incorporate generative reward model – DeepSeek-V3.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.
2. Now, incorporate generative reward model – DeepSeek-V3.
3. Sample multiple responses and retain correct ones.

Overall, we get from this $\approx$ 600K reasoning related training samples.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.
2. Now, incorporate generative reward model – DeepSeek-V3.
3. Sample multiple responses and retain correct ones.

Overall, we get from this $\approx$ 600K reasoning related training samples.

Next, add $\approx$ 200K tokens for factual QA, self-cognition, translation, etc.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.
2. Now, incorporate generative reward model – DeepSeek-V3.
3. Sample multiple responses and retain correct ones.

Overall, we get from this $\approx$ 600K reasoning related training samples.

Next, add $\approx$ 200K tokens for factual QA, self-cognition, translation, etc.

Add CoT where relevant, using DeepSeek-V3.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.
2. Now, incorporate generative reward model – DeepSeek-V3.
3. Sample multiple responses and retain correct ones.

Overall, we get from this $\approx$ 600K reasoning related training samples.

Next, add $\approx$ 200K tokens for factual QA, self-cognition, translation, etc.

Add CoT where relevant, using DeepSeek-V3.

SFT DeepSeek-V3-Base for 2 epochs over $\approx$ 800K sample dataset.

Rejection Sampling & Supervised Fine-Tuning

When RL converges, use the training checkpoint to sample data from domains beyond just reasoning: creative writing, role-playing, general purpose tasks.

Beyond Rule-Based Rewards Modelling:

1. Curate reasoning prompts and rejection sampling from the checkpoint.
2. Now, incorporate generative reward model – DeepSeek-V3.
3. Sample multiple responses and retain correct ones.

Overall, we get from this $\approx$ 600K reasoning related training samples.

Next, add $\approx$ 200K tokens for factual QA, self-cognition, translation, etc.

Add CoT where relevant, using DeepSeek-V3.

SFT DeepSeek-V3-Base for 2 epochs over $\approx$ 800K sample dataset. **Profit**.

Secondary RL Stage

This part is *exactly* RLHF.

Secondary RL Stage

This part is *exactly* RLHF.

Helps with: improving helpfulness and harmlessness.

Secondary RL Stage

This part is *exactly* RLHF.

Helps with: improving helpfulness and harmlessness.

Quiet Part Out Loud

Probably was used to filter out Tiananmen Square Massacre information, whatever additional censorship they desire.

Distillation

Further, they fine-tuned open source models like Qwen & LLaMa over DeepSeek-R1's responses.

Distillation

Further, they fine-tuned open source models like Qwen & LLaMa over DeepSeek-R1's responses.

Here, used SFT and not RL even though they think RL would peform better.

Distillation

Further, they fine-tuned open source models like Qwen & LLaMa over DeepSeek-R1's responses.

Here, used SFT and not RL even though they think RL would peform better.

Opinion

I disagree, they could've used RL to fine-tune DeepSeek-V3-Base with RL-checkpointed data but chose SFT.

What Didn't Work & Why

Process Reward Model: Prone to reward hacking, requires fine-grained setup, evaluating intermediate steps are challenging for general reasoning.

What Didn't Work & Why

Process Reward Model: Prone to reward hacking, requires fine-grained setup, evaluating intermediate steps are challenging for general reasoning.

Monte-Carlo Tree Search: Enable explore the solution space systematically; but search space is ill-designed and scaling is restrictive.

What Didn't Work & Why

Process Reward Model: Prone to reward hacking, requires fine-grained setup, evaluating intermediate steps are challenging for general reasoning.

Monte-Carlo Tree Search: Enable explore the solution space systematically; but search space is ill-designed and scaling is restrictive.

Setting restrictions leads to spurious local optima, and also requires a pre-trained value model which is expensive.

What Didn't Work & Why

Process Reward Model: Prone to reward hacking, requires fine-grained setup, evaluating intermediate steps are challenging for general reasoning.

Monte-Carlo Tree Search: Enable explore the solution space systematically; but search space is ill-designed and scaling is restrictive.

Setting restrictions leads to spurious local optima, and also requires a pre-trained value model which is expensive.

Aside

Approaches do exist that combine these two and exhibit strong performance in specific specialized domains.[a][b]

[a] https://arxiv.org/abs/2406.03816
[b] https://arxiv.org/abs/2501.07301

Drawbacks

The authors note the following limitations of their approach:

1. Their reasoning focused approach triggers some catastrophic forgetting.

Drawbacks

The authors note the following limitations of their approach:

1. Their reasoning focused approach triggers some catastrophic forgetting.
2. It reasons in English / Chinese even when query language is neither.

Drawbacks

The authors note the following limitations of their approach:

1. Their reasoning focused approach triggers some catastrophic forgetting.
2. It reasons in English / Chinese even when query language is neither.
3. It doesn't work well with few-shot prompting.

Drawbacks

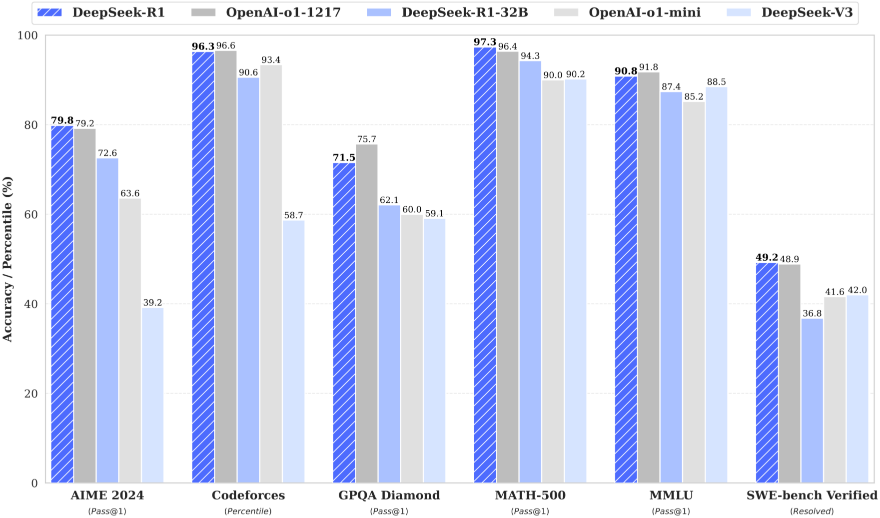
The authors note the following limitations of their approach:

1. Their reasoning focused approach triggers some catastrophic forgetting.
2. It reasons in English / Chinese even when query language is neither.
3. It doesn't work well with few-shot prompting.
4. It doesn't work well for SWE tasks.

Outline

1. RL Review
2. Training Details
3. Performance Evals

Reasoning



DeepSeek-R1
OpenAI-o1-1217
DeepSeek-R1-32B
OpenAI-o1-mini
DeepSeek-V3
Accuracy / Percentile (%)
100
80
60
40
20
0
79.8
79.2
72.6
63.6
39.2
96.3
96.6
90.6
93.4
58.7
71.5
75.7
62.1
60.0
59.1
97.3
96.4
94.3
90.0
90.2
90.8
91.8
87.4
85.2
88.5
49.2
48.9
36.8
41.6
42.0
AIME 2024
(Pass@1)
Codeforces
(Percentile)
GPQA Diamond
(Pass@1)
MATH-500
(Pass@1)
MMLU
(Pass@1)
SWE-bench Verified
(Resolved)

Other Stuff

[Refer to paper directly]

Thank you!

Have an awesome rest of your day!

Slides: `https://jinen.setpal.net/slides/dsr1.pdf`