SEMINAR
REASONING IN LANGUAGE MODELS

Advanced NLP - Monsoon'26

By Abdul Kadir Nalawala - 2026701019

downloaded from - okabdul.com/reasoning

16 Sept. 2026

TABLE OF CONTENTS

- What is Reasoning?
- Why we need Reasoning?
- Methods for Eliciting & Training Reasoning in LLMs
- Efficient Reasoning
- Recent Advances

There are a lot of things wrong with this world...
but too much intelligence is not one of them.

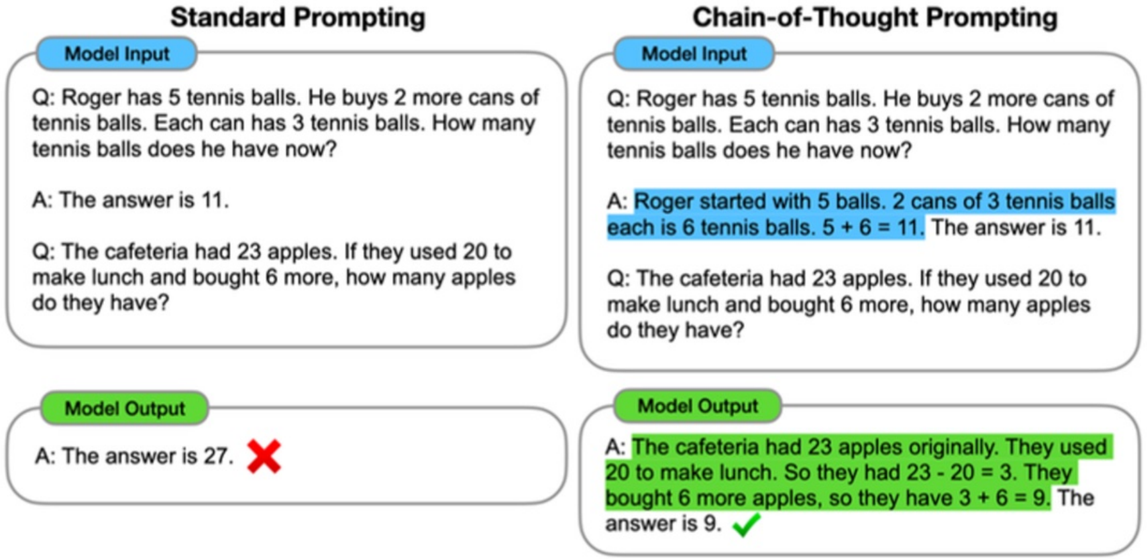
- *Richard Sutton*

What is Reasoning?

– The ability to solve a problem or navigate through a scenario by piecing together prior knowledge and facts

Why we need Reasoning?

– Consider how a human solves a complex task such as multi-step math problem: by reasoning through a series of intermediate steps before arriving at the final answer. [1] showed that eliciting such intermediate reasoning steps in LMs can greatly improve performance on a range of complex reasoning tasks.



Chain of Thought example - [1]

[1] Wei et al. - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Neural Information Processing Systems, 2022.

– Models can fail when familiar surface patterns are preserved but the underlying problem-solving requirements change.

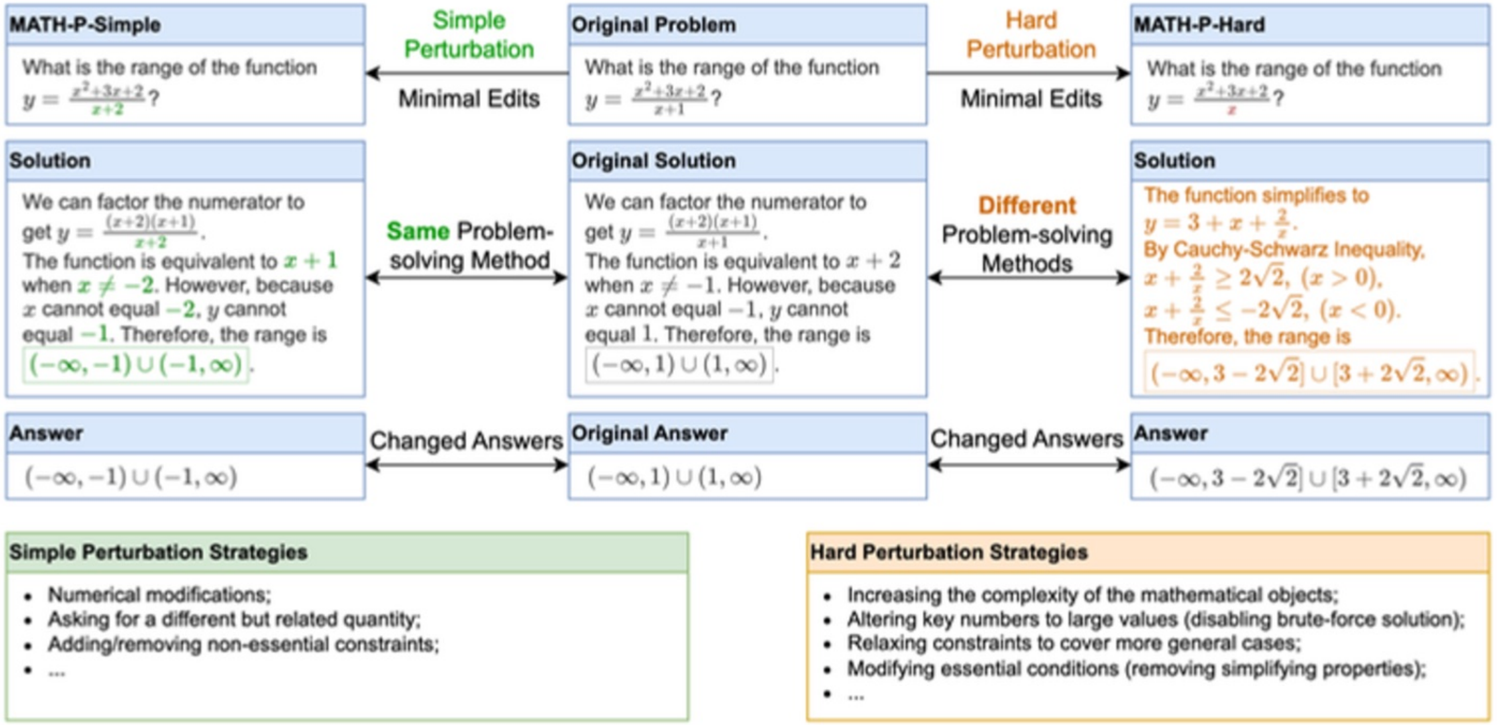


Illustration of the annotation process for MATH-P-Simple and MATH-P-Hard - [2]

[2] Huang et al. - Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations, 2025

Model	Mode	Original			MATH-P-Easy				MATH-P-Hard			
		All	Train	Test	All	Train	Test	Δ All	All	Train	Test	Δ All
Gemini-2.0-flash-thinking-exp	Reasoning	92.47	92.68	92.17	91.04	87.80	95.65	1.43	78.14	77.44	79.13	14.33
o1-preview	Reasoning	87.81	88.41	86.96	87.81	87.80	87.83	0.00	72.40	73.78	70.43	15.41
o1-mini	Reasoning	94.27	93.90	94.78	94.98	93.29	97.39	-0.71	78.49	79.27	77.39	15.78
Gemini-2.0-flash-exp	Non-Reasoning	88.17	87.20	89.57	82.80	81.71	84.35	5.37	67.03	68.29	65.22	21.14
Gemini-1.5-pro	Non-Reasoning	77.78	77.44	78.26	77.42	76.83	78.26	0.36	56.63	56.10	57.39	21.15
GPT-4o	Non-Reasoning	67.03	68.90	64.35	62.01	60.98	63.48	5.02	39.43	37.80	41.74	27.60
GPT-4-turbo	Non-Reasoning	56.99	55.49	59.13	55.20	56.71	53.04	1.79	34.41	36.59	31.30	22.58
Claude-3.5-Sonnet	Non-Reasoning	64.52	62.80	66.96	58.42	57.32	60.00	6.10	38.71	38.41	39.13	25.81
Claude-3-Opus	Non-Reasoning	41.94	39.02	46.09	41.94	39.63	45.22	0.00	26.52	25.00	28.70	15.42

Zero-shot CoT performance of LLMs across three sets: All, train & test - [2]

[2] Huang et al. - Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations, 2025

Methods for Eliciting & Training Reasoning in LLMs

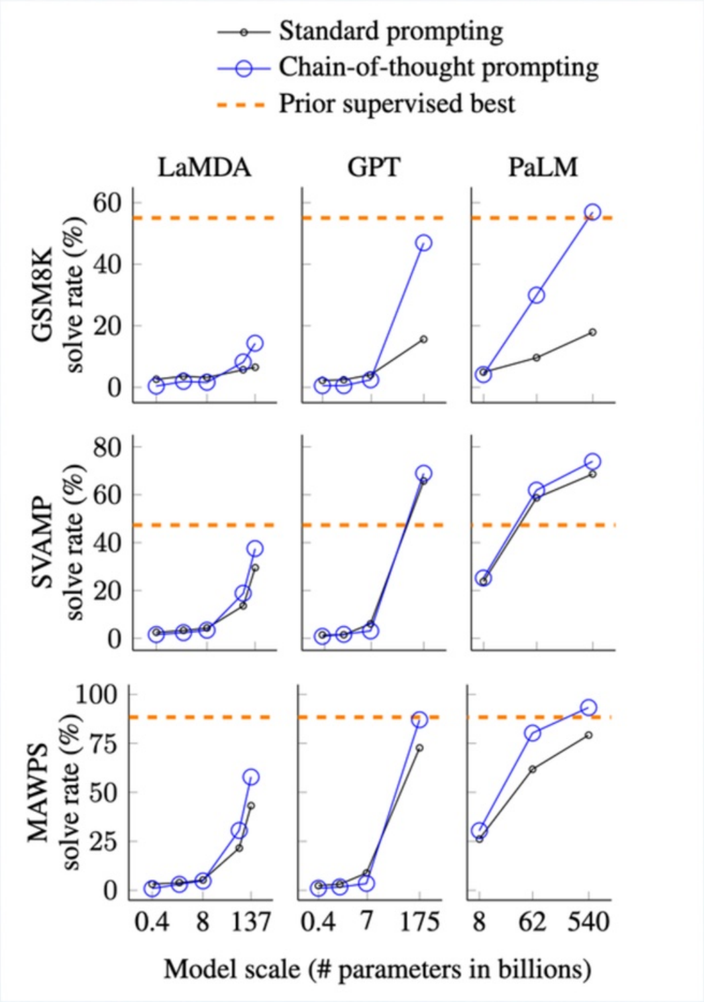
1. Chain of Thought (CoT) - [1]

– A series of intermediate reasoning steps (CoT) significantly improves the ability of large language models to perform complex reasoning.

– Works on LLMs since they are capable of in-context few-shot learning via prompting.

[1] Wei et al. - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Neural Information Processing Systems, 2022.

1. Chain-of-Thought (CoT) - [1]



Chain-of-Thought prompting enables LLMs to solve challenging math problems. It can also be observed that chain-of-thought reasoning is an emergent ability of increasing model size - [1]

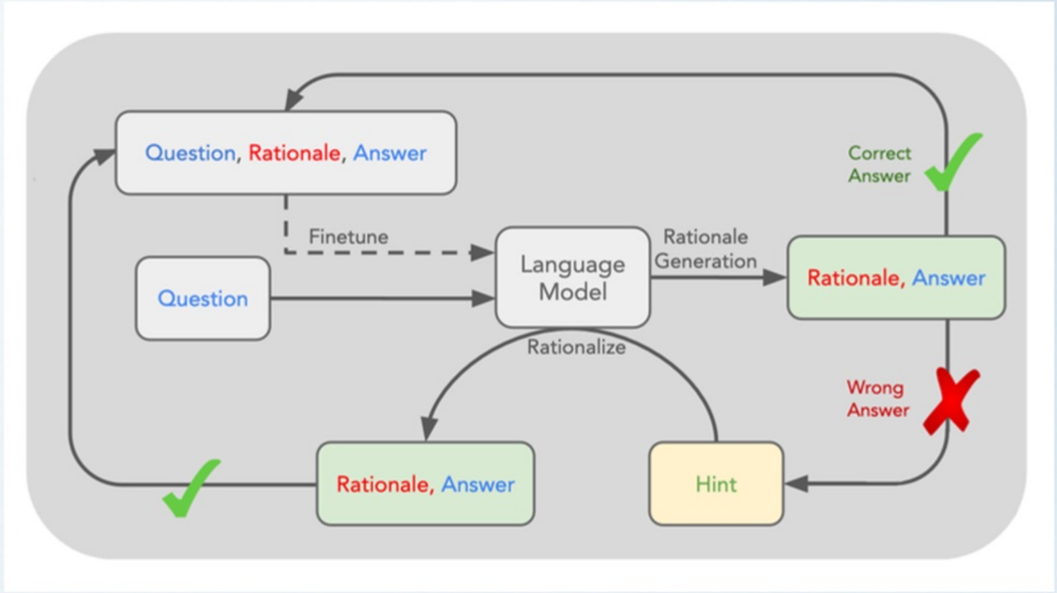
[1] Wei et al. - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Neural Information Processing Systems, 2022.

2. Supervised FineTuning (SFT) on Reasoning Traces

– Instead of elicting reasoning via prompting we train a model via SFT.

– [3] proposed a method in which we generate multiple reasoning traces and fine-tune the model on reasoning-traces that have correct answer.

– Since this loop eventually fails to solve any new problems in the training set because it receives no direct training signal for the questions it couldn't solve (wrong answer), they propose *rationalization*. This makes it a Self-Taught Reasoner (STaR)



Overview of STaR and STaR-generated rationale - [3]

[3] Zelikman, E., Wu, Y., Mu, J., & Goodman, N. D. (2022). STaR: Bootstrapping Reasoning With Reasoning. ArXiv, abs/2203.14465.

2. Supervised FineTuning (SFT) on Reasoning Traces

	GSM8K Test Accuracy (%)	Train Data Used (%)
Few-shot Direct GPT-J	3.0	~0
Few-shot CoT GPT-J	3.1	~0
GPT-J Direct Finetuned	5.8	100
STaR without rationalization	10.1	25.0
STaR with rationalization	**10.7**	28.7

Performance of STaR vs other SOTA methods -[3]

[3] Zelikman, E., Wu, Y., Mu, J., & Goodman, N. D. (2022). STaR: Bootstrapping Reasoning With Reasoning. ArXiv, abs/2203.14465.

3. Reinforcement Learning (RL)

– Law of Effect: *The greater the satisfaction or discomfort, the greater the strengthening or weakening of the bond.* - [4]

– Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal. - [5]

– How is it different from Supervised and Unsupervised learning?

– Explore & Exploit

[4] Edward L. Thorndike - Animal Intelligence: Experimental Studies, 1911
[5] Richard S. Sutton and Andrew G. Barto. - Reinforcement Learning: An Introduction. 2018

Elements of RL:

Element	General Meaning	For LMs
Agent π_θ	Learner/decision-maker that interacts with an environment to maximize expected cumulative reward.	The LM with parameters θ that generates tokens according to policy π_θ
Environment $\mathcal{E}$	External system the agent interacts with.	Everything outside the agent that responds to its action.
State s_t	Representation of the current situation the agent observes	Current prompt + tokens generated so far
Action a_t	A choice made by the agent in a state (acc. to its policy)	Generating the next token

Elements of RL:

Element	General Meaning	For LMs
Policy $\pi_\theta(a\|s)$	Specifies how agent chooses an action given a state	LM's next-token probability distribution - P_θ
Reward r_t/R	Scalar feedback showing how desirable/good or undesirable/bad the action is.	$R(x, y)$ scalar feedback assigned to the generated response/trajectory.
Trajectory τ	Sequence of states, actions & rewards in one interaction	$\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T\}$ Generation Prompt + Response + Rewards

RL Objective:

– Goal of RL while training LMs: $$\pi^* = \arg\max_{\pi} \underbrace{J(\pi)}_{\substack{\text{Expected return} \\ \text{generated using } \pi}}$$

– Standard SGD: $$\Theta_{k+1} = \Theta_k - \eta \nabla_\Theta L(\Theta_k)$$

– Vanilla Gradient Ascent: $$\Theta_{k+1} = \Theta_k + \eta \nabla_\Theta J(\Theta_k)$$

– Average Reward over all Trajectories, that the policy might generate:

$$J(\pi) = \int_\tau P(\tau \mid \pi)\, R(\tau) = \mathbb{E}_{\tau \sim P(\tau|\pi)} \left[R(\tau) \right]$$

RL Objective:

– Probability of a Trajectory: $P(\tau \mid \pi) = \underbrace{p_0(s_0)}_{\substack{\text{prob. that env.} \\ \text{starts in state } s_0}} \prod_{t=1}^{T} \underbrace{P(s_t \mid s_{t-1}, a_{t-1})}_{\substack{\text{prob. that env.} \\ \text{transitions to } s_t \\ \text{given state } s_{t-1} \\ \text{and action } a_{t-1}}} \underbrace{\pi(a_{t-1} \mid s_{t-1})}_{\substack{\text{prob. that} \\ \pi \text{ chooses } a_{t-1} \\ \text{after seeing } s_{t-1}}}$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[R(\tau) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_{t-1} \mid s_{t-1}) \right]$$

– The Expectation is over all possible Trajectories: $|\tau| = |\mathcal{V}|^N$ (non-tractable)

REINFORCE - [6]

– We use Monte-Carlo estimate (MC): sample a set N of trajectories and then estimate the gradient:

$$\nabla_\theta J(\pi_{\theta_0}) \approx \frac{1}{|N|} \sum_{\tau \in N} \sum_{t=1}^{T} \nabla_\theta \log \pi_{\theta_0}(a_{t-1} \mid s_{t-1})\, R(\tau)$$

[6] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8, 229–256.

TRPO - [7]

TRPO - [7]

– While REINFORCE tells us which direction improves the policy it doesn't sufficiently restrict how 'far' the new policy moves and it also has high-variance.

– Importance Ratio: $r_t(\theta) = \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$

$$\max_\theta \ \mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left[\sum_{t=0}^{T} \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} A_{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right]$$

$$\text{subject to} \qquad \mathbb{E}_t \left[D_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t) \right) \right] \leq \delta$$

[7] Schulman, J., Levine, S., Abbeel, P., Jordan, M.I., & Moritz, P. (2015). Trust Region Policy Optimization. International Conference on Machine Learning.

Some More Elements of RL:

$$A_{\pi_\theta}(s,a) = \underbrace{Q_{\pi_\theta}(s,a)}_{\text{Action-Value Function}} - \underbrace{V_{\pi_\theta}(s)}_{\text{State-Value Function}}$$

$Q_{\pi_\theta}(s,a)$
– Action - Value Function: expected cumulative reward when you start in state s & take action a and act according to the current policy π_θ.

$V_{\pi_\theta}(s)$
– State-Value Function: expected cumulative reward when you start in state s.

$A_{\pi_\theta}(s,a)$
– Captures how much better or worse the action a is relative to the policy's expected performance from state s.

PPO - [8]

PPO - [8]

– To reduce high variance of REINFORCE we use a more informative *baseline*

– We drop the KL constrain & introduce a clipping mechanism into the objective that discourages large updates.

$$\max_{\theta} \mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left[\sum_{t=0}^{T} \min \left(\underbrace{\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} A_{\pi_{\theta_{\text{old}}}}(s_t, a_t)}_{\text{Unclipped objective}}, \underbrace{\text{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon \right) A_{\pi_{\theta_{\text{old}}}}(s_t, a_t)}_{\text{Clipped objective}} \right) \right]$$

[8] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. ArXiv, abs/1707.06347.

$$J_{\text{PPO}}(\theta, \gamma) = \underbrace{\mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left[\sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} A_{\pi_{\theta_{\text{old}}}}(s_t, a_t),\ \text{clip} \left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon \right) A_{\pi_{\theta_{\text{old}}}}(s_t, a_t) \right) \right]}_{\text{Maximise reward}}$$

$$+ \underbrace{w_1 H(\pi_\theta)}_{\text{Maximise entropy}} \underbrace{- w_2 \, \mathbb{E}_t \left[D_{\text{KL}} \left(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_{\text{ref}}(\cdot \mid s_t) \right) \right]}_{\text{Penalise KL divergence}} \underbrace{- w_3 \, \mathcal{L}_V(\gamma)}_{\text{Critic L2}}$$

– We use a critic as the baseline: $A(x, y) = R(x, y) - V(x)$

RLHF - [9]

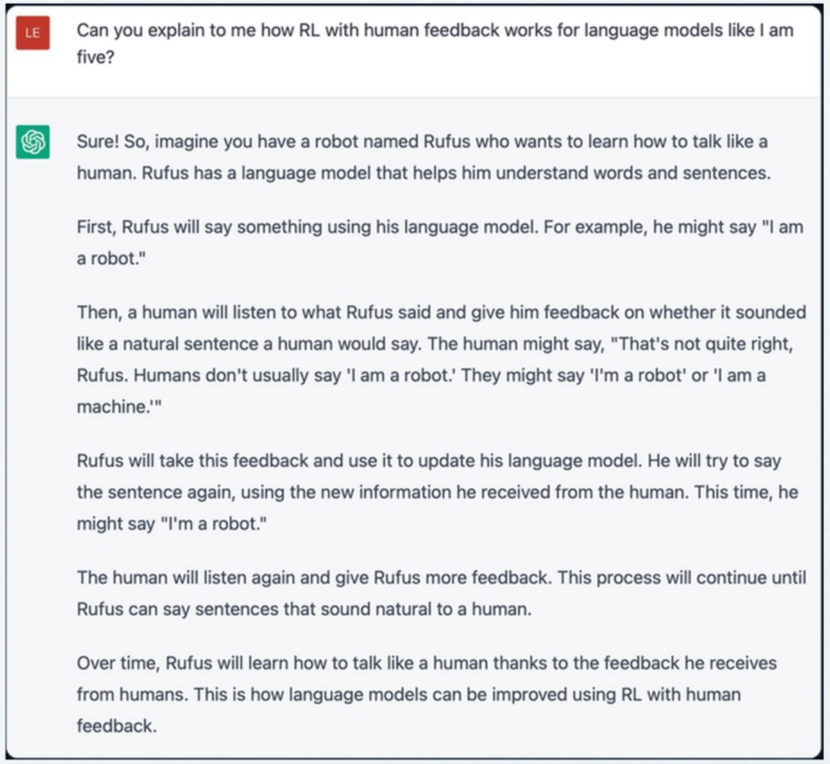
RLHF - [9]

– Cross-Entropy loss can't capture human preference/taste so we rely on metrics like BLEU & ROUGE

– But can we use Human Feedback on generated text as a measure of performance & then use that feedback as a reward signal that can optimize the LM?

[9] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

RLHF - [9]



LE

Can you explain to me how RL with human feedback works for language models like I am five?

Sure! So, imagine you have a robot named Rufus who wants to learn how to talk like a human. Rufus has a language model that helps him understand words and sentences.

First, Rufus will say something using his language model. For example, he might say "I am a robot."

Then, a human will listen to what Rufus said and give him feedback on whether it sounded like a natural sentence a human would say. The human might say, "That's not quite right, Rufus. Humans don't usually say 'I am a robot.' They might say 'I'm a robot' or 'I am a machine.'"

Rufus will take this feedback and use it to update his language model. He will try to say the sentence again, using the new information he received from the human. This time, he might say "I'm a robot."

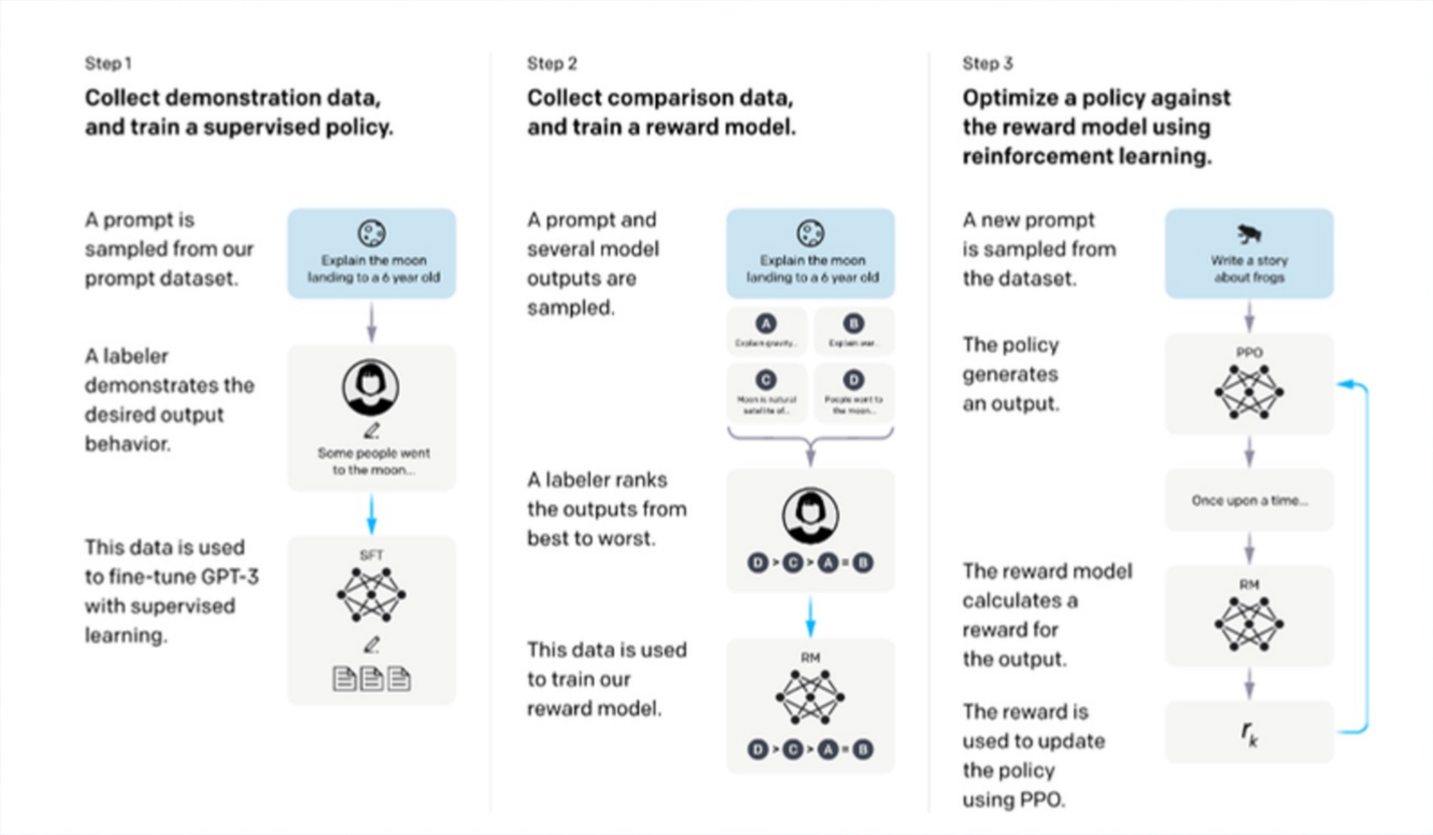
The human will listen again and give Rufus more feedback. This process will continue until Rufus can say sentences that sound natural to a human.

Over time, Rufus will learn how to talk like a human thanks to the feedback he receives from humans. This is how language models can be improved using RL with human feedback.

ChatGPT-3.5's response when asked to explain RLHF, from blog

[9] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

RLHF - [9]



RLHF method for training a LM proposed in [9]

[9] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

RLHF - [9]

– We train a Reward Model (RM) to minimize the following the objective:

$$\mathcal{L}(\theta) = -\log \sigma \left(R_\theta(x, y_i) - R_\theta(x, y_j) \right)$$

– It has been derived from the Bradley-Terry Model:

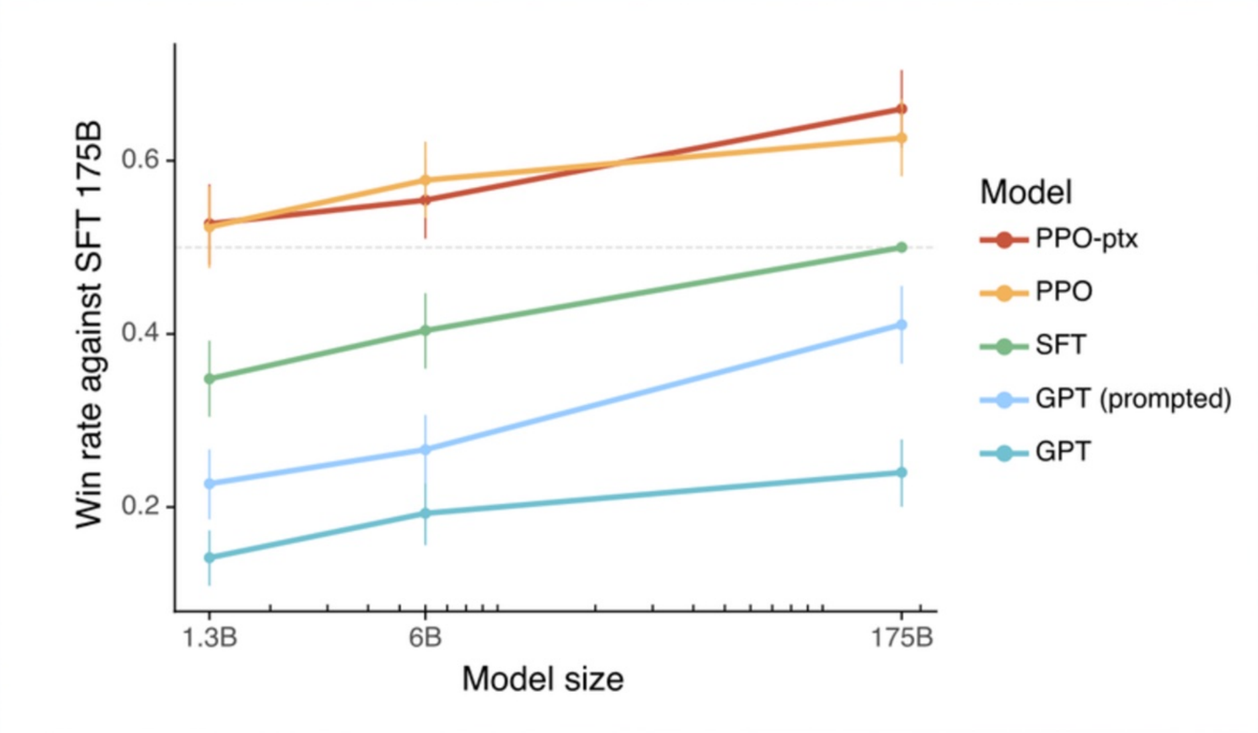
$$P\left(y_i \succ y_j \mid x\right) = \frac{\exp\left(R_\theta(x, y_i)\right)}{\exp\left(R_\theta(x, y_i)\right) + \exp\left(R_\theta(x, y_j)\right)}$$

– Training objective for RL phase (as given in [8]):

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta(y|x)} \left[R_\phi(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} \right] + \gamma \mathbb{E}_{x \sim \mathcal{D}_{\text{pretrain}}} \left[\log \pi_\theta(x) \right]$$

[9] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

RLHF - [9]



Human evaluations of various models (RLHF & SFT) - [9]

[9] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

RLOO - [10]

RLOO - [10]

– REINFORCE suffers from High Variance (uses entire trajectory's reward to scale gradient of every action)

– RLOO reduces the variance of the policy-gradient estimator without training a separate critic (aka value function) by using the rewards of the other sampled responses as a leave-one-out baseline.

$$A(x, y_i) = R(x, y_i) - \underbrace{\frac{1}{K-1} \sum_{\substack{j=1 \\ j \neq i}}^{K} R(x, y_j)}_{\text{Leave One Out Baseline}} = \frac{K}{K-1} \left(R(x, y_i) - \frac{1}{K} \sum_{j=1}^{K} R(x, y_j) \right)$$

[10] Wouter Kool, Herke van Hoof, Max Welling: Buy 4 REINFORCE Samples, Get a Baseline for Free! DeepRLStructPred@ICLR 2019

RLOO - [10]

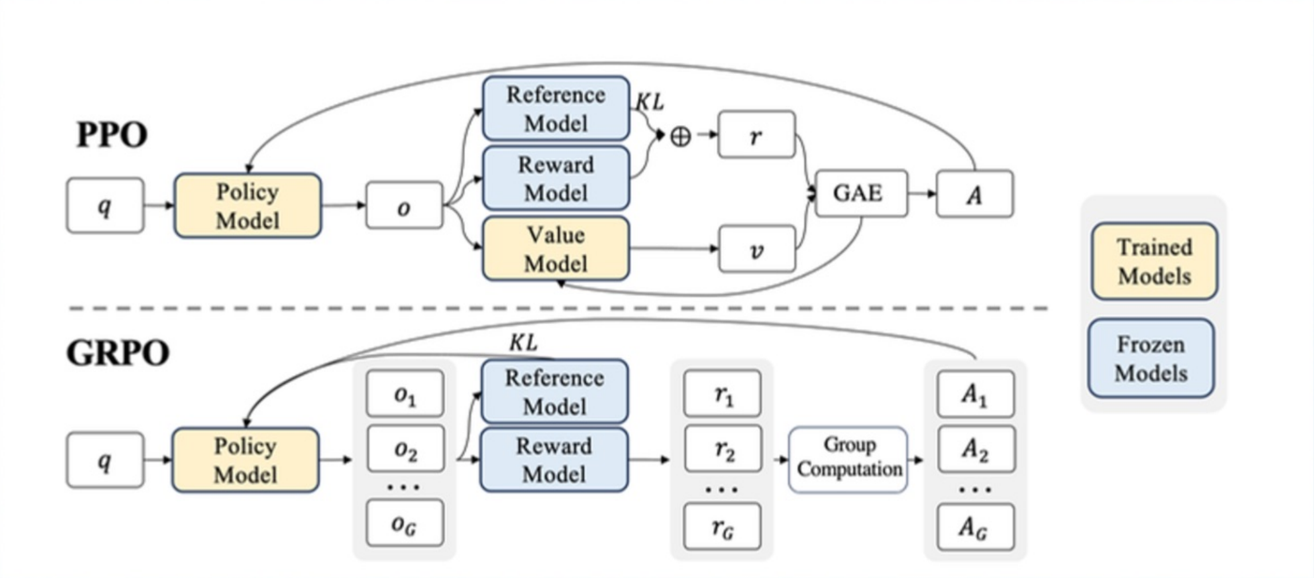
$$\nabla_\theta J(\pi_{\theta_0}) \approx \frac{1}{|N|} \sum_{\tau \in N} \sum_{t=1}^{T} \nabla_\theta \log \pi_{\theta_0}(a_{t-1} \mid s_{t-1}) \underbrace{A(\tau)}_{\text{Advantage}}$$

[10] Wouter Kool, Herke van Hoof, Max Welling: Buy 4 REINFORCE Samples, Get a Baseline for Free! DeepRLStructPred@ICLR 2019

GRPO - [11]

– PPO requires a separate critic/value model to estimate $V(s)$ and compute advantage.

– GRPO eliminates the critic by sampling multiple responses for the same prompt and using their group-relative rewards to estimate the advantage.



Demonstration of PPO and GRPO - [11]

[11] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. ArXiv, abs/2402.03300.

GRPO - [11]

$$J_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\left(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right) - \beta D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\right]$$

Advantage

$$A_i = \frac{R_i - \mu_R}{\sigma_R}$$

$$\sigma_R = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_j - \mu_R)^2}$$

$$\mu_R = \frac{1}{G}\sum_{j=1}^{G} R_j$$

Importance Ratio

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}$$

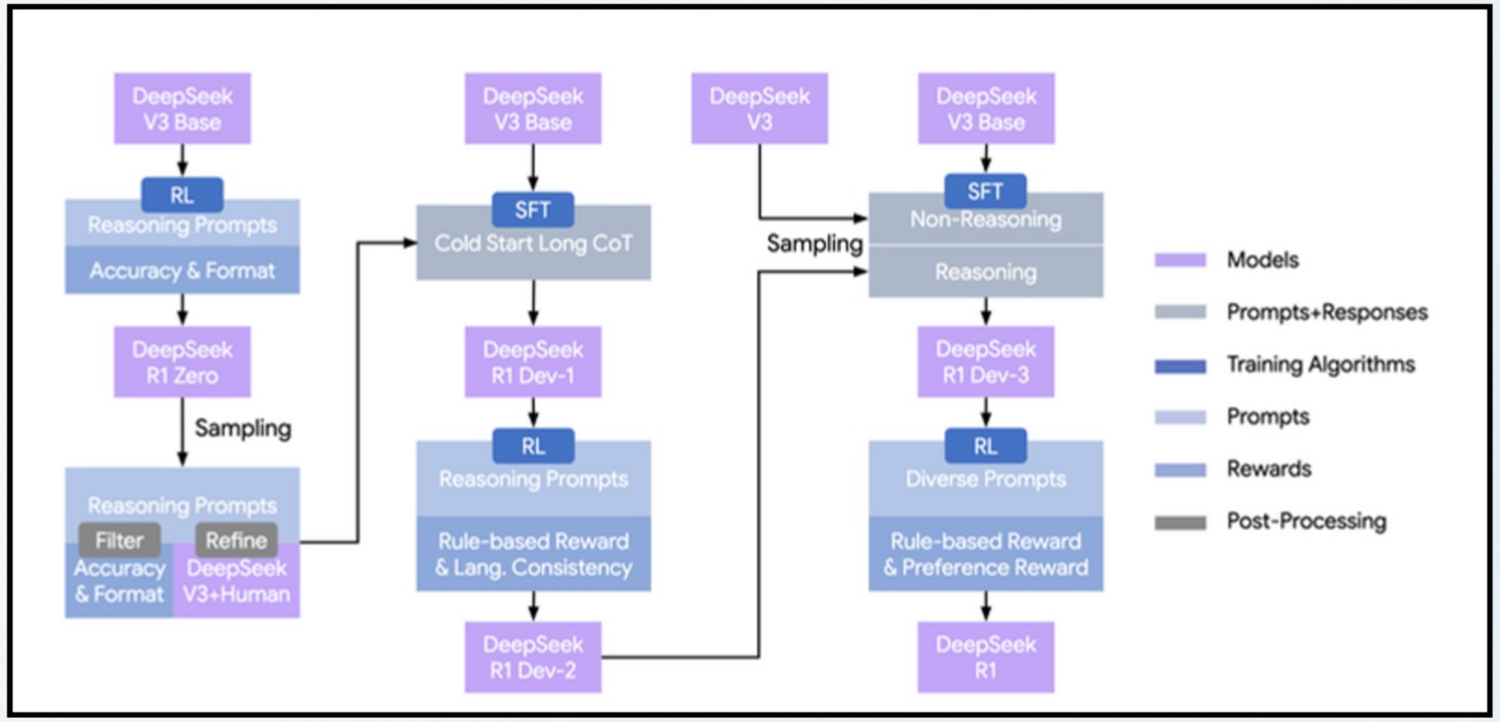
KL estimate

$$D_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}] = \frac{\pi_{\text{ref}}(y_{i,t} \mid x, y_{i,<t})}{\pi_\theta(y_{i,t} \mid x, y_{i,<t})} - \log\frac{\pi_{\text{ref}}(y_{i,t} \mid x, y_{i,<t})}{\pi_\theta(y_{i,t} \mid x, y_{i,<t})} - 1$$

[11] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. ArXiv, abs/2402.03300.

GRPO - [11]

– Use of rule-based reward system: $Reward_{rule} = Reward_{acc} + Reward_{format}$



Multi-stage pipeline of DeepSeek-R1 - [19]

[11] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. ArXiv, abs/2402.03300.

GRPO - [11]

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: $\cdots$

$\ldots$

"aha moment" of an intermediate version of DeepSeek-R1-Zero - [19]

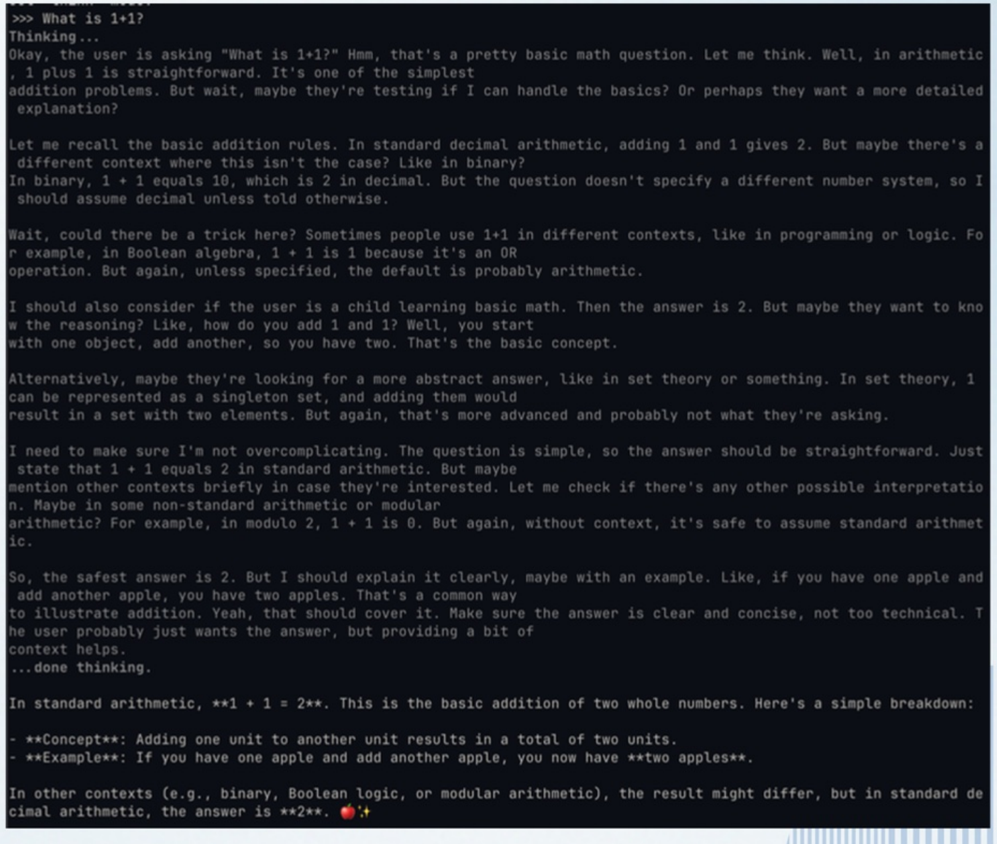
[11] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. ArXiv, abs/2402.03300.

Efficient Reasoning

– Reasoning is expensive

– Reasoning effort should change based on the difficulty of the task

```
abdulk@Abduls-MacBook-Air ~ % ollama run qwen3:8b-q8_0
>>> /set nothink
Set 'nothink' mode.
>>> What is 1+1?
1 + 1 equals 2.
```



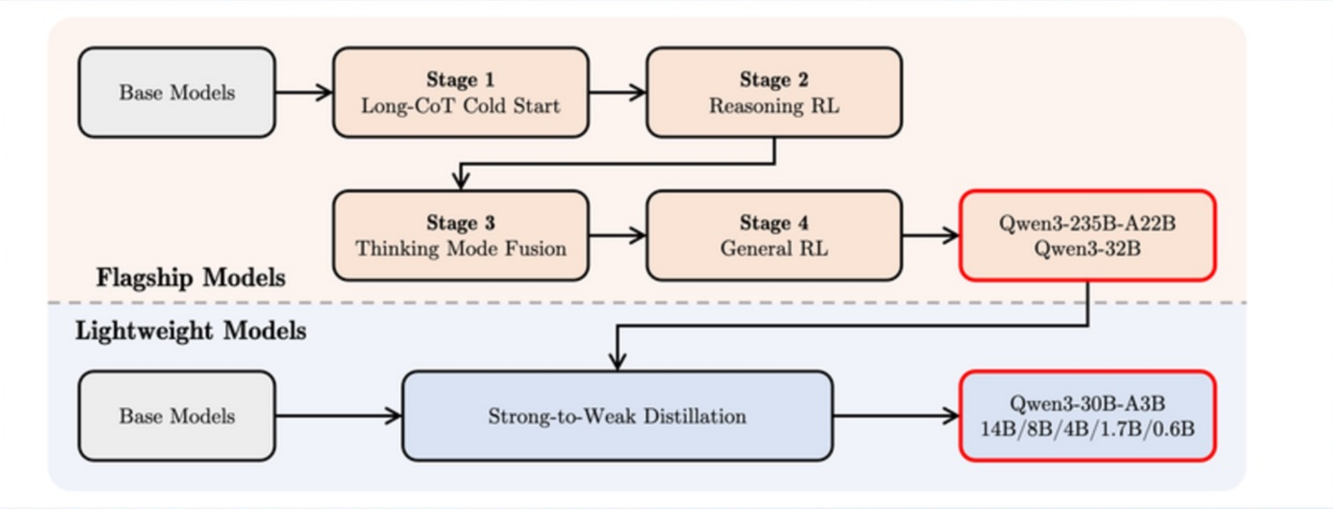
Thinking vs Non-thinking response via Qwen3:8b-q8 for the prompt “What is 1+1?”

Efficient Reasoning

Qwen3 - [12]

– Thinking Control: Integration of two modes - “thinking” and “non-thinking”

– Thinking Budget: Ability to handle intermediate cases i.e. generating responses based on incomplete thinking.



Post-training pipeline of the Qwen3 series models - [11]

Thinking Mode	Non-Thinking Mode
<|im_start|>user {query} /think<|im_end|> <|im_start|>assistant <think> {thinking_content} </think> {response}<|im_end|>	<|im_start|>user {query} /no_think<|im_end|> <|im_start|>assistant <think> </think> {response}<|im_end|>

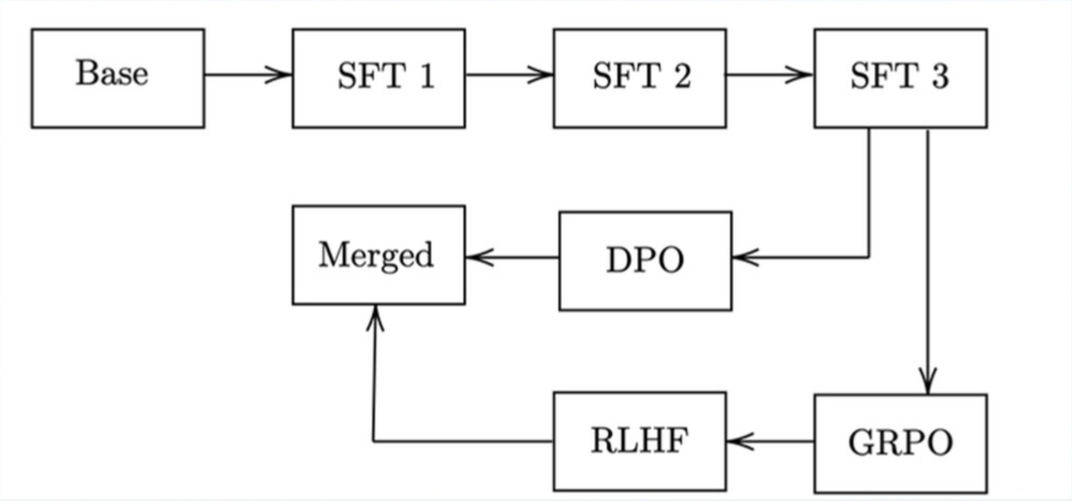
Examples of SFT data for thinking and non-thinking modes - [11]

[12] Yang, A., et al. (2025). Qwen3 Technical Report. ArXiv, abs/2505.09388.

Efficient Reasoning

Nemotron Nano 2 - [13]

– Explicit Thinking Budget



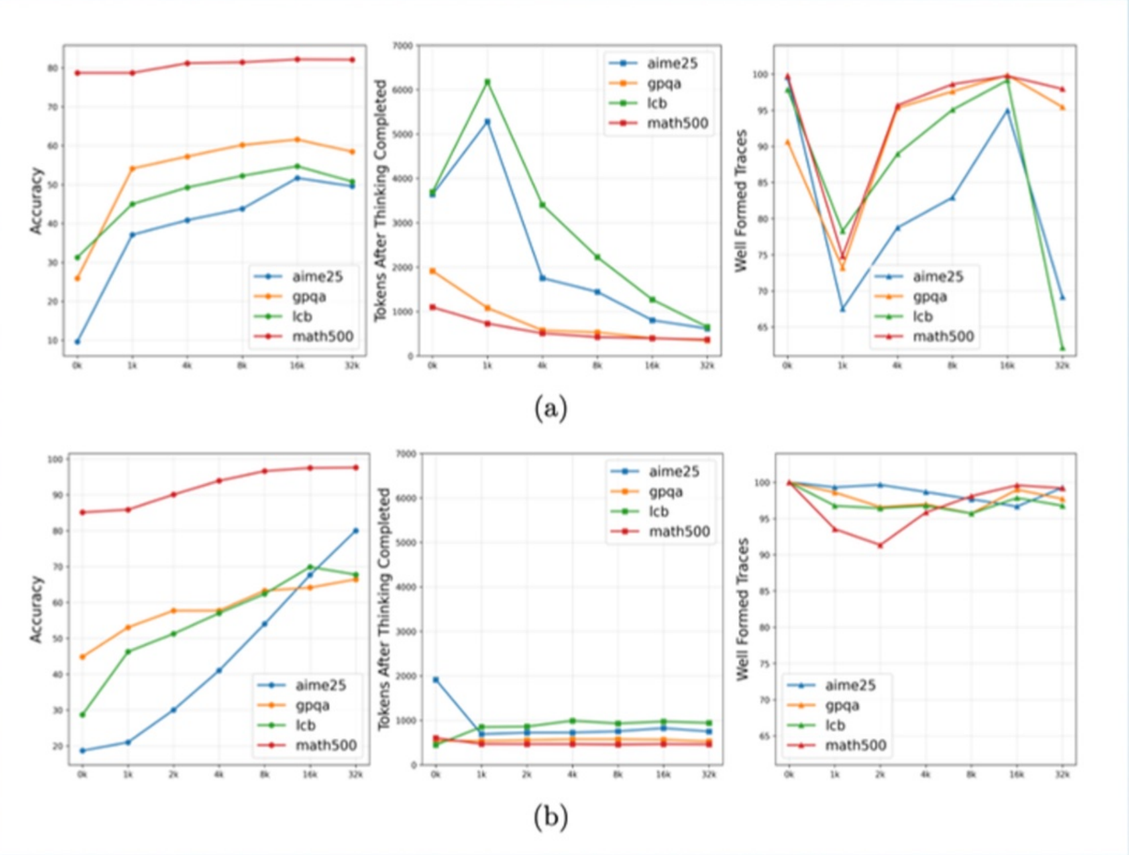
Flow of Post-training for Nemotron Nano 2 - [13]

[13] NVIDIA, A. Basant, A. Khairnar, et al. (2025). NVIDIA Nemotron Nano 2: An Accurate and Efficient Hybrid Mamba-Transformer Reasoning Model. ArXiv, abs/2508.14444.

Efficient Reasoning

Nemotron Nano 2 - [13]

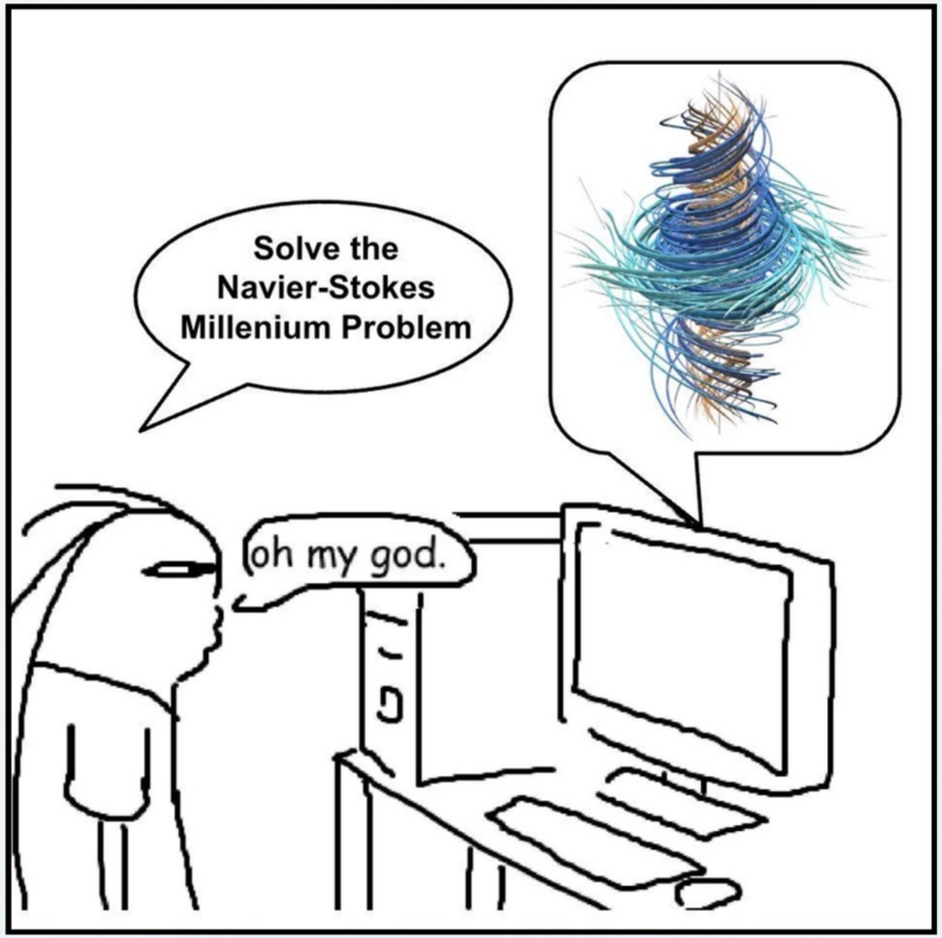
– Training model to tolerate truncation



Comparison of budget control truncation training (a) and after truncation training (b) - [13]

[13] NVIDIA, A. Basant, A. Khairnar, et al. (2025). NVIDIA Nemotron Nano 2: An Accurate and Efficient Hybrid Mamba-Transformer Reasoning Model. ArXiv, abs/2508.14444.

Recent Advances

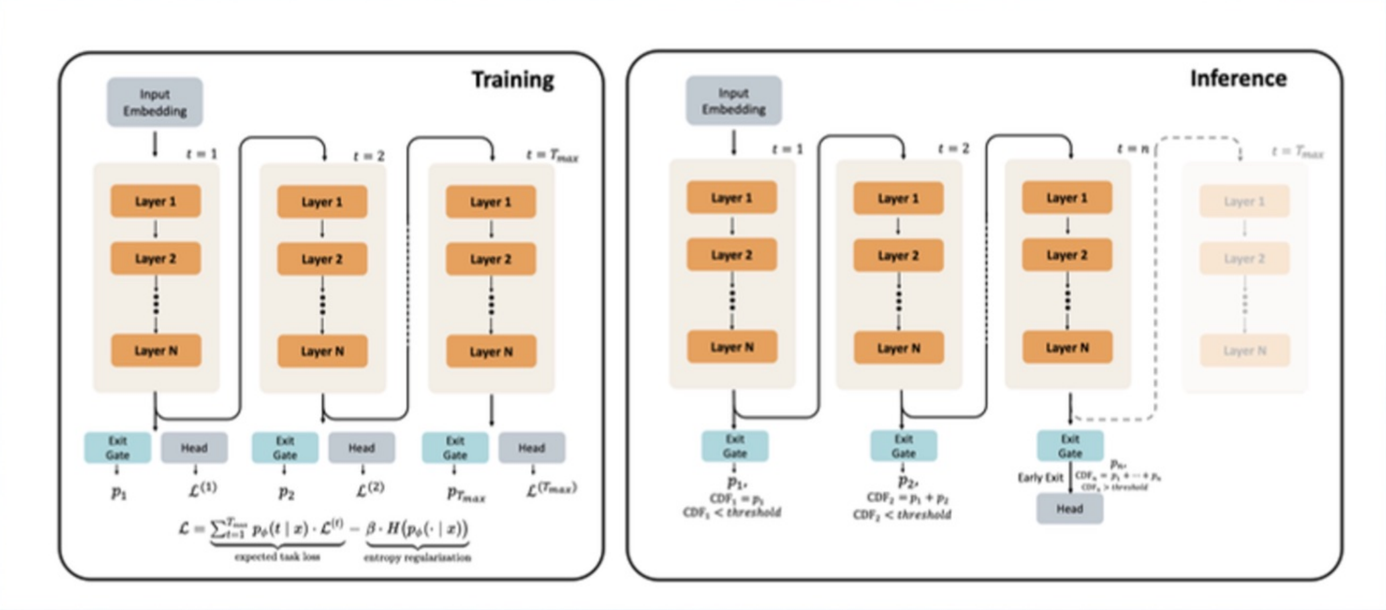


Recent Advances

– Agentic Reasoning: Foundational capabilities, Self-evolving adaptation & Collective coordination - [14]

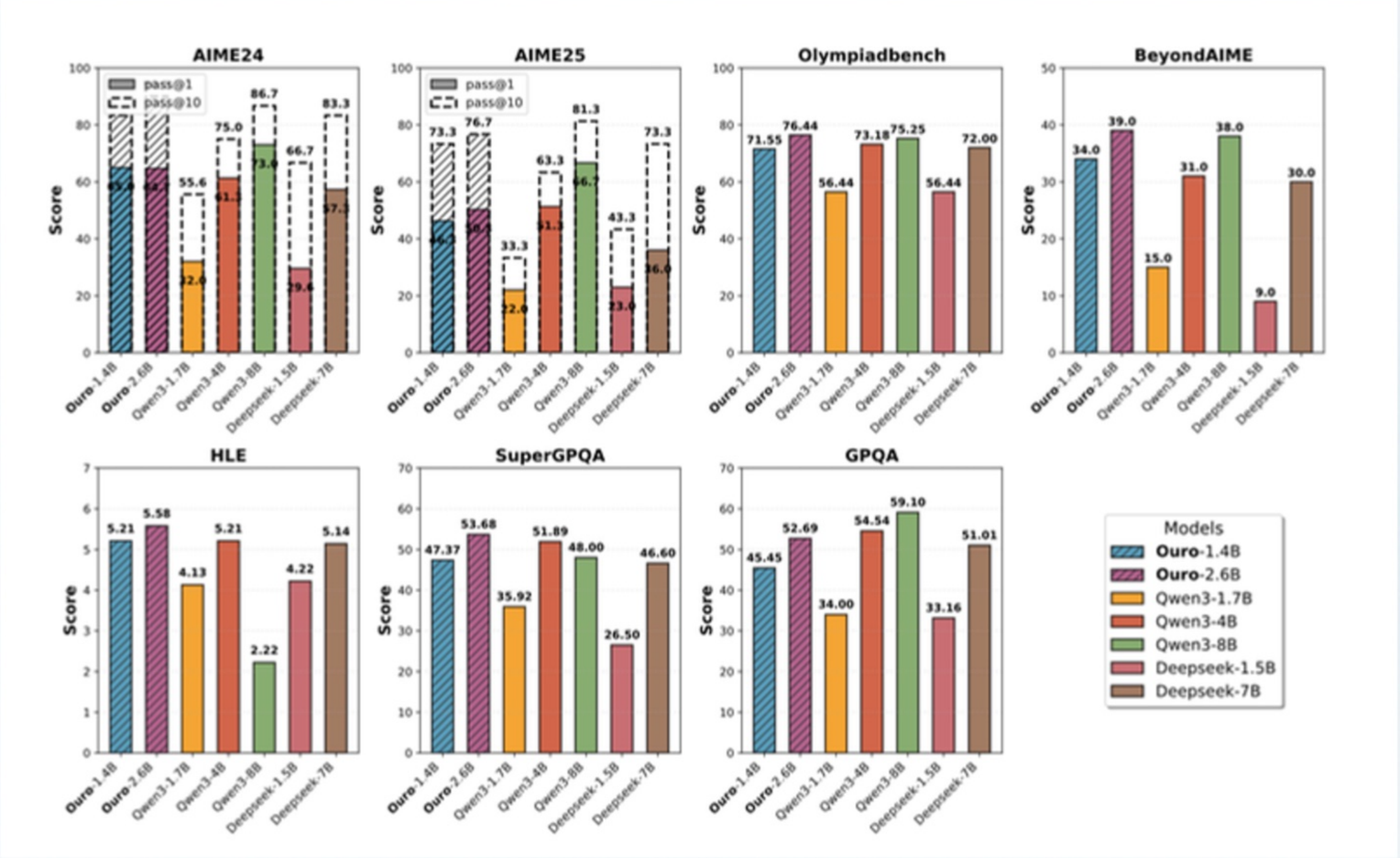
– Multimodal Reasoning - [15]

– Latent Reasoning via Looped Transformers - [16], [17]



Ouro Architecture - [18]

Recent Advances



Performance comparison of Ouro on advance reasoning benchmarks - [18]

References:

[14] Wei, T., et al. (2026). Agentic Reasoning for Large Language Models. ArXiv, abs/2601.12538.

[15] Lin, Z., Gao, Y., Zhao, X., Yang, Y., & Sang, J. (2025). Mind with Eyes: From Language Reasoning to Multimodal Reasoning. ArXiv, abs/2503.18071.

[16] Dehghani, M., et al. (2018). Universal Transformers. ArXiv, abs/1807.03819.

[17] Fan, Y., Svete, A., & Lee, K. (2026). Bridging the Gap Between Latent and Explicit Reasoning with Looped Transformers. ArXiv, abs/2606.31779.

[18] Zhu, R.-J., et al. (2025). Scaling Latent Reasoning via Looped Language Models. ArXiv, abs/2510.25741.

[19] DeepSeek-AI, et al. (2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. ArXiv, abs/2501.12948.

[20] Huang, J., & Chang, K. C.-C. (2023). Towards Reasoning in Large Language Models: A Survey. ArXiv, abs/2212.10403.

[21] Xu, F., et al. (2025). Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models. ArXiv, abs/2501.09686.

Resources:

– *Huang, J.-B. (2025). GRPO: The RL Trick Behind DeepSeek-R1's Reasoning. YouTube Video*

– *Qingfeng. (2025). From REINFORCE to Dr. GRPO: A Unified Perspective on LLM Post-Training. Blog post. Blog post (lancelqf.github.io)*

– *Shi, Y. J. (2025). A Vision Researcher's Guide to Some RL Stuff: PPO & GRPO. Blog post. Blog post (yugeten.github.io)*

– *Raschka, S. (2026). Controlling Reasoning Effort in LLMs: How LLMs Learn Low-, Medium-, and High-Effort Reasoning Modes. Blog post. Blog post (magazine.sebastianraschka.com)*

– *Cameron, R. W. (2026). Reinforcement Learning for LLMs: The Complete Guide. Blog post. Blog post (cameronrwolfe.substack.com)*

– *Sutton, R. S. (2019). The Bitter Lesson. Incomplete Ideas. The Bitter Lesson*

– *Fortes, A. (2025). Awesome LLM Reasoning. GitHub repository. GitHub Repository (github.com)*

Resources:

– *Huang, J.-B. (2025). GRPO: The RL Trick Behind DeepSeek-R1's Reasoning. YouTube Video*

– *Qingfeng. (2025). From REINFORCE to Dr. GRPO: A Unified Perspective on LLM Post-Training. Blog post. Blog post (lancelqf.github.io)*

– *Shi, Y. J. (2025). A Vision Researcher's Guide to Some RL Stuff: PPO & GRPO. Blog post. Blog post (yugeten.github.io)*

– *Raschka, S. (2026). Controlling Reasoning Effort in LLMs: How LLMs Learn Low-, Medium-, and High-Effort Reasoning Modes. Blog post. Blog post (magazine.sebastianraschka.com)*

– *Cameron, R. W. (2026). Reinforcement Learning for LLMs: The Complete Guide. Blog post. Blog post (cameronrwolfe.substack.com)*

– *Sutton, R. S. (2019). The Bitter Lesson. Incomplete Ideas. The Bitter Lesson*

– *Fortes, A. (2025). Awesome LLM Reasoning. GitHub repository. GitHub Repository (github.com)*