MUNI

Group Relative Policy Optimization

Hugo Adamove

IV125 Formela lab seminar

June 9, 2025

Topics

- Recap: RL Framework for LLMs
- From PPO to GRPO
- DeepSeekMath
- DeepSeek-R1

RL Framework for LLMs (consistent notation)

- **State** $s_t = (q,\ o_{<t})$: the query q together with already generated tokens $o_{<t}$.
- **Action** o_t**:** single next token predicted by the LLM.
- **Policy** π_θ**:** per-token distribution $\pi_\theta(o_t \mid q, o_{<t})$. Probability of a full answer

$$\pi_\theta(o \mid q) = \prod_{t=1}^{|o|} \pi_\theta(o_t \mid q, o_{<t}).$$

RL Framework for LLMs (consistent notation)

- **State** $s_t = (q,\ o_{<t})$: the query q together with already generated tokens $o_{<t}$.
- **Action** o_t**:** single next token predicted by the LLM.
- **Policy** π_θ**:** per-token distribution $\pi_\theta(o_t \mid q, o_{<t})$. Probability of a full answer

$$\pi_\theta(o \mid q) = \prod_{t=1}^{|o|} \pi_\theta(o_t \mid q, o_{<t}).$$

- **Reward** $R(q, o)$**:** scalar score emitted after the full answer
  - obtained from a trained Reward model or rule-based Reward functions.

RL Framework for LLMs (consistent notation)

- **State** $s_t = (q,\ o_{<t})$: the query q together with already generated tokens $o_{<t}$.
- **Action** o_t**:** single next token predicted by the LLM.
- **Policy** π_θ**:** per-token distribution $\pi_\theta(o_t \mid q, o_{<t})$. Probability of a full answer

$$\pi_\theta(o \mid q) = \prod_{t=1}^{|o|} \pi_\theta(o_t \mid q, o_{<t}).$$

- **Reward** $R(q, o)$**:** scalar score emitted after the full answer
  - obtained from a trained Reward model or rule-based Reward functions.

- **Objective:** maximize expected reward over the question distribution $P(Q)$:

$$J(\theta) = \mathbb{E}_{q\sim\mathcal{P}(Q),\ o\sim\pi_\theta(\cdot|q)}\big[R(q, o)\big].$$

PPO for LLM Fine-Tuning I

PPO surrogate objective for LLMs

The PPO surrogate objective for LLMs can be written as:

$$J_{\text{PPO}}(\theta) = \mathbb{E}_{\substack{q\sim P(Q),\\ o\sim\pi_{\theta_{old}}(o|q)}} \left[\frac{1}{|o|} \sum_{t=1}^{|o|} \min\left(A_t\rho_t, A_t\text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\right) \right] \tag{1}$$

where the probability ratio ρ_t under current policy π_θ and old policy $\pi_{\theta_{old}}$ is defined as:

$$\rho_t = \frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} \tag{2}$$

- A_t: Estimated advantage at token o_t (next slide).

PPO for LLM Fine-Tuning II — Advantage Estimation

(MC) Advantage Estimation

$$A_t = r_t - V_\psi(q, o_{<t}) \tag{3}$$

- V_ψ is a value model trained with an MSE loss $\mathcal{L}_V = \frac{1}{2}\left(V_\psi(q, o_{<t}) - R(q, o)\right)^2$.
- In practice, Generalized Advantage Estimation (GAE) [1] is also often used.

PPO for LLM Fine-Tuning II — Advantage Estimation

(MC) Advantage Estimation

$$A_t \;=\; r_t \;-\; V_\psi\big(q,\, o_{<t}\big) \tag{3}$$

- V_ψ is a value model trained with an MSE loss $\mathcal{L}_V = \frac{1}{2}\big(V_\psi(q, o_{<t}) - R(q, o)\big)^2$.
- In practice, Generalized Advantage Estimation (GAE) [1] is also often used.

Reward with divergence penalty

$$r_t \;=\; R_\varphi\big(q,\, o_{\le t}\big) \;-\; \beta \,\log \frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\text{ref}}(o_t \mid q, o_{<t})} \tag{4}$$

- R_φ — pretrained reward model scoring the (sub-)sequence.
- π_{ref} — frozen reference policy (e.g. the SFT model before applying RL).
- β — *KL-coefficient*: penalizes divergence from π_{ref}.

PPO for LLM Fine-Tuning III - "Big Picture"

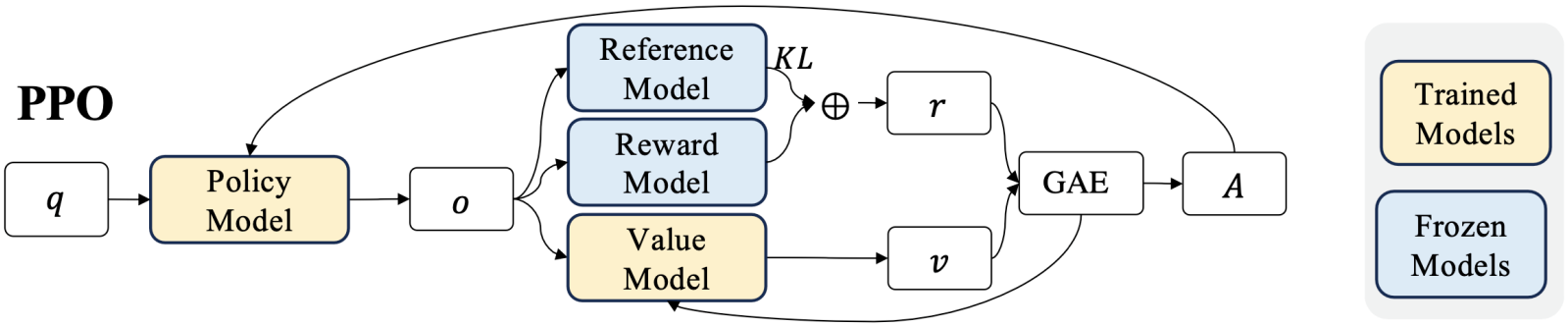

Figure: PPO diagram from [2].

- π_{ref} - Reference Model
- R_φ - Reward Model
- V_ψ - Value Model

Group Relative Policy Optimization (GRPO)

GRPO surrogate objective

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim P(Q),\\ \{o_i\}_1^G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(\hat{A}_{i,t}\ \rho_{i,t}, \text{clip}(\hat{A}_{i,t}\ \rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\Big) - \beta\, D_{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\right], \tag{5}$$

How GRPO differs from PPO

Group Relative Policy Optimization (GRPO)

GRPO surrogate objective

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim P(Q),\\ \{o_i\}_1^G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(\hat{A}_{i,t}\ \rho_{i,t}, \text{clip}(\hat{A}_{i,t}\ \rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\Big) - \beta\, D_{KL}\big(\pi_\theta \,||\, \pi_{\text{ref}}\big)\right], \tag{5}$$

How GRPO differs from PPO

- **Advantage function**: $\hat{A}_{i,t}$ is computed relatively within a group of G sampled answers, we no longer need V_ψ (next slide).

Group Relative Policy Optimization (GRPO)

GRPO surrogate objective

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim P(Q),\\ \{o_i\}_1^G\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(\hat{A}_{i,t}\ \rho_{i,t}, \text{clip}(\hat{A}_{i,t}\ \rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\Big) - \beta\, D_{KL}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big)\right], \tag{5}$$

How GRPO differs from PPO

- **Advantage function**: $\hat{A}_{i,t}$ is computed relatively within a group of G sampled answers, we no longer need V_ψ (next slide).
- **KL regularisation**: KL term is added directly to the loss instead of being folded into the reward via r_t.

GRPO – Group-Relative Advantage $\hat{A}_{i,t}$

- For each question q, GRPO samples a group of G outputs $\{o_1, \ldots, o_G\}$ from the frozen policy $\pi_{\theta_{\text{old}}}$.
- A reward model R_φ scores every full output o_i, producing rewards $\{r_1, \ldots, r_G\}$.

Group-Relative Advantage

Rewards are z-normalised within the group:

$$\hat{A}_{i,t} = \hat{r}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)} \tag{6}$$

GRPO – Group-Relative Advantage $\hat{A}_{i,t}$

- For each question q, GRPO samples a group of G outputs $\{o_1, \ldots, o_G\}$ from the frozen policy $\pi_{\theta_{\text{old}}}$.
- A reward model R_φ scores every full output o_i, producing rewards $\{r_1, \ldots, r_G\}$.

Group-Relative Advantage

Rewards are z-normalised within the group:

$$\hat{A}_{i,t} = \hat{r}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)} \tag{6}$$

- Normalization centers the scores around 0: positive values signal better-than-average, negative values worse-than-average.
- Good completions in the batch are reinforced, poor ones are suppressed.

PPO vs GRPO - "Big Picture"

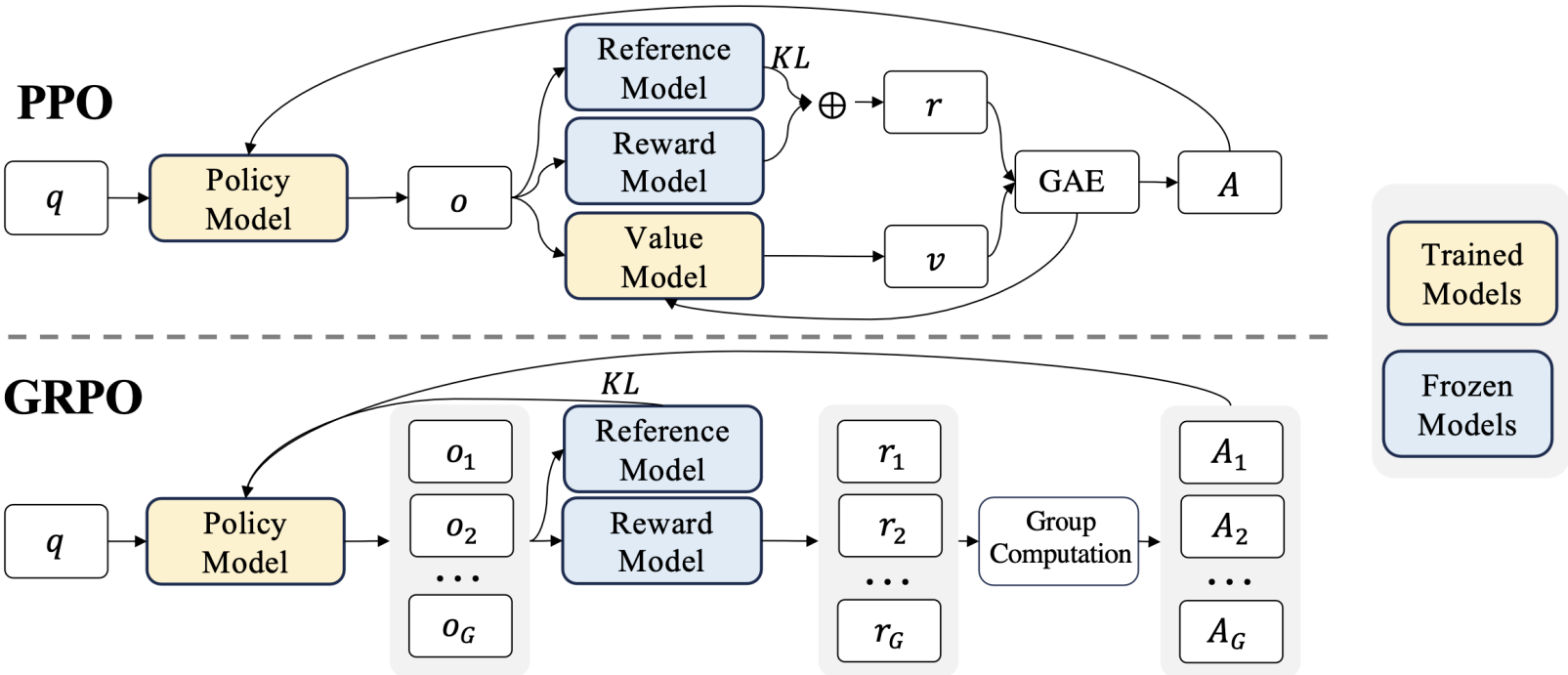

Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

Figure: PPO and GRPO diagram from [2].

PPO vs GRPO - "Big Picture"

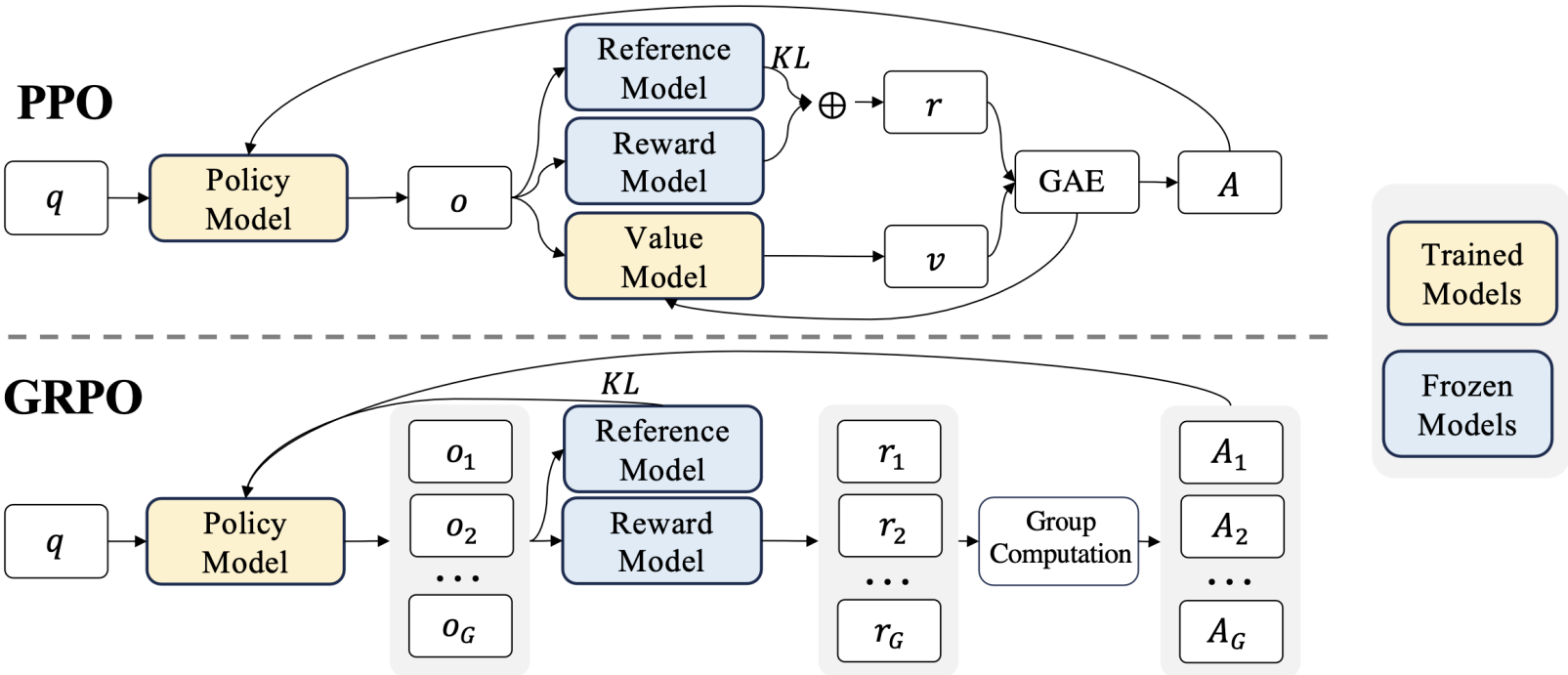

Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

Figure: PPO and GRPO diagram from [2].

- **No Value Model => significantly reduced training resources compared to PPO.**

Iterative GRPO – Pseudocode

Algorithm 1 Iterative Group Relative Policy Optimization

Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_φ; task prompts $\mathcal{D}$; hyperparameters ε, β, μ

policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, . . . , I **do**
    reference model $\pi_{ref} \leftarrow \pi_\theta$
    for step = 1, . . . , M **do**
        Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
        Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
        Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
        Compute rewards $\{r_i\}_{i=1}^{G}$ for each sampled output o_i by running r_φ
        Compute $\hat{A}_{i,t}$ for the t-th token of o_i through group relative advantage estimation.
        for GRPO iteration = 1, . . . , μ **do**
            Update the policy model π_θ by maximizing the GRPO objective (Equation 21)
    Update r_φ through continuous training using a replay mechanism.

Output π_θ

Figure: Iterative GRPO pseudocode from [2].

DeepSeekMath

- First paper to introduce GRPO for RL fine-tuning of an LLM.
- Specialized in mathematical reasoning and problem-solving.

DeepSeekMath

- First paper to introduce GRPO for RL fine-tuning of an LLM.
- Specialized in mathematical reasoning and problem-solving.

Training pipeline:

1. Initialized with **DeepSeek-Coder-Base-v1.5 7B** [3]
2. **Continued pre-training (unsupervised)**
   - Broad mathematical knowledge => **DeepSeekMath-Base 7B**
3. **Supervised instruction fine-tuning (SFT)**
   - Prompt following and explicit reasoning => **DeepSeekMath-Instruct 7B**
4. **GRPO reinforcement learning**
   - Improved reasoning capabilities => **DeepSeekMath-RL 7B**

Phase 1 – Continual Pre-Training (DeepSeekMath-Base)

Motivation

DeepSeek-Coder-Base-v1.5 lacked explicit mathematical domain knowledge.

- **Data**: 500B tokens
  - **56% DeepSeekMath Corpus** (filtered Common Crawl)
  - 4% AlgebraicStack
  - 10% arXiv
  - 20% GitHub code
  - 10% CommonCrawl NL

Phase 1 – Continual Pre-Training (DeepSeekMath-Base)

Motivation

DeepSeek-Coder-Base-v1.5 lacked explicit mathematical domain knowledge.

- **Data**: 500B tokens
  - **56% DeepSeekMath Corpus** (filtered Common Crawl)
  - 4% AlgebraicStack
  - 10% arXiv
  - 20% GitHub code
  - 10% CommonCrawl NL

- **Objective**: classical LM loss, maximize probability of next token prediction:
  $\mathcal{L}_{\text{LM}} = -\sum_{t} \log \pi_\theta(x_t \mid x_{<t})$
  - "read the Internet and guess the next word"

Phase 1 – Continual Pre-Training (DeepSeekMath-Base)

Motivation

DeepSeek-Coder-Base-v1.5 lacked explicit mathematical domain knowledge.

- **Data**: 500B tokens
  - **56% DeepSeekMath Corpus** (filtered Common Crawl)
  - 4% AlgebraicStack
  - 10% arXiv
  - 20% GitHub code
  - 10% CommonCrawl NL
- **Objective**: classical LM loss, maximize probability of next token prediction: $\mathcal{L}_{\text{LM}} = -\sum_t \log \pi_\theta(x_t \mid x_{<t})$
  - "read the Internet and guess the next word"
- **Results**: Tops open-source base models on GSM8K & MATH, surpasses 540B Minerva (see in a couple of slides).

Phase 2 – SFT (DeepSeekMath-Instruct)

Motivation

Base model not yet capable of instruction following or chain-of-thought.

- **Data**: 776K problems (English & Chinese) with
  - **CoT**: Natural-language reasoning steps.
  - **PoT**: Answer natural-language with python code that solves the task.

Phase 2 – SFT (DeepSeekMath-Instruct)

Motivation

Base model not yet capable of instruction following or chain-of-thought.

- **Data**: 776K problems (English & Chinese) with
  - **CoT**: Natural-language reasoning steps.
  - **PoT**: Answer natural-language with python code that solves the task.

- **Objective**: Supervised cross-entropy on target answer:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{|o^*|} \log \pi_\theta(o_t^* \mid q, o_{<t}^*)$$

  - o^*: gold solution written by human expert
  - "write the expert answer I show you"

Phase 2 – SFT (DeepSeekMath-Instruct)

Motivation

Base model not yet capable of instruction following or chain-of-thought.

- **Data**: 776K problems (English & Chinese) with
  - **CoT**: Natural-language reasoning steps.
  - **PoT**: Answer natural-language with python code that solves the task.

- **Objective**: Supervised cross-entropy on target answer:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{|o^*|} \log \pi_\theta(o_t^* \mid q, o_{<t}^*)$$

  - o^*: gold solution written by human expert
  - "write the expert answer I show you"

- **Results**: Beats all 7B peers and rivals 70B open-source instruction-tuned models (in two slides).

Phase 3 – RL with GRPO (DeepSeekMath-RL)

Motivation

Further align the model's reasoning quality without relying on a value critic.

- **Data**: 144K math prompts from **GSM8K** and **MATH**

Phase 3 – RL with GRPO (DeepSeekMath-RL)

Motivation

Further align the model's reasoning quality without relying on a value critic.

- **Data**: 144K math prompts from **GSM8K** and **MATH**
- **Objective**: GRPO surrogate objective (5)

Phase 3 – RL with GRPO (DeepSeekMath-RL)

Motivation

Further align the model's reasoning quality without relying on a value critic.

- **Data**: 144K math prompts from **GSM8K** and **MATH**
- **Objective**: GRPO surrogate objective (5)
  - Reward model initialized from **DeepSeekMath-Base** – with a scalar head attached, and fine-tuned on preference data.

Phase 3 – RL with GRPO (DeepSeekMath-RL)

Motivation

Further align the model's reasoning quality without relying on a value critic.

- **Data**: 144K math prompts from **GSM8K** and **MATH**
- **Objective**: GRPO surrogate objective (5)
  - Reward model initialized from **DeepSeekMath-Base** – with a scalar head attached, and fine-tuned on preference data.
  - For every prompt the policy samples a **64 candidate solutions (**$G = 64$**)** to compute group relative advantage.

Phase 3 – RL with GRPO (DeepSeekMath-RL)

Motivation

Further align the model's reasoning quality without relying on a value critic.

- **Data**: 144K math prompts from **GSM8K** and **MATH**
- **Objective**: GRPO surrogate objective (5)
  - Reward model initialized from **DeepSeekMath-Base** – with a scalar head attached, and fine-tuned on preference data.
  - For every prompt the policy samples a **64 candidate solutions (**$G = 64$**)** to compute group relative advantage.
  - **Iterative GRPO outperforms a single-pass variant**

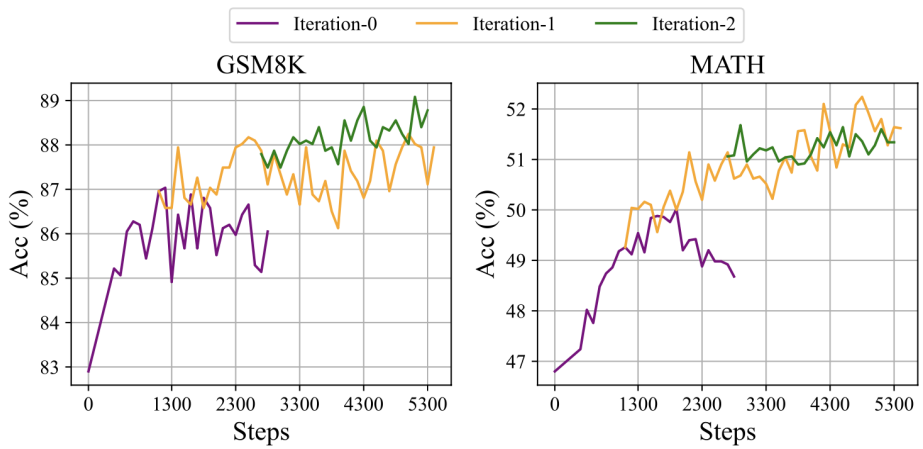

Figure 6 | Performance of iterative reinforcement learning with DeepSeekMath-Instruct 7B on two benchmarks.

Figure: From [2].

DeepSeekMath - Results

Model	Size	English Benchmarks		Chinese Benchmarks	
		GSM8K	MATH	MGSM-zh	CMATH
Chain-of-Thought Reasoning					
Closed-Source Model					
Gemini Ultra	-	94.4%	53.2%	-	-
GPT-4	-	92.0%	52.9%	-	86.0%
Inflection-2	-	81.4%	34.8%	-	-
GPT-3.5	-	80.8%	34.1%	-	73.8%
Gemini Pro	-	86.5%	32.6%	-	-
Grok-1	-	62.9%	23.9%	-	-
Baichuan-3	-	88.2%	49.2%	-	-
GLM-4	-	87.6%	47.9%	-	-
Open-Source Model					
InternLM2-Math	20B	82.6%	37.7%	-	-
Qwen	72B	78.9%	35.2%	-	-
Math-Shepherd-Mistral	7B	84.1%	33.0%	-	-
WizardMath-v1.1	7B	83.2%	33.0%	-	-
DeepSeek-LLM-Chat	67B	84.1%	32.6%	74.0%	80.3%
MetaMath	70B	82.3%	26.6%	66.4%	70.9%
SeaLLM-v2	7B	78.2%	27.5%	64.8%	-
ChatGLM3	6B	72.3%	25.7%	-	-
WizardMath-v1.0	70B	81.6%	22.7%	64.8%	65.4%
DeepSeekMath-Instruct	7B	82.9%	46.8%	73.2%	84.6%
DeepSeekMath-RL	7B	**88.2%**	**51.7%**	**79.6%**	**88.8%**

Figure: DeepSeekMath-RL 7B improves over DeepSeekMath-Instruct 7B and beats all open-source models from 7B to 70B, as well as the majority of closed-source models. From [2].

DeepSeek-R1

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

research@deepseek.com

Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

DeepSeek-R1 overview

- Introduced two reasoning models (671B):
  - **DeepSeek-R1-Zero** (pure RL with GRPO)
  - **DeepSeek-R1** (multi-stage RL with GRPO + SFT)
- And a family of distilled models based on **Qwen** [4] and **Llama** [5] 1.5B-70B.

DeepSeek-R1 overview

- Introduced two reasoning models (671B):
  - **DeepSeek-R1-Zero** (pure RL with GRPO)
  - **DeepSeek-R1** (multi-stage RL with GRPO + SFT)
- And a family of distilled models based on **Qwen** [4] and **Llama** [5] 1.5B-70B.
- Reaches **GPT-o1** level on math / code benchmarks
- Often mislabeled as "open-source"
  - Model weights are open, the datasets and code used to train the model are not!

DeepSeek-R1-Zero (pure RL with GRPO)

- Initialized with **DeepSeek-V3-Base** [6], only fine-tuned to reason with GRPO without any supervised data.

DeepSeek-R1-Zero (pure RL with GRPO)

- Initialized with **DeepSeek-V3-Base** [6], only fine-tuned to reason with GRPO without any supervised data.

- Reward model replaced with **rule based rewards**:
  - **Accuracy rewards**: Response is correct (+1), i.e. math and code problems with deterministic results.
  - **Format rewards**: Model is required to put its thinking process between <think></think> tags.

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

Table 1 | Template for DeepSeek-R1-Zero. prompt will be replaced with the specific reasoning question during training.

Figure: DeepSeek-R1-Zero training prompt, from [7].

DeepSeek-R1-Zero training – Accuracy

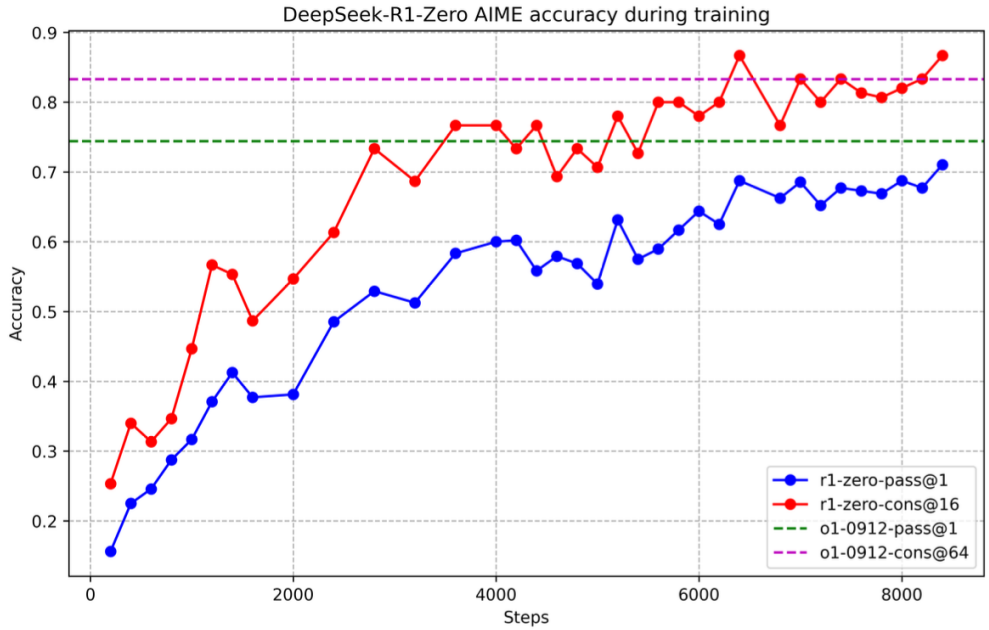

Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

Figure: From [7].

DeepSeek-R1-Zero training – Response length

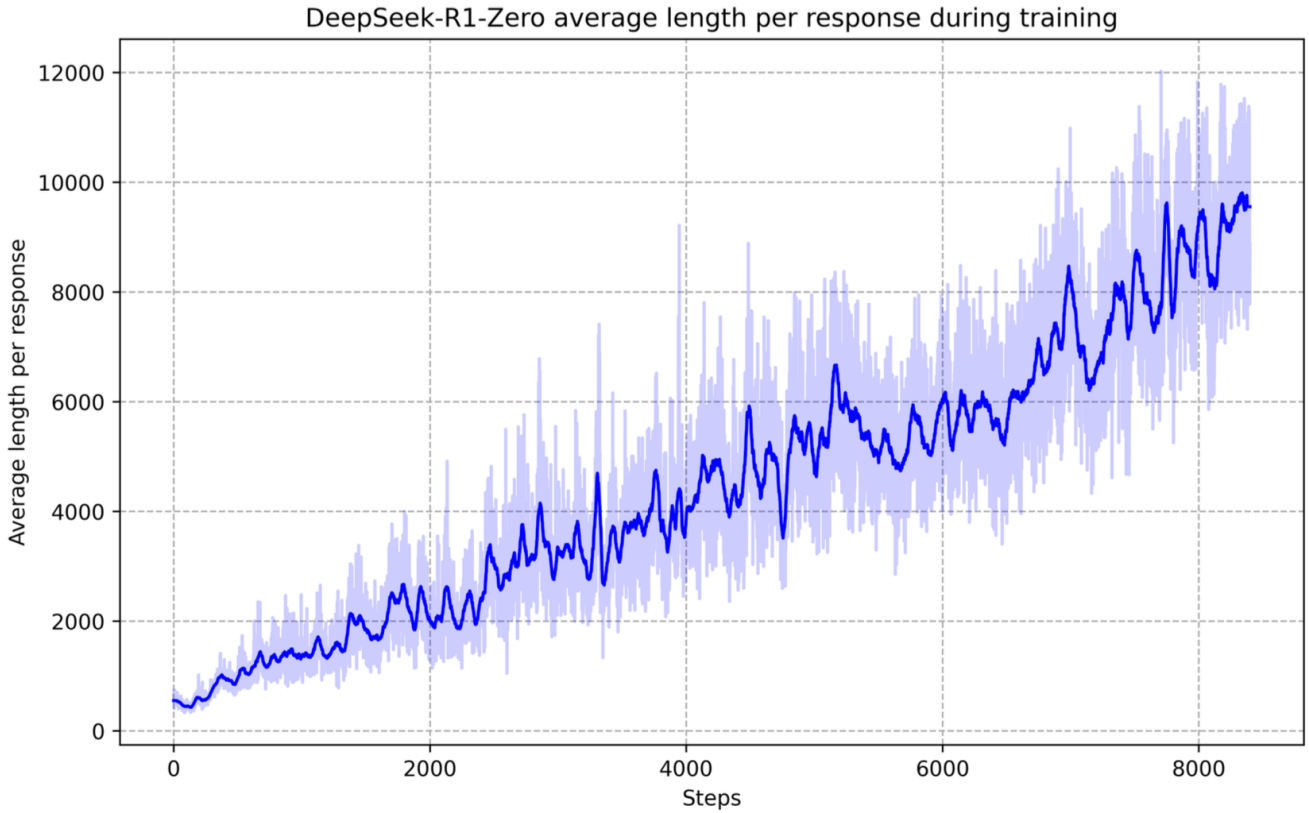

Figure: From [7].

- "DeepSeek-R1-Zero naturally acquires the ability to solve increasingly complex reasoning tasks by leveraging extended test-time computation"

DeepSeek-R1-Zero training – "Aha moment"

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

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

Figure: From [7].

- "DeepSeek-R1-Zero learns to allocate more thinking time to a problem by reevaluating its initial approach."

DeepSeek-R1 — Four-stage pipeline (SFT + GRPO)

Motivation

Problems with DeepSeek-R1-Zero: poor readability, language switching.

DeepSeek-R1 — Four-stage pipeline (SFT + GRPO)

Motivation

Problems with DeepSeek-R1-Zero: poor readability, language switching.

1. **SFT**: finetune on thousands of curated CoT examples (exact data sources not released).

2. **Reasoning-oriented RL with GRPO**: same rule-based rewards as R1-Zero + language-consistency rule (to prevent language switching).

DeepSeek-R1 — Four-stage pipeline (SFT + GRPO)

Motivation

Problems with DeepSeek-R1-Zero: poor readability, language switching.

1. **SFT**: finetune on thousands of curated CoT examples (exact data sources not released).

2. **Reasoning-oriented RL with GRPO**: same rule-based rewards as R1-Zero + language-consistency rule (to prevent language switching).

3. **Rejection-sampling SFT**:
   - 600K reasoning samples generated by model from previous stage. Filtered using rejection sampling (rule-based rewards for math & code, DeepSeek-V3 judges equivalence with ground-truth for the rest).

DeepSeek-R1 — Four-stage pipeline (SFT + GRPO)

Motivation

Problems with DeepSeek-R1-Zero: poor readability, language switching.

1. **SFT**: finetune on thousands of curated CoT examples (exact data sources not released).

2. **Reasoning-oriented RL with GRPO**: same rule-based rewards as R1-Zero + language-consistency rule (to prevent language switching).

3. **Rejection-sampling SFT**:
   - 600K reasoning samples generated by model from previous stage. Filtered using rejection sampling (rule-based rewards for math & code, DeepSeek-V3 judges equivalence with ground-truth for the rest).
   - 200K non-reasoning samples, e.g. factual QA, translation, "hello", ... (reused from DeepSeek-V3 pipeline).

DeepSeek-R1 — Four-stage pipeline (SFT + GRPO)

Motivation

Problems with DeepSeek-R1-Zero: poor readability, language switching.

1. **SFT**: finetune on thousands of curated CoT examples (exact data sources not released).

2. **Reasoning-oriented RL with GRPO**: same rule-based rewards as R1-Zero + language-consistency rule (to prevent language switching).

3. **Rejection-sampling SFT**:
   - 600K reasoning samples generated by model from previous stage. Filtered using rejection sampling (rule-based rewards for math & code, DeepSeek-V3 judges equivalence with ground-truth for the rest).
   - 200K non-reasoning samples, e.g. factual QA, translation, "hello", ... (reused from DeepSeek-V3 pipeline).

4. **RL with GRPO for all scenarios**: rule-based rewards for reasoning, helpful/harmless neural Reward Models for general queries

DeepSeek-R1 Results

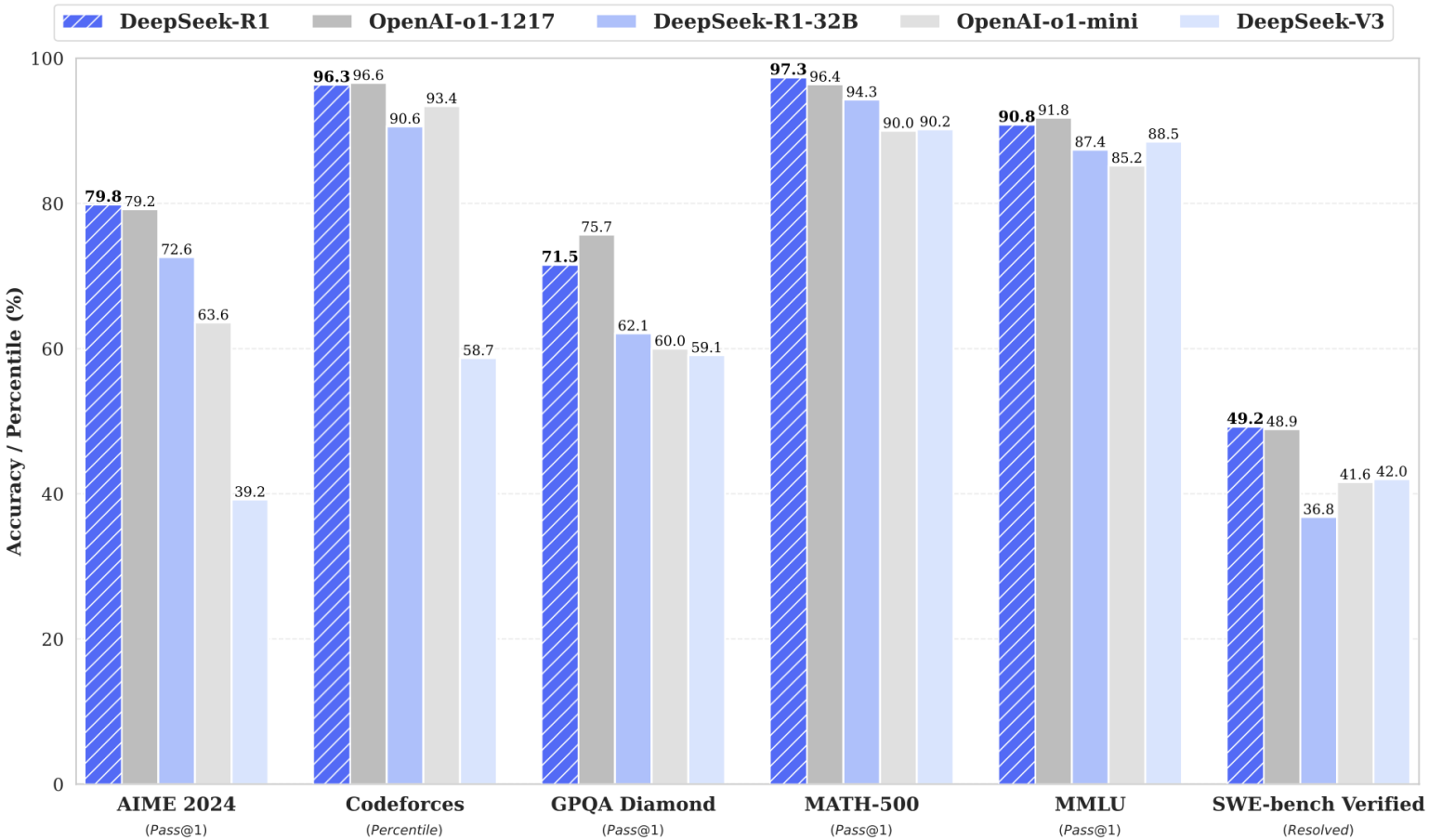

Figure 1 | Benchmark performance of DeepSeek-R1.

Figure: From [7].

DeepSeek-R1-Distill Models

- Models from **Qwen** and **Llama** family trained by SFT with reasoning samples generated by **DeepSeek-R1**.

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.

Figure: From [7].

DeepSeek-R1-Distill Models II

- **Interesting finding:** Applying large-scale RL on **Qwen-32B-Base** in the same way **DeepSeek-R1-Zero** was trained (10K steps) results in worse performance than SFT distillation!

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

Table 6 | Comparison of distilled and RL Models on Reasoning-Related Benchmarks.

Figure: From [7].

Takeaways

To train a strong reasoning model:

- You might not need a Value Model.

Takeaways

To train a strong reasoning model:

- You might not need a Value Model.
- You might (sometimes) not even need a Reward Model.

Takeaways

To train a strong reasoning model:

- You might not need a Value Model.
- You might (sometimes) not even need a Reward Model.
- You still need lot of engineering hacks (?)

May 9, 2025

Absolute Zero: Reinforced Self-play Reasoning with Zero Data

Andrew Zhao [1], Yiran Wu [3], Yang Yue [1], Tong Wu [2], Quentin Xu [1], Yang Yue [1], Matthieu Lin [1], Shenzhi Wang [1], Qingyun Wu [3], Zilong Zheng [2,✉] and Gao Huang [1,✉]

[1] Tsinghua University [2] Beijing Institute for General Artificial Intelligence [3] Pennsylvania State University

zqc21@mails.tsinghua.edu.cn, yiran.wu@psu.edu, zlzheng@bigai.ai, gaohuang@tsinghua.edu.cn

References I

[1] John Schulman et al. *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. 2015. arXiv: 1506.02438 [cs.LG]. URL: https://arxiv.org/abs/1506.02438.

[2] Zhihong Shao et al. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. 2024. arXiv: 2402.03300 [cs.CL]. URL: https://arxiv.org/abs/2402.03300.

[3] Daya Guo et al. *DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence*. 2024. arXiv: 2401.14196 [cs.SE]. URL: https://arxiv.org/abs/2401.14196.

[4] Qwen et al. *Qwen2.5 Technical Report*. 2025. arXiv: 2412.15115 [cs.CL]. URL: https://arxiv.org/abs/2412.15115.

[5] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[6] DeepSeek-AI et al. *DeepSeek-V3 Technical Report*. 2025. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

References II

[7] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

MASARYK
UNIVERSITY