Fine-tuning LLMs with Synthetic Data

Pu Yang
School of Mathematical Sciences, Peking University

2024.10.30

Table of contents

Introduction: Learning from Synthetic Data

- What is synthetic data?
  - Data generated by models rather than humans
- Why synthetic data?
  - SFT data is *expensive*, while synthetic data is much cheaper
  - SFT data is hard to make models *outperform* human (like imitation learning), while synthetic data allows models to explore
- How to learning form synthetic data? Two main question:
  - How to generate synthetic data? e.g. Teacher-student, **self-explore**
  - **How to use synthetic data for training?**

Generate via Self-explore: Rejection Sampling Fine-Tuning (RFT)

- LLMs are tasked to self-generated *positive* data for rejection sampling fine-tuning (RFT).
  Given a generative policy π and a binary reward function $r(\boldsymbol{y}, \hat{\boldsymbol{y}}) \rightarrow \{0, 1\}$ which verifies if a new generation $\hat{\boldsymbol{y}}$ is correct or not, A positive dataset $\mathcal{D}_\pi^+ = \{(\boldsymbol{x}, +\boldsymbol{y})\}$ where $+\boldsymbol{y}$ is positive generated from $\pi(\cdot \mid \boldsymbol{x})$.
  Then, we apply the next token prediction loss on $\mathcal{D}_{\pi_{\text{sft}}}^+$

$$\max \mathbb{E}_{(\boldsymbol{x}, +\boldsymbol{y}) \sim \mathcal{D}_{\mathrm{sft}}^+} \left[p(+\boldsymbol{y} \mid \boldsymbol{x}) \right] \tag{1}$$

  - e.g., code generation, where r is a Python interpreter.

Generate via Self-explore: Direct Preference Optimization (DPO)

- LLMs are tasked to self-generated positive and negative data - which can together form a *pairwise* dataset for preference optimization, e.g. DPO.
  A pairwise dataset $\mathcal{D}_\pi^\pm = \{(\boldsymbol{x}, +\boldsymbol{y}, -\boldsymbol{y})\}$ where $-\boldsymbol{y}$ is negative generated from $\pi(\cdot \mid \boldsymbol{x})$.
  Then we apply DPO loss on $\mathcal{D}_\pi^\pm$

$$\min_\pi \mathcal{L}_{\mathrm{DPO}}(\pi) := \mathbb{E}_{(\boldsymbol{x}, +\boldsymbol{y}, -\boldsymbol{y}) \sim \mathcal{D}_{\mathrm{sft}}^\pm} \left[\sigma \left(\beta \log \frac{\pi(+\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\mathrm{sft}}(+\boldsymbol{y} \mid \boldsymbol{x})} - \beta \log \frac{\pi(-\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\mathrm{stt}}(-\boldsymbol{y} \mid \boldsymbol{x})} \right) \right] \quad (2)$$

- In general, performance: DPO $>$ RFT $>$ SFT $>$ ICL

Relation to Data Augmentation (DA)

- Traditional DA techniques aim at expanding the training dataset in a somewhat mechanical manner, e.g.
  - paraphrasing
  - back-translation
- Generating synthetic data with LLMs serves as an advanced DA method, which focuses on the generation of novel, context-rich training data tailored to specific domains and skills.

RFT + Multiple-step[1]

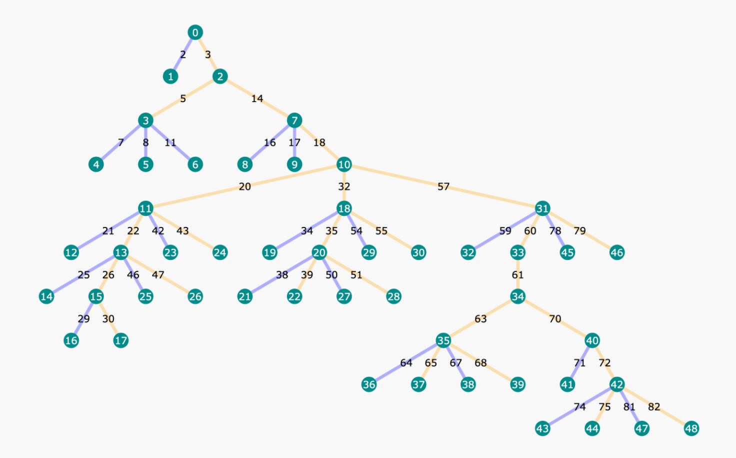
Google DeepMind 2024-05-22

Improve Mathematical Reasoning in Language Models by Automated Process Supervision

Liangchen Luo[1*], Yinxiao Liu[1*], Rosanne Liu[1], Samrat Phatale[1], Harsh Lara[1], Yunxuan Li[2], Lei Shu[1], Yun Zhu[1], Lei Meng[2], Jiao Sun[2] and Abhinav Rastogi[1]
[1]Google DeepMind, [2]Google

[1] Liangchen Luo et al. "Improve Mathematical Reasoning in Language Models by Automated Process Supervision". In: arXiv preprint arXiv:2406.06592 (2024).

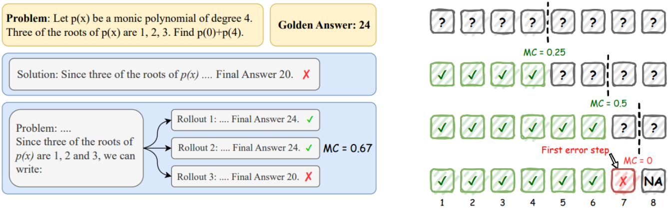
An Example



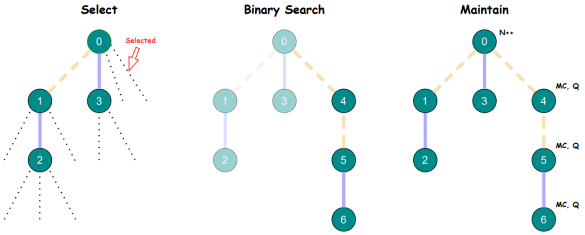
- Node: A state of partial chain-of-thought solution $(\boldsymbol{x}, \boldsymbol{y}_{1:i-1})$
- Edge: An action, i.e., a reasoning step $\boldsymbol{y}_i$
- Transition: Simply concate

Algorithm: Omega Process Reward Model (PRM)

- Build positive training examples via Monte Carlo Tree Search (MCTS)



(a) Monte Carlo estimation of a prefix solution. (b) Error locating using binary search.



(c) Three stages in an iteration of the MCTS process.

Algorithm: OmegaPRM

- Training with the positive training examples

$$\max \mathbb{E}_{(\boldsymbol{x},\textbf{Tree})\sim\mathcal{D}_{\mathrm{MCTS}}}\left[p(+\boldsymbol{y}_i \mid \boldsymbol{x}, +\boldsymbol{y}_{1:i-1})\right] \tag{3}$$

DPO + Multiple-step[2]

RL on Incorrect Synthetic Data Scales the Efficiency of LLM Math Reasoning by Eight-Fold

Amrith Setlur[1], Saurabh Garg[1], Xinyang (Young) Geng[2], Naman Garg[3], Virginia Smith[1] and Aviral Kumar[2]
[1]Carnegie Mellon University, [2]Google DeepMind, [3]MultiOn

[2] Amrith Setlur et al. “RL on Incorrect Synthetic Data Scales the Efficiency of LLM Math Reasoning by Eight-Fold”. In: arXiv preprint arXiv:2406.14532 (2024).

Overview

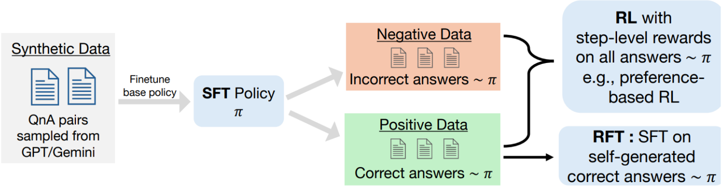
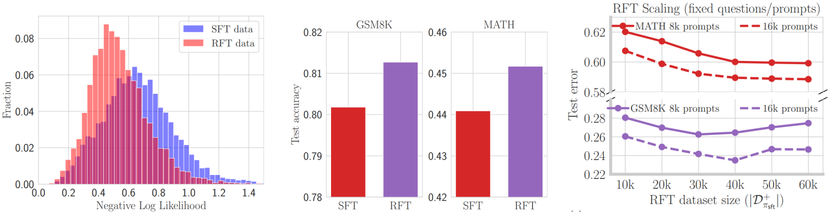


Figure 1: definitions for positive and negative synthetic data and how they are fed to SFT, RFT and step-level RL algorithms.

Positive Data Improves Coverage, But Amplifies Spurious Correlations



- Under base LLM, RFT data sampled from π_{sft}, has higher likelihood than SFT data.
- RFT data with a single self-generated correct solution per problem outperforms SFT data of the same size.
- However, continuing to scale RFT data leads to test error saturation, or even worse test error (Contrary to the scaling law). Why? Incorrect/Irrelevant steps are not detected by our verifier.

Negative Synthetic Data Enables Per-Step Credit Assignment

- Reasoning steps: The trace $\boldsymbol{y}_i$ consists of several intermediate steps, $\boldsymbol{y}_i = [\boldsymbol{y}_{i,1}, \cdots, \boldsymbol{y}_{i,L}]$.
- We formalize the notion of per-step credit using value functions from RL

$$Q_{\tilde{\pi}}(\underbrace{\boldsymbol{x}, \hat{\boldsymbol{y}}_{1:i-1}}_{\text{state}}; \underbrace{\hat{\boldsymbol{y}}_i}_{\text{action}}) = \underbrace{\mathbb{E}_{\boldsymbol{y}^{\text{new}}_{i+1:L} \sim \tilde{\pi}(\cdot|\boldsymbol{x}, \hat{\boldsymbol{y}}_{1:i})} \left[r\left(\left[\hat{\boldsymbol{y}}_{1:i}, \boldsymbol{y}^{\text{new}}_{i+1:L} \right], \boldsymbol{y} \right) \right]}_{\text{expected future reward under new actions (i.e., steps) sampled by policy } \tilde{\pi}} \quad (4)$$

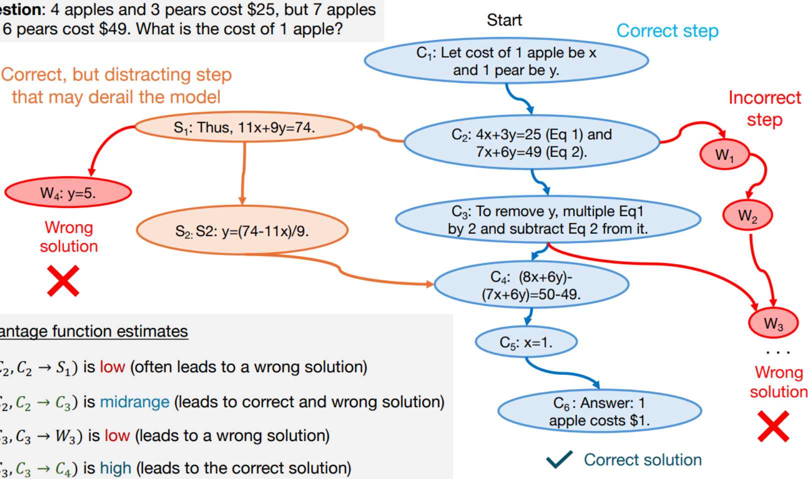
  - State: Problem $\boldsymbol{x}$ and previous reasoning steps $\hat{\boldsymbol{y}}_{1:i-1}$
  - Action: Current reasoning step $\hat{\boldsymbol{y}}_i$
- Advantage of a given step $\hat{\boldsymbol{y}}_i$:

$$A_{\tilde{\pi}}(\boldsymbol{x}, \hat{\boldsymbol{y}}_{1:i-1}; \hat{\boldsymbol{y}}_i) = Q_{\tilde{\pi}}(\boldsymbol{x}, \hat{\boldsymbol{y}}_{1:i-1}; \hat{\boldsymbol{y}}_i) - Q_{\tilde{\pi}}(\boldsymbol{x}, \hat{\boldsymbol{y}}_{1:i-2}; \hat{\boldsymbol{y}}_{i-1}). \quad (5)$$

  It is the gap between the Q-value of a state-action pair and the value function of the state.

Illustration of Advantage Estimation from Negative Data on a Didactic Example

Question: 4 apples and 3 pears cost $25, but 7 apples and 6 pears cost $49. What is the cost of 1 apple?



Advantage function estimates

$A_{\tilde{\pi}}(C_2, C_2 \rightarrow S_1)$ is low (often leads to a wrong solution)

$A_{\tilde{\pi}}(C_2, C_2 \rightarrow C_3)$ is midrange (leads to correct and wrong solution)

$A_{\tilde{\pi}}(C_3, C_3 \rightarrow W_3)$ is low (leads to a wrong solution)

$A_{\tilde{\pi}}(C_3, C_3 \rightarrow C_4)$ is high (leads to the correct solution)

Equivalence of Advantage-weighted RL and DPO with Per-Step Pairs

- Callback:

$$\min_{\pi} \mathcal{L}_{\mathrm{DPO}}(\pi) := \mathbb{E}_{(\boldsymbol{x},+\boldsymbol{y},-\boldsymbol{y})\sim\mathcal{D}^{\pm}_{\mathrm{sft}}}\left[\sigma\left(\beta \log \frac{\pi(+\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\mathrm{sft}}(+\boldsymbol{y} \mid \boldsymbol{x})} - \beta \log \frac{\pi(-\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\mathrm{stt}}(-\boldsymbol{y} \mid \boldsymbol{x})}\right)\right]$$

- We have the following theorem:

Theorem 6.1 (Equivalence of advantage-weighted RL and DPO with per-step pairs)**.** *The optimal policy from Equation 1 with $\mathcal{D}^{\pm}_{\pi_{\mathrm{sft}}}$ given by $(\boldsymbol{x}, [\boldsymbol{y}_{1:i}, +\boldsymbol{y}_{i+1}], [\boldsymbol{y}_{1:i}, -\boldsymbol{y}_{i+1}])$ where the positive and negative traces share prefix $\boldsymbol{y}_{1:i} \sim \pi_{\mathrm{sft}}$, and $-\boldsymbol{y}_{i+1} \sim \pi_{\mathrm{sft}}(\cdot|\boldsymbol{x}, \boldsymbol{y}_{1:i})$, $+\boldsymbol{y}_{i+1} \sim \sigma(A_{\tilde{\pi}}(\boldsymbol{x}, \boldsymbol{y}_{1:i}; \cdot) - A_{\tilde{\pi}}(\boldsymbol{x}, \boldsymbol{y}_{1:i}; -\boldsymbol{y}_{i+1}))$, is identical to the optima of the advantage-weighted RL objective:*

$$\max_{\pi} \; \mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{syn}}(\boldsymbol{x}), \boldsymbol{y}\sim\pi_{\mathrm{sft}}(\cdot|\boldsymbol{x})}\left[\sum_{i=1}^{L} \log \pi(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{0:i-1}) \cdot \exp\left(A_{\tilde{\pi}}(\boldsymbol{x}, \boldsymbol{y}_{0:i-1}; \boldsymbol{y}_i)/\beta\right)\right]. \quad (4)$$

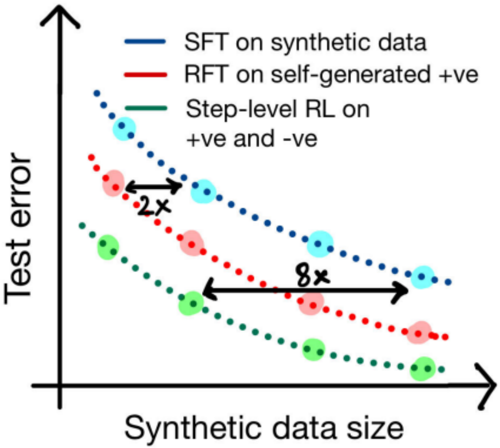
Per-step DPO Algorithm

Algorithm 1 Per-step DPO (Part 1: Practical version for most experiments; Parts 1 + 2: Complete version)

Require: Synthetic dataset: $\mathcal{D}_{\text{syn}}$, SFT policy finetuned on $\mathcal{D}_{\text{syn}}$: π_{sft}, sampling policy $\tilde{\pi}$.

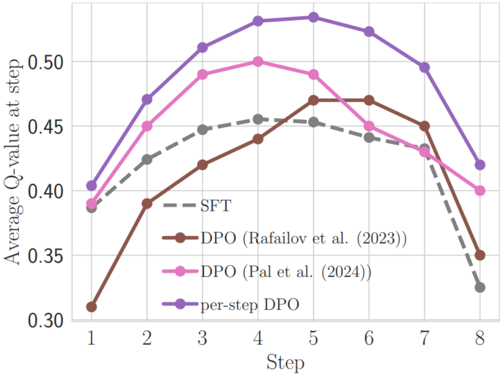
Initialize per-step DPO dataset $\mathcal{D}^{\pm}_{\pi_{\text{sft}}} \leftarrow \{\}$.
for $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_{\text{syn}} \cup \mathcal{D}^{+}_{\pi_{\text{sft}}}$ **do**
    # **Part 1:** Identify critical steps in incorrect responses
    Sample multiple incorrect answers $-\hat{\boldsymbol{y}} \sim \pi_{\text{sft}}(\cdot \mid \boldsymbol{x})$, and collect them in set $\mathcal{C}(\boldsymbol{x})$.
    for $-\hat{\boldsymbol{y}} := [-\hat{\boldsymbol{y}}_1, \ldots, -\hat{\boldsymbol{y}}_L] \in \mathcal{C}(\boldsymbol{x})$ **do**
        Compute the Monte Carlo estimate for $Q_{\tilde{\pi}}(\boldsymbol{x}, -\hat{\boldsymbol{y}}_{1:i-1}; -\hat{\boldsymbol{y}}_i)$ for each step $-\hat{\boldsymbol{y}}_i$.
        If $-\hat{\boldsymbol{y}}_c$ is the first step with least $Q_{\tilde{\pi}}(\boldsymbol{x}, -\hat{\boldsymbol{y}}_{1:i-1}; -\hat{\boldsymbol{y}}_i)$, then $\mathcal{D}^{\pm}_{\pi_{\text{sft}}} \leftarrow \mathcal{D}^{\pm}_{\pi_{\text{sft}}} \cup \{(\boldsymbol{x}, \boldsymbol{y}, -\hat{\boldsymbol{y}}_{1:c})\}$.
    end for
    # **Part 2:** Identify spurious steps in correct responses
    Sample multiple correct answers $+\hat{\boldsymbol{y}} \sim \pi_{\text{sft}}(\cdot \mid \boldsymbol{x})$, and collect them in set $\mathcal{C}'(\boldsymbol{x})$.
    for $+\hat{\boldsymbol{y}} := [+\hat{\boldsymbol{y}}_1, \ldots, +\hat{\boldsymbol{y}}_L] \in \mathcal{C}'(\boldsymbol{x})$ **do**
        Compute the Monte Carlo estimate for $Q_{\tilde{\pi}}(\boldsymbol{x}, +\hat{\boldsymbol{y}}_{1:i-1}; +\hat{\boldsymbol{y}}_i)$ for each step $+\hat{\boldsymbol{y}}_i$.
        If $+\hat{\boldsymbol{y}}_c$ is the first step with least $Q_{\tilde{\pi}}(\boldsymbol{x}, +\hat{\boldsymbol{y}}_{1:i-1}; +\hat{\boldsymbol{y}}_i)$, then $\mathcal{D}^{\pm}_{\pi_{\text{sft}}} \leftarrow \mathcal{D}^{\pm}_{\pi_{\text{sft}}} \cup \{(\boldsymbol{x}, \boldsymbol{y}, +\hat{\boldsymbol{y}}_{1:c})\}$.
    end for
end for
Optimize DPO loss in Equation (1) on $\mathcal{D}^{\pm}_{\pi_{\text{sft}}}$ with π_{sft} as the reference policy.

Results



- Per-step DPO > RFT > SFT

Ablation: Number of Steps



- When the number of steps is small, multi-step reasoning is treated as one step.
- When the number of steps is large, the Monte Carlo estimation may not be accurate.

Example: Advantage Estimatation

Advantage Estimates Computed Over a Positive Model Generated Sample

Question:
Betty wants a new wallet which costs $100. Betty has only half of the money she needs. Her parents give her $15 for that purpose, and her grandparents twice as much as her parents. How much more money does Betty need?
Positive response with identified spurious step:
Betty's parents gave her 15, so her grandparents gave her 2 * 15 = «15*2=30»30. In total, Betty received 30 + 15 = «30+15=45»45. The total amount of money Betty needs is 100 * 2 = «100*2=200»200. Betty needs 100 / 2 = «100/2=50»50 in total. Betty still needs to save 50 - 45 = «50-45=5»5. The answer is 5

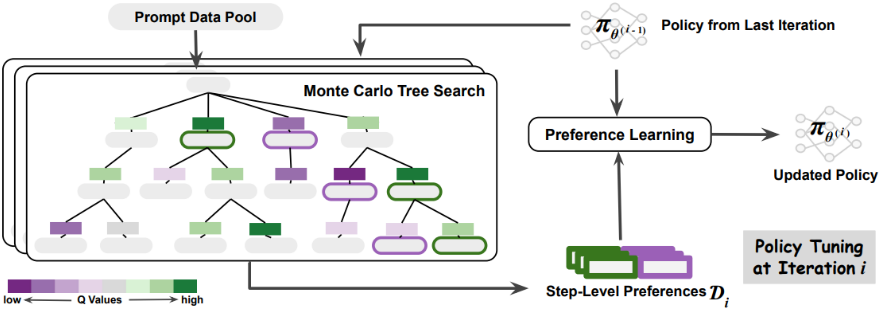
DPO + Multiple-Step + MCTS[3]

Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning

Yuxi Xie[1*] **Anirudh Goyal** **Wenyue Zheng**[1] **Min-Yen Kan**[1]
Timothy Lillicrap[2] **Kenji Kawaguchi**[1] **Michael Shieh**[1]
[1] National University of Singapore
[2] Google DeepMind

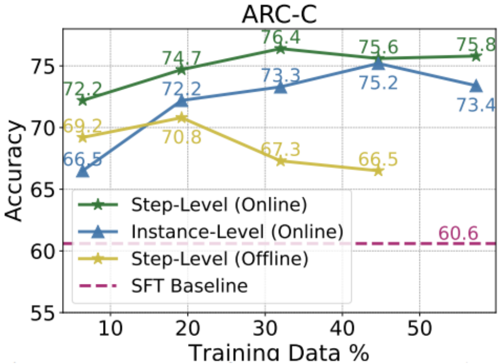
[3] Yuxi Xie et al. “Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning”. In: arXiv preprint arXiv:2405.00451 (2024).

Overview



- Use action values Q estimated by MCTS to assign the preferences

Main Results



- Offline setting can fail with high probability if the sampling policy differs too much from the current policy.
- we can indeed avoid this failure case in the online setting.

Summary: Exploitation and Exploration

We talk about how to learn from synthetic data.

- Explore more efficiently
- Make full use of synthetic data