Reflect-RL: Two-Player Online RL Fine-Tuning for LMs

Runlong Zhou

Jul 15 2024

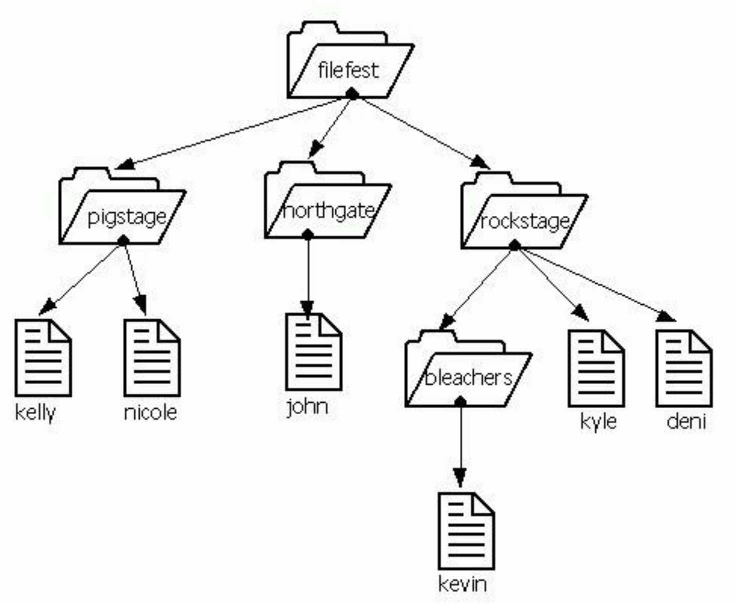
ACL 2024, Bangkok

Acknowledgement

This is a joint work with

Simon Du and **Beibin Li**

Motivations



- Tasks from production teams
  - A **repository**
    - Code
    - Document
    - Database files
  - User Questions
    - What is the alternative method for X if there is no Y?
      - **Document retrieval**
    - Create a map, colored by salary
      - **Coding, database query**
    - Show me the consequences when demand A increases by 50%
      - **Coding**

Challenges

- Privacy
  - Keep the proprietary data private (local), never leave the house

- Cost
  - Smaller model & shorter prompt: dedicated to the tasks, faster response

Task Abstract

- **Autonomous exploration** in a file system given natural language queries
  - **Locate**, modify and execute the correct files
- A local, small language model
  - Privacy, cost
- Contextual **reinforcement learning**
  - Find shortest path given context (query)

We focus on file locating in this work

Current LM/RL Approaches

- Token-generation as RL:
  - Each token as an action
  - Instead of embodied tasks, games, or interactive decision-making

Not suitable for our production tasks

Current LM/RL Approaches

- Bandit v.s. MDP:
  - Most RLHF work consider one-step bandit problems

Complex environments usually contains multiple steps

  - Frozen LMs as assistive agents to help other policy agents in MDPs

Why not directly interact with MDPs?

Current LM/RL Approaches

- Offline (v.s. Online):
  - Offline dataset / SFT insufficient for exploration in **complex environments**
  - LMs unable to self-correct during interactions without **external feedback** (Huang et al., 2023)

Need online interaction

Markov Decision Processes (MDPs)

- **State** space $\mathcal{S}$
- **Action** space $\mathcal{A}(s)$ for each s $\in \mathcal{S}$
- **Planning horizon** H
- **Reward** function $R(s, a) \in [-1,1]$
- **Transition** model $\mathcal{T}(s, a) \in \mathcal{S}$
- **Initial state** distribution μ

We study deterministic MDPs which are sufficient for our tasks

Policy and Value Function

Probability simplex

- Policy $\pi: \mathcal{S} \to \Delta(\mathcal{A})$
- Value functions and Q-functions:

$$V_h^\pi(s) := \mathbb{E}_\pi\left[\sum_{t=h}^{H} r_t \,\middle|\, s_h = s\right],$$

$$Q_h^\pi(s,a) := \mathbb{E}_\pi\left[\sum_{t=h}^{H} r_t \,\middle|\, (s_h, a_h) = (s,a)\right].$$

- **Expected return:** $J^\pi := \mathbb{E}_{s_1 \sim \mu}[V_1^\pi(s_1)]$

Policy Gradient

- REINFORCE algorithm (Sutton et al. (1999)):
  - **Policy gradient theorem:**

$$\nabla_\theta J^{\pi_\theta} = \sum_{h=1}^{H} \mathbb{E}_{s,a\sim d_h^{\pi_\theta}} \left[Q_h^{\pi_\theta}(s,a) \nabla_\theta \ln \pi_\theta(a|s) \right]$$

  - Update step:

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta J^{\pi_{\theta_t}}$$

LM as an RL policy

- State represented in tokens $s = (s_1, s_2, \dots, s_L) \in \mathcal{S}$
- Action also represented in tokens $a = (a_1, a_2, \dots, a_K) \in \mathcal{A}(s)$
- Autoregressive policy: $a_{i+1} \sim \pi_\theta(\cdot \mid s, a_{1:i})$

Training

- Stage 1: Supervised fine-tuning (**SFT**)
  - Gather an offline dataset $\mathcal{D}$ from human (GPT-4) or algorithmic solutions
  - For better **instruction following (generating valid reflections)**
- Stage 2: Reinforcement learning fine-tuning (**RLFT**)
  - Online policy gradient (or PPO)
  - For **better exploration**

Will be mentioned later

Reflection

- Assume access to a (external) reflection model (e.g., GPT-4) $R(s)$
  - Policy now: $a_{i+1} \sim \pi_\theta(\cdot\,|s, R(s), a_{1:i})$
  - For simplicity, assume from now $R(s)$ is included in s
  - Add reflection data into $\mathcal{D}$
- Train (SFT) a local reflect model $\hat{R}_\phi$ using $\mathcal{D}$

$$\mathcal{L}_{\text{reflect}}(\phi) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{L_i} -\log \widehat{R}_\phi(R_{i,j}|s_i, R_{i,:j-1})$$

Simplifying Action Generation

- Highly possible for LMs to generate **invalid** actions due to different $\mathcal{A}(s)$
- Remedy 1: SayCan / action prompt normalization (Ahn et al., 2022; Tan et al., 2024)
  - Enumerate $a \in \mathcal{A}(s)$ and normalize: $p_\theta(a|s) = \frac{\pi_\theta(a|s)}{\sum_{a' \in \mathcal{A}(s)} \pi_\theta(a'|s)}$
  - Time complexity: $\Theta\left(|\mathcal{A}(s)||s|^2 + \sum_{a \in \mathcal{A}(s)} |a|^2\right)$
    - In our experiments $|\mathcal{A}(s)| \approx 20, |s| \approx 500, |a| \approx 5$

Simplifying Action Generation

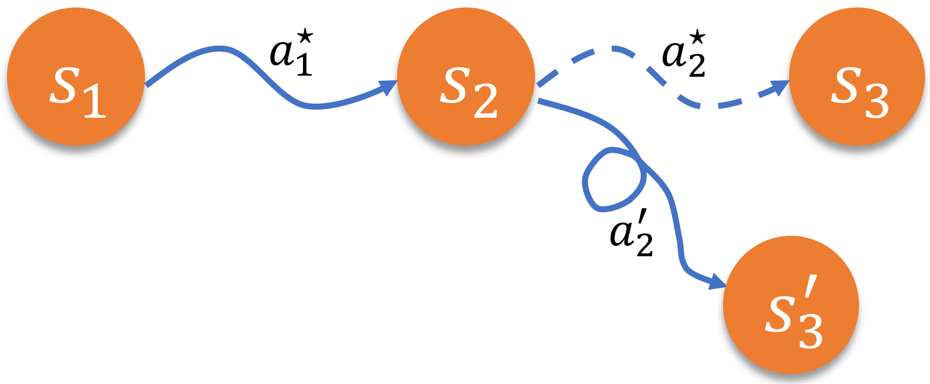
- **Remedy 2 (ours):** Single-prompt action enumeration
  - Like language classification tasks
  - Action enumeration function $\alpha\big(\mathcal{A}(s)\big) = (1, a_1; 2, a_2; \ldots)$
  - Input $\Big(s, \alpha\big(\mathcal{A}(s)\big)\Big)$, output only the choice number
    - Retain only the logits corresponding to the choice token
  - Time complexity: $\Theta\big(|s|^2 + \sum_{a \in \mathcal{A}(s)} |a|^2\big)$
    - 20x faster!

Curriculum Learning

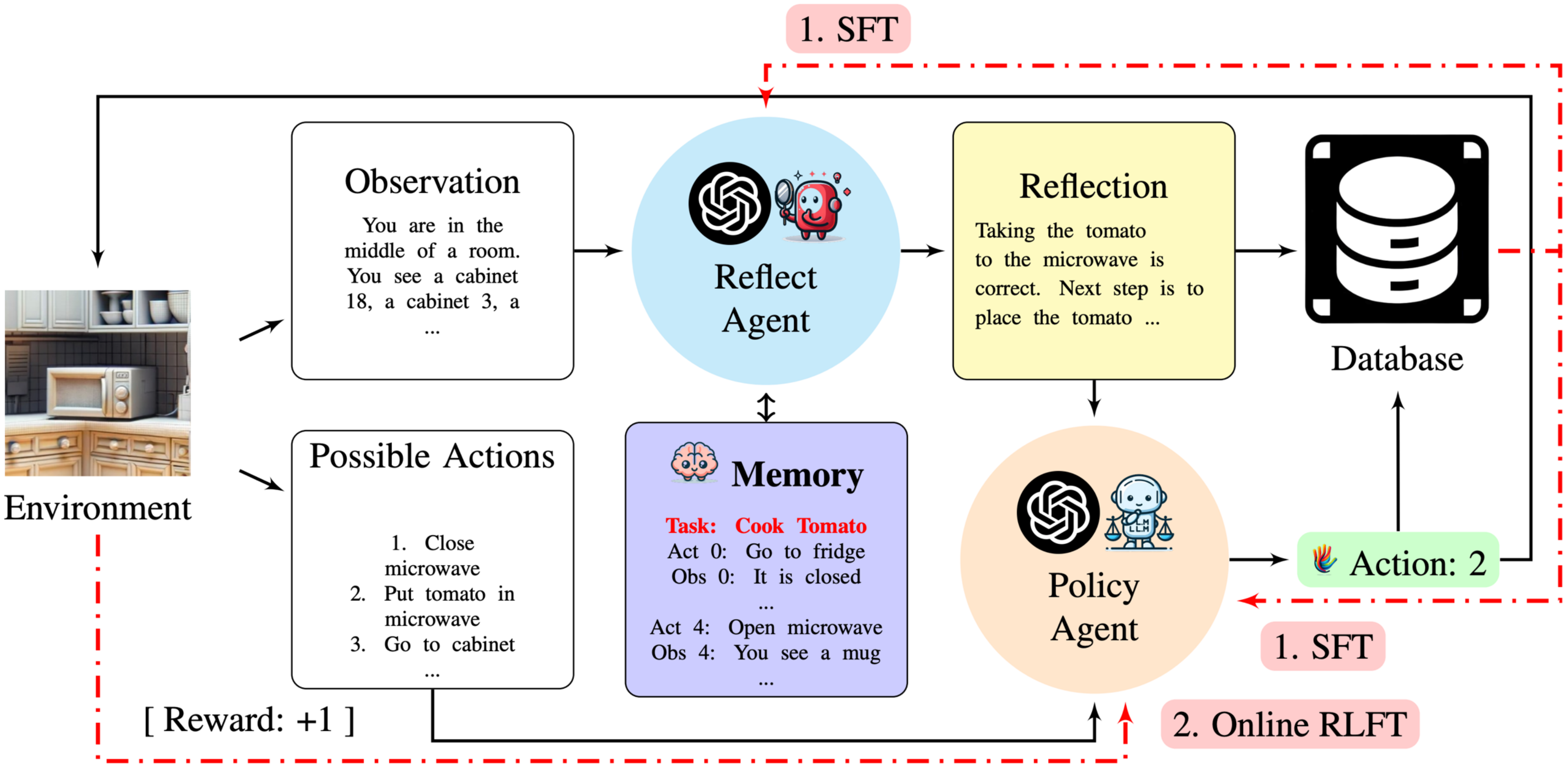
- Use an ordering of tasks to help training (Elman, 1993; Bengio et al., 2009)
- Add extra reward signals when reaching some milestones
- Start from problems with shorter horizons then increase

Negative Data

- Reflection model must be able to correct errors
  - Only optimal or oracle trajectories in SFT data is insufficient
- Perturb each action on an optimal trajectory and tell GPT-4 it is sub-optimal



Full Pipeline

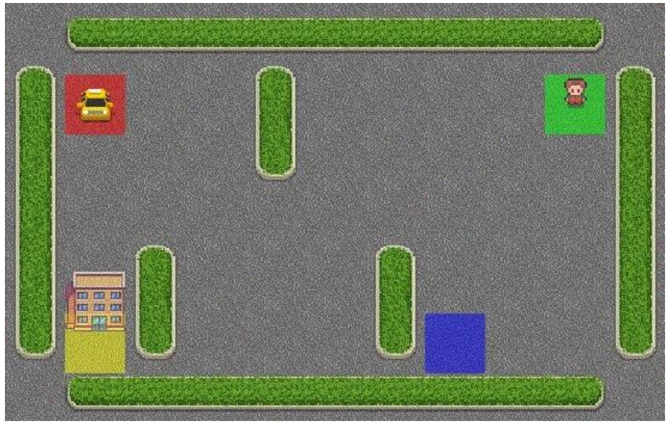


Benchmarks - AutoExplore

- Find the correct file in a file system for a natural language query
- Sandbox
  - Protect original files
  - Track changed files
- Copilot
  - Build prompts
  - Mediate between sandbox and language model

Benchmarks - Taxi

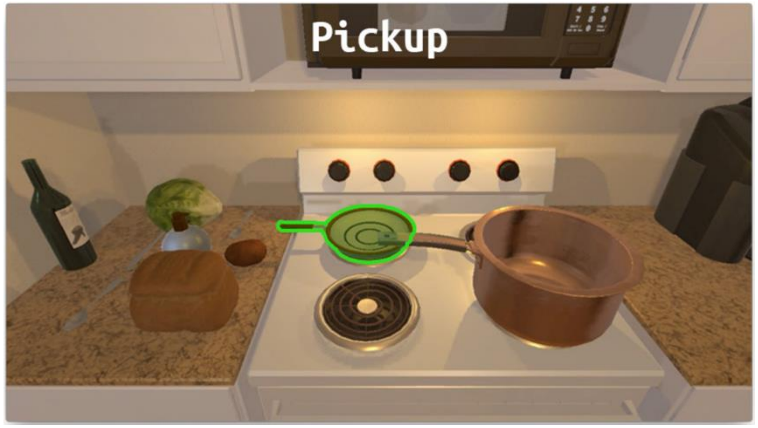
- Represent OpenAI Gym's taxi environment in text
- More challenges
  - Invalid pickup / dropoff kills the game
  - Bumping into wall kills the game



```
+---------+
|T: | : :P|
| : | : : |
| : : : : |
| | : | : |
|D| : | : |
+---------+
```

Benchmarks - ALFWorld

- Adapted from Textworld (Cote et al., 2019) / Alfworld (Shridhar et al., 2020)
- Navigate in a **text** environment to complete a given goal



Results

	Model	AutoExplore Depth 1	AutoExplore Depth 2	DangerousTaxi Pickup	DangerousTaxi +Dropoff*	ALFWorld
Open Source	Mistral 7B	34%	3%	7%	0%	0%
	Llama2 7B-chat	2%	1%	3%	0%	0%
	Orca-2 7B	6%	1%	1%	0%	0%
SFT Only	GPT-2 XL 1.56B	4%	9%	7%	0%	0%
RLFT Only	GPT-2 XL 1.56B	12%	3%	2%	0%	0%
SFT+RLFT (w/o reflection)	GPT-2 XL 1.56B	20%	4%	6%	0%	66%
SFT+RLFT (w/o negative)	GPT-2 XL 1.56B	33%	12%	-	-	-
Reflect-RL (Ours)	GPT-2 XL 1.56B	**36%**	**17%**	**58%**	**29%**	**74%**

References

- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. arXiv preprint arXiv:2310.01798.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In Advances in Neural Information Processing Systems, volume 12. MIT Press.
- Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances.
- Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. 2024. True Knowledge Comes from Practice: Aligning LLMs with Embodied Environments via Reinforcement Learning.
- Jeffrey L. Elman. 1993. Learning and development in neural networks: the importance of starting small. Cognition, 48(1):71–99.
- Yoshua Bengio, Jer´ ome Louradour, Ronan Collobert, ˆ and Jason Weston. 2009. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, ICML ’09, page 41–48, New York, NY, USA. Association for Computing Machinery.
- Marc-Alexandre Cotˆ e, Akos K ´ ad´ ar, Xingdi Yuan, Ben ´ Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2019. Textworld: A learning environment for text-based games. In Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7, pages 41–75. Springer.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cotˆ e,´ Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. arXiv preprint arXiv:2010.03768.

Thank You