Week 6

PPO

Let’s get confused together :)

RLHF

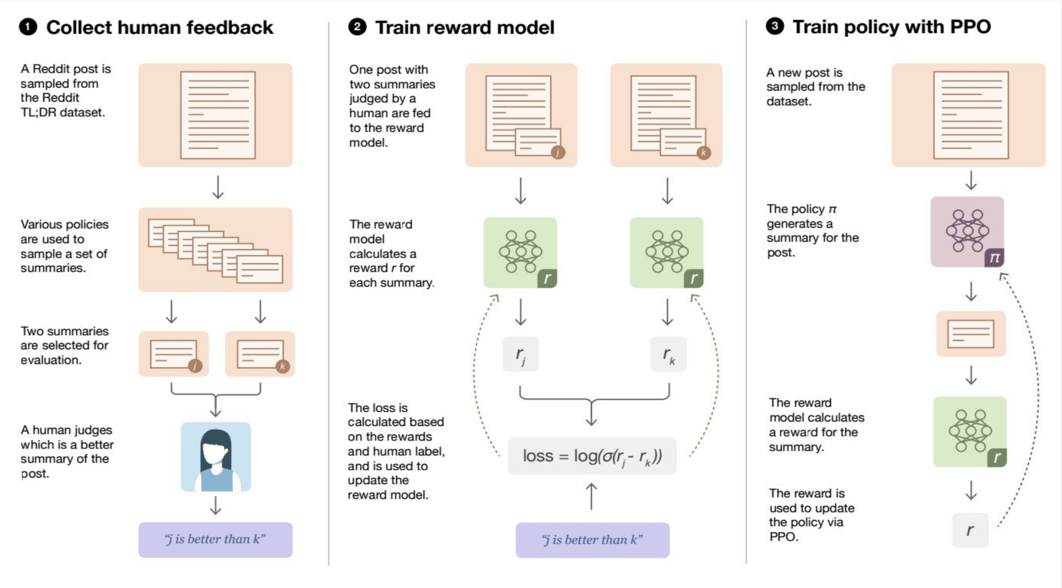
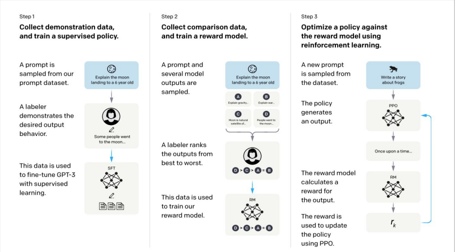

1 Collect human feedback
A Reddit post is sampled from the Reddit TL;DR dataset.
Various policies are used to sample a set of summaries.
Two summaries are selected for evaluation.
j
k
A human judges which is a better summary of the post.
"j is better than k"
2 Train reward model
One post with two summaries judged by a human are fed to the reward model.
j
k
The reward model calculates a reward r for each summary.
r
r
r_j
r_k
The loss is calculated based on the rewards and human label, and is used to update the reward model.
loss = $\log(\sigma(r_j - r_k))$
"j is better than k"
3 Train policy with PPO
A new post is sampled from the dataset.
The policy π generates a summary for the post.
π
The reward model calculates a reward for the summary.
r
The reward is used to update the policy via PPO.
r

Figure 2: Diagram of our human feedback, reward model training, and policy training procedure.

2020


Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

2022

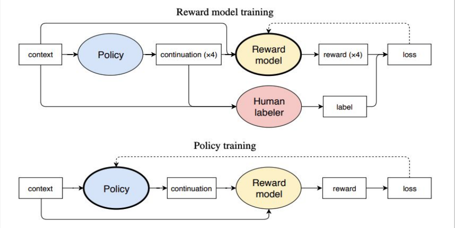
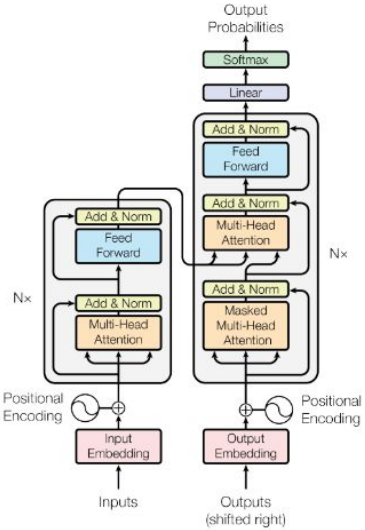


Figure 1: Our training processes for reward model and policy. In the online case, the processes are interleaved.

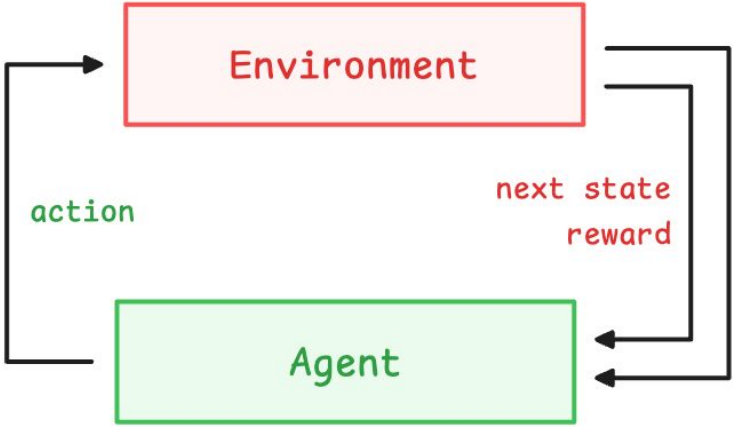
2019

Problems

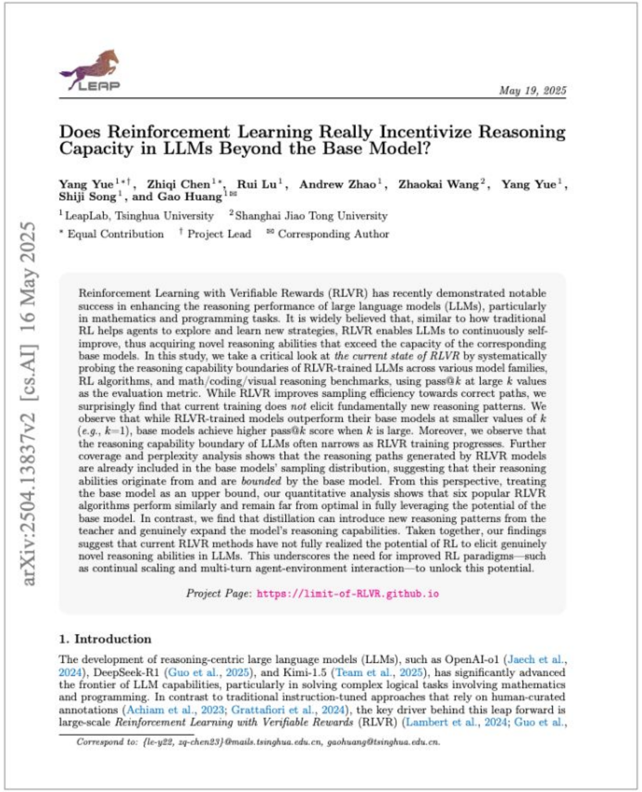
Able to store massive amounts of data



Efficient knowledge extraction



AGI with RL?



May 19, 2025

Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?

Yang Yue[1*†], Zhiqi Chen[1*], Rui Lu[1], Andrew Zhao[1], Zhaokai Wang[2], Yang Yue[1], Shiji Song[1], and Gao Huang[1✉]

[1] LeapLab, Tsinghua University [2] Shanghai Jiao Tong University

[*] Equal Contribution [†] Project Lead [✉] Corresponding Author

arXiv:2504.13837v2 [cs.AI] 16 May 2025

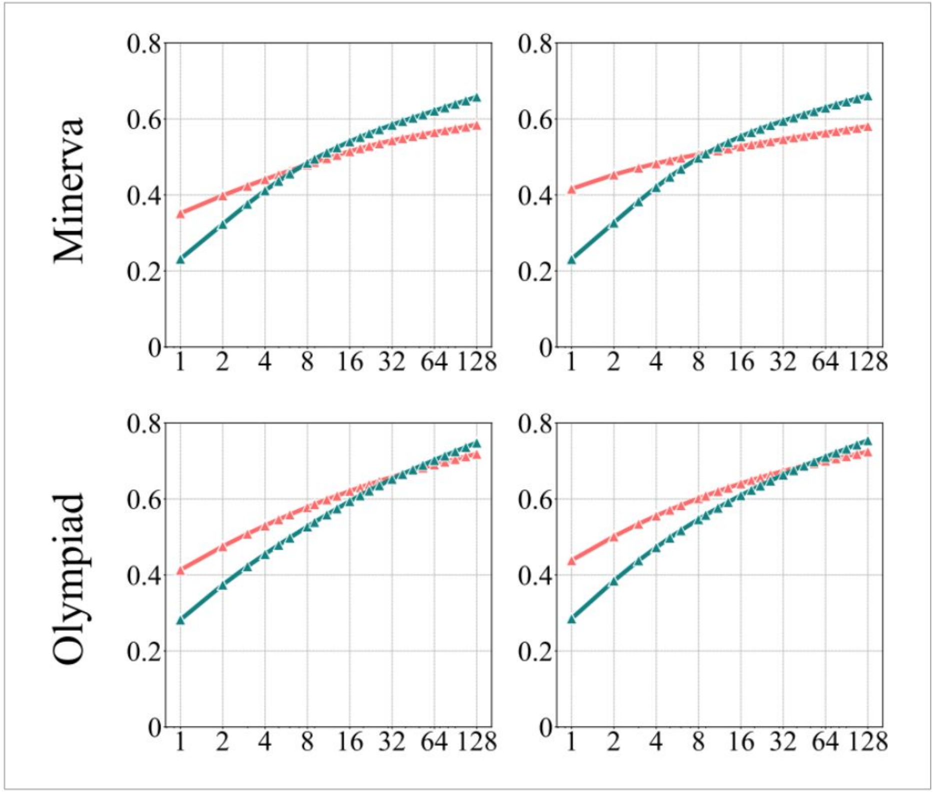
Reinforcement Learning with Verifiable Rewards (RLVR) has recently demonstrated notable success in enhancing the reasoning performance of large language models (LLMs), particularly in mathematics and programming tasks. It is widely believed that, similar to how traditional RL helps agents to explore and learn new strategies, RLVR enables LLMs to continuously self-improve, thus acquiring novel reasoning abilities that exceed the capacity of the corresponding base models. In this study, we take a critical look at *the current state of RLVR* by systematically probing the reasoning capability boundaries of RLVR-trained LLMs across various model families, RL algorithms, and math/coding/visual reasoning benchmarks, using pass@k at large k values as the evaluation metric. While RLVR improves sampling efficiency towards correct paths, we surprisingly find that current training does *not* elicit fundamentally new reasoning patterns. We observe that while RLVR-trained models outperform their base models at smaller values of k (*e.g.*, k=1), base models achieve higher pass@k score when k is large. Moreover, we observe that the reasoning capability boundary of LLMs often narrows as RLVR training progresses. Further coverage and perplexity analysis shows that the reasoning paths generated by RLVR models are already included in the base models' sampling distribution, suggesting that their reasoning abilities originate from and are *bounded* by the base model. From this perspective, treating the base model as an upper bound, our quantitative analysis shows that six popular RLVR algorithms perform similarly and remain far from optimal in fully leveraging the potential of the base model. In contrast, we find that distillation can introduce new reasoning patterns from the teacher and genuinely expand the model's reasoning capabilities. Taken together, our findings suggest that current RLVR methods have not fully realized the potential of RL to elicit genuinely novel reasoning abilities in LLMs. This underscores the need for improved RL paradigms—such as continual scaling and multi-turn agent-environment interaction—to unlock this potential.

Project Page: `https://limit-of-RLVR.github.io`

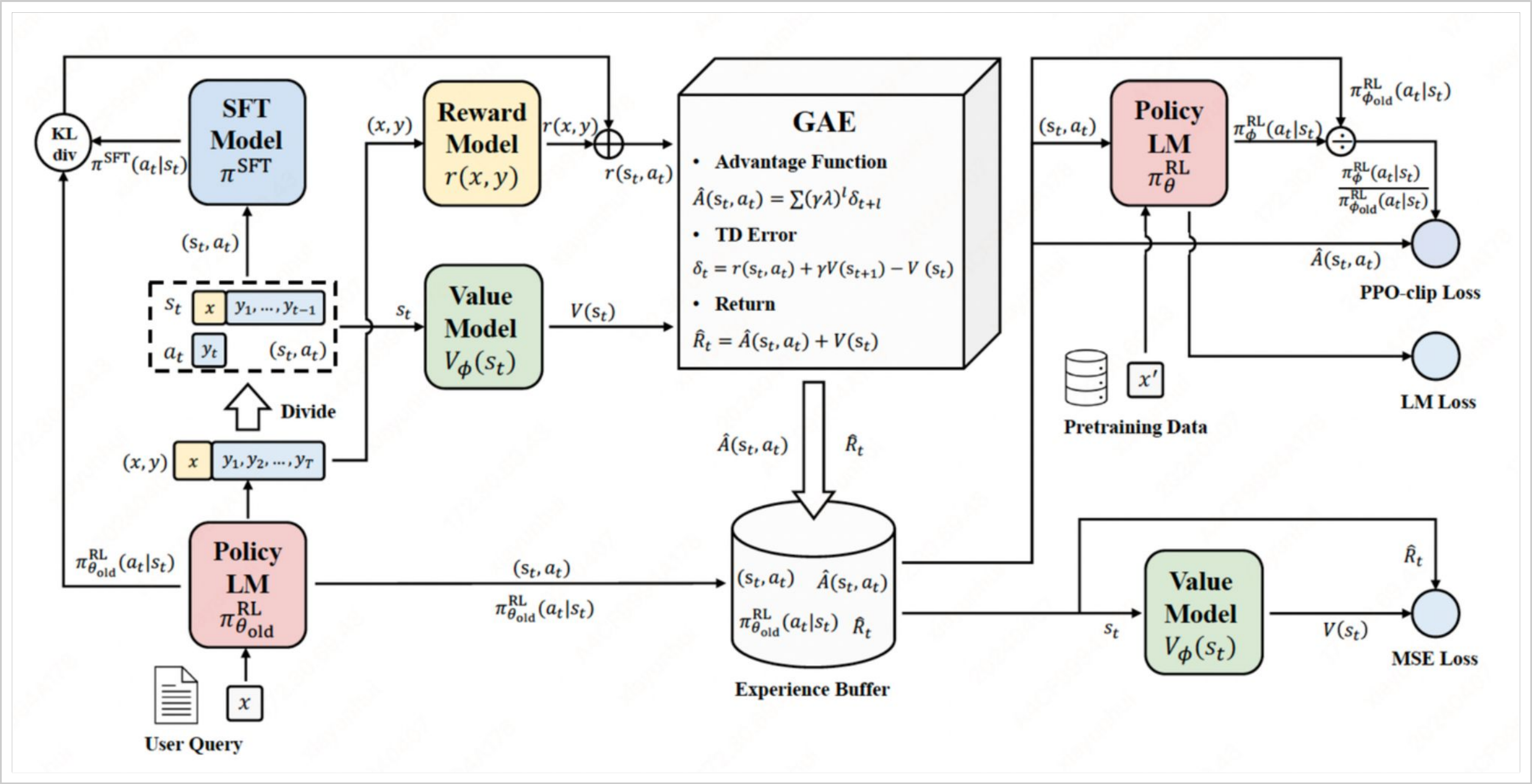
1. Introduction

The development of reasoning-centric large language models (LLMs), such as OpenAI-o1 (Jaech et al., 2024), DeepSeek-R1 (Guo et al., 2025), and Kimi-1.5 (Team et al., 2025), has significantly advanced the frontier of LLM capabilities, particularly in solving complex logical tasks involving mathematics and programming. In contrast to traditional instruction-tuned approaches that rely on human-curated annotations (Achiam et al., 2023; Grattafiori et al., 2024), the key driver behind this leap forward is large-scale *Reinforcement Learning with Verifiable Rewards* (RLVR) (Lambert et al., 2024; Guo et al.,

Correspond to: {le-y22, zq-chen23}@mails.tsinghua.edu.cn, gaohuang@tsinghua.edu.cn.



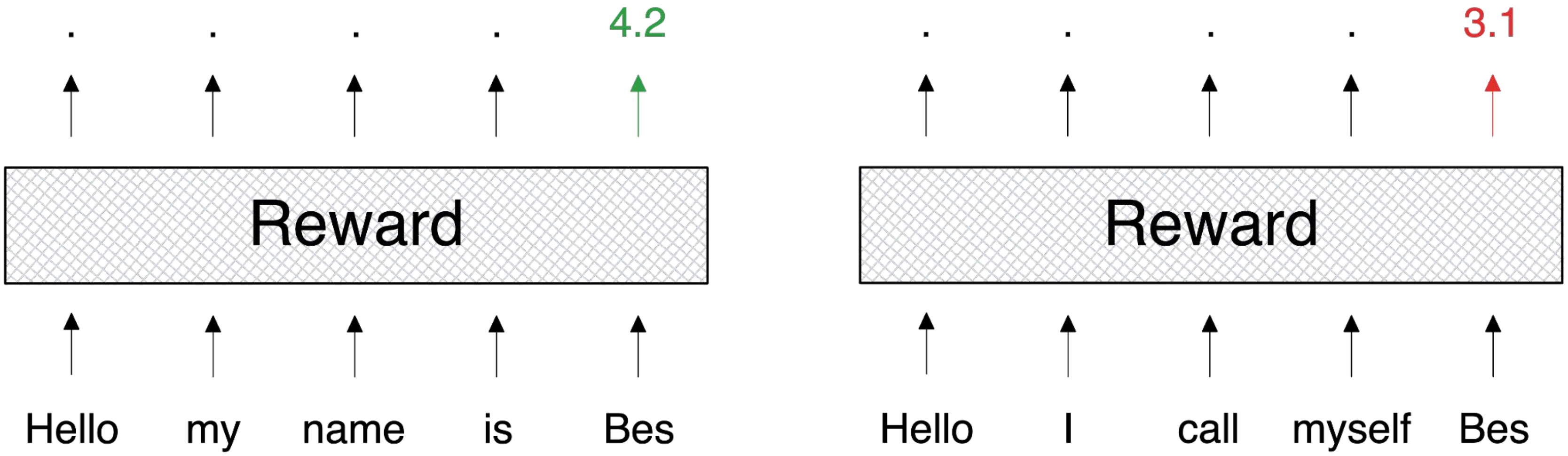
Overview


KL div
SFT Model π^{SFT}
$\pi^{SFT}(a_t|s_t)$
(x, y)
Reward Model $r(x, y)$
$r(x, y)$
$r(s_t, a_t)$
GAE
Advantage Function
$\hat{A}(s_t, a_t) = \sum(\gamma\lambda)^l \delta_{t+l}$
TD Error
$\delta_t = r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t)$
Return
$\hat{R}_t = \hat{A}(s_t, a_t) + V(s_t)$
(s_t, a_t)
s_t x $y_1, ..., y_{t-1}$
a_t y_t (s_t, a_t)
s_t
Value Model $V_\phi(s_t)$
$V(s_t)$
Divide
(x, y) x $y_1, y_2, ..., y_T$
$\pi^{RL}_{\theta_{old}}(a_t|s_t)$
Policy LM $\pi^{RL}_{\theta_{old}}$
(s_t, a_t)
$\pi^{RL}_{\theta_{old}}(a_t|s_t)$
User Query
x
$\hat{A}(s_t, a_t)$
$\hat{R}_t$
(s_t, a_t) $\hat{A}(s_t, a_t)$
$\pi^{RL}_{\theta_{old}}(a_t|s_t)$ $\hat{R}_t$
Experience Buffer
(s_t, a_t)
Policy LM π^{RL}_θ
$\pi^{RL}_\phi(a_t|s_t)$
$\pi^{RL}_{\phi_{old}}(a_t|s_t)$
$\frac{\pi^{RL}_\phi(a_t|s_t)}{\pi^{RL}_{\phi_{old}}(a_t|s_t)}$
$\hat{A}(s_t, a_t)$
PPO-clip Loss
x'
Pretraining Data
LM Loss
$\hat{R}_t$
s_t
Value Model $V_\phi(s_t)$
$V(s_t)$
MSE Loss

Reward

Bradley–Terry

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,y_w,y_l)}\Big[\log\sigma\big(r_\theta(x,y_w) - r_\theta(x,y_l)\big)\Big] = -\mathbb{E}_{(x,y_w,y_l)}\log\left[\frac{e^{r_\theta(x,y_w)}}{e^{r_\theta(x,y_w)} + e^{r_\theta(x,y_l)}}\right]$$



. . . . 4.2

Reward

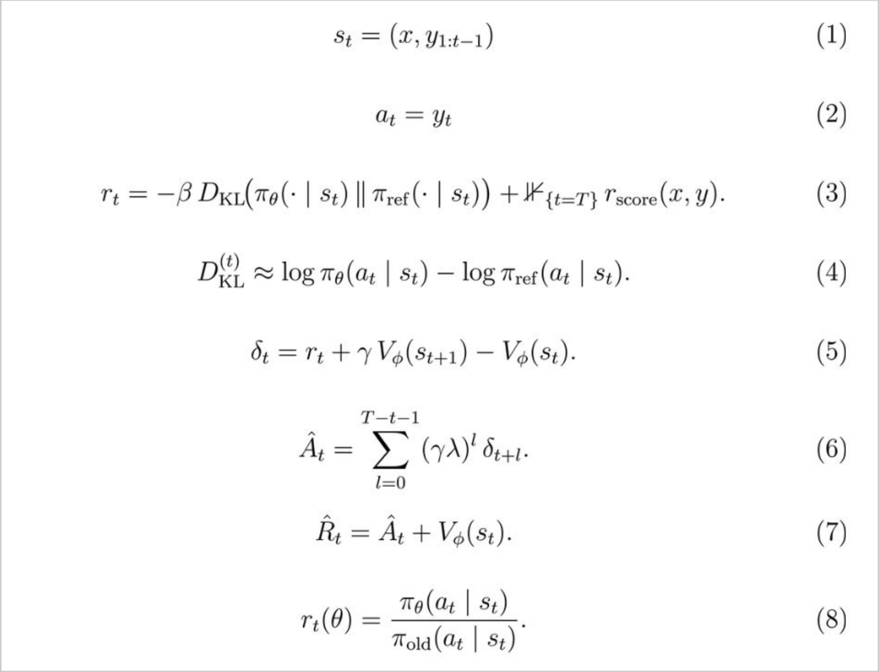
Hello my name is Bes

. . . . 3.1

Reward

Hello I call myself Bes

Math



$$s_t = (x, y_{1:t-1}) \tag{1}$$

$$a_t = y_t \tag{2}$$

$$r_t = -\beta\, D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid s_t)\big) + \nVdash_{\{t=T\}}\, r_{\mathrm{score}}(x, y). \tag{3}$$

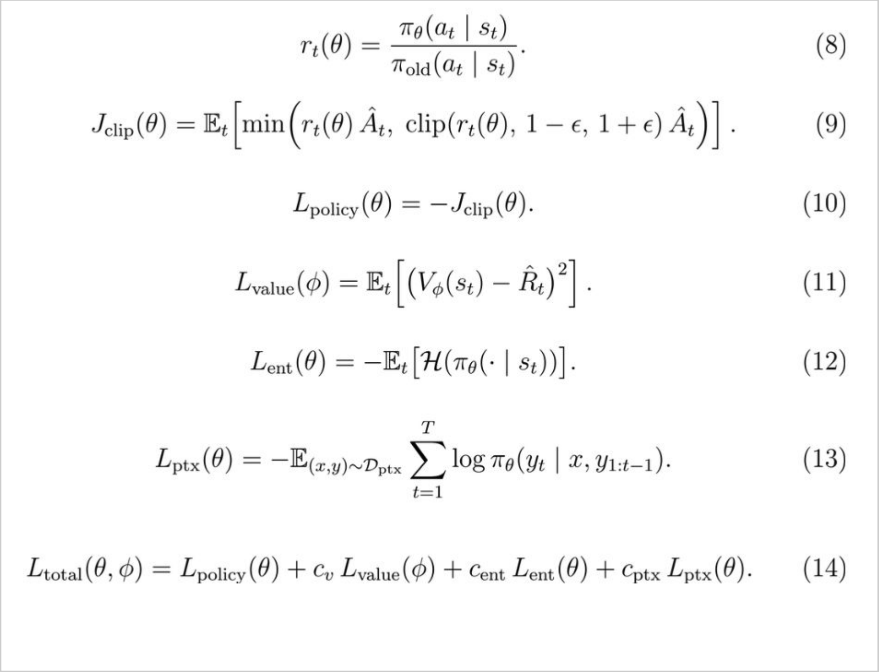
$$D_{\mathrm{KL}}^{(t)} \approx \log \pi_\theta(a_t \mid s_t) - \log \pi_{\mathrm{ref}}(a_t \mid s_t). \tag{4}$$

$$\delta_t = r_t + \gamma\, V_\phi(s_{t+1}) - V_\phi(s_t). \tag{5}$$

$$\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l\, \delta_{t+l}. \tag{6}$$

$$\hat{R}_t = \hat{A}_t + V_\phi(s_t). \tag{7}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\mathrm{old}}(a_t \mid s_t)}. \tag{8}$$



$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\mathrm{old}}(a_t \mid s_t)}. \tag{8}$$

$$J_{\mathrm{clip}}(\theta) = \mathbb{E}_t\Big[\min\Big(r_t(\theta)\,\hat{A}_t,\ \mathrm{clip}(r_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\,\hat{A}_t\Big)\Big]\,. \tag{9}$$

$$L_{\mathrm{policy}}(\theta) = -J_{\mathrm{clip}}(\theta). \tag{10}$$

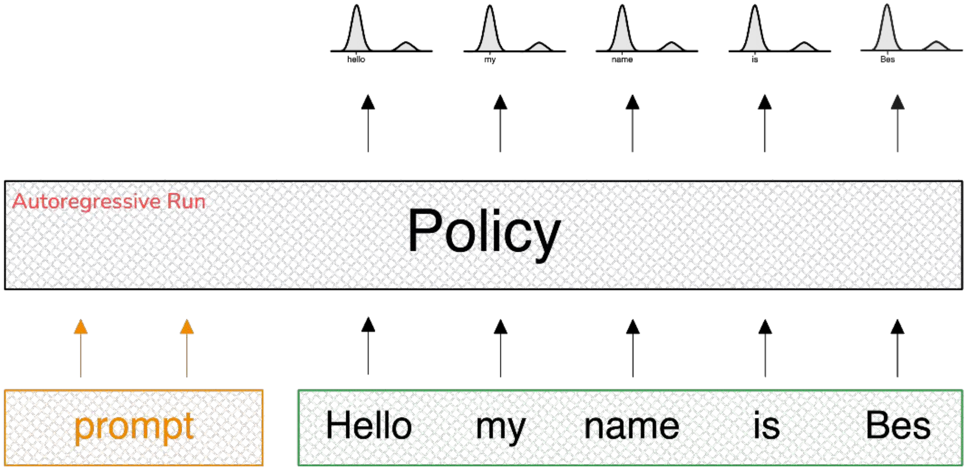
$$L_{\mathrm{value}}(\phi) = \mathbb{E}_t\Big[\big(V_\phi(s_t) - \hat{R}_t\big)^2\Big]\,. \tag{11}$$

$$L_{\mathrm{ent}}(\theta) = -\mathbb{E}_t\big[\mathcal{H}(\pi_\theta(\cdot \mid s_t))\big]. \tag{12}$$

$$L_{\mathrm{ptx}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{ptx}}} \sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{1:t-1}). \tag{13}$$

$$L_{\mathrm{total}}(\theta, \phi) = L_{\mathrm{policy}}(\theta) + c_v\, L_{\mathrm{value}}(\phi) + c_{\mathrm{ent}}\, L_{\mathrm{ent}}(\theta) + c_{\mathrm{ptx}}\, L_{\mathrm{ptx}}(\theta). \tag{14}$$

Policy & Friends



yes, everything is calculated at the token level, it is a form of reward shaping

this is calculated for every single token, correct?

only the last token gets the extra reward score, previous tokens get only the approximate KL divergence

what is this?

$$r_t = -\beta \, D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid s_t)\big) + \mathbb{1}_{\{t=T\}} \, r_{\mathrm{score}}(x, y). \tag{3}$$

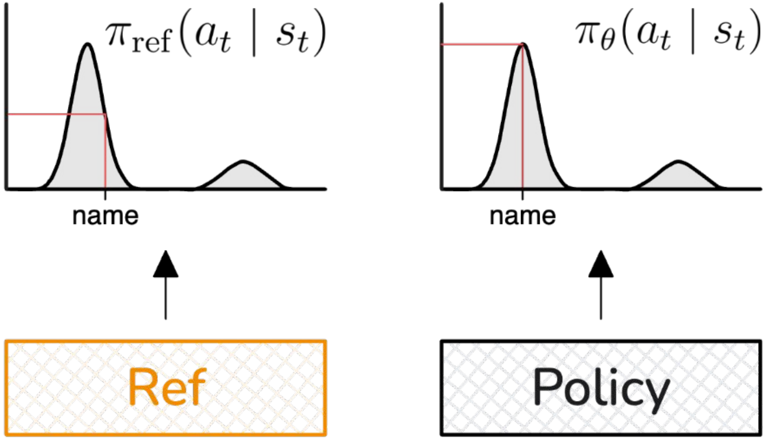
$$D_{\mathrm{KL}}^{(t)} \approx \log \pi_\theta(a_t \mid s_t) - \log \pi_{\mathrm{ref}}(a_t \mid s_t). \tag{4}$$

why an approximate?

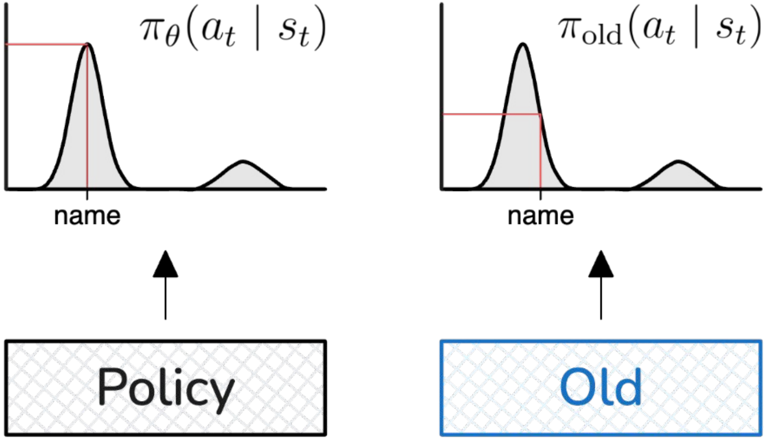
approximated via sampling, because the true KL divergence would require the all distribution or all the possible actions

KL

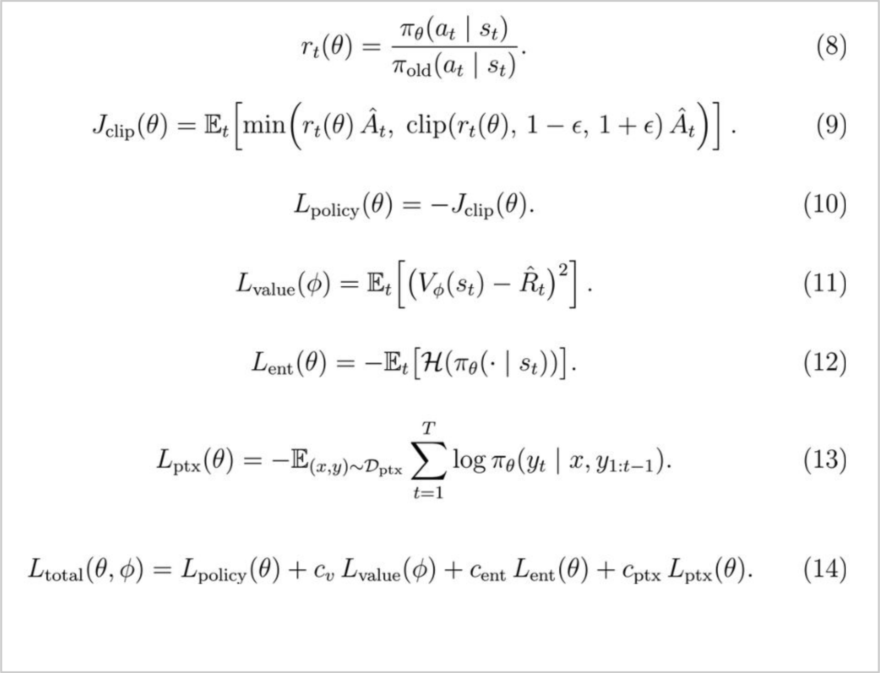
$$D_{\mathrm{KL}}^{(t)} \approx \log \pi_\theta(a_t \mid s_t) - \log \pi_{\mathrm{ref}}(a_t \mid s_t)$$


$\pi_{\mathrm{ref}}(a_t \mid s_t)$
name
$\pi_\theta(a_t \mid s_t)$
name
Ref
Policy

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\mathrm{old}}(a_t \mid s_t)}$$


$\pi_\theta(a_t \mid s_t)$
name
$\pi_{\mathrm{old}}(a_t \mid s_t)$
name
Policy
Old

clip



$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\text{old}}(a_t \mid s_t)}. \tag{8}$$

$$J_{\text{clip}}(\theta) = \mathbb{E}_t\Big[\min\Big(r_t(\theta)\,\hat{A}_t,\ \text{clip}(r_t(\theta),\,1-\epsilon,\,1+\epsilon)\,\hat{A}_t\Big)\Big]\,. \tag{9}$$

$$L_{\text{policy}}(\theta) = -J_{\text{clip}}(\theta). \tag{10}$$

$$L_{\text{value}}(\phi) = \mathbb{E}_t\Big[\big(V_\phi(s_t) - \hat{R}_t\big)^2\Big]\,. \tag{11}$$

$$L_{\text{ent}}(\theta) = -\mathbb{E}_t\big[\mathcal{H}(\pi_\theta(\cdot \mid s_t))\big]. \tag{12}$$

$$L_{\text{ptx}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{ptx}}} \sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{1:t-1}). \tag{13}$$

$$L_{\text{total}}(\theta,\phi) = L_{\text{policy}}(\theta) + c_v\,L_{\text{value}}(\phi) + c_{\text{ent}}\,L_{\text{ent}}(\theta) + c_{\text{ptx}}\,L_{\text{ptx}}(\theta). \tag{14}$$

$$\text{clip}(x,a,b) = \begin{cases} a, & x < a, \\ x, & a \le x \le b, \\ b, & x > b. \end{cases}$$