NeurIPS 2024

InfoRM: Mitigating Reward Hacking in RLHF via Information-Theoretic Reward Modeling

Speaker: Miao Yuchun

Email: szmyc1@163.com

Content

- ✓ **Background and Motivation**
- **Methodology**
- **Experiments**
- **Additional Strength**

Background and Motivation

Pipeline of LLM Alighment

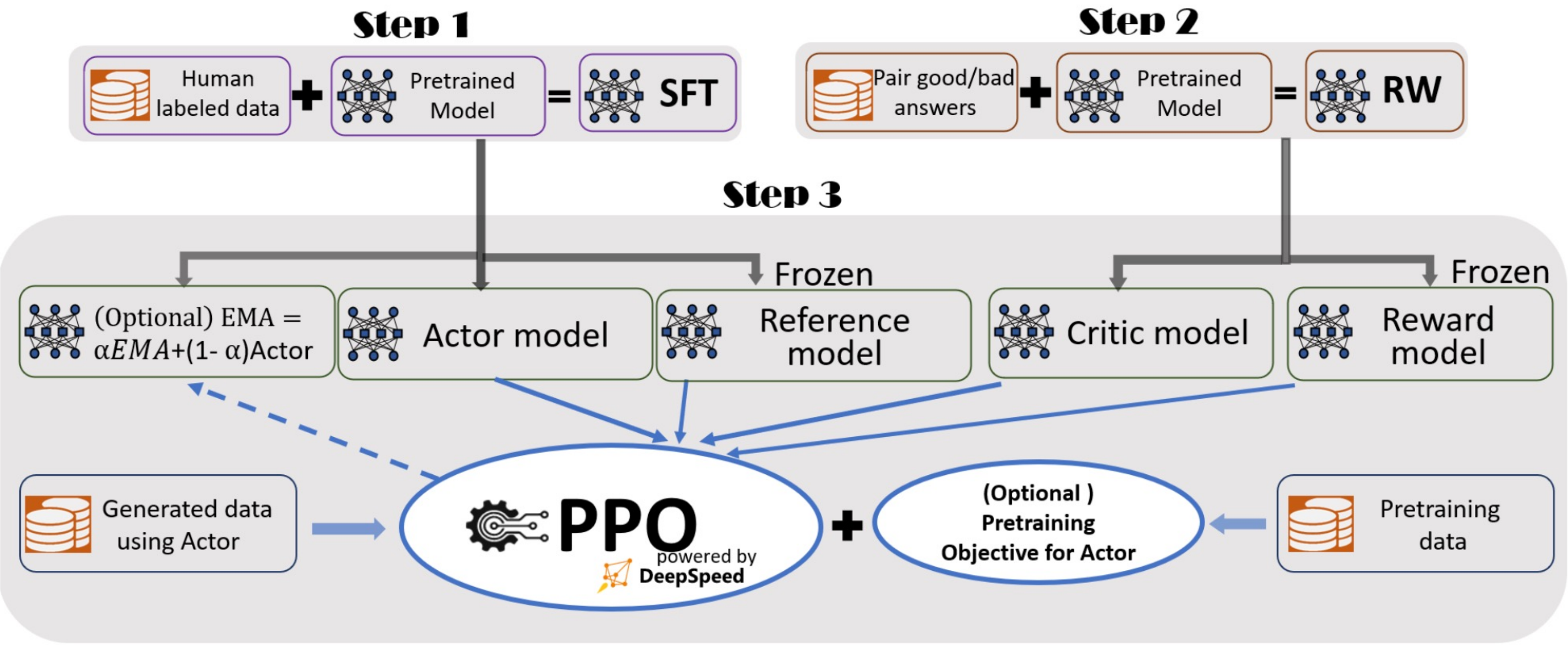


Figure 1: The illustration of DeepSpeed Chat's RLHF training pipeline with optional features.

Background and Motivation

Reward Hacking

Despite the success of reinforcement learning from human feedback (RLHF) in aligning language models with human values, reward hacking, also termed reward over-optimization, remains a critical challenge. This issue can be manifested in various ways, from copying styles without generating meaningful content to exhibiting excessive caution in responses

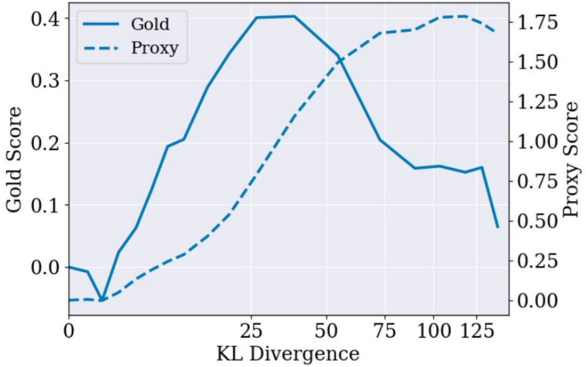


Figure 2: An example of overoptimization.

Background and Motivation

Motivation

One primary cause of reward over-optimization in the reward modeling process is reward mis-generalization, where RMs may incorrectly generalize training data, resulting in poor proxies for actual human preference. This occurs because the same human feedback can be interpreted in multiple ways by RMs, even with ample data.

Consequently, RMs often rely on spurious features, such as length bias, which correlate with ranking labels but are irrelevant to human preferences. Over-exploiting such information leads to RM overfitting, undermining generalizability and causing instability during the RL stage.

Content

Background and Motivation

✓ **Methodology**

Experiments

Additional Strength

Methodology

Main Idea

In this work, we propose a new reward modeling framework from an information-theoretic perspective, namely, InfoRM. The advantages of our framework are two-fold:

Firstly, leveraging MI modeling, InfoRM removes irrelevant information from the IB latent representation, ensuring generalizable human preference modeling. This directly addresses reward misgeneralization by retaining only the features that genuinely reflect human preferences.

Secondly, InfoRM excels in detecting over-optimization. We discovered a correlation between reward over-optimization and the emergence of outliers in InfoRM's IB latent space, a phenomenon absent in RM without IB. Based on this, we designed the Cluster Separation Index (CSI) to detect over-optimization by quantifying deviations in RLHF model-generated sample distributions.

Methodology

Overview

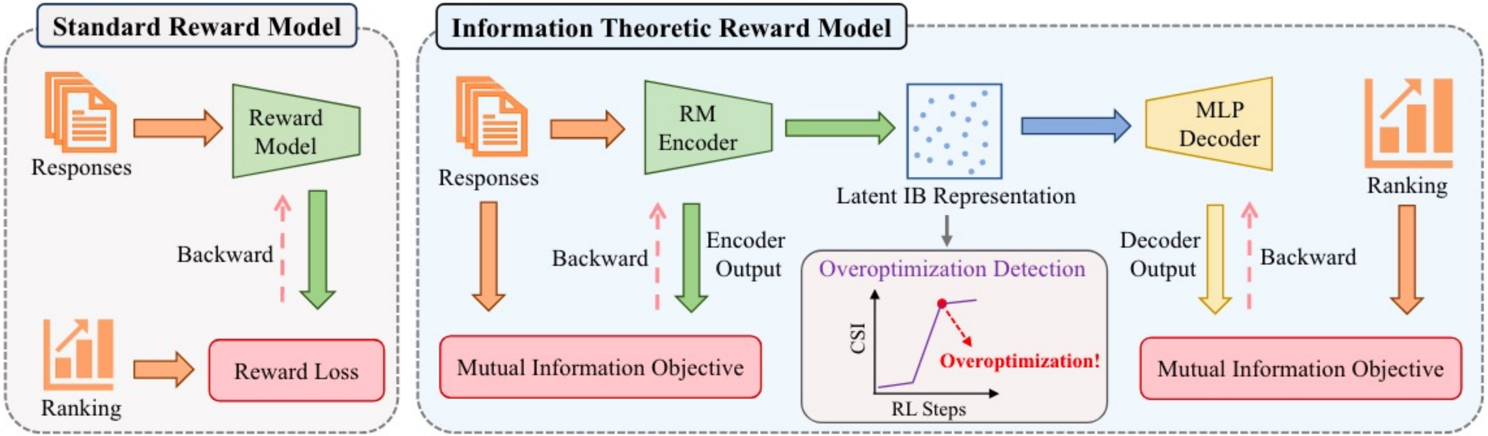


Figure 1: Comparison between `standard RM` and our information-theoretic reward model (`InfoRM`). `InfoRM` distinguishes itself by enhancing RM generalizability through mutual information modeling. Additionally, a distinct feature of `InfoRM` is its overoptimization detection mechanism, which can guide parameter selection and algorithm design in subsequent RLHF. Specifically, the RM encoder is derived from the standard RM, with modification to the final layer.

Methodology

Formulation

The objective of our information-theoretic reward modeling framework *J(θ)* can be formulated as follows:

$$\max_{\boldsymbol{\theta}} \; J(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \; I_{\text{preference}} - \beta I_{\text{bottleneck}} = \max_{\boldsymbol{\theta}} \; I(S, Y) - \beta I(X, S|Y),$$

The variational lower bound is:

$$\begin{aligned} J(\boldsymbol{\phi}, \boldsymbol{\psi}) &\geq J_{\text{VLB}}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{D}} \left[J_{\text{preference}} - \beta J_{\text{bottleneck}} \right] \\ J_{\text{preference}} &= \int p_{\phi}(\boldsymbol{s}|\boldsymbol{x}) \log q_{\psi}(y|\boldsymbol{s}) d\boldsymbol{s} \\ J_{\text{bottleneck}} &= \text{KL}\left[p_{\phi}(\boldsymbol{S}|\boldsymbol{x}), r(\boldsymbol{S}) \right], \end{aligned}$$

Thus, the final objective for our information-theoretic reward modeling reads

$$\begin{aligned} \max_{\{\boldsymbol{\phi}, \boldsymbol{\psi}\}} J_{\text{VLB}}(\boldsymbol{\phi}, \boldsymbol{\psi}) &\approx \max_{\{\boldsymbol{\phi}, \boldsymbol{\psi}\}} \mathbb{E}_{(\boldsymbol{x}^w, \boldsymbol{x}^l) \sim \mathcal{D}} \left[L_{\text{preference}} - \beta L_{\text{bottleneck}} \right] \\ L_{\text{preference}} &= \log \sigma \left(g_{\psi}(h_{\phi}(\boldsymbol{x}^w, \boldsymbol{\epsilon}^w)) - g_{\psi}(h_{\phi}(\boldsymbol{x}^l, \boldsymbol{\epsilon}^l)) \right) \\ L_{\text{bottleneck}} &= \text{KL}\left[p_{\phi}(\boldsymbol{S}|\boldsymbol{x}^w), r(\boldsymbol{S}) \right] + \text{KL}\left[p_{\phi}(\boldsymbol{S}|\boldsymbol{x}^l), r(\boldsymbol{S}) \right], \end{aligned}$$

Content

Background and Motivation

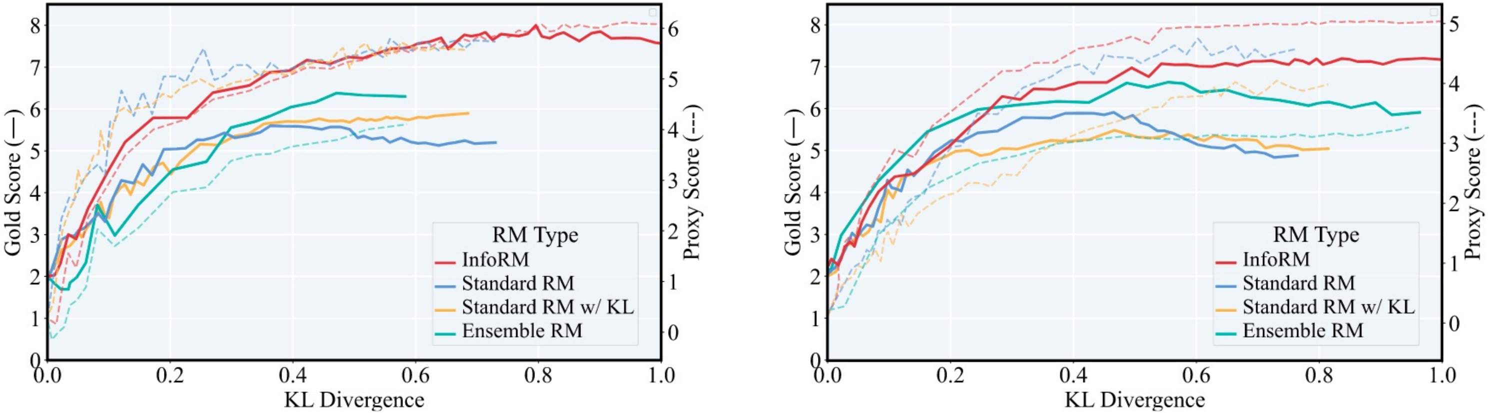
Methodology

✓ Experiments

Additional Strength

Experiments

Simulated Experiments



(a) Without label noise

(b) With 25% label noise

Figure 4: Simulated RLHF results for different proxy RMs (1.4B). Solid and dashed lines represent the gold and proxy scores, respectively. In later RL stages, as KL divergence increases, `Standard RM` shows a declining gold score and a rising proxy score, indicating overoptimization. Conversely, our `InfoRM` maintains consistent growth in both scores, effectively mitigating overoptimization.

Experiments

Real-World Experiments

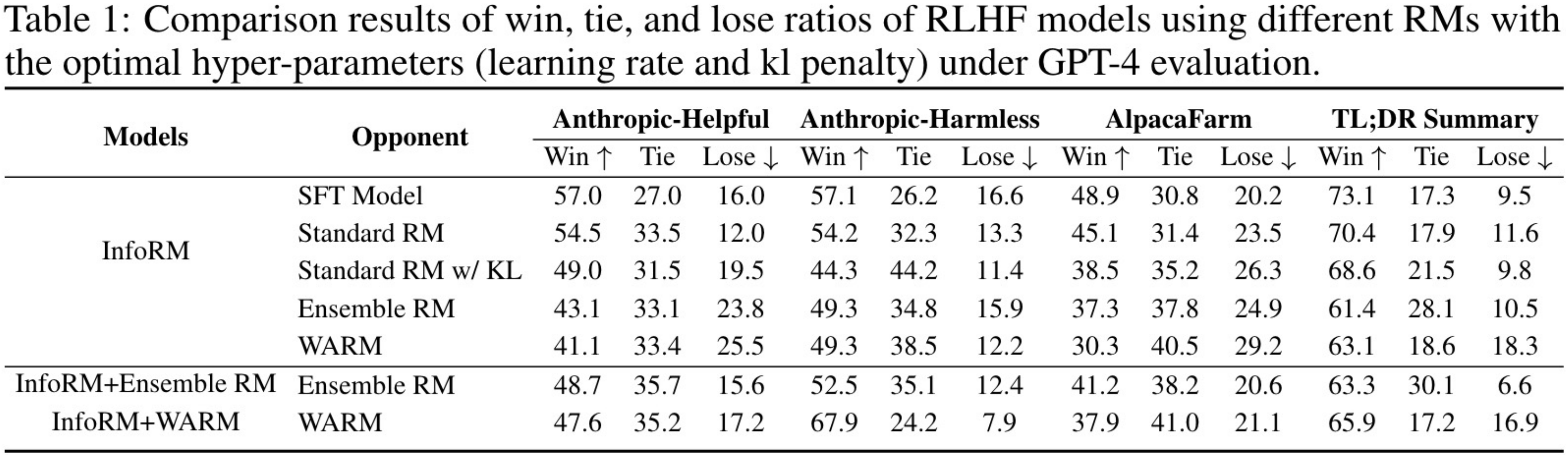


Table 1: Comparison results of win, tie, and lose ratios of RLHF models using different RMs with the optimal hyper-parameters (learning rate and kl penalty) under GPT-4 evaluation.

Models	Opponent	Anthropic-Helpful			Anthropic-Harmless			AlpacaFarm			TL;DR Summary		
		Win ↑	Tie	Lose ↓	Win ↑	Tie	Lose ↓	Win ↑	Tie	Lose ↓	Win ↑	Tie	Lose ↓
InfoRM	SFT Model	57.0	27.0	16.0	57.1	26.2	16.6	48.9	30.8	20.2	73.1	17.3	9.5
	Standard RM	54.5	33.5	12.0	54.2	32.3	13.3	45.1	31.4	23.5	70.4	17.9	11.6
	Standard RM w/ KL	49.0	31.5	19.5	44.3	44.2	11.4	38.5	35.2	26.3	68.6	21.5	9.8
	Ensemble RM	43.1	33.1	23.8	49.3	34.8	15.9	37.3	37.8	24.9	61.4	28.1	10.5
	WARM	41.1	33.4	25.5	49.3	38.5	12.2	30.3	40.5	29.2	63.1	18.6	18.3
InfoRM+Ensemble RM	Ensemble RM	48.7	35.7	15.6	52.5	35.1	12.4	41.2	38.2	20.6	63.3	30.1	6.6
InfoRM+WARM	WARM	47.6	35.2	17.2	67.9	24.2	7.9	37.9	41.0	21.1	65.9	17.2	16.9

Content

Background and Motivation

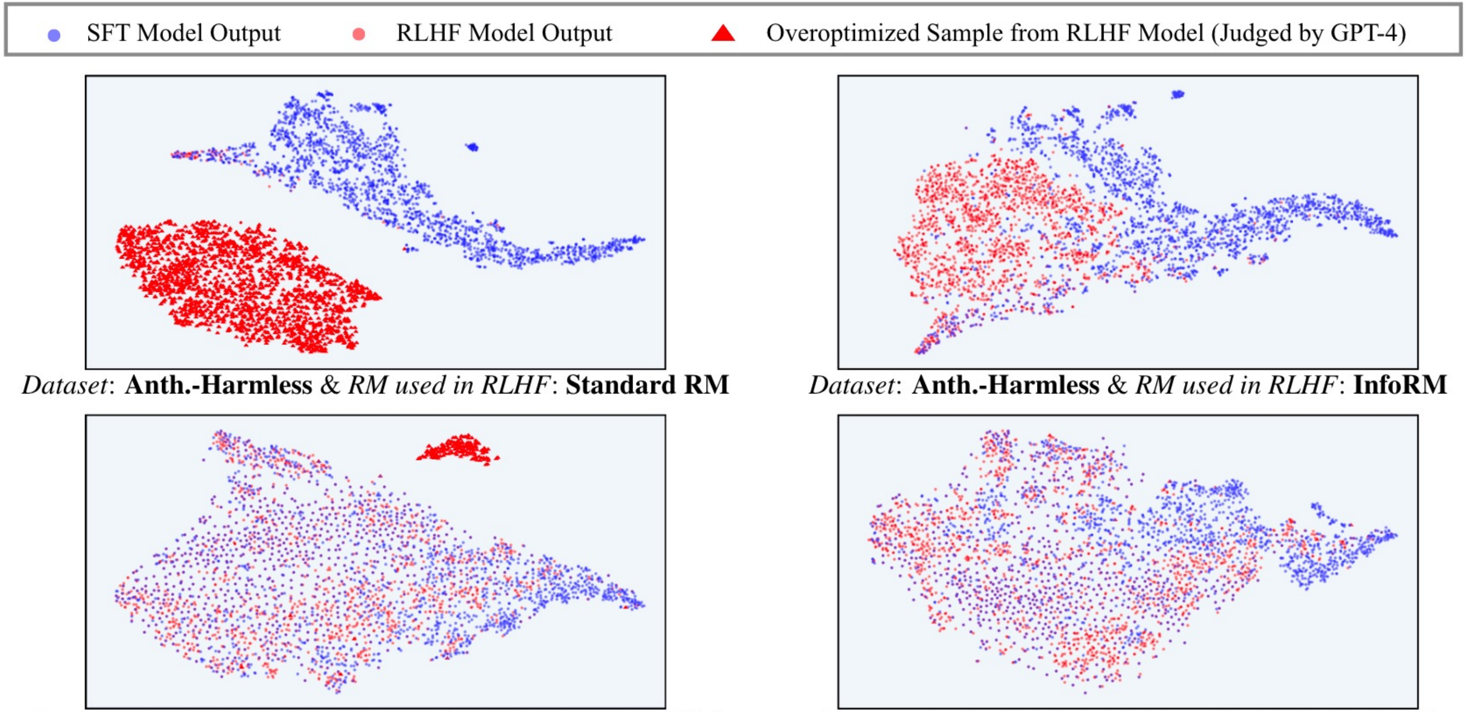
Methodology

Main Experiments

✓ Additional Strength

Additional Strength

Reward Over-optimization Detection



Dataset: **Anth.-Harmless** & *RM used in RLHF*: **Standard RM**

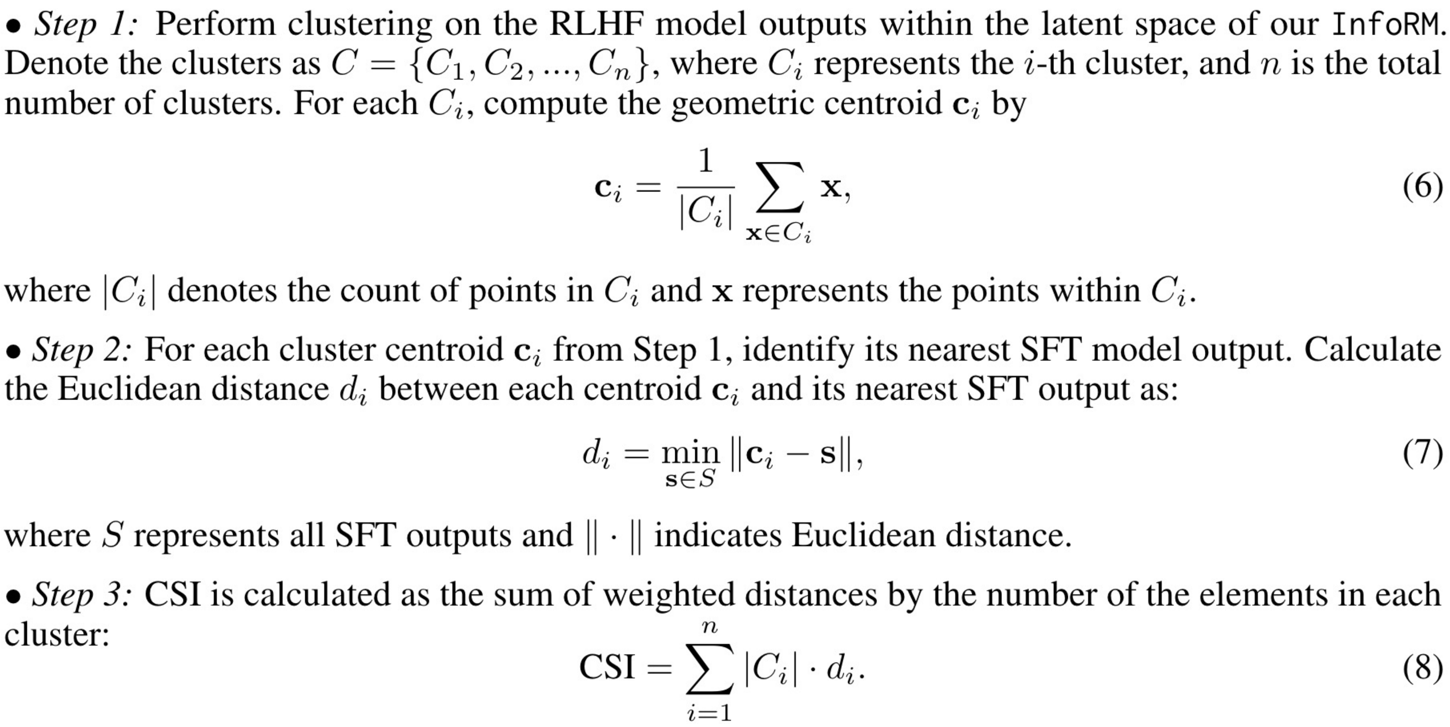
Dataset: **Anth.-Harmless** & *RM used in RLHF*: **InfoRM**

Dataset: **Anth.-Helpful** & *RM used in RLHF*: **Standard RM**

Dataset: **Anth.-Helpful** & *RM used in RLHF*: **InfoRM**

Additional Strength

Reward Over-optimization Detection



• *Step 1:* Perform clustering on the RLHF model outputs within the latent space of our `InfoRM`. Denote the clusters as $C = \{C_1, C_2, ..., C_n\}$, where C_i represents the i-th cluster, and n is the total number of clusters. For each C_i, compute the geometric centroid $\mathbf{c}_i$ by

$$\mathbf{c}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}, \tag{6}$$

where $|C_i|$ denotes the count of points in C_i and $\mathbf{x}$ represents the points within C_i.

• *Step 2:* For each cluster centroid $\mathbf{c}_i$ from Step 1, identify its nearest SFT model output. Calculate the Euclidean distance d_i between each centroid $\mathbf{c}_i$ and its nearest SFT output as:

$$d_i = \min_{\mathbf{s} \in S} \|\mathbf{c}_i - \mathbf{s}\|, \tag{7}$$

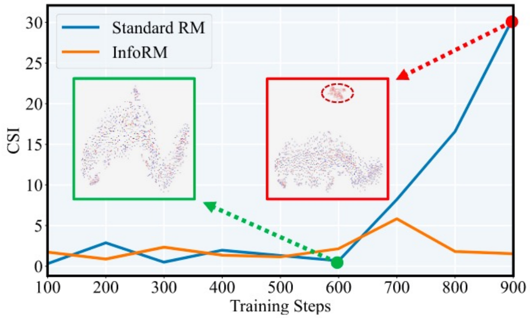
where S represents all SFT outputs and $\| \cdot \|$ indicates Euclidean distance.

• *Step 3:* CSI is calculated as the sum of weighted distances by the number of the elements in each cluster:

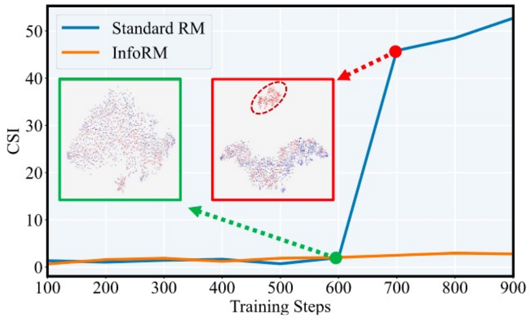
$$\text{CSI} = \sum_{i=1}^{n} |C_i| \cdot d_i. \tag{8}$$

Additional Strength

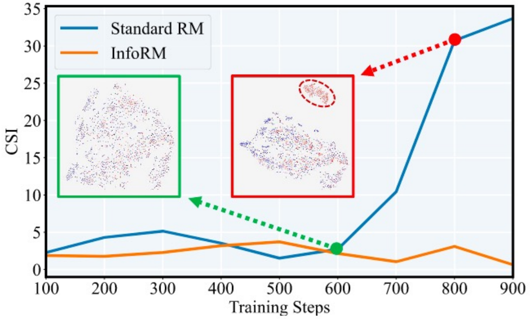
Reward Over-optimization Detection



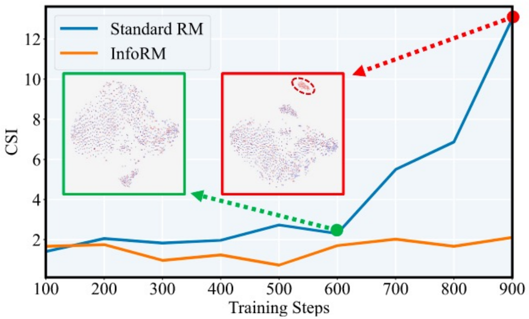
Dataset used for generation: **AlpacaFarm**



Dataset used for generation: **FalseQA**



Dataset used for generation: **Flan**



Dataset used for generation: **HelpSteer**

Thanks for Your Listening

Speaker: Miao Yuchun

Email: szmyc1@163.com