Google DeepMind

ICML International Conference On Machine Learning

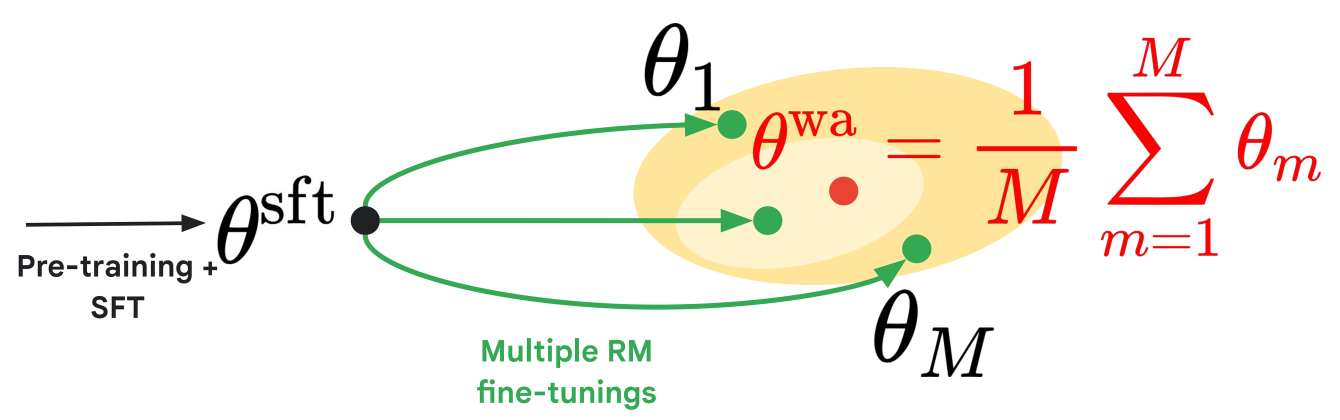
WARM: On the Benefits of Weight Averaged Reward Models

Alexandre Ramé, Nino Vieillard, Léonard Hussenot,
Robert Dadashi, Geoffrey Cideron, Olivier Bachem, Johan Ferret

Context and challenge

After pre-training and supervised fine-tuning, LLMs are aligned via reinforcement learning with human feedback (RLHF); the LLM policy optimizes a **reward model,** which is only an imperfect approximation of human preferences. This can lead to **reward hacking,** where increases in reward are not correlated with better/safer generations.

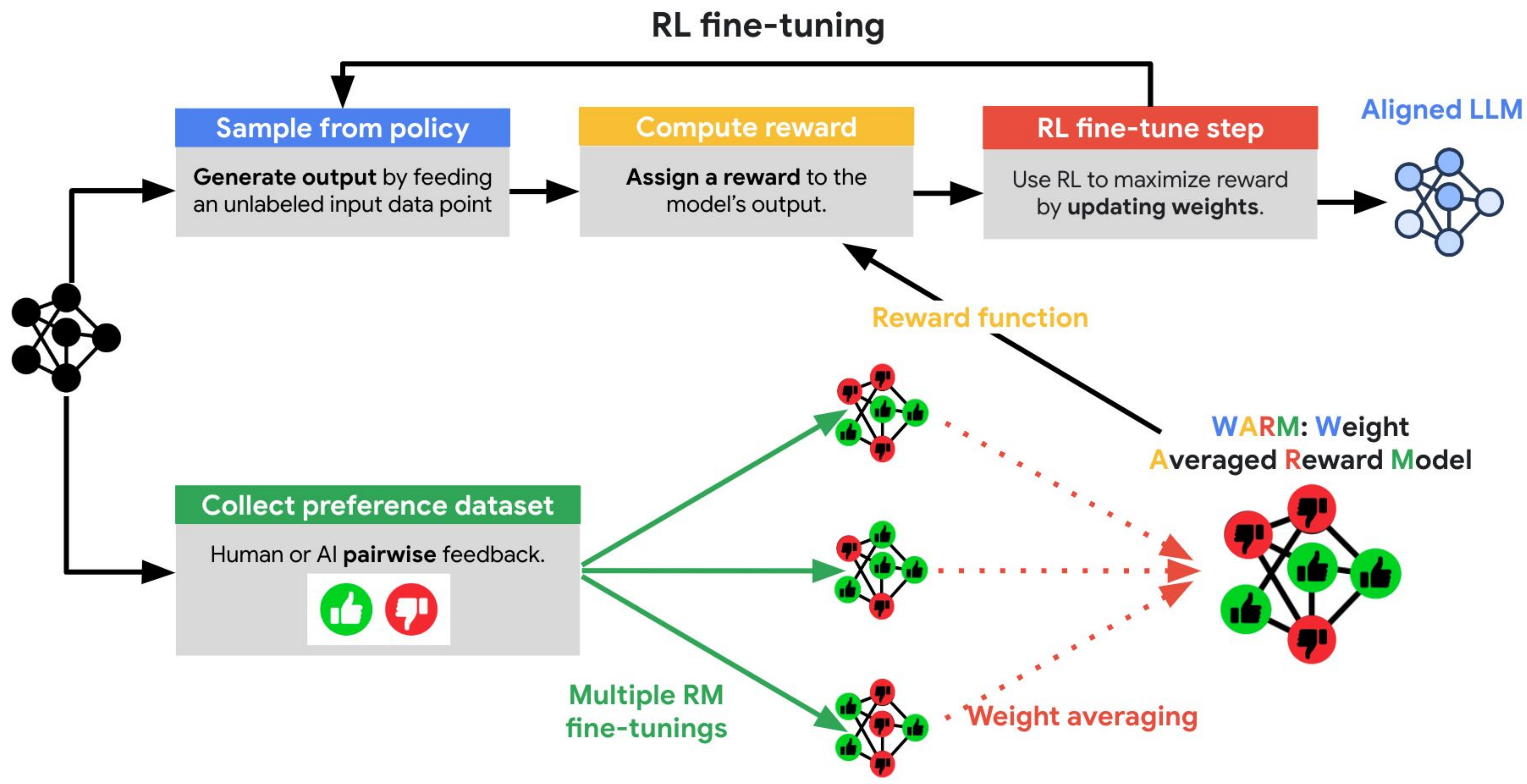
We improve reward modeling by (i) training M independent RMs from a shared pre-trained initialization and (ii) weight average them into WARM, (iii) finally used in RL.



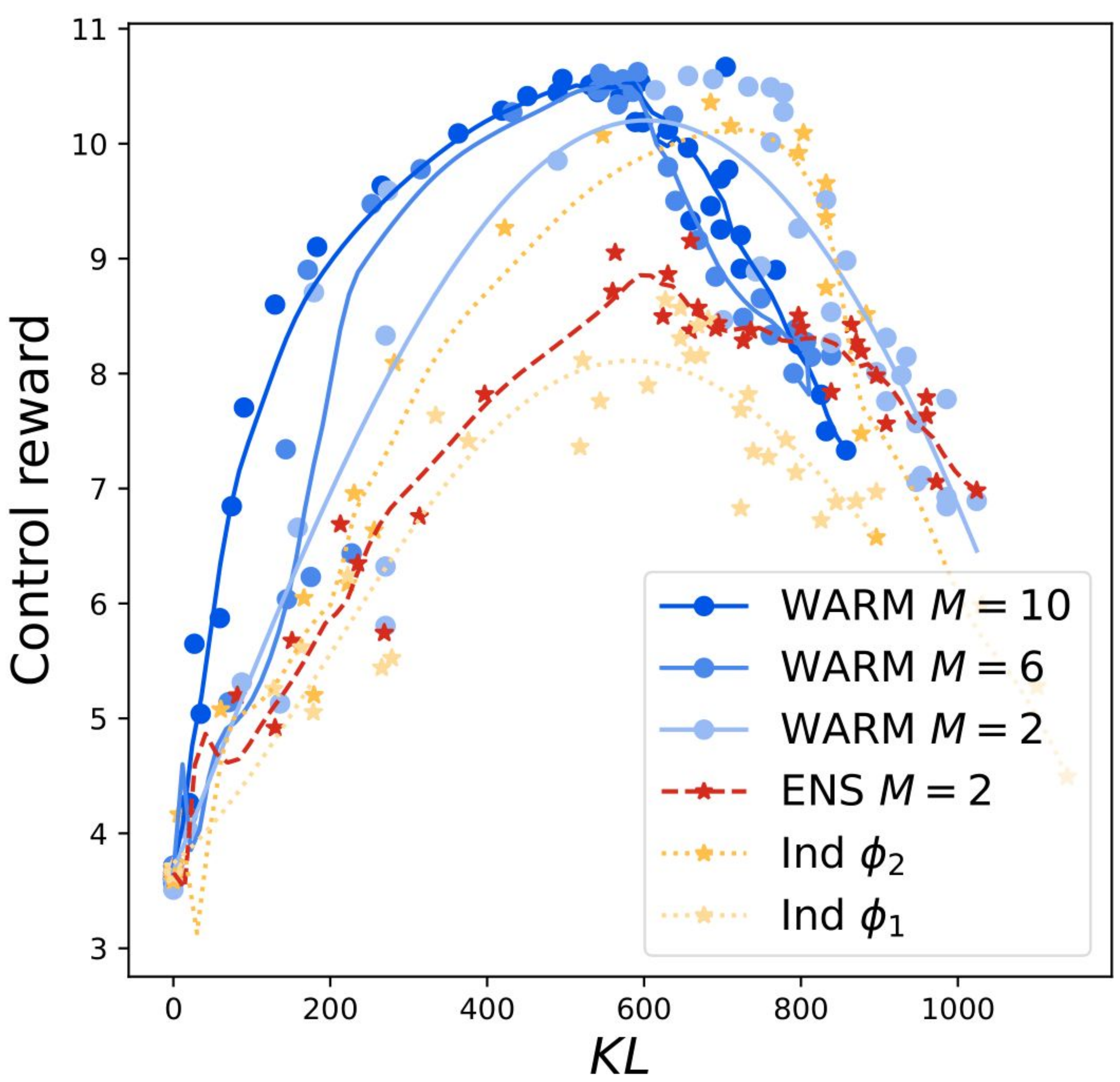
Thanks to linear mode connectivity, WARM benefits from:

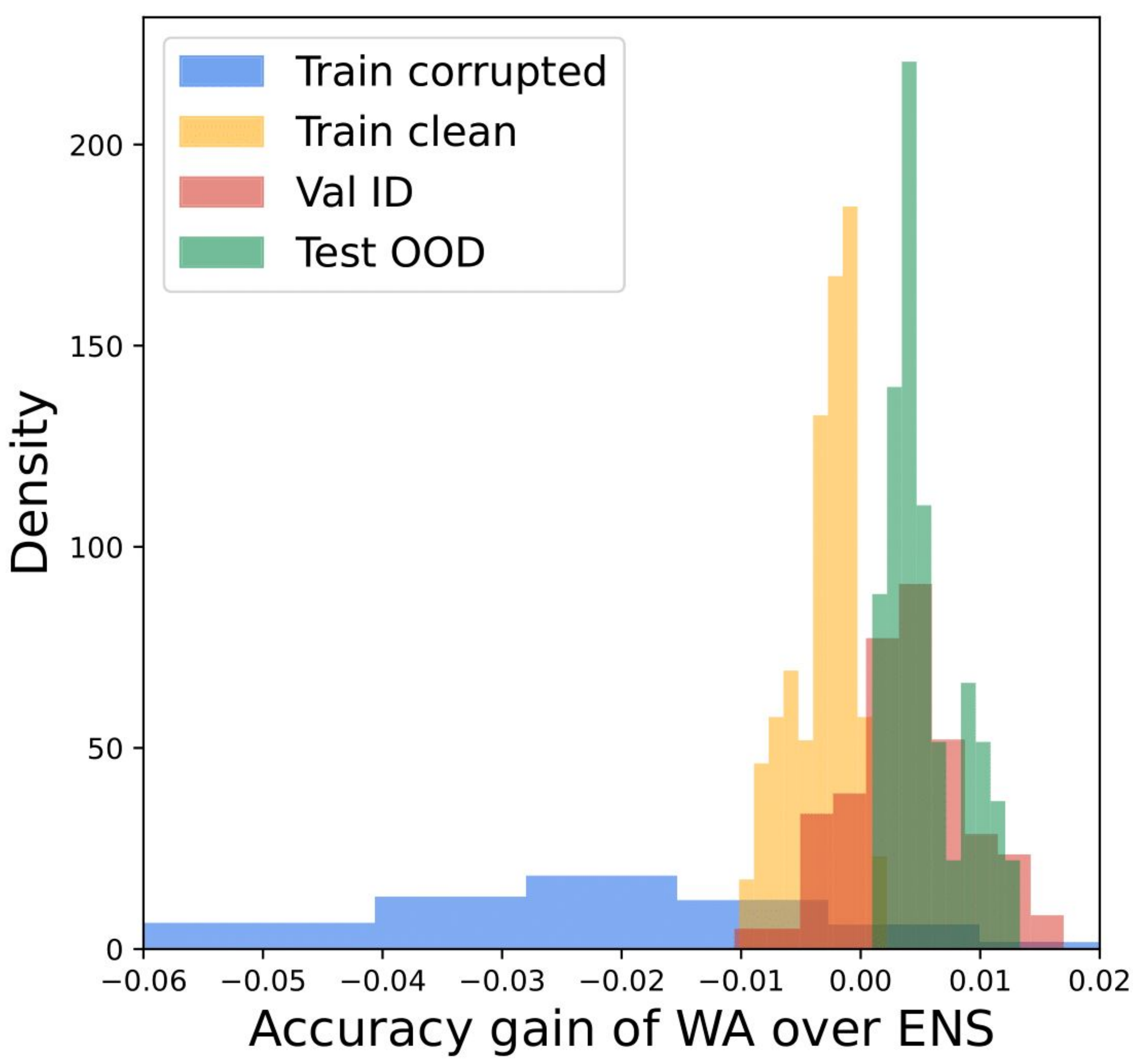
- **Efficiency**, removing the memory/inference overheads of (traditional) ensembling of the predictions of M models.
- **Robustness** to corruptions in preference labels, reducing memorization by enforcing invariance across runs.
- **Reliability** under distribution shifts, improving generalization by reducing variance.

Weight Averaged Reward Models (WARM)



WARM experiments on summarization





Mitigate reward hacking during RL

(1) by reducing memorization under label noise,

(2) and improving generalization under distribution shifts.

Setup
Data: TL;DR summarization
RL method: REINFORCE
Policy RLHFd: PaLM-XXS
Proxy reward: PaLM-XXS
Control reward: PaLM-XS

Follow-up work: Weight Averaged Rewarded Policies (WARP)

In WARP, we merge policies themselves (rather than reward models). The goal is to:

- maximize the reward model, to improve policy's alignment with human preferences,
- minimize the KL, to mitigate forgetting of general pre-trained knowledge.

We apply 3 variants of weight averaging at three distinct stages, **iteratively**.

- **Exponential moving average (EMA)** for dynamic anchor in the KL regularization.
- **Spherical linear interpolation (SLERP)** of task vectors of fine-tuned models.
- **Linearly interpolate towards the initialization (LITI)** to mitigate forgetting.

This strategy was used in Gemma 2!