Weight averaging for RLHF

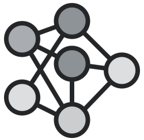
Alexandre Ramé

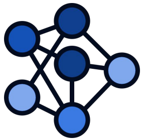
Google DeepMind

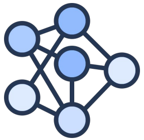
Google

What is model merging?

We consider 2 deep models, with different parameters θ, sharing the same non-linear architecture (with attention/relu/etc layers).





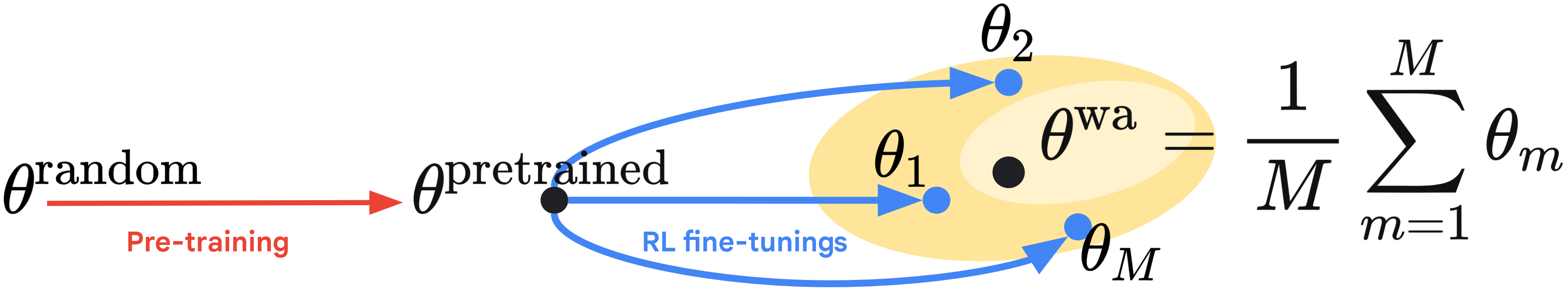


$$\theta^1 + \theta^2 = \frac{\theta^1 + \theta^2}{2}$$

We want to use them together; can we merge them?

Weight averaging? Really? Despite the non-linearities?

Model merging by weight averaging

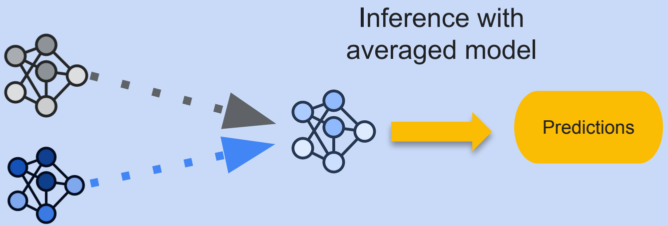
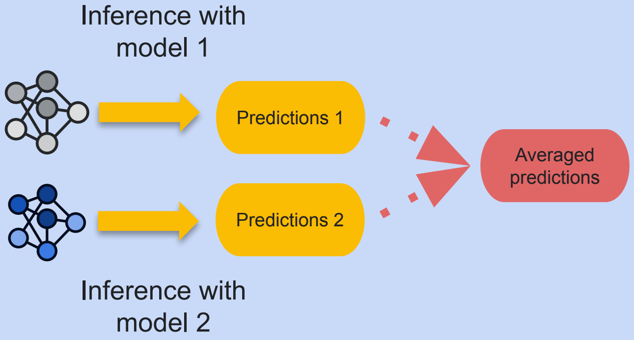


When fine-tuning from a shared pre-trained initialization,

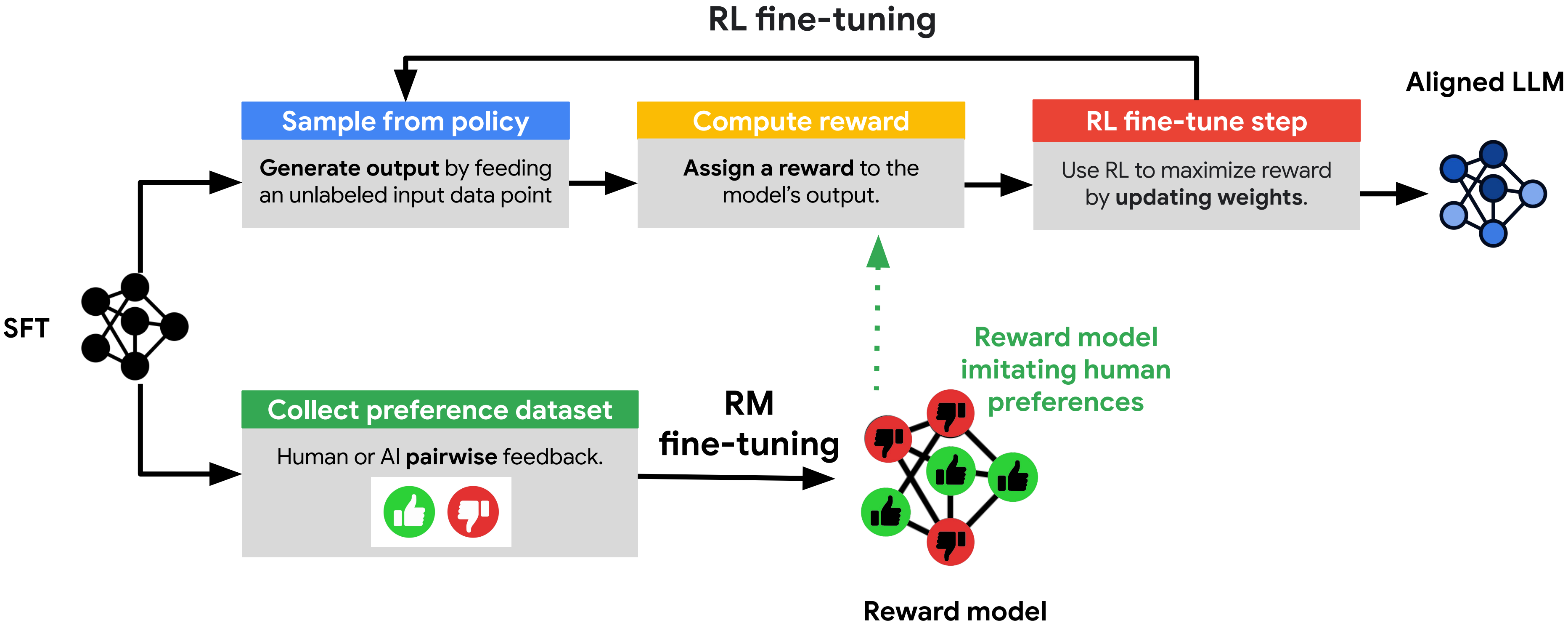
we can merge models (and their abilities) by weight averaging

"Linear Mode Connectivity and the Lottery Ticket Hypothesis" by Frankle *et al.*, ICML 2020.
"Model soups: averaging weights improves accuracy without increasing inference time" by Wortsman *et al.*, ICML 2022.
"Diverse Weight Averaging for Out-of-Distribution Generalization" by Ramé *et al.*, NeurIPS 2022.

Weight averaging as an efficient and improved ensembling strategy

Name	Weight averaging	Prediction averaging (traditional ensembling)
What		
Inference cost	1 single forward	2 forwards
Constraint	Weights fine-tuned from a shared pretrained init for a given architecture	No constraint
Reliability in OOD	Generalizes under distribution shifts thanks to variance reduction	Generalizes under distribution shifts thanks to variance reduction
Robustness to corruptions	Reduced memorization by removing run-specific features	Memorization of corrupted labels

RLHF in one slide

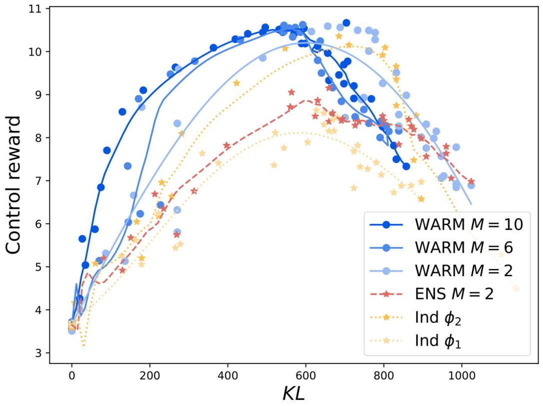


WARM: Weight Averaged Reward Models (ICML 2024)

The problem: reward hacking

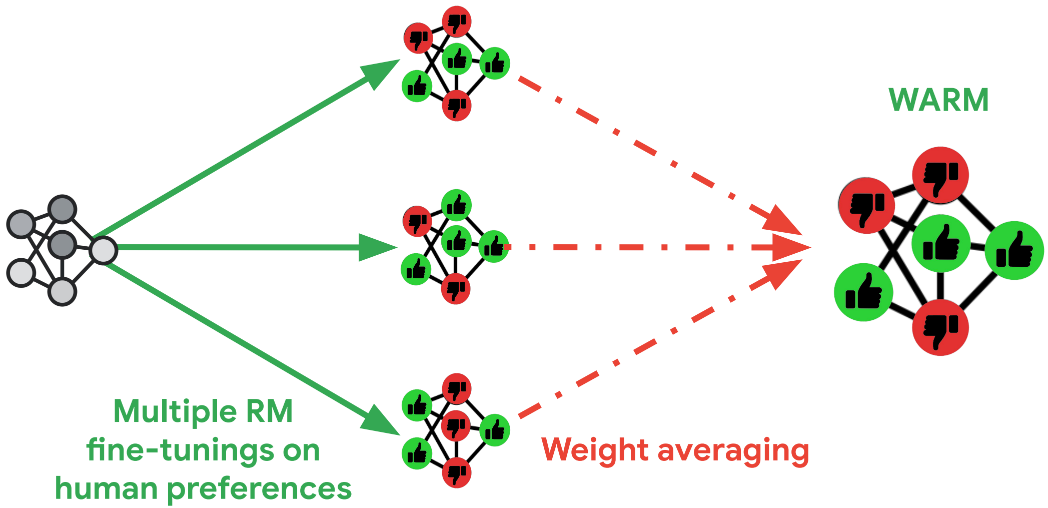
Misalignment as the policy exploits errors in the RM without really improving human preferences (because of label noise and distribution shifts).

Experiments: better when merging more RMs

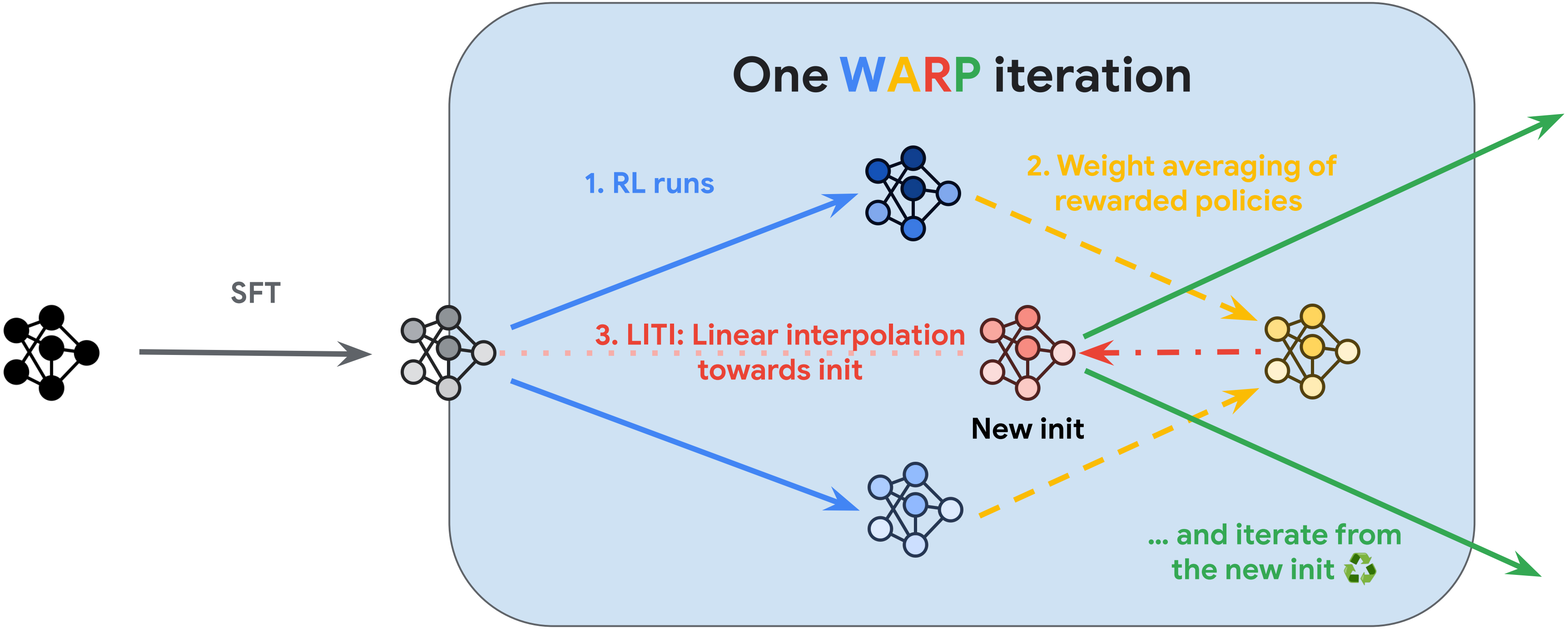


Our solution: merge reward models

Train several RMs, weight average them, and then run RLHF against the WARM.



WARP: Weight Averaged Rewarded Policies (arXiv 2024)



We use the merged policy as an advanced new initialization for subsequent WARP iterations.

Rewarded soups: towards Pareto-optimal alignment (NeurIPS 2023) and Conditioned Language Policy (arXiv since yesterday)

Goal: Maximizing a linear combination of rewards (for multi-objective RLHF).

Where the reward weightings λ are usually manually fixed before training

$$\theta = \operatorname{argmax} \boxed{\lambda_1}\boxed{R_1} + \boxed{\lambda_2}\boxed{R_2} + \cdots + \boxed{\lambda_M}\boxed{R_M}$$

e.g. quality factuality safety

$$\theta = \boxed{\lambda_1}\boxed{\theta_1} + \boxed{\lambda_2}\boxed{\theta_2} + \cdots + \boxed{\lambda_M}\boxed{\theta_M}$$

Our solution: Learn $\{\theta_i\}_{1\leqslant i\leqslant M}$ (one for each reward) and interpolate them for improved results and flexibility at deployment.

Thank you