Google DeepMind

WARM: Weight Averaged Reward Models.

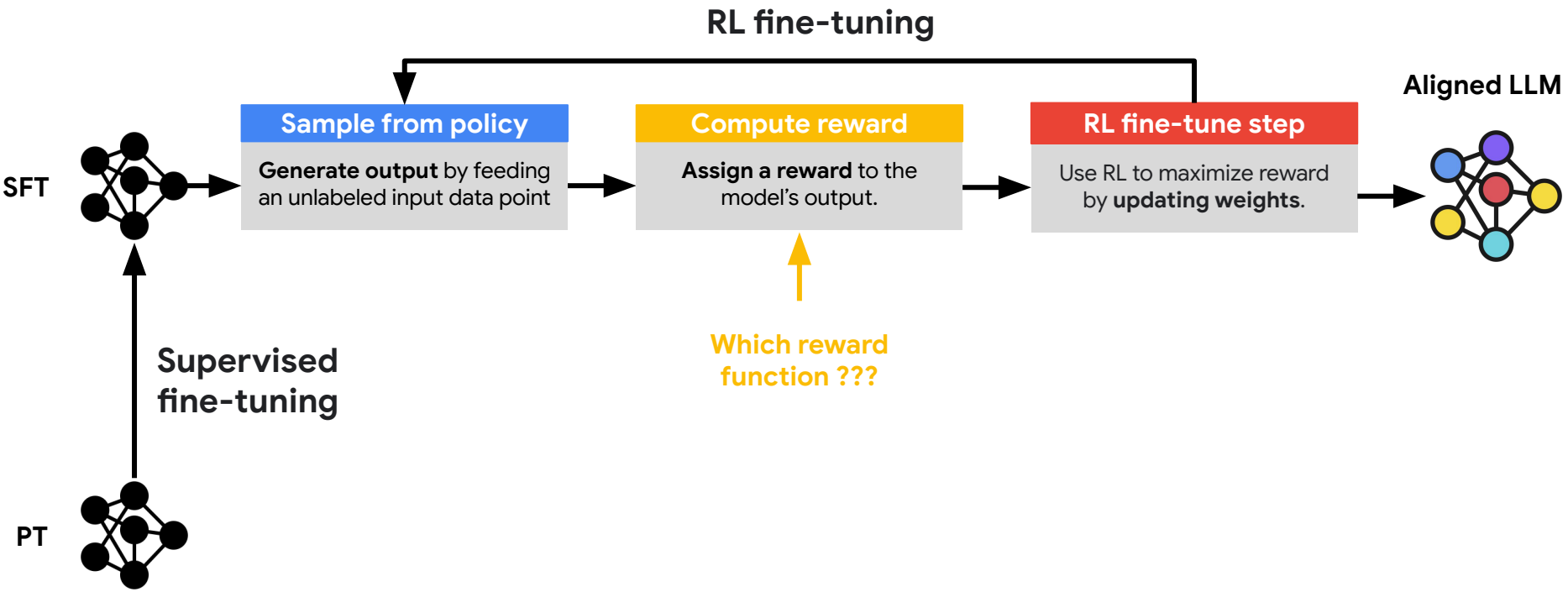
Alexandre Ramé: student researcher,
under the supervision of **Johan Ferret** and the **RL5X** team.

Idea summary: *go/warm-idea*
Paper draft: *go/warm-tex*
Short deck (this): *go/warm-gslides*
Long deck: *go/warm-gslides-internship*

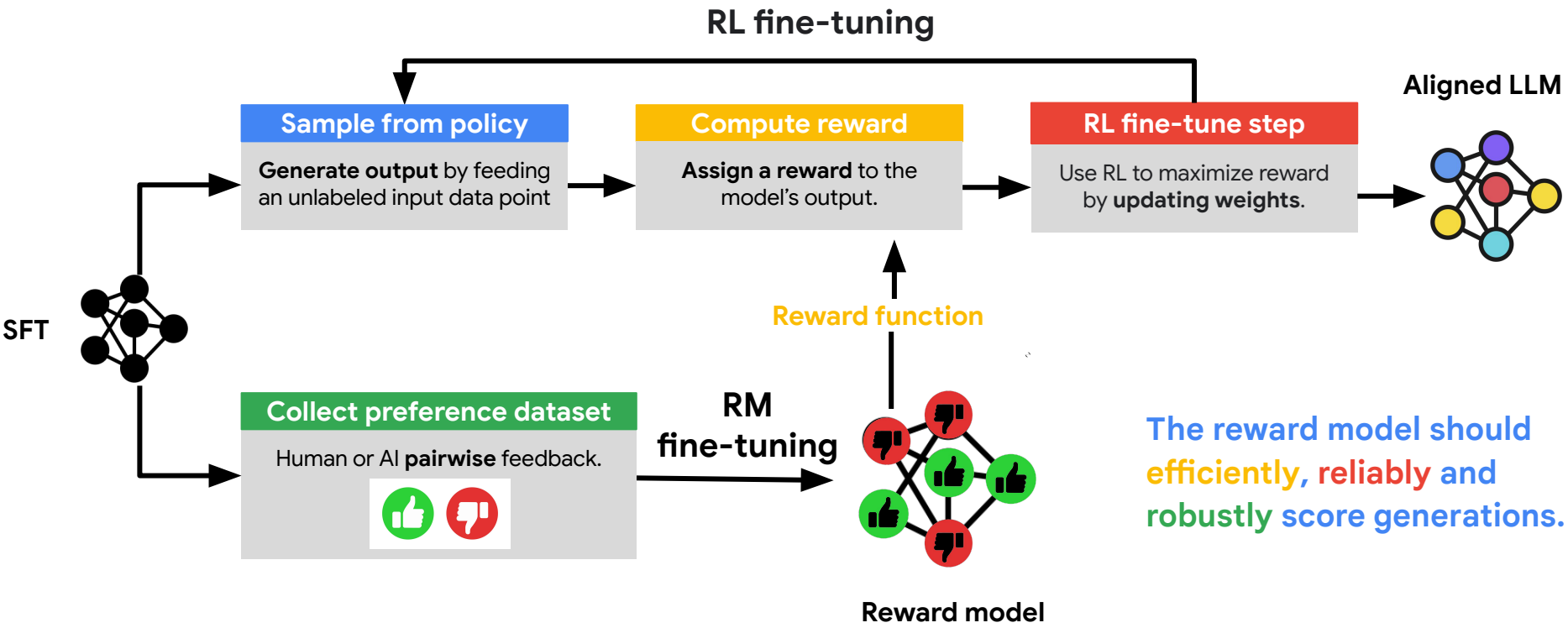
“

Explore the findings from the generalization literature (& my PhD) to the design of reward models for efficient, reliable, and robust RL alignment.

RL alignment of LLMs: reinforcement learning



RL alignment of LLMs: reinforcement learning from **pairwise** feedback



WARM policies are favored in pairwise comparisons

RL vs. SFT: **98.5 %** win rate*.

WARM RL vs. SFT: **99.8 %** win rate*.

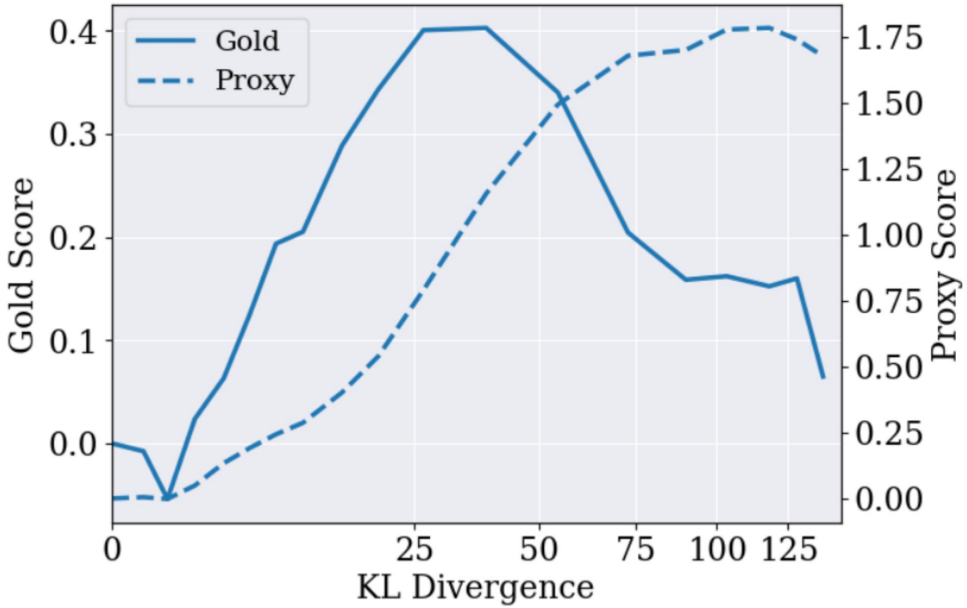
WARM RL vs. RL: **79.4%** win rate*.

*computed with a ULM 340B prompted as a preference labeler.

Reward overoptimization is a key challenge in alignment

The policy exploits reward misspecification to achieve high proxy rewards without improving gold human preferences.

- Key in Bard and Gemini efforts.
- Across all model scales.
- For every flavor of RL algorithms.

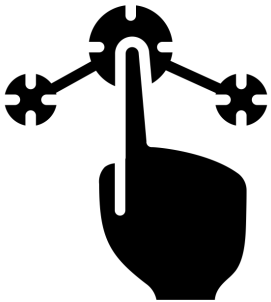


From “Reward model ensembles help mitigate overoptimization” by Coste *et al.*

Consequences of reward overoptimization

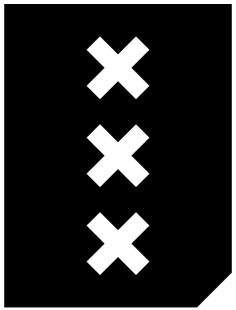
Complex checkpoint selection

Goodhart's law: "*When a measure becomes a target, it ceases to be a good measure*".



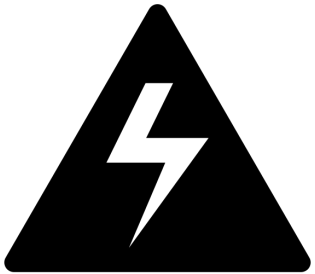
Bad performances

Incoherent linguistic outputs. Low diversity in predictions. Adversarial generations. Bad generalization to new prompts.

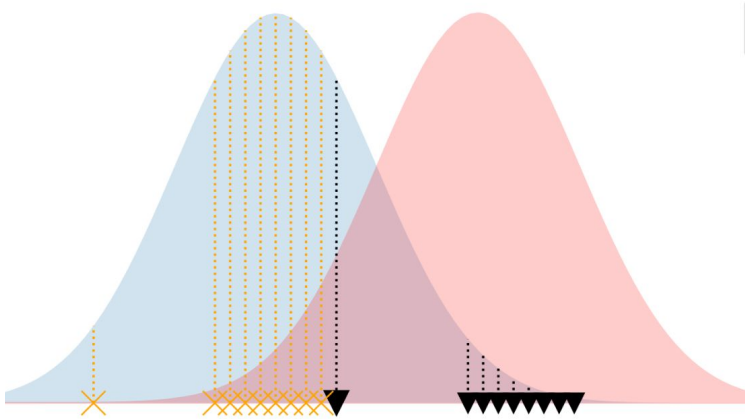


Bias and safety risks

Misalignment, thus unsafe AI, worsening social issues and misuses. Uncontrolled deployment in real-world applications.



Issue 1: distribution shifts in reward modeling



Offline preference dataset

The preference dataset is generated with a policy different from the one of interest.

Model drift

The policy changes during training, accentuating the distribution shifts.

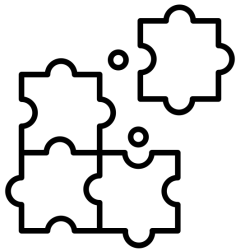
Issue 2: noisy and unreliable preference datasets

Labeler inconsistency

Errors caused by fatigue, imperfect (financial) incentives for non-rational human labelers. Or bad prompting for AIs.

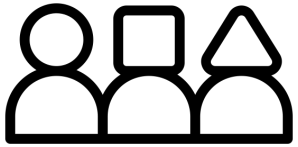
Complexity of the tasks

Especially when the AIs are more capable than the human labelers. Scalable oversight challenge.



Heterogeneity of opinions

Humans have different opinions on some subject (aesthetics, politics, etc), and multi objectives (harmlessness to engagement).



Overall, 65% inter-agreement across labelers.

The standard strategies against reward overoptimization are:

1. Strongly KL-regularized RL

Explicitly force RL-policies to remain closer to their SFT initialization (small KL).

Risk of underfitting.

2. Continual learning of the RM

The RM is continually updated with new active data, collected on-policy.

Not practical and expensive.

3. Prediction ensembling of RMs

The predicted rewards from multiple RMs are averaged, and then used in the RL.

Inefficient. Do not tackle corruption.

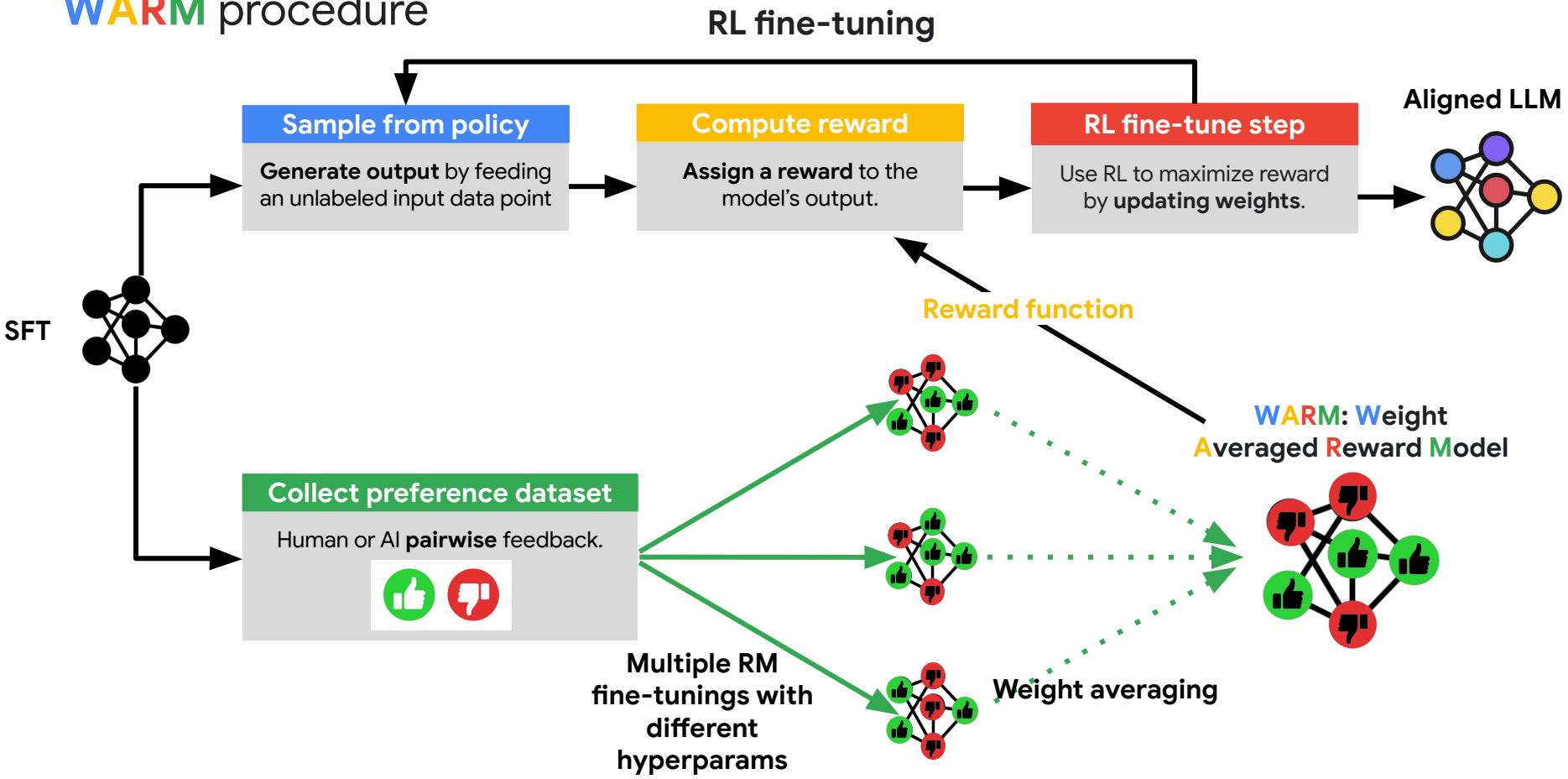
Such **strategies** only mitigate the issue to some extent.

“

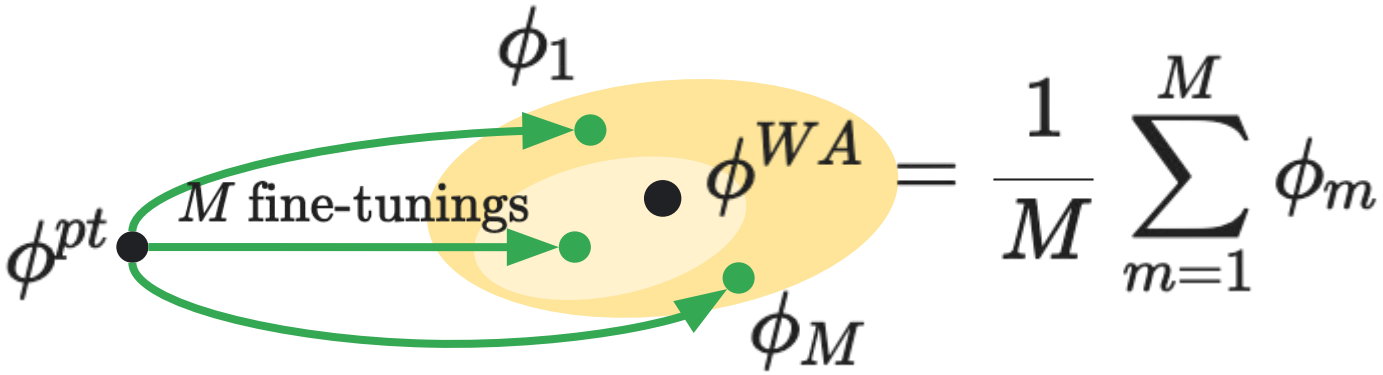
Our strategy: efficient ensembling of M reward models.

More efficient: zero overhead.
More reliable; less hacking.
More robust; less inconsistent.

WARM procedure



Weight interpolation relies on linear mode connectivity



When fine-tuned from a shared pre-trained initialization, weights remain linearly connected and thus can be interpolated despite the non-linearities in the architecture.

- "What is being transferred in transfer learning?" by Neyshabur *et al.*, NeurIPS 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.

Sources of diversity

Different data orders

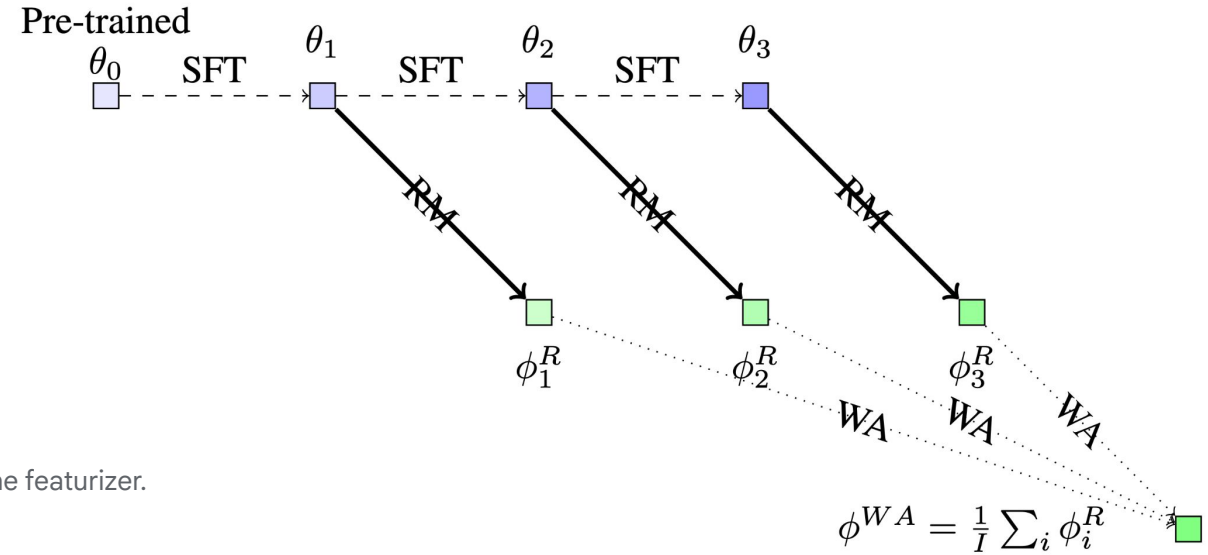
Stochasticity in the the batch ordering.

Different SFT initializations? *Baklava*

Different initializations collected along a single SFT.

Different hyperparameters

- Learning rates.
- Dropout proba.



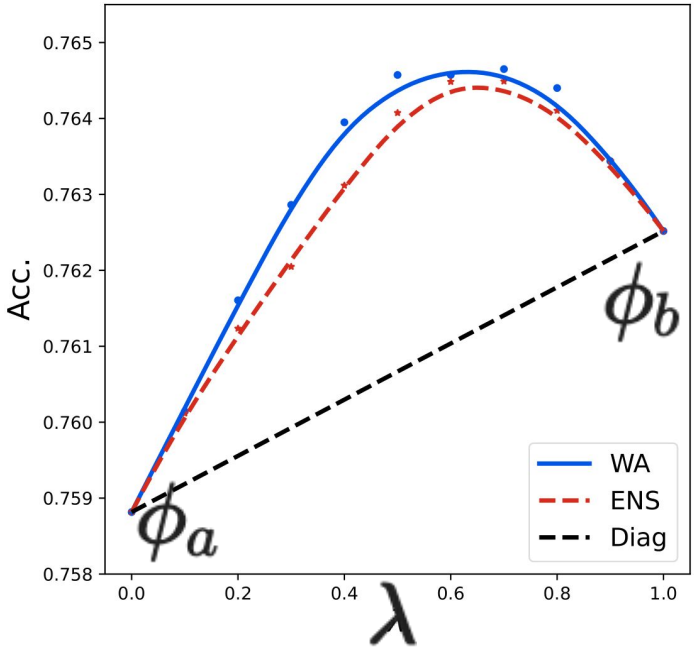
Diversity limitations:

- Same architecture and pre-training of the featurizer.
- Linear probing of the classifier.

As a **first** order analysis, weight averaging (WA) approximates prediction ensembling (ENS)

Considering reward model r parameterized by ϕ_a and ϕ_b, we slide the interpolating coefficient λ between 0 and 1.

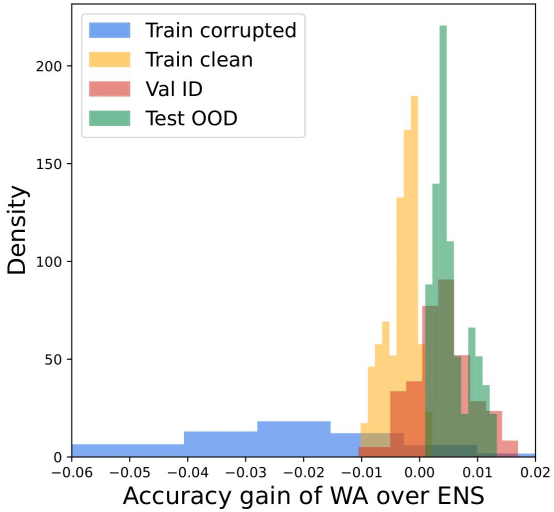
$$Acc(r_{(1-\lambda)\cdot\phi_a+\lambda\cdot\phi_b}) \approx Acc((1-\lambda)\times r_{\phi_a} + \lambda \times r_{\phi_b})$$

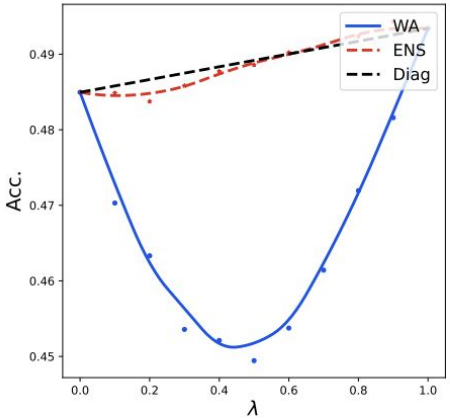


WA performs similarly to the more costly ENS.

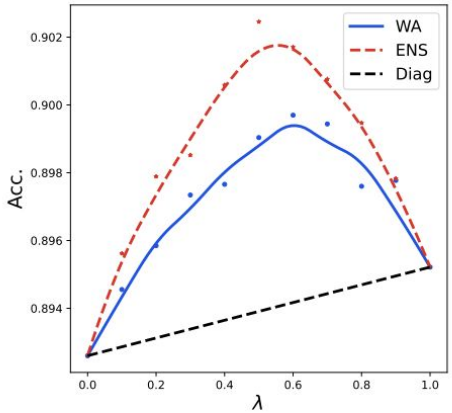
As a **second** order analysis,
WA generalizes while ENS memorizes.

The further away from the train distribution, the better the WA vs. ENS.

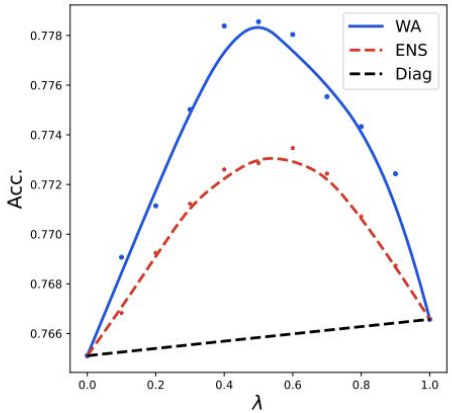




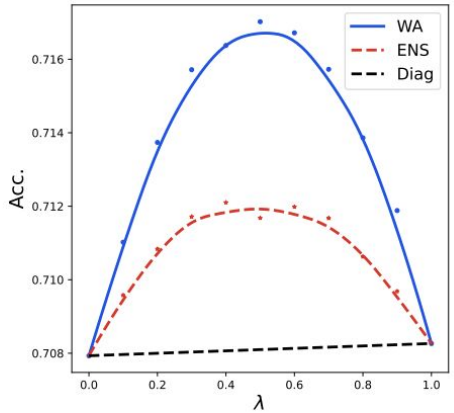
(a) Train (corrupt).



(b) Train (clean).



(c) Validation (ID).



(d) Test (OOD).

“

Theoretical insights:

WA removes run-specific features entailing memorization,

and favors run-invariant features entailing generalization,

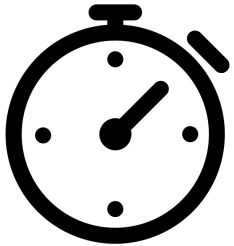
which facilitates smoothness and learnability of the reward.

(see paper go/warm-tex for more details)

What WARM brings is:

Efficiency

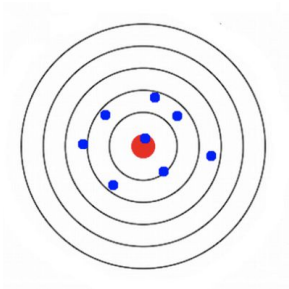
WA deletes the inference overhead, making M large possible.



Better success detectors at zero inference cost.

Reliability under distribution shifts

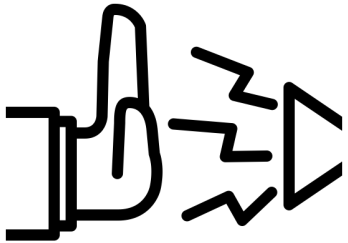
Variance reduction by combining M RMs.



Lead to appropriate policies by limiting reward hacking.

Robustness to labeling inconsistencies

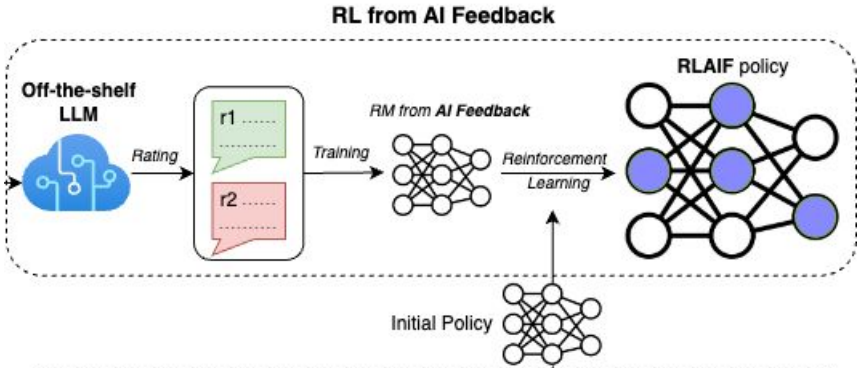
Resilience to noise/corruption in preference labels.



Provide robust trainable signals for the policy during RL.

Experimental setup

Task: TLDR summarization.
Preference: from a ULM-340B, following the "*RL from AI Feedback*" paper.
Summaries in train: generated by GPT-3 variants by OpenAI.
Summaries to evaluate (Agent LLM): ULM-8B
Trained reward model: ULM-1B

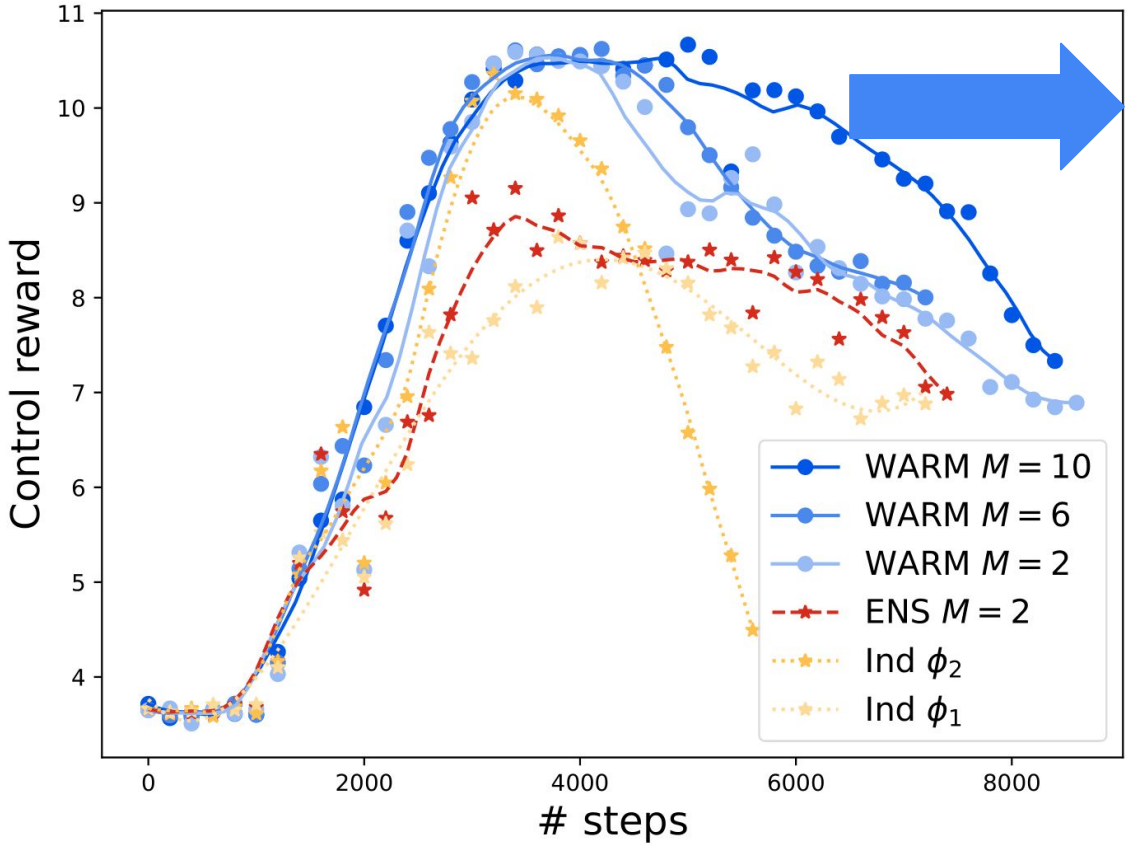


To simulate **pairwise** human preferences, we prompt a ULM-340B as a preference labeler.

Reward overoptimization

Increasing M (the number of weights in the average) delays the collapse of the control reward at the end of the training.

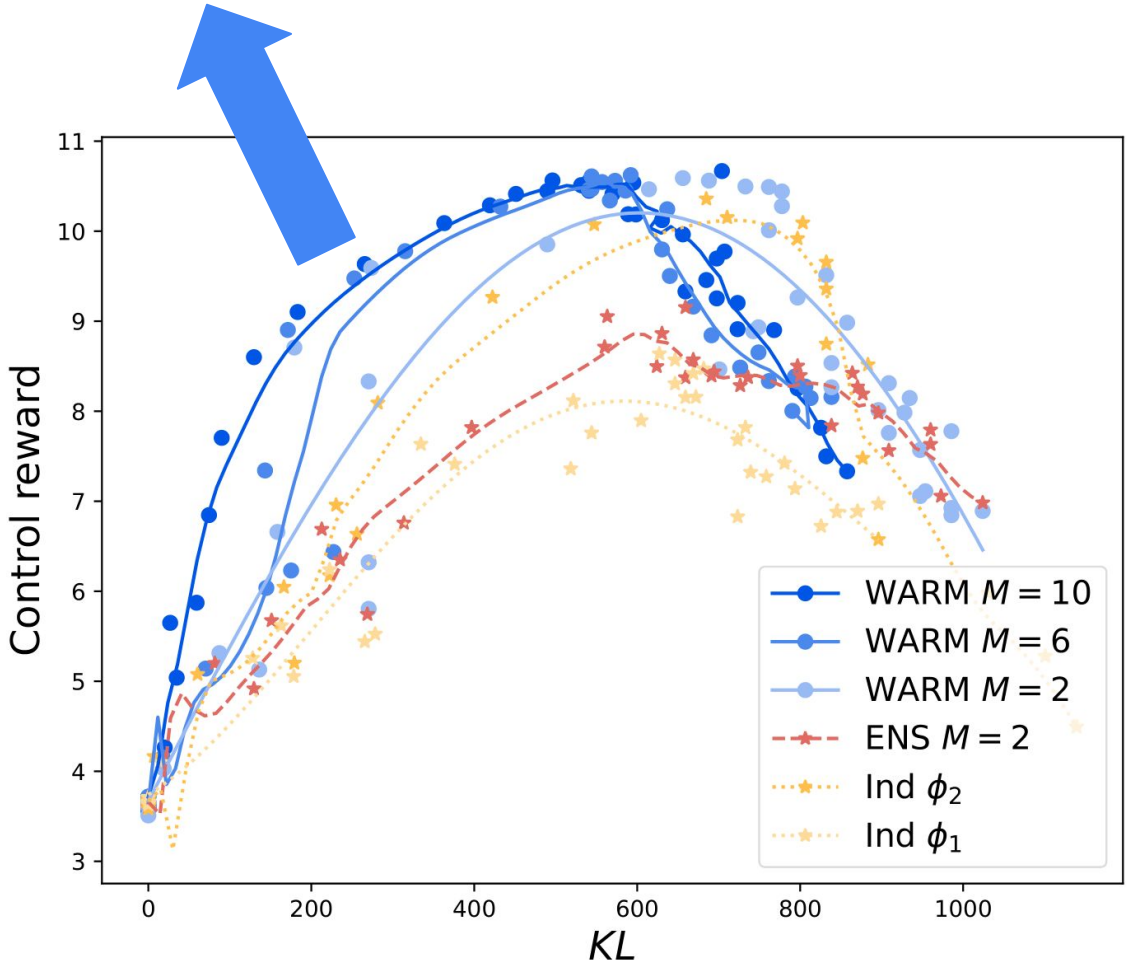
WARM reduces reward overoptimization and facilitates checkpoints selection.



Pareto optimality

Increasing M (the number of weights in the average) pushes the front of solutions to the top left.

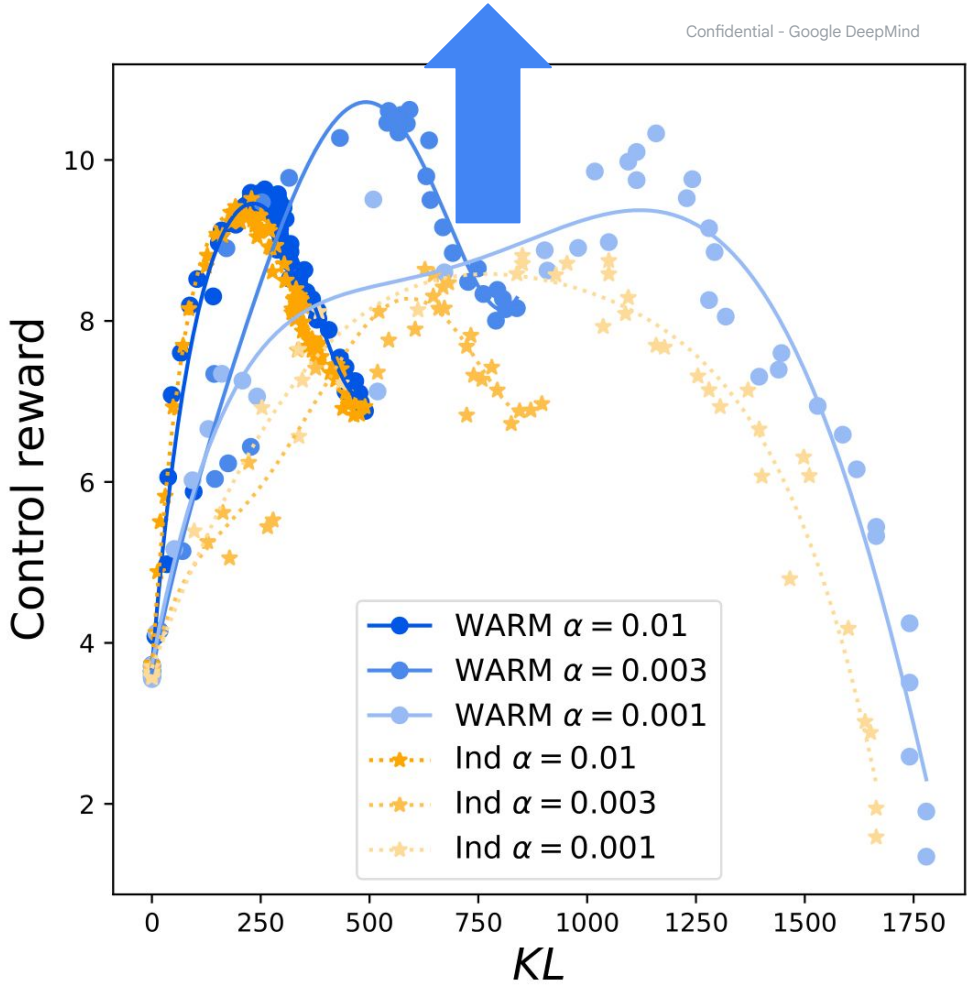
WARM improves the Pareto-optimal front of policies.



Impact of WARM for different regularization strengths

The α hyperparameter controls the KL regularization.

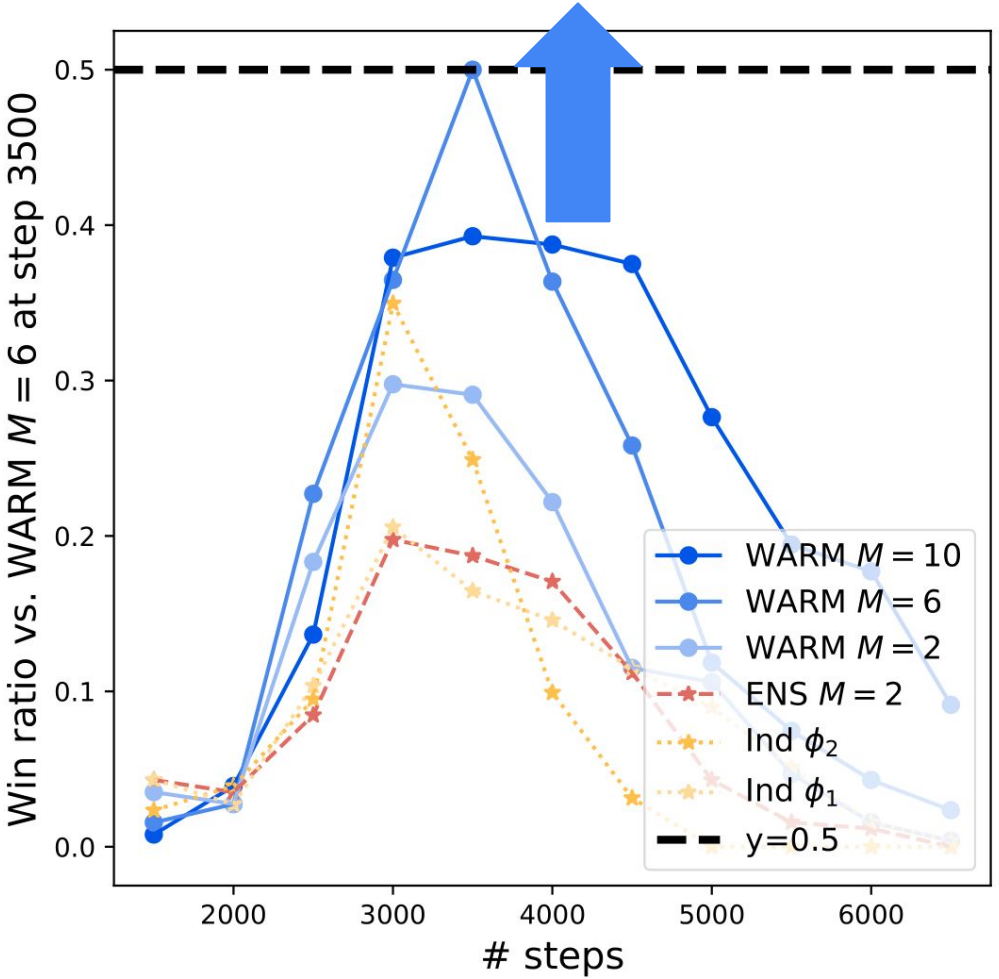
The policies trained with WARM lead to a better Pareto-front of solutions when considering a wide range of values for α.



Pairwise comparisons

All policies are compared against a reference trained with WARM M=6 for 3500 training steps (which is the best number of steps according to the control reward).

The policies trained with WARM are favored* in pairwise comparisons.



*computed with a ULM 340B prompted as a preference labeler.

Key takeaways

Simple and **efficient success detectors**: better accuracies at no inference cost.

Reliable reward: prevents **reward hacking** caused by distribution shifts.

Provide **robust & learnable signals**: reduces memorization in the noisy preference datasets.

Weight averaging everywhere

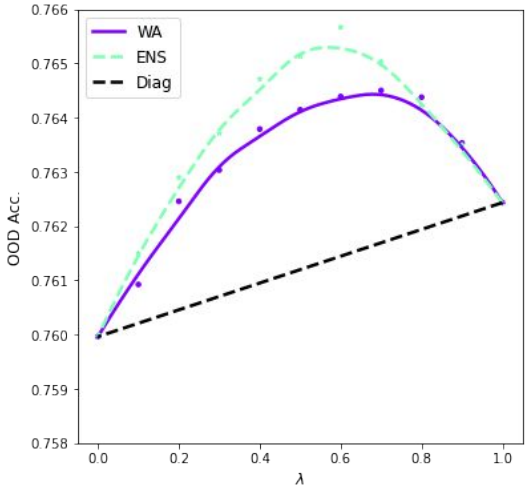
Thank you!

Alexandre Ramé
Student researcher

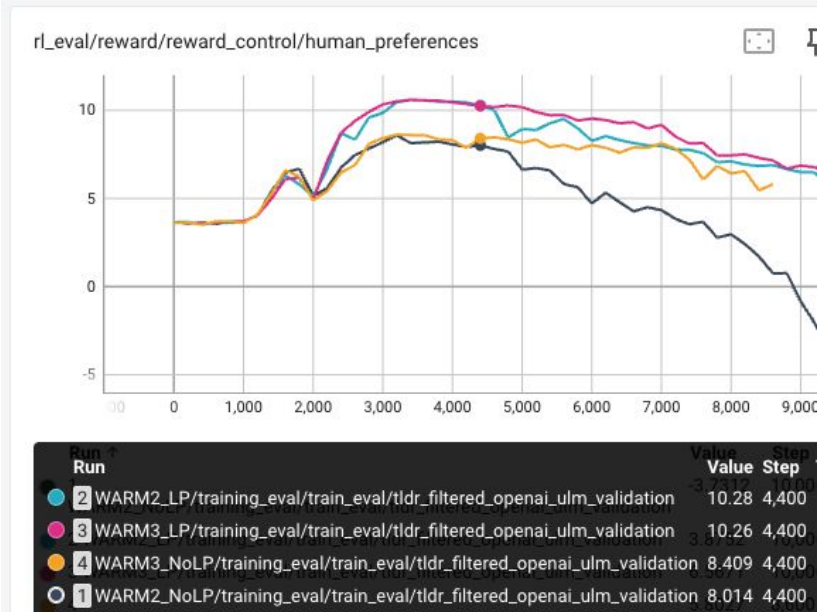
Supervised by Johan Ferret and Nino Vieillard, and helped by the RL5X team.

Google DeepMind

First learn the linear classifier before end-to-end fine-tuning



WA consistently matches or beats ensembling, except when no linear probing.



Following “Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution” by Kumar *et al.*, ICLR 2022.