WARM: On the Benefits of Weight Averaged Reward Models

Alexandre Ramé [1] **Nino Vieillard** [1] **Léonard Hussenot** [1] **Robert Dadashi** [1] **Geoffrey Cideron** [1] **Olivier Bachem** [1] **Johan Ferret** [1]

Abstract

Aligning large language models (LLMs) with human preferences through reinforcement learning (RLHF) can lead to reward hacking, where LLMs exploit failures in the reward model (RM) to achieve seemingly high rewards without meeting the underlying objectives. We identify two primary challenges when designing RMs to mitigate reward hacking: distribution shifts during the RL process and inconsistencies in human preferences. As a solution, we propose Weight Averaged Reward Models (*WARM*), first fine-tuning multiple RMs, then averaging them in the weight space. This strategy follows the observation that fine-tuned weights remain linearly connected when sharing the same pre-training. By averaging weights, *WARM* improves *efficiency* compared to the traditional ensembling of predictions, while improving *reliability* under distribution shifts and *robustness* to preference inconsistencies. Our experiments on summarization tasks, using best-of-N and RL methods, shows that *WARM* improves the quality and alignment of LLM predictions; for example, a policy RL fine-tuned with *WARM* has a 79.4% win rate against a policy RL fine-tuned with a single RM.

1. Introduction

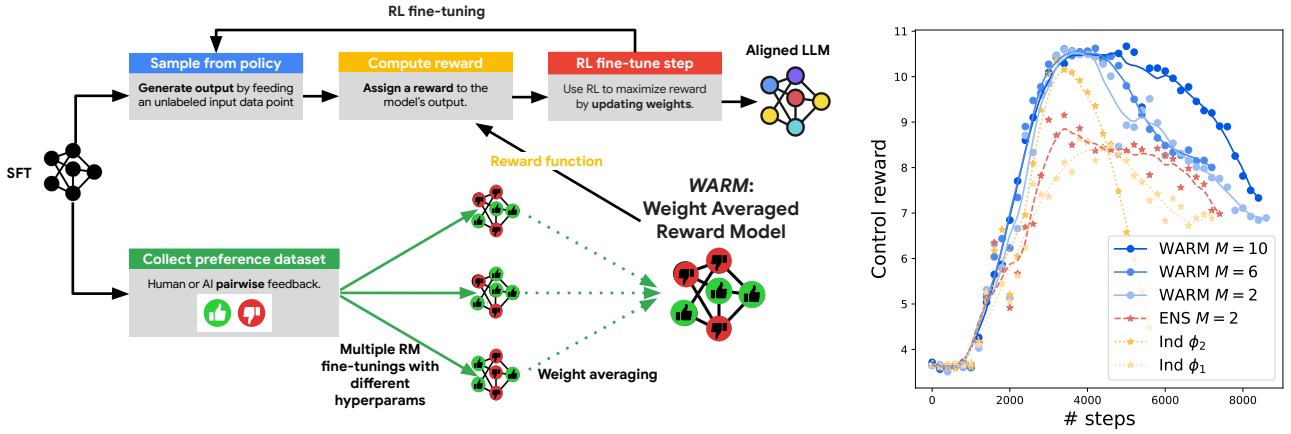
Reward modeling. Conversational assistants such as Gemini (Gemini Team, 2023) or GPT-4 (OpenAI, 2023) have revolutionized the AI community and beyond. These LLMs are capable of completing novel and intricate tasks, including mathematics, coding, and tool use (Bubeck et al., 2023). These advancements are underpinned by a systematic three stage training procedure: pre-training by next token prediction (Radford et al., 2018; Devlin et al., 2019), supervised fine-tuning (SFT) to learn to follow instructions (Wei et al., 2022a), and ultimately, reinforcement learning (RL) to maximize a reward encapsulating the desired behaviors (Ouyang et al., 2022). However, defining such rewards for real-world tasks is non-trivial (McKinney et al., 2023). In reinforcement learning from human feedback (RLHF, Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020), rewards are reward models (RMs), trained on binary preference datasets to emulate human judgment. The enhancement of LLM capabilities from RL is strongly tied to the quality of the RMs (Touvron et al., 2023).

Reward hacking. Particularly insidious in RLHF (Gao et al., 2023; Casper et al., 2023) is the *reward hacking* (a.k.a. reward overoptimization, Amodei et al., 2016; Clark & Amodei, 2016; Askell et al., 2021) arising from *reward misspecification* (Pan et al., 2022) between the proxy RM and actual human preferences. While optimizing the proxy RM initially provides improvements, in later stages the policy (i.e., the LLM being trained) actually exploits loopholes in the proxy RM, as illustrated in Figure 1(b). This reward hacking phenomenon poses numerous issues. First, it degrades performances, manifesting for example as linguistically flawed (Lewis et al., 2017) or unnecessarily verbose (Singhal et al., 2023) outputs. Second, it complicates checkpoint selection, echoing Goodhart's Law (Strathern, 1997): "when a measure becomes a target, it ceases to be a good measure". Third, misalignment (Taylor et al., 2016; Ngo et al., 2022) can escalate into safety risks (Amodei et al., 2016; Hendrycks & Mazeika, 2022), especially given the rapid integration of LLMs in critical decision-making. Such concerns underscore the need to mitigate reward hacking to ensure beneficial and safe deployment of LLMs.

Challenges. Two primary challenges underlie reward hacking. The first major issue are *the distribution shifts* encountered by the RM. Indeed, the generations from the policy might deviate substantially from those in the offline preference dataset, posing an out-of-distribution (OOD) challenge. Moreover, those distribution shifts are accentuated by the policy drift during the RL procedure: the policy moves away from its SFT initialization, continually altering the distribution of predictions the RM needs to interpret *reliably*. Second, *preferences are inconsistent*: the binary labels in the preference dataset are noisy. Indeed, human labelers often rely on simpler criteria (length, bullet points, polite-

[1]Google DeepMind. Correspondence to: Alexandre Ramé <alexandrerame@google.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.



(a) *WARM* procedure with $M = 3$.

(b) *WARM* mitigates reward hacking.

Figure 1: Figure 1(a) illustrates the alignment process with *WARM* for RLHF (Christiano et al., 2017). From a SFT-ed LLM, we RL fine-tune to optimize a proxy reward model (RM). The innovation of *WARM* lies in the design of the proxy RM, which is the weight average (WA) of M individual RMs, each fine-tuned from a shared pre-trained LLM on the same preference dataset, but with slight differences such as diverse hyperparameters. This WA approach is *efficient*, while enhancing the *reliability* under distribution shifts and *robustness* under inconsistent preferences. Figure 1(b) shows the impact during RL alignment. The control reward (detailed in Section 5) initially increases but eventually deteriorates, a phenomenon called reward hacking. Yet, when *WARM* serves as the proxy RM, increasing M (the number of averaged RMs) significantly improves absolute results while delaying the collapse, as indicated by the control rewards maintaining higher values for longer during training. Same plot with KL as the x-axis in Figure 4(a) and with label corruption in Figure 17.

ness) over more nuanced indicators. Moreover, errors can be exacerbated for complex tasks requiring specific expertise (Bowman et al., 2022), and because of the multi-objective nature of alignment (Ramé et al., 2023) requiring handling the heterogeneity of human opinions. Overall, this results in a low inter-labeler agreement (72.6% for InstructGPT (Ouyang et al., 2022)), altering the *robustness* of the RM.

Goal and ensembling baseline. RMs should ideally satisfy three key properties: guiding RL *efficiently*, *reliably* scoring generations despite the distribution shifts, and providing *robust* signals amidst label noise. To address these challenges, the seminal work on RLHF from Christiano et al. (2017) and more recent works (Eisenstein et al., 2023; Coste et al., 2023) leveraged *prediction ensembling* (ENS, Lakshminarayanan et al., 2017), averaging the rewards from multiple RMs. ENS improves the reward *reliability* and mitigates hacking. Yet, ENS suffers from memory and inference overheads reducing *efficiency*; we will also show that ENS fails to improve *robustness* to preference inconsistencies.

WARM. In this paper, we propose weight averaged reward models (*WARM*), a simple, *efficient* and scalable strategy to obtain a *reliable* and *robust* RM by combining multiple RMs, following the model soups (Wortsman et al., 2022a) strategy first introduced for image classification. Starting from a shared pre-trained LLM, we launch multiple RM fine-tunings: in practice, the different runs have different hyperparameters (as in grid search), and see the preference data in different orders, thus leading to diverse RMs. A key contribution is how the different RMs are merged: by *linear interpolation in the weight space*. This follows the findings from the linear mode connectivity (LMC, Frankle et al., 2020; Neyshabur et al., 2020) and weight averaging (WA, Izmailov et al., 2018; Matena & Raffel, 2022; Wortsman et al., 2022a; Ramé et al., 2022b) literature: when the different fine-tunings share the same pre-training, the fine-tuned weights can be linearly interpolated despite the non-linearities in the architecture.

On the benefits of *WARM*. Firstly, *WARM* stands out for its *efficiency* and practicality. By requiring a single model at inference time, it provides a scalable approximation to the traditional, costlier ensembling of predictions, without its memory and inference burdens. Secondly, *WARM* improves *reliability*, inheriting the well-documented generalization abilities of WA under distribution shifts (Cha et al., 2021; Ramé et al., 2022b; 2023). Lastly, *WARM* improves *robustness* to label corruption. We show that WA selects the invariant predictive mechanisms (Muandet et al., 2013; Arjovsky et al., 2019) across different runs, thus naturally diminishing the memorization of corrupted samples, occurring in each run in different ways. In contrast, ENS memorizes the

corrupted samples. These multifaceted benefits of *WARM* are explored in Section 4. We summarize our contributions as follows.

1. *Innovation in reward modeling*. We introduce *WARM*, the first instance of weight averaging for reward modeling. This novel strategy *efficiently* mitigates reward hacking, improves *reliability* under distribution shifts and *robustness* to label corruption.

2. *Theoretical and empirical insights into weight averaging*. We validate linear mode connectivity for reward models trained on binary preference datasets. We also reveal a key difference between weight and prediction averaging; weight averaging only maintains the invariant predictive mechanisms across runs, thereby diminishing memorization and enhancing the focus on generalizable features.

Our experiments on summarization tasks in Section 5 confirm that *WARM* improves performance, either when used as the reward selector in best-of-N, or as the proxy RM in RL. *WARM* mitigates reward hacking, and thus provides better downstream policies; specifically, it leads to a win rate of 79.4% (according to the preference oracle metric) against a policy trained with a standard RM.

2. Context and Challenges

2.1. Context

LLMs. We consider an LLM f_θ of a fixed non-linear architecture parameterized by θ, usually a Transformer with attention layers (Vaswani et al., 2017). It defines a policy by mapping prompt inputs x to $f_\theta(x)$. Following the foundation model paradigm (Bommasani et al., 2021) and the success of transfer learning (Oquab et al., 2014), the weights θ are first pre-trained (Radford et al., 2018) on the vast amount of web data into θ^{pt}, before supervised fine-tuning (SFT, Wei et al., 2022a) to learn to follow instructions into θ^{sft}. However, the high cost and limited scope of instruction data (i.e., prompts and responses) can create a misalignment (Amodei et al., 2016; Taylor et al., 2016; Ngo et al., 2022) between the LLM and its intended application. Reinforcement learning (RL) as a third step in the training process of LLMs was shown to help alignment of LLMs with the intended usage (Ouyang et al., 2022).

RMs. A notable aspect of RL is the absence of supervised samples to be imitated by the policy; instead, the focus shifts to maximizing the reward of generated samples, that should measure their quality. The challenge is that the oracle reward, perfectly encapsulating the desired behaviors, is not given by the environment. The key innovation from RLHF (Christiano et al., 2017) is that this reward is the output of a reward model (RM), trained in a supervised way to predict and thus reflect human preferences. Specifically, an RM is an LLM r_ϕ parameterized by ϕ, predicting a single scalar as the reward $r_\phi(x, y)$ for a prompt x and generation y. The weights ϕ are usually initialized from $\left(\theta^{sft}, \omega\right)$, where the final linear layer ω is added on top of the extracted features from the SFT model θ^{sft}. Then ϕ is trained on a preference dataset $\mathcal{D}_{train} = \{x_d, y_d^+, y_d^-\}_{d=1}^D$ where the generation y_d^+ has been preferred over y_d^- to continue x_d. Usually human labelers evaluate those generations, but recent works on RLAIF (Bai et al., 2022b; Lee et al., 2023) showed that similar performances can be obtained by prompting an LLM for AI feedback. Following the assumption from Bradley & Terry (1952) about the distribution of preferences, and by framing the problem as binary classification, the maximum likelihood principle motivates learning ϕ by minimizing the following negative log-likelihood loss (where σ is the logistic function): $\mathcal{L}_R(r_\phi, \mathcal{D}_{train}) = -\mathbb{E}_{(x,y^+,y^-)\in\mathcal{D}_{train}}[\log \sigma(r_\phi(x, y^+) - r_\phi(x, y^-))]$.

Alignment. With this RM, the literature suggests applying any kind of RL algorithm (usually REINFORCE (Williams, 1992) or PPO (Schulman et al., 2017)) to fine-tuned θ^{sft} into θ^{rl}. A training-free alternative is best-of-N (BoN) sampling, which returns the generation that has the highest reward among N generations from θ^{sft}. Both methods aim to align the policy with human preferences. Yet, the *reward misspecification* (Pan et al., 2022) between the proxy RM and the true human preferences can lead to *reward hacking* (Amodei et al., 2016; Clark & Amodei, 2016; Askell et al., 2021; Skalse et al., 2022), where the policy exploits loopholes in the proxy RM to artificially increase the score without matching human preferences.

Challenges in reward modeling. Designing RMs reflecting human preferences is a complex challenge for two main reasons: distribution shifts and noise in human preferences (as further detailed in Appendix A.2.1). Then, a good RM should ideally satisfy the three following properties.

> ***Property* 1: efficiency.** The RM should incur no memory or inference overhead. Then the policy can be optimized efficiently.

> ***Property* 2: reliability.** The RM should reliably reward predictions despite distribution shifts. Then the policy can explore while relying on the RM.

> ***Property* 3: robustness.** The RM should be robust to label inconsistencies in binary preferences. Then the policy can learn from robust signals from the RM.

2.2. Related work

Previous reward modeling works have explored a few research directions, further detailed in Appendix A.2.2. During RL, the standard strategy is to encourage the policy to remain close to its SFT initialization with Kullback-Leibler (KL) regularization (Jaques et al., 2017); KL reduces model drift (Lazaridou et al., 2020) but can cause underfitting and adds an extra hyperparameter (the regularization strength α). Collecting, labelling and then training on new data (reflecting the evolving policy) can improve the *reliability* of the RM (Touvron et al., 2023). Yet it poses significant *efficiency* challenges due to the continuous requirement for human annotation and computational resources. In contrast, *active learning* strategies (Reddy et al., 2020) proactively enrich the preference dataset by seeking out a diverse set of generations. Concurrent work (Wang et al., 2024) suggests applying label smoothing and flipping. Most similar to *WARM*, *prediction ensembling* (ENS, Lakshminarayanan et al., 2017) averages the logits from M RMs. From a bias-variance perspective (Kohavi et al., 1996), ENS reduces variance when members are diverse (Ueda & Nakano, 1996), and thus favors *reliability* under distribution shifts where variance is the key issue (Ramé et al., 2022b). From a RL perspective, ENS mitigates hacking risks (Christiano et al., 2017; Coste et al., 2023; Eisenstein et al., 2023). Despite its advantages, ENS faces *efficiency* challenges; the memory and inference costs grow linearly with M, making ENS incompatible with the scaling trend in RMs, where larger architectures consistently perform better (Kundu et al., 2023). Moreover, we will also show in Section 4.2 that ENS fails to improve *robustness* to preference inconsistencies.

3. *WARM*

3.1. Weight averaging of reward models

Facing those challenges in reward modeling and the limitations from existing approaches, we propose Weight Averaged Reward Models (*WARM*). *WARM* is a simple and *efficient* strategy that combines multiple models without the memory and inference overheads of prediction ensembling, enhancing reward *reliability* (under distribution shifts) and *robustness* (amidst noisy preference dataset). *WARM* is illustrated in Figure 1(a) and described below.

1. *Shared pre-trained initialization*. For a given pre-trained LLM, each RM is initialized from $\left(\theta^{sft}, \omega\right)$ combining SFT weights and a linear probed classifier.
2. *Diverse fine-tunings*. We launch M RM fine-tunings with diverse hyperparameters (as in a grid search), yielding M weights $\{\phi_i\}_{i=1}^{M}$.
3. *Weight averaging*. We average those M weights together to form $\phi^{\text{WARM}} = \frac{1}{M}\sum_{i=1}^{M}\phi_i$.

Then $r_{\phi^{\text{WARM}}}$ serves as the proxy RM to guide the RL procedure, as *efficiently* as an individual RM, but with the enhanced *reliability* and *robustness* provided by the WA strategy, that leverages the strengths and mitigates the weaknesses of the individual RMs.

3.2. Linear mode connectivity

Compared to ENS, the main difference lies in how *WARM* combines the different RMs: we do so through *linear interpolation in the weight space*. It relies on the linear mode connectivity (LMC, Frankle et al., 2020; Neyshabur et al., 2020) property across fine-tuned weights, i.e., the fact that the accuracy of the interpolated model is at least as good as the interpolation of the individual accuracies. Precisely, by defining the pairwise accuracy of an RM r_ϕ w.r.t. a dataset $\mathcal{D}$ as $\text{Acc}(r_\phi, \mathcal{D}) = \mathbb{E}_{(x,y^+,y^-)\in\mathcal{D}}\left[\mathbb{1}_{r_\phi(x,y^+)\geq r_\phi(x,y^-)}\right]$, the following Observation 1 underpins the success of *WARM*.

Observation 1 (LMC)**.** *Given two fine-tuned weights ϕ_1 and ϕ_2 with a shared pre-training and a test dataset $\mathcal{D}_{test}$, then for all $\lambda \in [0,1]$, $\text{Acc}\left(r_{(1-\lambda)\cdot\phi_1+\lambda\cdot\phi_2}, \mathcal{D}_{test}\right) \geq (1-\lambda)\times\text{Acc}(r_{\phi_1}, \mathcal{D}_{test}) + \lambda\times\text{Acc}(r_{\phi_2}, \mathcal{D}_{test})$.*

We empirically validate this LMC in Figure 2, by evaluating interpolated RMs on OOD test samples. This follows similar observations for multi-class classification in the context of computer vision (Frankle et al., 2020; Neyshabur et al., 2020), which led to a plethora of weight averaging (WA) works such as the model soups (Wortsman et al., 2022a; Ramé et al., 2022b; 2023) variants (detailed in our related work in Appendix A.1).

Remark 1 (Importance of pre-training and linear probing)**.** *The efficacy of WA can be surprising given the non-linearities (Vaswani et al., 2017) and permutation symmetries (Ainsworth et al., 2022) in deep neural network architectures. WA is actually possible only because of the shared pre-training which constrains the divergence during fine-tunings (Neyshabur et al., 2020), such as the weights remain in convex regions of the loss valley (Gueta et al., 2023). In contrast, the LMC does not hold when training weights from scratch (Neyshabur et al., 2020), even if the random initialization is shared. For these reasons and to facilitate the LMC, we follow Ramé et al. (2022b; 2023) and use linear probing to initialize the classifier ω; compared to random initialization, such linear probing prevents feature distortion (Kumar et al., 2022).*

3.3. Sources of diversity

On one hand, *WARM* requires shared pre-training so that the fine-tuned weights remain linearly connected. On the other hand, weights must not be identical: actually, the diversity across those fine-tuned weights significantly contributes to the accuracy gains observed in WA (Ramé et al., 2022b). Overall, an effective *WARM* requires a delicate trade-off

between ensuring LMC and diversity across weights.

In practice, we use the following sources of diversity (Gontijo-Lopes et al., 2022), leading the RM fine-tunings to *diverse yet linearly connected* models. First, the different fine-tunings see the data samples in *different orders*. Second, we sample slightly *different hyperparameters*, notably different learning rates and dropout probabilities, as detailed in Appendix C.3. Third, we investigate a new source of *diversity in initialization* named *Baklava*, illustrated in Figure 7. Specifically, we initialize the RMs' featurizers from different checkpoints $\{\theta_i^{sft}\}_{i=1}^M$ collected along a given SFT trajectory. *Baklava* relaxes the shared initialization constraint from model soups (Wortsman et al., 2022a) to simply sharing the same pre-training: *Baklava* is an *efficient* alternative to model ratatouille (Ramé et al., 2023) without the need of multiple auxiliary tasks. Overall, *Baklava* increases diversity compared to only initializing from the last SFT checkpoint, while adhering to the shared pre-training requisite for LMC, without incurring any overhead.

4. On the Benefits of *WARM*

We now explore the benefits from *WARM*. We ground our analysis on the empirical comparison between WA and ENS for reward modeling, and a novel general theoretical comparison in Section 4.3.

Experimental setup. We use the TL;DR summarization benchmark (Völske et al., 2017), a standard in reward modeling for LLMs, briefly described below and further detailed in Appendix C. The goal of the RMs is to score summaries such as they are ranked properly. In training, we use the dataset $\mathcal{D}_{train}$ from Stiennon et al. (2020) where the candidate summaries are generated by GPT-3 (Brown et al., 2020) variants. To obtain the labels, we follow the RLAIF procedure from Lee et al. (2023), where a PaLM-L (Anil et al., 2023) is prompted with chain-of-thought (Wei et al., 2022b) to mimick human preferences. This strategy performs similarly to human labelers with similar inter-agreement, and will be useful in Section 5 as an oracle metric. The RMs are PaLM-XXS models, pre-trained and SFT-ed on the preferred summaries from $\mathcal{D}_{train}$, on which we plug a linear probed (Kumar et al., 2022) classification layer. We train the RMs for 10k steps on $\mathcal{D}_{train}$, with hyperparameters and procedure detailed in Appendix C.3. We report RMs' accuracies on a novel out-of-distribution (OOD) test dataset $\mathcal{D}_{ood}$ with 92k pairwise comparisons where summaries are generated by multiple PaLM-XS policies with high temperature, some only pre-trained, others after SFT or RLHF.

4.1. Standard analysis: WA for *reliability* and *efficiency*

Previous works (Wortsman et al., 2022a; Ramé et al., 2022b) argued that WA is best understood as an efficient approximation of ENS, as clarified in Observation 2.

Observation 2 (WA and ENS: standard analysis)**.** *Weight averaging and prediction ensembling perform similarly: i.e., for all* $\lambda \in [0, 1]$ *and a test dataset* $\mathcal{D}_{test}$, $\text{Acc}\big(r_{(1-\lambda)\cdot\phi_1+\lambda\cdot\phi_2}, \mathcal{D}_{test}\big) \approx \text{Acc}((1-\lambda) \times r_{\phi_1} + \lambda \times r_{\phi_2}, \mathcal{D}_{test})$.

Theoretically, a simple Taylor expansion can justify this similarity when $\|\phi_1 - \phi_2\| \ll 1$. Empirically, this is validated in Figure 2 where the accuracy curves on $\mathcal{D}_{ood}$ for WA and ENS closely match. This similarity justifies that WA is a variance reduction method; then, because variance is the dominant issue under distribution shifts (Ramé et al., 2022b), this explains the significant gains in Figure 2 over the individual RMs ϕ_1 and ϕ_2 (validating Observation 1), in particular when weights are sufficiently diverse. This suggests improved *reliability* in *WARM*, with *efficiency* benefits over ENS: indeed, WA maintains a single set of weights, removing the memory and inference overheads from ENS.

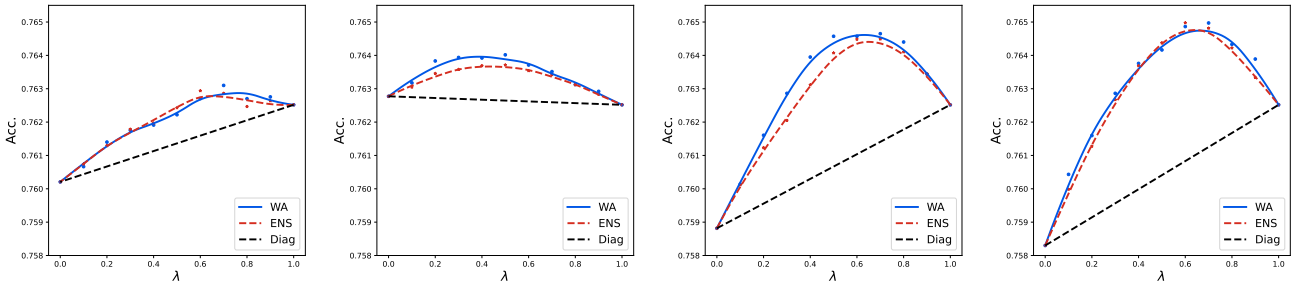
4.2. Refined analysis: WA for more *robustness*

A surprising fact remains poorly unexplained. WA is slightly superior to ENS under distribution shifts, which one can see on the plots from Figure 2, and more consistently in Figure B.1 from model soups (Wortsman et al., 2022a) or in Figure 1 from DiWA (Ramé et al., 2022b). More generally, WA is the state-of-the-art strategy for OOD generalization, consistently outperforming ENS; yet, this was only partially explained in Lin et al. (2024), urging for new insights about the difference between WA and ENS.

Corruption setup. To refine our understanding on the difference between WA and ENS, we propose a new setup where 25% of the binary labels are swapped in training. We then report the per-subset accuracies on Figure 3, and enrich those results in Appendix D.1 where we consistently observe the same phenomenon. On the corrupted subset of training data, the accuracy curve for WA is below the expected accuracies, while it is above on all other subsets. More precisely, we make the following Observation 3.

Observation 3 (WA and ENS: refined analysis)**.** *The accuracy gains of WA over ENS grow as data moves away from the training distribution.*

- *WA* $\ll$ *ENS on train corrupt: WA is worse than ENS on train samples with swapped labels, reducing memorization and improving* robustness *to label corruption.*
- *WA* $\leq$ *ENS on train clean: WA is worse than ENS on train samples with correct labels.*
- *WA* $\gtrapprox$ *ENS on ID val: WA is better or similar to ENS on samples without distribution shifts.*



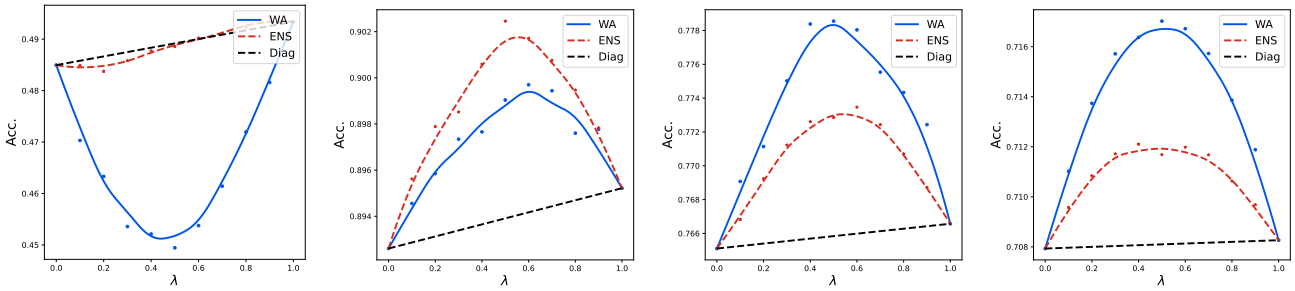
(a) 1 RM fine-tuning at 2 different training steps.

(b) 2 RM fine-tunings with shared config.

(c) 2 RM fine-tunings with different learning rates.

(d) 2 RM fine-tunings with different inits: *Baklava*.

Figure 2: **Experiments under distribution shifts validating Observations 1 and 2** on TL;DR (Völske et al., 2017). We report the accuracies on $\mathcal{D}_{ood}$ when interpolating between two RM ϕ_1 and ϕ_2 with the coefficient λ sliding between 0 and 1. *WA* stands for weight averaging $r_{(1-\lambda)\cdot\phi_1+\lambda\cdot\phi_2}$ while *ENS* combines the predictions $(1-\lambda)\times r_{\phi_1}+\lambda\times r_{\phi_2}$; *Diag* is the interpolated accuracy $(1-\lambda)\times \text{Acc}(r_{\phi_1})+\lambda\times\text{Acc}(r_{\phi_2})$. We consider sources of increasing diversity between ϕ_1 and ϕ_2: in Figure 2(a), they are collected at different number of training steps (8k and 10k) along a single RM fine-tuning; in Figure 2(b), they are from two independant RM fine-tunings, with the exact same config, but seeing the data in different orders; in Figure 2(c), they have different learning rates (1e-4 and 4e-5); in Figure 2(d), they are initalized from different SFT checkpoints collected at different number of SFT steps (8k and 12k), per *Baklava*.



(a) Train (corrupt).

(b) Train (clean).

(c) Validation (ID).

(d) Test (OOD).

Figure 3: **Corruption experiments validating Observation 3**. The two RMs are fine-tuned with the same config but this time with 25% corruption; we then report accuracies on the different data subsets. WA reduces memorization of the corrupted labels in Figure 3(a), and still performs slightly worse than ENS on the clean training samples in Figure 3(b); yet, WA generalizes better than ENS as we move away from the training distribution, in particular on $\mathcal{D}_{ood}$ in Figure 3(d).

- *WA $\geq$ ENS on OOD test: WA is far better than ENS on test samples from new distributions, improving* reliability *under distribution shifts.*

Overall, this suggests that weight averaging memorizes less and generalizes better than ensembling predictions.

4.3. Weight averaging enforces invariance across runs

We now provide theoretical support to this Observation 3, by suggesting that WA acts as a regularization towards the predictive mechanisms that are *invariant* across runs, i.e., learned simultaneously in each independent run. Then, in contrast with ENS, WA improves *robustness* to corruption because it underweights the run-specific features (with low probability of being learned) inducing memorization.

Setup. We consider a simplified binary classification setup with labels $y \in \{-1, 1\}$, related to F features $\{z^j\}_{j=1}^F$ such as $z^j \in \mathbb{R}^d$. From inputs x, we train a binary classifier $r(x) = \omega^\intercal f(x)$. Following Lin et al. (2024), we make three key assumptions. First, *features orthogonality*: we assume that $(z^j)^\intercal z^{j'} = 0$ when $j \neq j'$. Second, *input as bag of features*: we assume that the input $x = \left[x^j\right]_{j=1}^F \in \mathbb{R}^{F\times d}$ can be represented as the concatenation of x^j generated by $x^j \sim \mathcal{N}\left(y \cdot z^j, \sigma \cdot \mathbf{I}_d\right)$ with $\sigma \ll 1$. Finally, the *binary featurizer* assumption: we assume that the featurizer $f = \left[f^j\right]_{j=1}^F \in \{0, 1\}^F$ is a binary selector of the features that make the input. For example, if $y = 1$, $F = 3$, $x \approx [z^1, z^2, z^3]$, and $f = [1, 0, 1]$ learns to extract the first and third features, then $f(x) \approx z^1 + z^3$. We denote p_j the probability that the featurizer f learns to use the j-th

feature dimension (associated with z^j); this means f^j is 1 with probability p_j and 0 otherwise. Moreover, for infinite training samples and under some constraints on σ, Lemma 5 in Lin et al. (2024) proved that the optimal linear fit ω on the features selected from f is $\omega = \sum_{j=1}^{F} f^j \cdot z^j$.

Results. We consider M RMs $\{r_i = \omega_i^\intercal f_i\}_{i=1}^M$, and compare the limit behaviours of their prediction ensembling r_M^{ENS} and weight averaging r_M^{WA} when $M \to \infty$. In this limit case, the averaged prediction $r_M^{ENS} = \frac{1}{M}\sum_{i=1}^M \omega_i^\intercal f_i$ for an input x tends towards the expected prediction $\mathbb{E}[r(x)] = \mathbb{E}[\omega^\intercal f(x)] = \mathbb{E}_{\{f^j\}_{j=1}^F}\big[\big(\sum_{j=1}^F f^j \cdot z^j\big)^\intercal \big(\sum_{j'=1}^F f^{j'} \cdot x^{j'}\big)\big] \approx y \cdot \sum_{j=1}^F p_j \cdot |z^j|^2$, using $x^{j'} \approx y \cdot z^{j'}$ thus $(z^j)^\intercal x^{j'} \approx 0$ when $j \neq j'$, and $(f^j)^2 = f^j$.

$$r_M^{ENS}(x) \xrightarrow[M\to\infty]{} \mathbb{E}[\omega^\intercal f](x) \approx \sum_{j=1}^{F} \boldsymbol{p_j} \cdot (z^j)^\intercal x^j. \quad (1)$$

In contrast, when considering $r_M^{WA} = \Big(\frac{1}{M}\sum_{i=1}^M \omega_i\Big)^\intercal \Big(\frac{1}{M}\sum_{i=1}^M f_i\Big)$ with $M \to \infty$, we have $\frac{1}{M}\sum_{i=1}^M f_i \xrightarrow[M\to\infty]{} \mathbb{E}[f] = [p_j]_{j=1}^F$ and $\frac{1}{M}\sum_{i=1}^M \omega_i \xrightarrow[M\to\infty]{} \mathbb{E}[\omega] = \sum_{j=1}^F p_j \cdot z^j$, and thus:

$$r_M^{WA}(x) \xrightarrow[M\to\infty]{} \mathbb{E}[\omega]^\intercal \mathbb{E}[f](x) \approx \sum_{j=1}^{F} \boldsymbol{p_j^2} \cdot (z^j)^\intercal x^j. \quad (2)$$

Interpretation. For ENS, the coefficient for a given feature is $\boldsymbol{p_j}$, the same as the probability of this information being used by any individual network. In contrast, WA involves the square of the probability $\boldsymbol{p_j^2}$. Intuitively, WA applies an AND-mask on the information, that need to be found both in the feature space and the classification weights. Thus WA reduces the reliance on features with low probability, related to minor specific information (such as noise or context) which can be used to fit the corrupted training samples; this would reduce memorization, and thus explains the *robustness* of WA under label corruption. Reciprocally, WA tends to prioritize the most probable features, favoring the mechanisms that are consistently learned, in other words the *mechanisms invariant across runs*. Overall, WA acts as a regularization, improving *robustness* under label corruption by tackling run-specific mechanisms favoring memorization, and improving *reliability* under distribution shifts by preserving run-invariant mechanisms favoring generalization. We further analyze those insights in Appendix B.

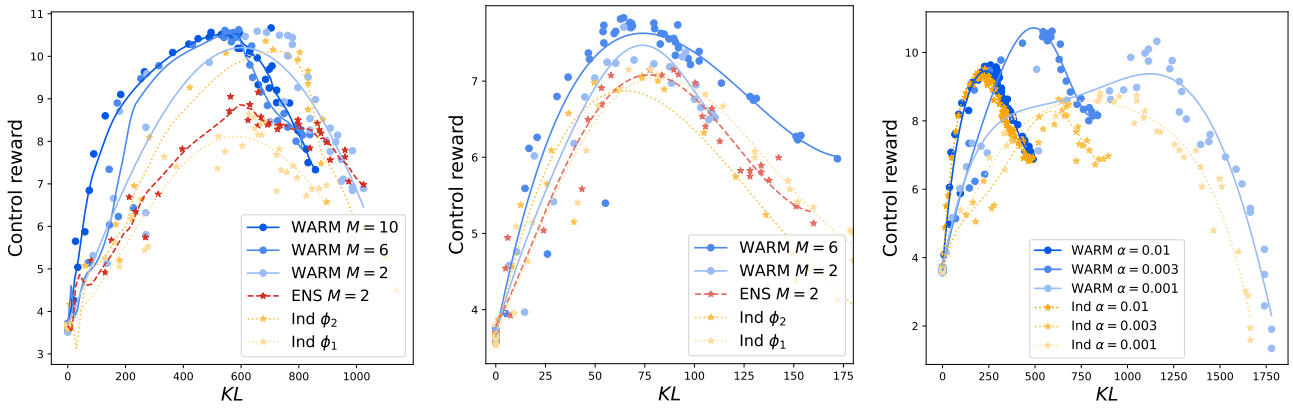
In conclusion, *WARM* has several benefits. First, WARM uses a single RM during RL, thus improves *efficiency*. Second, *WARM* reduces variance, thus improves *reliability* under distribution shifts. Lastly, *WARM* enforces invariance across runs, thus improves *robustness* to noisy preferences by reducing memorization of corrupted labels. Yet, *WARM* has nonetheless a few limitations, detailed in Section 6.

5. Experiments

Setup. To empirically validate *WARM*'s benefits described in previous section, we follow Lee et al. (2023) and train PaLM-XXS RMs on the TL;DR summarization benchmark (Völske et al., 2017) where preference labels are generated by a PaLM-L model prompted with chain-of-thought (Wei et al., 2022b). This AI labeling approach, increasingly common in recent research (Dubois et al., 2023; Eisenstein et al., 2023; Singhal et al., 2023) as an efficient alternative to human assessments, provides an automatic *pairwise oracle preference* metric to evaluate reward hacking (in a similar fashion to the distillation setup from (Gao et al., 2023), discussed in Appendix D.4). In addition, we leverage a PaLM-XS RM for *pointwise control reward* reaching 80.1% accuracy on the OOD dataset $\mathcal{D}_{ood}$. As verified in our experiments, this control RM also detects hacking, as it benefits from a larger architecture and a disjoint pretraining compared to the PaLM-XXS RMs of interest. We consider two setups, without (clean setup) and with (corrupt setup) 25% label corruption in the preference datasets, and denote in each setup the weights $\{\phi_i\}_{i=1}^M$ sorted in decreasing accuracy on $\mathcal{D}_{ood}$. Below we explore the main scenario, where *WARM* guides the RL procedure. We refer the readers to our best-of-N experiments in Appendix D.2.

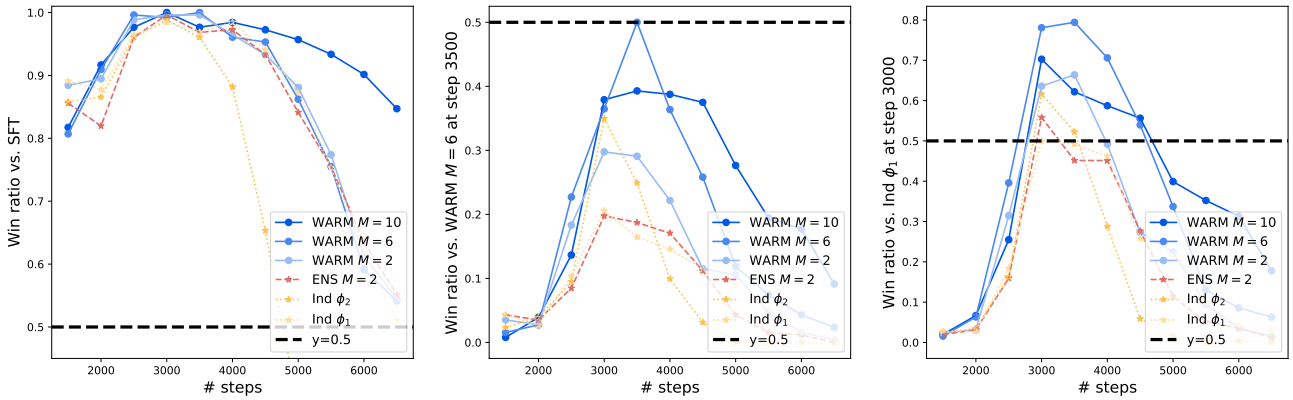
RL fine-tuning of policies. Following Lee et al. (2023), we use a modified version of REINFORCE (Williams, 1992) with a baseline value score for variance reduction. Both policy and value LLMs are PaLM-XS, initialized from the same SFT model. We then generate samples with the policy, compute the reward with the RMs and update the weights to optimize this reward. More details are available in Appendix C.4. To reduce forgetting and encourage the policy to remain close to its SFT initialization, we incorporate a KL regularization (Jaques et al., 2017; Geist et al., 2019) controlled by a coefficient α, ablated in Figure 4(c), yet otherwise set to 0.003 in the clean setup and 0.01 in the corrupt setup. This KL serves as the x-axis in our plots to estimate model drift, as done in the literature; same curves with the number of training steps as the x-axis in Figures 1(b) and 17.

Control reward. In Figure 4, we observe reward hacking; as the policy moves away from its SFT initialization, the in terms of *pointwise control reward collapses*. Critically, *WARM* improves performances: in particular, increasing M pushes the Pareto front of solutions to the top left in Figures 4(a) and 4(b). In comparison, policies trained with ENS (with $M = 2$ for computational reasons) are still susceptible to early reward hacking, while reaching absolute control rewards significantly worse than with *WARM* (even with $M = 2$). In Figure 4(c), we confirm that the α hyperparameter plays a crucial role; low values of α such as 0.001 correspond to high KL, while high values of α such as 0.01 entail low KL but a risk of underfitting. From a practical



(a) RL (clean). (b) RL (corrupt). (c) Ablating α for RL (clean).

Figure 4: **Control reward for RL experiments**: clean preference dataset in Figures 4(a) and 4(c) and 25% corruptions in Figure 4(b). The blue lines show the RL fine-tuning of policies when averaging M weights; the darker, the higher the M. *WARM* performs higher than when using individual RMs (in yellows) or when ensembling their predictions (in red). Figure 4(c) compares policies RL fine-tuned with *WARM* $M = 6$ or ϕ_1, for different α controlling the KL strength.



(a) SFT. (b) WARM $M = 6$. (c) ϕ_1 (best individual RM).

Figure 5: **Oracle preference metric for RL experiments**: clean preference dataset. We plot the win rates along RL fine-tuning against three reference policies: the SFT policy, the policy RL fine-tuned with *WARM* $M = 6$ after 3500 steps, and the policy RL fine-tuned with ϕ_1 after 3000 steps. More results are provided in Appendix D.3.

perspective, this highlights that the optimal value of α for *WARM* is lower than for a single RM; this is because *WARM* can mitigate reward hacking, and thus the optimal policies are obtained for larger values of KL.

Oracle preference. In Figure 5, we compare the different policies according to our pairwise AI labeler. In Figure 5(a), the reference policy is the SFT initialization; all the RL fine-tuned policies outperform this baseline, with *WARM* $M = 6$ reaching a win rate of 99.8% after 3500 steps (the highest win rate among all policies). We use this policy as the reference in Figure 5(b); no other policy could beat it. Interestingly, we observe that using $M = 10$ rewards can delay reward hacking but does not improve the peak performance; we speculate this is related to our weight selection procedure, as the weights $\{\phi_i\}_{i=7}^{10}$ have lower individual accuracy on $\mathcal{D}_{ood}$ than $\{\phi_i\}_{i=1}^{6}$ (details in Figure 6). Finally, in Figure 5(c), the reference policy is obtained after 3000 steps of RL fine-tuning with ϕ_1 (the best individual RM on $\mathcal{D}_{ood}$). There is a large region of steps in which *WARM* policies (even for $M = 2$) perform better; the previous reference from Figure 5(b) has a 79.4% win rate against it.

6. Discussion

Benefits. This paper has detailed several of its benefits, and below, we detail more exploratory advantages. *WARM* follows the *updatable machine learning paradigm* (Raffel,

2023), eliminating the need for inter-server communication, thus enabling *embarrassingly simple parallelization* (Li et al., 2022) of RMs. This facilitates its use in *federated learning* scenario (McMahan et al., 2017) where the data should remain private; moreover, WA would add a layer of privacy and bias mitigation by reducing the memorization of private preference (Zaman et al., 2023). Then, a straightforward extension of *WARM* would combine RMs trained on different datasets, for example, coming from different (clusters of) labelers. This diversity could help *WARM* performances, but also from a multi objective perspective (Wu et al., 2023); by non-uniform interpolation of RMs, we could learn a set of *personalized policies* (Ramé et al., 2023). Furthermore, as WA has been shown to limit catastrophic forgetting (Stojanovski et al., 2022; Eeckt et al., 2022), *WARM* could seamlessly support iterative and evolving preferences. Finally, a promising research direction is extending *WARM* to DPO strategies (Rafailov et al., 2023), where averaging the RMs casts back to averaging the DPO policies (Labonne, 2024).

Limitations. *WARM* faces some limitations, notably two when compared to prediction ensembling methods; first, prediction ensembling can benefit from the diversity brought by combining RMs from various architectures and pretrainings; second, prediction ensembling can incorporate prediction disagreement into the reward to provide uncertainty estimation and limit model drift. However, it's been noted in Eisenstein et al. (2023) that simple averaging of logits often performs comparably to more complex prediction aggregation functions that include uncertainty elements. Another limitation is that, while *WARM* effectively reduces certain types of memorization, it does not completely eradicate all forms of spurious correlations or biases inherent in the preference data. For instance, if each individual RM predominantly relies on summary length as a criterion, *WARM* is likely to replicate this tendency. Therefore, alternative methods (from the OOD generalization literature?) might be required, for example those based on invariance regularization (Arjovsky et al., 2019; Ramé et al., 2022a) or last layer retraining (Kirichenko et al., 2023).

7. Conclusion

We introduce *WARM* to address two challenges in reward modeling: *reliability* under distribution shifts and *robustness* under label corruption. By averaging the weights of multiple RMs, *WARM* appears as an *efficient* solution to mitigate reward hacking in RLHF. Our empirical results on summarization demonstrate its effectiveness. We hope that *WARM* will inspire the alignment community to investigate deeper into model merging and the generalization literature.

Impact Statement

WARM is a flexible and pragmatic method to improve the alignment of AI with human values and societal norms, with several benefits detailed along this work, but also a few limitations further described in Section 6. In particular, one key limitation is that *WARM* only enhances the reward modeling stage without tackling the other challenges in RLHF (Casper et al., 2023). Thus, to mitigate the safety risks (Amodei et al., 2016; Hendrycks & Mazeika, 2022; Hendrycks, 2023) from misalignment (Taylor et al., 2016; Ngo et al., 2022), *WARM* must be considered within the larger context of responsible and safe AI.

References

Ainsworth, S. K., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. In *ICLR*, 2022. (p. 4)

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint*, 2016. (pp. 1, 3, and 9)

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. PaLM 2 technical report. *arXiv preprint*, 2023. (pp. 5, 18, 20, and 22)

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint*, 2019. (pp. 2, 9, and 17)

Arpit, D., Wang, H., Zhou, Y., and Xiong, C. Ensemble of averages: Improving model selection and boosting performance in domain generalization. In *NeurIPS*, 2021. (pp. 16 and 19)

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Kernion, J., Ndousse, K., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., and Kaplan, J. A general language assistant as a laboratory for alignment. *arXiv preprint*, 2021. (pp. 1 and 3)

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint*, 2023. (p. 17)

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint*, 2022a. (p. 26)

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint*, 2022b. (pp. 3 and 17)

Barnett, P., Freedman, R., Svegliato, J., and Russell, S. Active reward learning from multiple teachers. *arXiv preprint*, 2023. (p. 17)

Beirami, A., Agarwal, A., Berant, J., D'Amour, A., Eisenstein, J., Nagpal, C., and Suresh, A. T. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint*, 2024. (p. 22)

Blondé, L., Strasser, P., and Kalousis, A. Lipschitzness is all you need to tame off-policy generative adversarial imitation learning. *Machine Learning*, 2022. (p. 18)

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint*, 2021. (pp. 3 and 16)

Bowman, S. R., Hyun, J., Perez, E., Chen, E., Pettit, C., Heiner, S., Lukosuite, K., Askell, A., Jones, A., Chen, A., et al. Measuring progress on scalable oversight for large language models. *arXiv preprint*, 2022. (pp. 2 and 17)

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 1952. (p. 3)

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *NeurIPS*, 2020. (pp. 5 and 18)

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint*, 2023. (p. 1)

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *TMLR*, 2023. (pp. 1 and 9)

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. SWAD: Domain generalization by seeking flat minima. In *NeurIPS*, 2021. (pp. 2 and 16)

Cheng, J., Xiong, G., Dai, X., Miao, Q., Lv, Y., and Wang, F.-Y. Rime: Robust preference-based reinforcement learning with noisy preferences. *arXiv preprint*, 2024. (p. 17)

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017. (pp. 1, 2, 3, 4, and 17)

Clark, J. and Amodei, D. Faulty Reward Functions in the Wild. https://openai.com/research/faulty-reward-functions, 2016. (pp. 1 and 3)

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019. (p. 18)

Condorcet. Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. 1785. (p. 17)

Coste, T., Anwar, U., Kirk, R., and Krueger, D. Reward model ensembles help mitigate overoptimization. *arXiv preprint*, 2023. (pp. 2, 4, and 17)

Croce, F., Rebuffi, S.-A., Shelhamer, E., and Gowal, S. Seasoning model soups for robustness to adversarial and natural distribution shifts. In *CVPR*, 2023. (p. 16)

Daheim, N., Dziri, N., Sachan, M., Gurevych, I., and Ponti, E. M. Elastic weight removal for faithful and abstractive dialogue generation. *arXiv preprint*, 2023. (p. 16)

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *JMLR*, 2020. (pp. 16 and 17)

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 2021. (p. 16)

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. (p. 1)

Don-Yehiya, S., Venezian, E., Raffel, C., Slonim, N., Katz, Y., and Choshen, L. ColD fusion: Collaborative descent for distributed multitask finetuning. In *ACL*, 2023. (p. 16)

Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint*, 2023. (p. 7)

Eeckt, S. V. et al. Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition. *arXiv preprint*, 2022. (p. 9)

Eisenstein, J., Nagpal, C., Agarwal, A., Beirami, A., D'Amour, A., Dvijotham, D., Fisch, A., Heller, K., Pfohl, S., Ramachandran, D., et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint*, 2023. (pp. 2, 4, 7, 9, 17, and 22)

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *ICML*, 2020. (pp. 2, 4, and 16)

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint*, 2022. (p. 17)

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *ICML*, 2023. (pp. 1, 7, and 26)

Gaya, J.-B., Soulier, L., and Denoyer, L. Learning a subspace of policies for online adaptation in reinforcement learning. In *ICLR*, 2022. (p. 16)

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized markov decision processes. In *ICML*, 2019. (p. 7)

Gemini Team, G. Gemini: A family of highly capable multimodal models. 2023. (p. 1)

Ghosh, A., Kumar, H., and Sastry, P. S. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017. (p. 16)

Gontijo-Lopes, R., Dauphin, Y., and Cubuk, E. D. No one representation to rule them all: Overlapping features of training methods. In *ICLR*, 2022. (p. 5)

Gooding, S. and Mansoor, H. The impact of preference agreement in reinforcement learning from human feedback: A case study in summarization. *arXiv preprint*, 2023. (p. 17)

Gueta, A., Venezian, E., Raffel, C., Slonim, N., Katz, Y., and Choshen, L. Knowledge is a region in weight space for fine-tuned language models. In *EMNLP*, 2023. (p. 4)

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *ICLR*, 2021. (p. 17)

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *ICML*, 2017. (p. 17)

Hafner, R. and Riedmiller, M. Reinforcement learning in feedback control: Challenges and benchmarks from technical process control. *Machine learning*, 2011. (p. 18)

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *NeurIPS*, 2018. (p. 16)

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. *NeurIPS*, 2017. (p. 18)

Hendrycks, D. Natural selection favors AIs over humans. *arXiv preprint*, 2023. (p. 9)

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. (p. 16)

Hendrycks, D. and Mazeika, M. X-risk analysis for AI research. *arXiv preprint*, 2022. (pp. 1 and 9)

Hilton, J. KL divergence of max-of-n, 2023. (p. 22)

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. In *NeurIPS*, 2022. (p. 16)

Ilharco, G., Tulio Ribeiro, M., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *ICLR*, 2023. (p. 16)

Irvine, R., Boubert, D., Raina, V., Liusie, A., Mudupalli, V., Korshuk, A., Liu, Z., Cremer, F., Assassi, V., Beauchamp, C.-C., et al. Rewarding chatbots for real-world engagement with millions of users. *arXiv preprint*, 2023. (p. 17)

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. (pp. 2, 16, and 19)

Jain, N., Chiang, P.-y., Wen, Y., Kirchenbauer, J., Chu, H.-M., Somepalli, G., Bartoldson, B. R., Kailkhura, B., Schwarzschild, A., Saha, A., et al. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint*, 2023. (p. 16)

Jaques, N., Gu, S., Bahdanau, D., Hernández-Lobato, J. M., Turner, R. E., and Eck, D. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *ICML*, 2017. (pp. 4 and 7)

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. (p. 16)

Juneja, J., Bansal, R., Cho, K., Sedoc, J., and Saphra, N. Linear connectivity reveals generalization strategies. In *ICLR*, 2023. (p. 16)

Kaufmann, T., Ball, S., Beck, J., Hüllermeier, E., and Kreuter, F. On the challenges and practices of reinforcement learning from real human feedback. 2023. (p. 17)

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. In *ICLR*, 2023. (p. 9)

Knox, W. B., Hatgis-Kessell, S., Adalgeirsson, S. O., Booth, S., Dragan, A., Stone, P., and Niekum, S. Learning optimal advantage from preferences and mistaking it for reward. *arXiv preprint*, 2023. (p. 17)

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S. M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021. (p. 17)

Kohavi, R., Wolpert, D. H., et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, 1996. (p. 4)

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *ICLR*, 2022. (pp. 4, 5, and 18)

Kundu, S., Bai, Y., Kadavath, S., Askell, A., Callahan, A., Chen, A., Goldie, A., Balwit, A., Mirhoseini, A., McLean, B., et al. Specific versus general principles for constitutional ai. *arXiv preprint*, 2023. (p. 4)

Laakom, F., Raitoharju, J., Iosifidis, A., and Gabbouj, M. Learning distinct features helps, provably. *arXiv preprint*, 2021. (p. 17)

Labonne, M. NeuralBeagle14-7B. https://huggingface.co/mlabonne/NeuralBeagle14-7B-GGUF, 2024. (pp. 9 and 17)

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. (pp. 2 and 4)

Lawson, D. and Qureshi, A. H. Merging decision transformers: Weight averaging for forming multi-task policies. In *ICLR RRL Workshop*, 2023. (p. 16)

Lazaridou, A., Potapenko, A., and Tieleman, O. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *ACL*, 2020. (p. 4)

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Carbune, V., and Rastogi, A. RLAIF: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint*, 2023. (pp. 3, 5, 7, 17, 18, 20, and 22)

Lewis, M., Yarats, D., Dauphin, Y. N., Parikh, D., and Batra, D. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint*, 2017. (p. 1)

Li, L., Chai, Y., Wang, S., Sun, Y., Tian, H., Zhang, N., and Wu, H. Tool-augmented reward modeling. In *ICLR*, 2023. (p. 17)

Li, M., Gururangan, S., Dettmers, T., Lewis, M., Althoff, T., Smith, N. A., and Zettlemoyer, L. Branch-Train-Merge: Embarrassingly parallel training of expert language models. *arXiv preprint*, 2022. (p. 9)

Lin, Y., Tan, L., Hao, Y., Wong, H., Dong, H., Zhang, W., Yang, Y., and Zhang, T. Spurious feature diversification improves out-of-distribution generalization. In *ICLR*, 2024. (pp. 5, 6, 7, and 16)

Matena, M. and Raffel, C. Merging models with Fisher-weighted averaging. In *NeurIPS*, 2022. (p. 2)

McKinney, L., Duan, Y., Krueger, D., and Gleave, A. On the fragility of learned reward functions. *arXiv preprint*, 2023. (p. 1)

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017. (p. 9)

Mizrahi, M., Kaplan, G., Malkin, D., Dror, R., Shahaf, D., and Stanovsky, G. State of what art? a call for multi-prompt llm evaluation. *arXiv preprint*, 2023. (p. 17)

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *ICML*, 2013. (pp. 2 and 17)

Nayman, N., Golbert, A., Noy, A., Ping, T., and Zelnik-Manor, L. Diverse ImageNet models transfer better. *arXiv preprint*, 2022. (p. 17)

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? In *NeurIPS*, 2020. (pp. 2, 4, and 16)

Ng, A. Y., Russell, S., et al. Algorithms for inverse reinforcement learning. In *ICML*, 2000. (p. 16)

Ngo, R., Chan, L., and Mindermann, S. The alignment problem from a deep learning perspective. *arXiv preprint*, 2022. (pp. 1, 3, 9, and 16)

Nikishin, E., Izmailov, P., Athiwaratkun, B., Podoprikhin, D., Garipov, T., Shvechikov, P., Vetrov, D., and Wilson, A. G. Improving stability in deep reinforcement learning with weight averaging. 2018. (p. 16)

Noukhovitch, M., Lavoie, S., Strub, F., and Courville, A. Language model alignment with elastic reset. In *NeurIPS*, 2023. (p. 16)

OpenAI. Gpt-4 technical report. 2023. (p. 1)

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. (p. 3)

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. (pp. 1, 2, 3, and 17)

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. (p. 17)

Pan, A., Bhatia, K., and Steinhardt, J. The effects of reward misspecification: Mapping and mitigating misaligned models. In *ICLR*, 2022. (pp. 1 and 3)

Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. In *NeurIPS*, 2020. (p. 17)

Pirotta, M., Restelli, M., and Bascetta, L. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 2015. (p. 18)

Pitis, S. Failure modes of learning reward models for llms and other sequence models. In *ICML*, 2023. (p. 17)

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018. (pp. 1 and 3)

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint*, 2023. (pp. 9 and 17)

Raffel, C. Building Machine Learning Models Like Open Source Software. *ACM*, 2023. (p. 8)

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. (pp. 22 and 26)

Ramé, A., Dancette, C., and Cord, M. Fishr: Invariant gradient variances for out-of-distribution generalization. In *ICML*, 2022a. (pp. 9 and 17)

Ramé, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P., and Cord, M. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022b. (pp. 2, 4, 5, 16, and 18)

Ramé, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model Ratatouille: Recycling diverse models for out-of-distribution generalization. In *ICML*, 2023. (pp. 2, 4, 5, 16, 18, and 19)

Ramé, A., Couairon, G., Shukor, M., Dancette, C., Gaya, J.-B., Soulier, L., and Cord, M. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *NeurIPS*, 2023. (pp. 2, 9, and 16)

Razin, N., Zhou, H., Saremi, O., Thilak, V., Bradley, A., Nakkiran, P., Susskind, J., and Littwin, E. Vanishing gradients in reinforcement finetuning of language models. *arXiv preprint*, 2023. (p. 18)

Reddy, S., Dragan, A., Levine, S., Legg, S., and Leike, J. Learning human objectives by evaluating hypothetical behavior. In *ICML*, 2020. (pp. 4 and 17)

Rosca, M., Weber, T., Gretton, A., and Mohamed, S. A case for new neural network smoothness constraints. In *NeurIPS ICBINB*, 2020. (p. 18)

Sabzevari, M. *Ensemble learning in the presence of noise*. PhD thesis, Universidad Autónoma de Madrid, 2019. (p. 16)

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint*, 2017. (pp. 3 and 20)

Sclar, M., Choi, Y., Tsvetkov, Y., and Suhr, A. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint*, 2023. (p. 17)

Shah, R., Gundotra, N., Abbeel, P., and Dragan, A. On the feasibility of learning, rather than assuming, human biases for reward inference. In *ICML*, 2019. (p. 17)

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICML*, 2018. (pp. 18 and 20)

Shukor, M., Dancette, C., Ramé, A., and Cord, M. Unival: Unified model for image, video, audio and language. *TMLR*, 2023. (p. 16)

Simon, H. A. Bounded rationality. *Utility and probability*, 1990. (p. 17)

Singhal, P., Goyal, T., Xu, J., and Durrett, G. A long way to go: Investigating length correlations in rlhf. *arXiv preprint*, 2023. (pp. 1 and 7)

Skalse, J. M. V., Howe, N. H. R., Krasheninnikov, D., and Krueger, D. Defining and characterizing reward gaming. In *NeurIPS*, 2022. (p. 3)

Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 2017. (p. 18)

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. *TNNLS*, 2022. (p. 16)

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *NeurIPS*, 2020. (pp. 1, 5, 17, and 18)

Stojanovski, Z., Roth, K., and Akata, Z. Momentum-based weight interpolation of strong zero-shot models for continual learning. In *NeurIPS Workshop*, 2022. (pp. 9 and 16)

Strathern, M. Improving ratings: audit in the british university system. *European Review*, 1997. (p. 1)

Sun, Z., Shen, S., Cao, S., Liu, H., Li, C., Shen, Y., Gan, C., Gui, L.-Y., Wang, Y.-X., Yang, Y., et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint*, 2023. (p. 17)

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint*, 2013. (p. 18)

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. Learning from noisy labels by regularized estimation of annotator confusion. In *CVPR*, 2019. (p. 16)

Taylor, J., Yudkowsky, E., LaVictoire, P., and Critch, A. Alignment for advanced machine learning systems. *Ethics of AI*, 2016. (pp. 1, 3, and 9)

Teney, D., Lin, Y., Oh, S. J., and Abbasnejad, E. ID and OOD performance are sometimes inversely correlated on real-world datasets. In *NeurIPS Workshop*, 2023. (p. 17)

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*, 2023. (pp. 1, 4, and 17)

Ueda, N. and Nakano, R. Generalization error of ensemble estimators. In *ICNN*, 1996. (p. 4)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017. (pp. 3 and 4)

Völske, M., Potthast, M., Syed, S., and Stein, B. Tl; dr: Mining reddit to learn automatic summarization. In *ACL Workshop*, 2017. (pp. 5, 6, and 7)

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. On calibration and out-of-domain generalization. In *NeurIPS*, 2021. (p. 17)

Wang, B., Zheng, R., Chen, L., Liu, Y., Dou, S., Huang, C., Shen, W., Jin, S., Zhou, E., Shi, C., et al. Secrets of RLHF in large language models part II: Reward modeling. *arXiv preprint*, 2024. (pp. 4, 17, and 18)

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. In *ICLR*, 2022a. (pp. 1 and 3)

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. Chain-of-Thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022b. (pp. 5, 7, and 18)

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 1992. (pp. 3, 7, and 20)

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022a. (pp. 2, 4, 5, 16, and 19)

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Hajishirzi, H., Farhadi, A., Namkoong, H., and Schmidt, L. Robust fine-tuning of zero-shot models. In *CVPR*, 2022b. (p. 16)

Wu, J., Ouyang, L., Ziegler, D. M., Stiennon, N., Lowe, R., Leike, J., and Christiano, P. Recursively summarizing books with human feedback. *arXiv preprint*, 2021. (p. 17)

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. In *NeuriPS*, 2023. (p. 9)

Xia, X., Liu, T., Han, B., Gong, M., Yu, J., Niu, G., and Sugiyama, M. Sample selection with uncertainty of losses for learning with noisy labels. In *ICLR*, 2022. (p. 16)

Yang, Y.-Y., Rashtchian, C., Zhang, H., Salakhutdinov, R., and Chaudhuri, K. Adversarial robustness through local lipschitzness. *arXiv preprint*, 2020. (p. 18)

Zaman, K., Choshen, L., and Srivastava, S. Fuse to forget: Bias reduction and selective memorization through model fusion. *arXiv preprint*, 2023. (pp. 9 and 16)

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., and Oermann, E. K. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLOS Medicine*, 2018. (p. 16)

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR*, 2017. (p. 16)

Zhuang, S. and Hadfield-Menell, D. Consequences of misaligned AI. *NeurIPS*, 2020. (p. 17)

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint*, 2019. (pp. 1 and 17)

WARM: On the Benefits of Weight Averaged Reward Models

Supplementary material

This supplementary material is organized as follows:

- Appendix A enriches our related work section.
- Section 6 further discusses the theoretical insights from this work, along with *WARM*'s benefits and limitations.
- Appendix C clarifies some experimental details.
- Appendix D enriches our experiments.

A. Related Work

This paper leverages the insights from the OOD generalization literature, in particular from linear mode connectivity (see Appendix A.1), and applies them to the design of *efficient*, *reliable* and *robust* reward models (see Appendix A.2.2).

A.1. Out-of-distribution generalization, linear mode connectivity and memorization

LMC in fine-tuning. Fine-tuning foundation models (Bommasani et al., 2021) into specialized models that generalize well to new distributions is critical for many real-world applications (Hendrycks & Dietterich, 2019; Zech et al., 2018; DeGrave et al., 2021). Recently, different variants of weight averaging (WA) were able to improve performance, such as moving average (Izmailov et al., 2018; Cha et al., 2021; Arpit et al., 2021), WiSE fine-tuning (Wortsman et al., 2022b), model soups (Wortsman et al., 2022a), DiWA (Ramé et al., 2022b) and model ratatouille (Ramé et al., 2023). These works rely on the LMC (Frankle et al., 2020; Neyshabur et al., 2020) across fine-tuned weights, which was extended to fine-tunings on different tasks (Ilharco et al., 2022; Don-Yehiya et al., 2023; Ramé et al., 2023), modalities (Shukor et al., 2023) or with different losses (Ramé et al., 2022b; Croce et al., 2023), although (Juneja et al., 2023) highlighted some limitations. WA was also used recently in RL setups (Nikishin et al., 2018; Gaya et al., 2022; Lawson & Qureshi, 2023; Ramé et al., 2023; Noukhovitch et al., 2023), in particular in RLHF in (Ramé et al., 2023; Noukhovitch et al., 2023) but only to combine policies and not rewards.

Insights into WA. Specifically, WA comes with several benefits. First, WA flattens the loss landscape (Cha et al., 2021). Second, WA approximates prediction ensembling, thus reduces variance of the estimator (Wortsman et al., 2022a; Ramé et al., 2022b) and tackles model misspecification (D'Amour et al., 2020). Third, WA combines models' abilities (Ilharco et al., 2023; Daheim et al., 2023), which can be useful for multi-task (Ilharco et al., 2022), multi-objective (Ramé et al., 2023) or in continual learning (Stojanovski et al., 2022) setups. Lastly, it has recently been shown that WA can provide some benefits under spurious correlations (Lin et al., 2024; Zaman et al., 2023), with a phenomenon called *FalseFalseTrue* in Lin et al. (2024). These works (Lin et al., 2024; Zaman et al., 2023) share similarities with our memorization experiments from Section 4.2, but we are the first to analyze WA regularization properties under label corruption, and their consequences on generalization. In contrast, in Zaman et al. (2023) the networks are trained on different datasets while Lin et al. (2024) consider spurious correlations. Overall, our theoretical insights clarify and simplify those from Lin et al. (2024).

Memorization. Traditional approaches (Song et al., 2022) tackling memorization of corrupted labels (Zhang et al., 2017) usually require explicit regularization (Tanno et al., 2019), specific data augmentation (Jain et al., 2023), loss adjustment (Ghosh et al., 2017) or sample selection (Xia et al., 2022). Some other strategies are based on ensembling: they filter out potentially corrupted samples with self-labeling filtering (Jiang et al., 2018; Han et al., 2018) or bagging diversity procedures (Sabzevari, 2019). As far as we know, with WA we propose the first strategy combining multiple models trained on the same dataset that manages to tackle corruption.

A.2. Reward modeling

One of the central challenge in aligning LLMs is the absence of explicit rewards from the environment, a.k.a. the outer alignment challenge (Ngo et al., 2022). While Inverse Reinforcement Learning (Ng et al., 2000) attempts to derive the RM

from expert demonstrations, most recent efforts (Christiano et al., 2017; Ziegler et al., 2019; Stiennon et al., 2020; Wu et al., 2021; Ouyang et al., 2022) primarily focus on learning from human preferences. Despite its importance to enhance LLM performances post-RL and for safe deployment in real-world applications, how to best design RMs has arguably receive less attention than it warrants. First in Appendix A.2.1 we clarify the challenges in designing those RMs, and then in Appendix A.2.2 we discuss the existing approaches.

A.2.1. Challenges in reward modeling

Distribution shifts. The primary challenge is the distribution shifts resulting from the offline nature of preference data. Indeed, the generations in the preference dataset and those from the policy θ^{sft} do not necessarily follow the same distributions, and the shifts can become even more pronounced due to model drift during RL. The OOD generalization literature has extensively analyzed the repercussions of these shifts. Firstly, they often lead to a reduction in performance (Gulrajani & Lopez-Paz, 2021; Koh et al., 2021). RMs (of limited capacity) trained on narrow data distributions may rely on spurious correlations (Arjovsky et al., 2019) or a limited number of features (Pezeshki et al., 2020), thus failing when encountering OOD examples (Laakom et al., 2021; Nayman et al., 2022). Secondly, they complicate the selection of RMs, as ID validation metrics may poorly correlate with real-world OOD performances (D'Amour et al., 2020; Teney et al., 2023) and the ability to guide the RL (Eisenstein et al., 2023). Lastly, RMs can become poorly calibrated (Guo et al., 2017) in OOD scenarios (Ovadia et al., 2019; Wald et al., 2021), and predict more extreme values as rewards. Such miscalibration exacerbates the problem in a negative feedback loop, further intensifying model drift and distribution shifts. In conclusion, limited data coverage during reward modeling reduces the *reliability* of the RM and facilitates reward hacking (Zhuang & Hadfield-Menell, 2020) in regions where the RM is badly specified.

Inconsistent preferences. The second major challenge is the label noise in preference datasets. Human labelers, often grappling with fatigue, misunderstandings (Simon, 1990; Shah et al., 2019) and imperfect incentives (Kaufmann et al., 2023), might default to simpler criteria such as length, bullet points, or politeness rather than more causal indicators. This tendency is exacerbated for complex tasks (Bowman et al., 2022) or when considering multiple objectives, ranging from harmlessness (Ganguli et al., 2022) to engagement (Irvine et al., 2023) and representing the heterogeneity of human opinions. Consequently, these factors lead to low inter-rater agreement, where human data appears as an imperfect representation of the underlying ground truth (Condorcet, 1785; Pitis, 2023). To mitigate these issues, there has been a shift towards AI-generated preferences (Bai et al., 2022b; Lee et al., 2023), which, while reducing human labor costs, introduces its own set of noise and failure cases, such as sensitivity to prompting strategies (Sclar et al., 2023; Mizrahi et al., 2023). These layers of noise and inconsistency challenge the *robustness* of the RM, and its ability to provide stable signals.

A.2.2. Reward modeling approaches

Some works (Knox et al., 2023) seek to refine the reward modeling loss function. Other approaches are more data oriented: for example, LLaMA-2 (Touvron et al., 2023) involves continual learning of the RM to adjust to new generation distributions; Reddy et al. (2020); Barnett et al. (2023) follow an active learning paradigm (Gooding & Mansoor, 2023). Augmenting rewards with tools (Li et al., 2023) or additional information (Sun et al., 2023) represents an even more recent and very promising trend. Limited efforts have been made at the intersection of label corruption and reward modeling; Cheng et al. (2024) tried to filter the preference dataset for small academic locomotion tasks, while Wang et al. (2024) suggests applying label smoothing and flipping. Actually, reward ensembling is the most discussed method to mitigate reward hacking (Coste et al., 2023; Eisenstein et al., 2023); we show that *WARM* can beat ENS while removing its overheads. Finally, following DPO (Rafailov et al., 2023), a recent trend merges reward modeling with policy learning; though, the policies still tend to hack the preference data (Azar et al., 2023), and thus require only a few training steps and very small learning rates. The WA of DPO policies, theoretically equivalent to the WA of RMs, is a promising research direction with already significant empirical results on public benchmarks, as demonstrated in Labonne (2024).

B. Additional Remarks on the Theoretical Insights

We discuss in more details the results from Section 4.3.

Invariance. We argue that weight averaging only keeps the invariant predictive mechanisms across runs. This is in analogy with the invariance literature (Muandet et al., 2013), popular for domain generalization (Arjovsky et al., 2019; Ramé et al., 2022a) under spurious correlations, where the key idea is that the predictive mechanisms which are *invariant across domains* are the causal ones that are stable under distribution shifts. This theoretically connects two key paradigms for OOD

generalization, ensembling and invariance, and shows that weight averaging actually benefits from both.

Extension to a deeper structure with L layers. We obtain a square in $\boldsymbol{p_j^2}$ in Equation (2) due to our simplified two-layer architecture. Yet, using a deeper structure with L layers, and applying similar assumptions at each layer, we would obtain $\boldsymbol{p_j^L}$. The analysis of these insights under relaxed assumptions is a promising direction for future works.

From reward *robustness* to learnability. When applied to the design of RMs in *WARM*, we now argue that WA facilitates *WARM*'s stability (Wang et al., 2024) by mitigating the reliance on some non-*robust* features. Indeed, WA makes the *WARM* reward more *robust* to small (potentially adversarial (Szegedy et al., 2013)) perturbations (Yang et al., 2020), i.e., smoother (Rosca et al., 2020) in the input space. This relates to the Lipschitzness property of the reward (Hein & Andriushchenko, 2017; Sokolić et al., 2017; Cohen et al., 2019), where the difference in predicted rewards is bounded by the distance in input space. Fortunately, such smoothness is useful in RL (Hafner & Riedmiller, 2011), in particular for the stability of the policy gradient (Pirotta et al., 2015) because "sharp changes in reward value are hard to represent and internalize" (Blondé et al., 2022). This is studied in *Lipschitzness is all you need* (Blondé et al., 2022) where the authors argue that "the local Lipschitzness of the reward is a sine qua non condition for good performance". In summary, *robustness* improves stability and hinders the cascade of errors occurring when minor input variations can cause large reward differences.

C. Implementation Details

C.1. Dataset details

For summarization, we use the Reddit TL;DR dataset (Stiennon et al., 2020), containing posts from Reddit that have been filtered to ensure high quality. The training summaries from Stiennon et al. (2020) are generated by OpenAI GPT-3 (Brown et al., 2020) variants. The dataset contains 123k posts, and $\sim$5% is held out as the ID validation set. To generate the candidate responses in $\mathcal{D}_{ood}$ with 92k pairwise comparisons, we considered multiple PaLM-XS policies with high temperature, some are pre-trained only, others SFT-ed and others RLHF-ed; the goal was to get a diverse set of summaries.

C.2. AI labeling details

While the ideal approach for evaluating our models would involve human preferences, we resort to the cheaper AI labeling procedure from RLAIF (Lee et al., 2023). We query an instruct fine-tuned PaLM-L (Anil et al., 2023) LLM through Cloud's Vertex AI, prompted to generate preference mimicking human preferences. Specifically, we follow the "Detailed + CoT 0-shot" prompting strategy from RLAIF (Lee et al., 2023), the best one according to their results, involving zero-shot prompting with chain-of-thought (Wei et al., 2022b), a maximum decoding length of 512 tokens and temperature $T = 0.0$ (i.e., greedy decoding). To avoid position bias, we run the AI labeler in the two possible orderings. This strategy was shown to perform similarly to human labellers, with similar inter-agreement. For the corruption experiments, we swap the labels for 25% of the training samples.

C.3. Reward modeling details

C.3.1. General details

The RMs are PaLM-XXS models (Anil et al., 2023). They are first pre-trained, and then supervised fine-tuned on the Reddit TL;DR dataset for 12k steps with a batch size of 128 and the Adafactor (Shazeer & Stern, 2018) optimizer with a learning rate of 10^{-5}. Following the *Baklava* recipe, we actually launch the reward modeling from different checkpoints along this SFT fine-tuning, at steps {8k, 10k, 12k}; taking a too-early checkpoint would drastically reduce RM accuracy, as observed in (Razin et al., 2023). To convert this LLM into a classifier, we plug a linear probed classification layer (the same for all RMs); said differently, even though the featurizers are actually from different SFT checkpoints, they share the same linear probed classification linear layer. As explained in Kumar et al. (2022), it prevents features from moving too much away from their initializations, which facilitates the LMC required for WA.

We train all RMs for 10k steps, a batch size of 128, the Adafactor (Shazeer & Stern, 2018) optimizer, a learning rate sampled in {1e-5,4e-5,1e-4}, and a dropout probability in {0.05, 0.1}. This follows the practical recommandations from (Ramé et al., 2022b) to leverage hyperparameters in a mild range to preserve the LMC. Training for a longer number of steps could help, as it did not alter the LMC in previous works (Ramé et al., 2023).

In practice, for the main experiments with clean labels, we launch 10 reward modelings; when ranked in decreasing accuracy

on $\mathcal{D}_{ood}$, we denote them $\{\phi_i\}_{i=1}^{10}$. Therefore, the RMs named ϕ_1 and ϕ_2 in the different plots are the two best according to their individual performances under distribution shifts. Then, *WARM* $M = 2$ is actually the RM defined per $\frac{\phi_1+\phi_2}{2}$, while ENS $M = 2$ averages their predictions. More generally, *WARM* with M weights is the WA of the M best weights $\{\phi_i\}_{i=1}^{M}$. The main motivation of this weight selection procedure is to remove potentially bad RMs, as validated in Figure 6, in which we consider different permutations across those 10 RMs. As a side note, we speculate that a greedy procedure as in Wortsman et al. (2022a) could further improve performances.

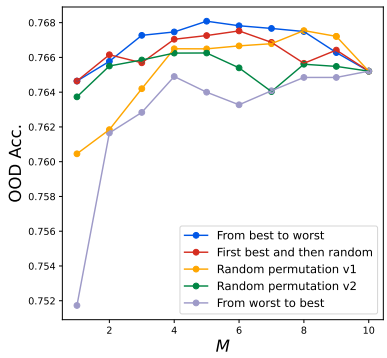


Figure 6: **Analysis of the weight selection procedure**. We plot the accuracy resulting from averaging M weights (out of 10), where these weights are chosen based on various selection procedures. This effectively validates that choosing models from best to worst serves as a reliable heuristic.

C.3.2. BAKLAVA STRATEGY DETAILS

Figure 7 illustrates the Baklava strategy. The different initializations have different level of specialization on the SFT task, which subsequently increases the diversity across fine-tuned RMs.

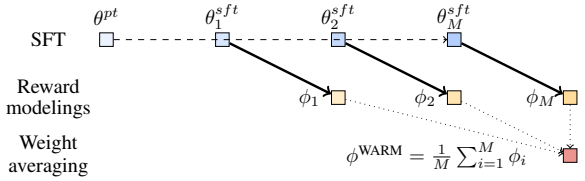


Figure 7: ***Baklava* diversity procedure**. From a pre-trained LLM θ^{pt}, we consider different checkpoints $\{\theta_i^{sft}\}_{i=1}^{M}$ along a single SFT run (dashed arrow ⇢) collected at different number of SFT training steps. Those checkpoints serve as initializations for M RM fine-tunings on the preference dataset (thick solid arrows ⟶) to learn the $\{\phi_i\}_{i=1}^{M}$. Finally, those RMs are weight averaged (dotted arrows ⇢) into the final model ϕ^{WARM}. Following the culinary analogy from model soups (Wortsman et al., 2022a) and model ratatouille (Ramé et al., 2023), we named this method *Baklava* because of its diamond geometric shape.

C.3.3. NEGATIVE RESULT: DIVERSITY THROUGH MOVING AVERAGE

Following stochastic weight average (Izmailov et al., 2018) or moving average (Arpit et al., 2021), we also tried to average checkpoints collected along a single RM fine-tuning. Though interesting because less costly for training, the lower results in Figure 2(a) suggest that the accuracy-diversity trade-off was not favorable: incorporating early checkpoints would compromise individual accuracies, and considering only later checkpoints would not bring the necessary diversity. As a result, we opted to use in *WARM* only the last checkpoint from each RM fine-tuning.

C.4. Reinforcement learning details

Both policy and value models are PaLM-XS (Anil et al., 2023), initialized from the same SFT model. We then generate samples from the policy with temperature $T = 0.9$, batch size of 128, the Adafactor (Shazeer & Stern, 2018) optimizer, a learning rate of 10^{-5} and a policy warmup of 2k steps. We set $\alpha = 0.003$ for the KL regularization in the main experiment without label corruption, and $\alpha = 0.01$ with label corruption. Following Lee et al. (2023), we used a modified version of REINFORCE (Williams, 1992) with a baseline value function for variance reduction: this algorithm is simpler than PPO (Schulman et al., 2017) yet still effective for LLMs.

D. Additional Experiments

D.1. Refined analysis: weight averaging for more *robust* ensembling

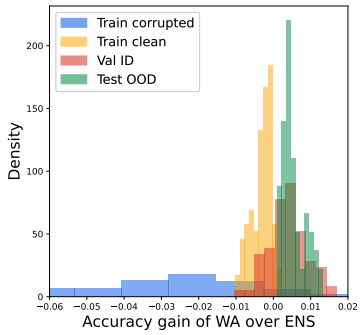


Figure 8: Histograms of the differences in accuracy between WA and ENS on different data subsets.

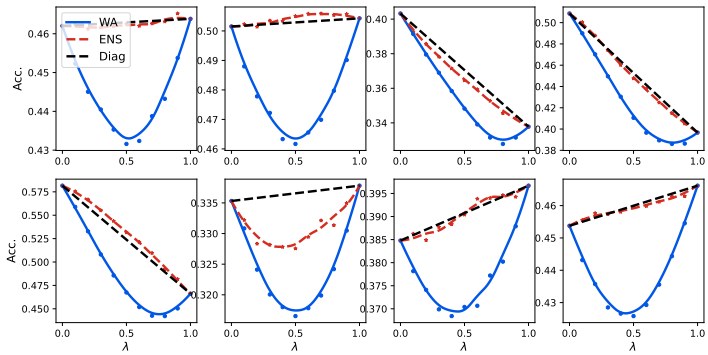


Figure 9: Train (corrupt). More results enriching Figure 3(a) with different pairs of RMs.

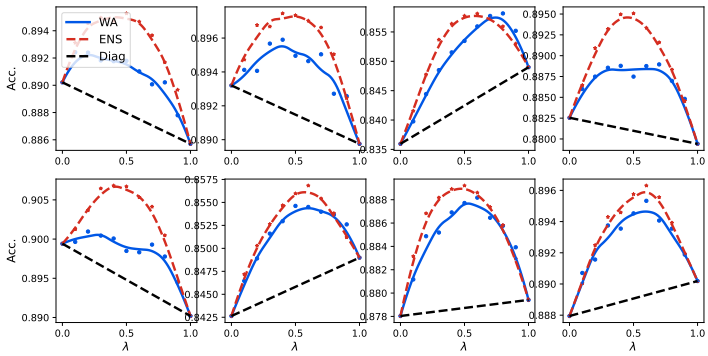


Figure 10: Train (clean). More results enriching Figure 3(b) with different pairs of RMs.

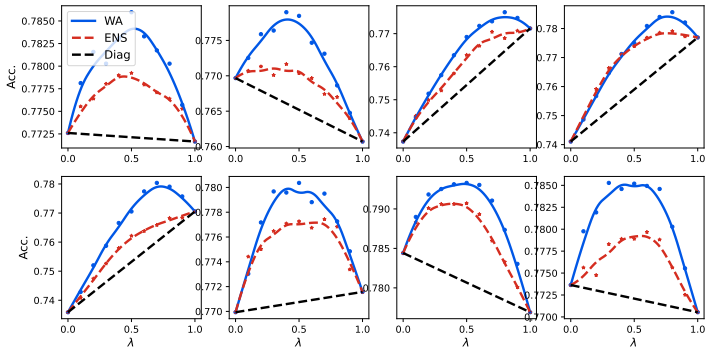


Figure 11: Validation (ID). More results enriching Figure 3(c) with different pairs of RMs.

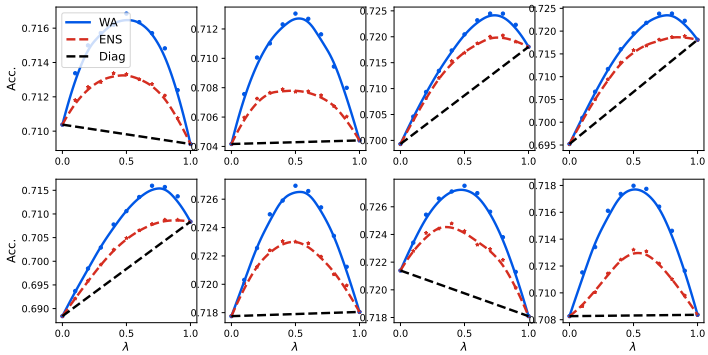


Figure 12: Test (OOD). More results enriching Figure 3(d) with different pairs of RMs.

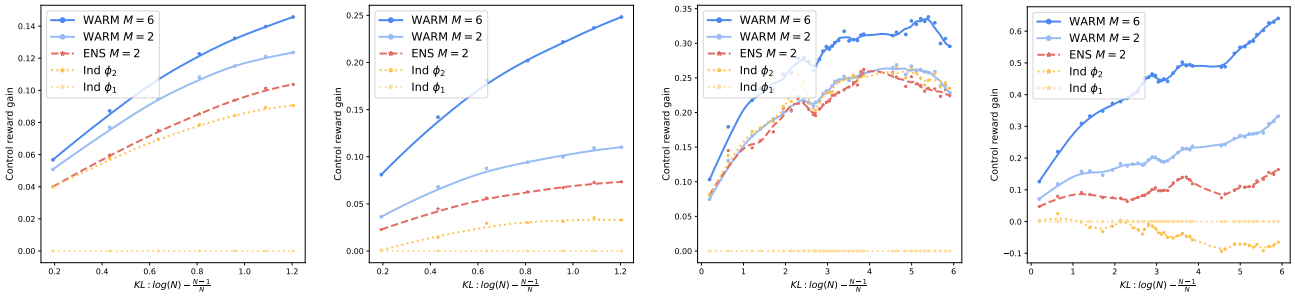
D.2. BoN experiments

D.2.1. Main BoN experiments

Setup. Figures 13 and 14 report the performances of *WARM* for best-of-N (BoN). Given a dataset of D text prompts, for each prompt we generate N summaries from a SFT policy, and then returns the summary with the highest reward according to different RMs. We actually consider two SFT policies; one based on PaLM architecture (Anil et al., 2023) ($N = 8$, $D = 15000$), the other on T5 architecture (Raffel et al., 2020) ($N = 1000$, $D = 1000$). For the x-axis, we plot the KL between the BoN policy and the SFT policy, which can be approximated by $\log(N) - \frac{N-1}{N}$ (Hilton, 2023; Beirami et al., 2024). We consider two setups, without and with 25% label corruption.

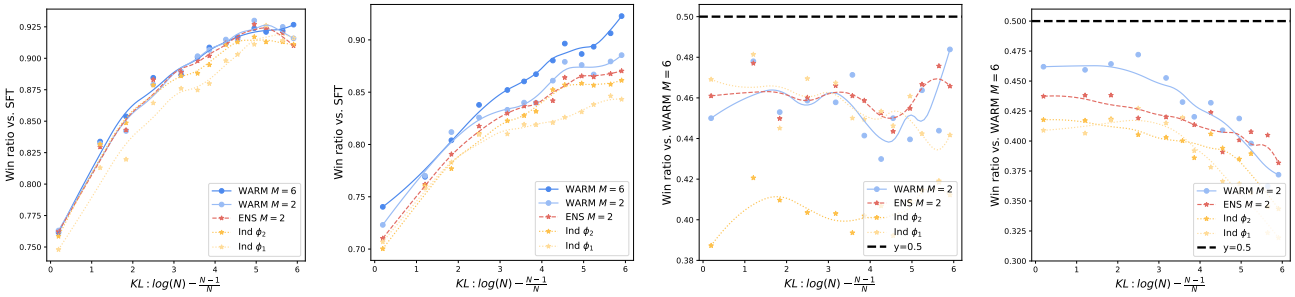
Control reward. Figure 13 shows that, in terms of *pointwise control reward*, *WARM* performs consistently better than ENS (only with $M = 2$ for computational reasons) and the two best individual RMs ϕ_1 and ϕ_2; moreover, the gains get bigger for $M = 6$. As a side note, we also observe that the individual RM ϕ_2 performs better in BoN in Figure 13(c) than ϕ_1 though ϕ_1 was better than ϕ_2 on $\mathcal{D}_{ood}$, highlighting that selecting the appropriate individual RM is not trivial (Eisenstein et al., 2023).

Oracle preference. In Figure 14, we leverage the *pairwise oracle preference* (Lee et al., 2023) metric to validate better performance with *WARM*. We observe in Figures 14(a) and 14(b) that summaries selected with *WARM* have a win rate of up to 92.5% against the random selection of a summary (from SFT). We also see in Figures 14(c) and 14(d) that reciprocally, all selection strategies have a win rate lower than 50% against the summaries selected by *WARM* $M = 6$.



(a) PaLM (clean). (b) PaLM (corrupt). (c) T5 (clean). (d) T5 (corrupt).

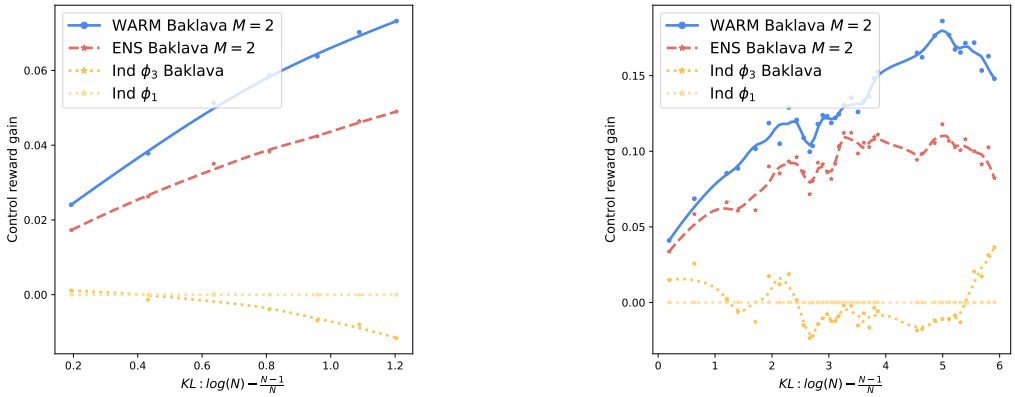
Figure 13: **Control reward for BoN experiments**: clean preference dataset in Figures 13(a) and 13(c) and 25% corruptions in Figures 13(b) and 13(d). We consider two SFT policies to generate candidate summaries: one based on PaLM (Anil et al., 2023), the other on T5 (Raffel et al., 2020). The x-axis is the KL between the BoN and SFT policy; the y-axis is the control reward gains w.r.t. an RM ϕ_1 (the best individual RM on $\mathcal{D}_{ood}$). The blue lines represent *WARM* with M weights: *WARM* outperforms the individual RMs (in yellows) or the ensembling of their predictions (ENS in red).



(a) SFT (clean). (b) SFT (corrupt). (c) *WARM* (clean). (d) *WARM* (corrupt).

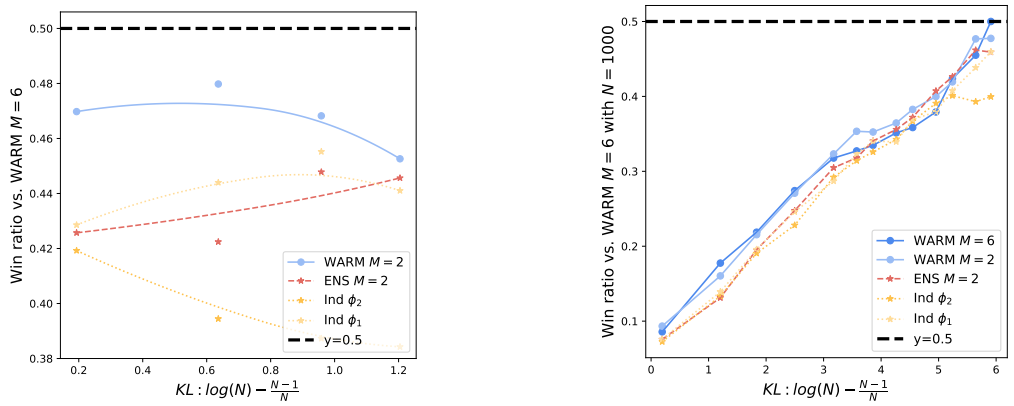
Figure 14: **Oracle preference metric for BoN experiments on T5 generations**: clean preference dataset in Figures 14(a) and 14(c) and 25% corruptions in Figures 14(b) and 14(d). We plot the win rates for different values of N vs. two reference strategies: SFT (i.e., random selection or equivalently BoN with $N = 1$), or selecting the best summary according to *WARM* $M = 6$. All strategies beat the SFT reference (they are all above 50% win rate), but that none beats *WARM* $M = 6$.

D.2.2. Additional BoN experiments



(a) *Baklava* with PaLM. (b) *Baklava* with T5.

Figure 15: **Control reward for BoN experiments** (clean setup) with *Baklava* when the two fine-tunings ϕ_1 and ϕ_3 have different featurizer initializations, collected respectively at steps 12k and 8k from a shared SFT.



(a) PaLM. (b) T5 vs. *WARM* with $N = 1000$.

Figure 16: **Oracle preference metric for BoN experiments** (clean setup). Figure 16(a) confirms Figure 14(c) but on generations from PaLM SFT. Figure 16(b) shows win rates for BoN on T5 generations for *WARM* with $M = 6$ and always $N = 1000$ for BoN vs. other RMs with $1 \leq N \leq 1000$. We validate that BoN limits reward hacking compared to RL, as performances get better when increasing N.

D.3. RL experiments

D.3.1. Experiments with corrupted preference dataset

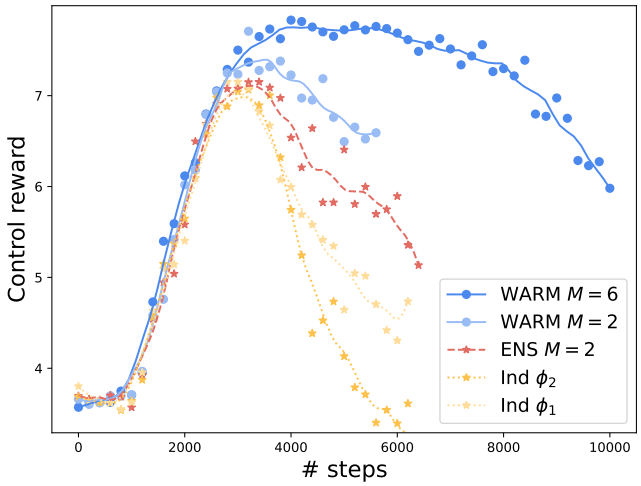
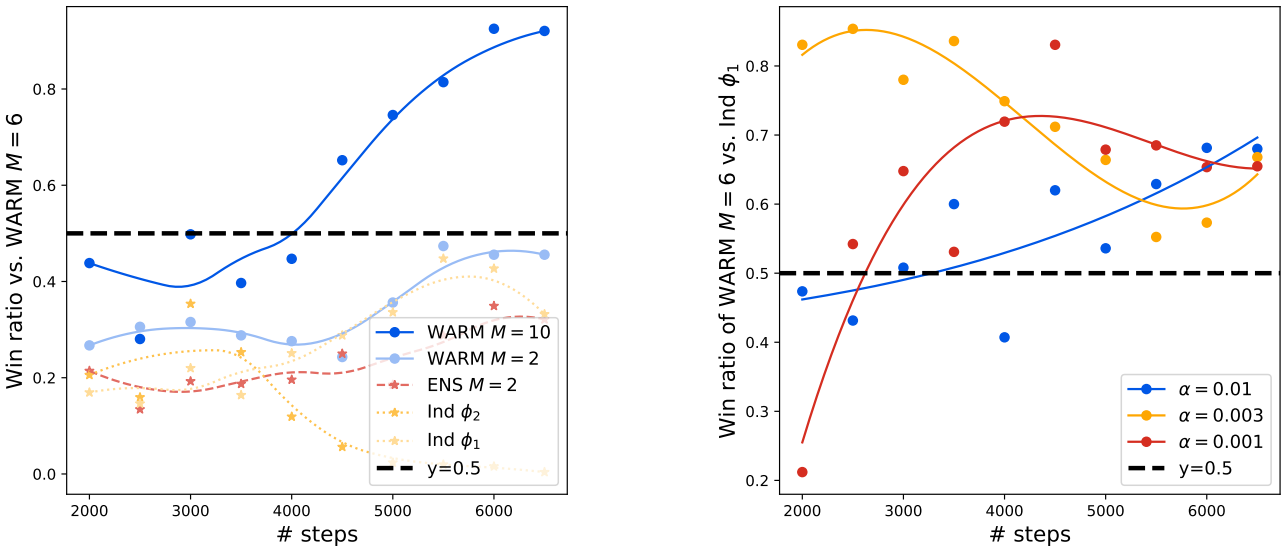


Figure 17: **RL experiments**. Same as Figure 1(b) but with 25% corruption in the preference dataset.

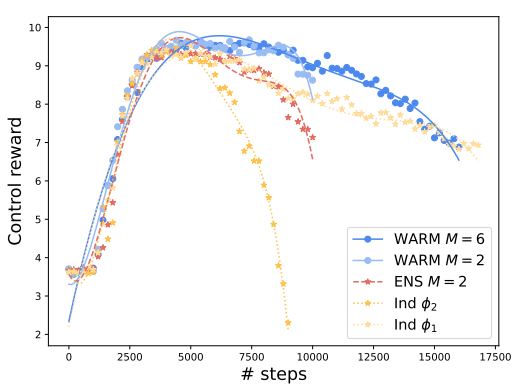
D.3.2. Experiments with clean preference dataset



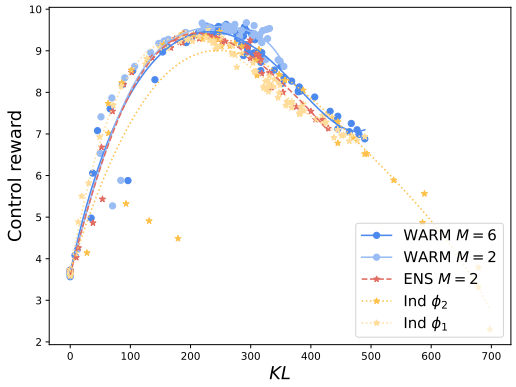
(a) *WARM* $M = 6$.

(b) Impact of α.

Figure 18: **Oracle preference metric for RL experiments** at fixed number of training steps (clean setup). Figure 18(a) plots the win rate of the policy with *WARM* $M = 6$ vs. the other policies, all at the same number of training steps. Figure 18(b) shows the win rate of *WARM* $M = 6$ against the policy trained with a single RM ϕ_1 (the best according to OOD accuracy) along training for different values of α controlling the KL regularization strength.

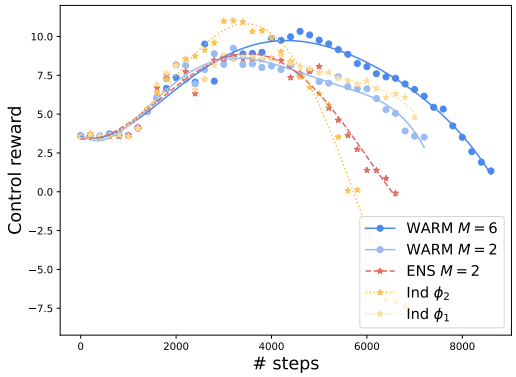


(a) Control reward vs. training steps.

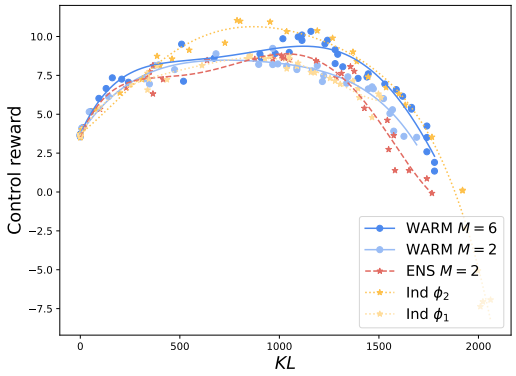


(b) Control reward vs. KL.

Figure 19: **Control reward for RL experiments** with $\alpha = 0.01$ (clean setup).



(a) Control reward vs. training steps.



(b) Control reward vs. KL.

Figure 20: **Control reward for RL experiments** with $\alpha = 0.001$ (clean setup).

D.4. Distillation experiments

In Figure 21 we reproduce the distillation setup from Gao et al. (2023), where the control PaLM-XS RM generates the labels to train PaLM-XXS RMs. As a side note, we observed that distillation changes the diversity across fine-tuned RMs, thus potentially altering the significance of the distillation setup, motivating us in exploring the more realistic RLAIF setup.

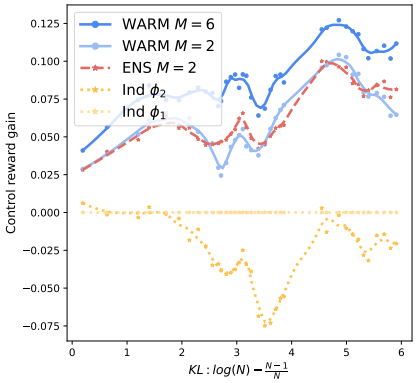
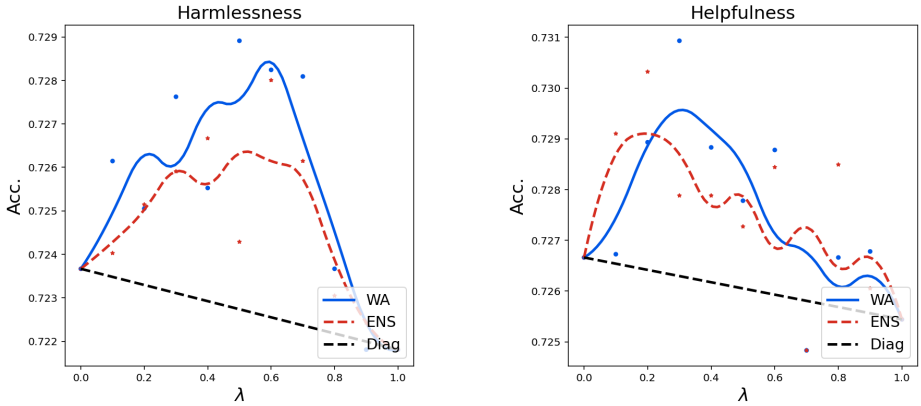


Figure 21: **BoN experiment in the distillation setup from Gao et al. (2023).** The labels in the preference dataset are given by the control RM, the same RM which gives the y-axis. The candidate summaries are generated by a SFT with the T5 architecture (Raffel et al., 2020). The blue lines represent *WARM* with M weights: *WARM* performs higher than the individual RMs (in yellows) or when ensembling their predictions (ENS in red).

D.5. Experiments on Anthropic HH datasets



(a) Harmlessness dataset.

(b) Helpfulness dataset.

Figure 22: **Anthropic HH datasets (Bai et al., 2022a).** We consider ϕ_1 and ϕ_2 trained on the concatenation of the harmlessness and the helpfulness training datasets, combine them either by weight interpolation (WA) or by prediction ensembling (ENS), and report their performances. We observe that the linear mode connectivity Observation 1 holds as the curves are above the diagonal (Diag), both when the evaluation occurs on the harmlessness validation dataset or on the helpfulness validation dataset. We notably uncover an interesting phenomenon: when interpolating between two RMs fine-tuned respectively on harmlessness only, and the other on helpfulness only, we can obtain a WARM that encapsulates both criteria (in a multitask fashion) by weight averaging. This opens a promising direction for future research on multi-objective alignment.