Rewarded soups: towards Pareto-optimality by interpolating weights fine-tuned on diverse rewards

Alexandre Rame[1,*] **Guillaume Couairon**[1,2,†] **Corentin Dancette**[1,†]
Jean-Baptiste Gaya[1,2,†] **Mustafa Shukor**[1,†] **Laure Soulier**[1], **Matthieu Cord**[1,3]
[1]Sorbonne Université, CNRS, ISIR, Paris, France [2]Meta AI [3]Valeo.ai

Abstract

Foundation models are first pre-trained on vast unsupervised datasets and then fine-tuned on labeled data. Reinforcement learning, notably from human feedback (RLHF), can further align the network with the intended usage. Yet the imperfections in the proxy reward may hinder the training and lead to suboptimal results; the diversity of objectives in real-world tasks and human opinions exacerbate the issue. This paper proposes embracing the heterogeneity of diverse rewards by following a multi-policy strategy. Rather than focusing on a single a priori reward, we aim for Pareto-optimal generalization across the entire space of preferences. To this end, we propose *rewarded soup*, first specializing multiple networks independently (one for each proxy reward) and then interpolating their weights linearly. This succeeds empirically because we show that the weights remain linearly connected when fine-tuned on diverse rewards from a shared pre-trained initialization. We demonstrate the effectiveness of our approach for text-to-text (summarization, Q&A, helpful assistant, review), text-image (image captioning, text-to-image generation, visual grounding), and control (locomotion) tasks. We hope to enhance the alignment of deep models, and how they interact with the world in all its diversity.

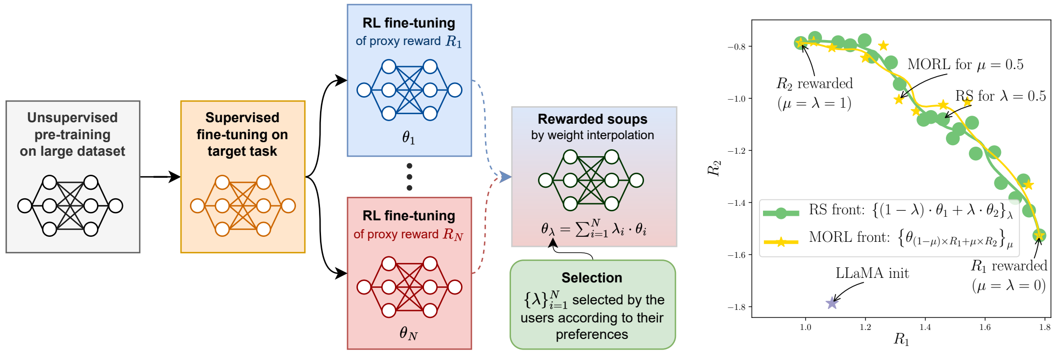
1 Introduction

Foundation models [1] have emerged as the standard paradigm to learn neural networks' weights. They are typically first pre-trained through self-supervision [2, 3, 4, 5] and then fine-tuned [6, 7] via supervised learning [8]. Yet, collecting labels is expensive, and thus supervision may not cover all possibilities and fail to perfectly align [9, 10, 11] the trained network with the intended applications. Recent works [12, 13, 14] showed that deep reinforcement learning (DRL) helps by learning from various types of rewards. A prominent example is reinforcement learning from human feedback (RLHF) [12, 15, 16, 17], which appears as the current go-to strategy to refine large language models (LLMs) into powerful conversational agents such as ChatGPT [13, 18]. After pre-training on next token prediction [19] using Web data, the LLMs are fine-tuned to follow instructions [20, 21, 22] before reward maximization. This RL strategy enhances alignment by evaluating the entire generated sentence instead of each token independently, handling the diversity of correct answers and allowing for negative feedback [23]. Similar strategies have been useful in computer vision (CV) [14, 24], for instance to integrate human aesthetics into image generation [25, 26, 27].

Diversity of proxy rewards. RL is usually seen as more challenging than supervised training [28], notably because the real reward—ideally reflecting the users' preferences—is often not specified at training time. Proxy rewards are therefore developed to guide the learning, either as hand-engineered metrics [29, 30, 31] or more recently in RLHF as models trained to reflect human preferences

[*]Project lead, main contributor. Correspondence to alexandre.rame.cl@gmail.com.
[†]Equal technical contribution.

Preprint. Under review.



(a) Illustration of our proposed rewarded soup (RS). (b) LLaMA RLHF for summarization.

Figure 1: Figure 1(a) details the different steps in rewarded soup. After unsupervised pre-training and supervised fine-tuning, we launch N independent RL fine-tunings on the proxy rewards $\{R_i\}_{i=1}^{N}$. Then we combine the trained networks by interpolation in the weight space. The final weights are adapted at test time by selecting the coefficient λ. Figure 1(b) shows our results (extended in Figure 2(a)) with LLaMA-7b [45] instruct fine-tuned on Alpaca [22], when RL fine-tuning for news summarization [12] with $N = 2$ reward models assessing diverse preferences of summaries. With only two trainings (R_1 and R_2 rewarded on Figure 1(b)), the λ-interpolation ($0 \leq \lambda \leq 1$) reveals the green front of Pareto-optimal solutions, i.e., that cannot be improved for one reward without sacrificing the other. RS matches the costly yellow front of multi-objective (MORL) [46, 47] requiring multiple trainings on different linear weightings over the rewards $(1-\mu) \times R_1 + \mu \times R_2$ with $0 \leq \mu \leq 1$.

[15, 32, 33]. Nonetheless, designing reliable proxy rewards for evaluation is difficult. This *reward misspecification* [9, 34] between the proxy reward and the users' actual rewards can lead to unforeseen consequences [35]. Moreover, the diversity of objectives in real-world applications complicates the challenge. In particular, human opinions can vary significantly [36, 37, 38] on subjects such as aesthetics [39], politics or fairness [40]. Humans have also different expectations from machines: for example, while [41, 42] stressed aligning LLMs towards helpful, honest, and harmless [43] feedback, others' interests are to make LLMs mostly engaging and enjoyable [44]. Even hand-engineered metrics can be in tension: generating shorter descriptions with higher precision can increase the BLEU [29] score but decrease the ROUGE [30] score due to reduced recall.

Towards multi-policy strategies. Considering these challenges, it may not be feasible to develop a single model simultaneously aligned with everyone's preferences [13]. Current strategies tend to align towards a consensus-based user [48, 49], inherently prioritizing certain values over others, potentially resulting in unfair representations of marginalized groups [50]. Moreover, these trade-offs [51] are decided a priori before training, shifting the responsibility to the engineers and reducing transparency and explainability [52]. These limitations, further discussed in Appendix A.1, highlight a key limitation of single-policy alignment strategies; their inability to handle the diversity of human preferences. Yet, "human-aligned artificial intelligence is a multi-objective problem" [53]. Thus, we draw inspiration from the multi-objective reinforcement learning (MORL) literature [46, 47, 54, 55, 56, 57] and notably [52] arguing that tackling diverse rewards requires shifting from single-policy to multi-policy approaches. As optimality depends on the relative preferences across those rewards, the goal is not to learn a single network but rather a **set of Pareto-optimal networks** [58].

In this paper, we propose **rewarded soup** (RS), an efficient and flexible multi-policy strategy to fine-tune any foundation model. As shown in Figure 1(a), we first use RL to learn one network for each proxy reward; then, we combine these expert networks according to user preferences. This a posteriori selection allows for better-informed trade-offs, improved transparency and increased fairness [52, 59]. The method to combine those networks is our main contribution: we do this through **linear interpolation in the weight space**, despite the non-linearities in the network. This is in line with recent findings on linear mode connectivity (LMC) [60, 61]: weights fine-tuned from a shared pre-trained initialization remain linearly connected and thus can be interpolated. This LMC inspired a plethora of weight interpolation (WI) strategies [62, 63, 64, 65, 66, 67], discussed in Section 4. Unlike previous works, which focused on supervised learning, we explore LMC in RL, in a challenging setup where each training run uses a different reward. Perhaps surprisingly, we show that we can trade off the capabilities of multiple weights in a single final model, thus without any computational overhead. This enables the creation of custom weights for any preference over the diverse rewards.

- We propose a new practical strategy named rewarded soup for fine-tuning foundation models with diverse rewards. It defines a continuous set of (close to) Pareto-optimal solutions by weight interpolation, approximating more costly multi-policy strategies.
- We analyze the linear mode connectivity between weights fine-tuned on diverse rewards.
- We validate that our strategy mitigates reward misspecification.

In Section 3, we demonstrate the consistent effectiveness of rewarded soup across a variety of tasks: RLHF fine-tuning of LLaMA, multimodal tasks such as image captioning, text-to-image generation with diffusion models or visual grounding, as well as locomotion tasks.

2 Rewarded soups

2.1 RL fine-tuning with diverse rewards

We consider a deep neural network f of a fixed non-linear architecture (e.g., with batch normalization [68], ReLU layers [69] or self-attention [70]). It defines a policy by mapping inputs x to $f(x, \theta)$ when parametrized by θ. For a reward $\hat{R}$ (evaluating the correctness of the prediction according to some preferences) and a test distribution T of deployment, our goal is to maximize $\int_{x \in T} \hat{R}(f(x, \theta))$. For example, with f a LLM, x would be textual prompts, $\hat{R}$ would evaluate if the generated text is harmless [43], and T would be the distribution of users' prompts. Learning the weights θ is now commonly a three-step process: unsupervised pre-training, supervised fine-tuning, and reward optimization. Yet $\hat{R}$ is usually not specified before test time, meaning we can only optimize a proxy reward R during training. This **reward misspecification** between R and $\hat{R}$ may hinder the alignment of the network with $\hat{R}$. Moreover, the **diversity of human preferences** complicates the design of R.

Rather than optimizing one single proxy reward, our paper's first key idea is to consider a family of N diverse proxy rewards $\{R_i\}_{i=1}^N$. Each of these rewards evaluates the prediction according to different (potentially conflicting) criteria. The goal then becomes obtaining a coverage set of policies that trade-off between these rewards. To this end, we first introduce the costly MORL baseline. Its inefficiency motivates our rewarded soups, which leverages our second key idea: weight interpolation.

MORL baseline. The standard MORL scalarization strategy [46, 47] linearizes the problem by interpolating the proxy rewards using M different weightings. Specifically, during the *training phase*, M trainings are launched, with the j-th optimizing the reward $\sum_{i=1}^N \mu_i^j R_i$, where $\forall j \in \{1, \ldots, M\}, \{\mu_i^j\}_{i=1}^N \in \Delta_N$ the N-simplex s.t. $\sum_{i=1}^N \mu_i^j = 1$ and $0 \leq \mu_i^j \leq 1$. Then, during the *selection phase*, the user's reward $\hat{R}$ becomes known and the j-th policy that maximizes $\hat{R}$ on some validation dataset is selected. We typically expect to select j such that $\sum_{i=1}^N \mu_i^j R_i \approx \hat{R}$ linearly approximates the user's reward. Finally, this j-th weight is used during the *inference phase* on test samples. Yet, a critical issue is that "minor [preference] variations may result in significant changes in the solution" [71]. Thus, a high level of granularity in the mesh of Δ_N is necessary. This requires explicitly maintaining a large set of $M \gg N$ networks, practically one for each possible preference. Ultimately, this MORL strategy is unscalable in deep learning due to the **computational, memory, and engineering costs** involved (see further discussion in Appendix A.2).

Rewarded soup (RS). In this paper, we draw inspiration from the weight interpolation literature. The idea is to learn expert weights and interpolate them linearly to combine their abilities. Specifically, we propose RS, illustrated in Figure 1(a) and whose recipe is described below. RS alleviates MORL's scaling issue as it requires only $M = N$ trainings while being flexible and transparent.

1. During the *training phase*, we optimize a set of N expert weights $\{\theta_i\}_{i=1}^N$, each corresponding to one of the N proxy rewards $\{R_i\}_{i=1}^N$, and all from a shared pre-trained initialization.
2. For the *selection phase*, we linearly interpolate those weights to define a continuous set of rewarded soups policies: $\{\sum_{i=1}^N \lambda_i \cdot \theta_i\}_{\{\lambda_i\}_{i=1}^N \in \Delta_N}$. Practically, we uniformly sample M interpolating coefficients $\{\{\lambda_i^j\}_{i=1}^N\}_{j=1}^M$ from the N-simplex Δ_N and select the j-th that maximizes the user's reward $\hat{R}$ on validation samples, i.e., $\mathrm{argmax}_{j=1}^M \hat{R}\left(\sum_{i=1}^N \lambda_i^j \theta_i\right)$.
3. For the *inference phase*, we predict using the network f parameterized by $\sum_{i=1}^N \lambda_i^j \theta_i$.

While MORL interpolates the rewards, RS interpolates the weights. This is a considerable advantage as the appropriate weighting λ, which depends on the desired trade-off, can be selected *a posteriori*; the selection is achieved without additional training, only via inference on some samples. In the next Section 2.2 we explicitly state the Hypotheses 1 and 2 underlying in RS. These are considered *Working Hypotheses* as they enabled the development of our RS strategy. Their empirical verification will be the main motivation for our experiments on various tasks in Section 3.

2.2 Exploring the properties of the rewarded soups set of solutions

2.2.1 Linear mode connectivity of weights fine-tuned on diverse rewards

We consider $\{\theta_i\}_{i=1}^N$ fine-tuned on $\{R_i\}_{i=1}^N$ from a shared pre-trained initialization. Previous works [60, 61, 62, 67] defined linear mode connectivity (LMC) w.r.t. a single performance measure (e.g., accuracy or loss) in supervised learning. We extend this notion in RL with N rewards, and define that the LMC holds if all rewards for the interpolated weights exceed the interpolated rewards. It follows that the LMC condition which underpins RS's viability is the Hypothesis 1 below.

Working Hypothesis 1 (LMC)**.** $\forall \{\lambda_i\}_i \in \Delta_N$ *and* $k \in \{1, \dots, N\}$, $R_k(\sum_i \lambda_i \cdot \theta_i) \geq \sum_i \lambda_i R_k(\theta_i)$.

2.2.2 Pareto optimality of rewarded soups

The Pareto front (PF) is the set of undominated weights, for which no other weights can improve a reward without sacrificing another, i.e., $\{\theta \mid \nexists \theta' \in \Theta \text{ s.t. } \{R_i(\theta')\}_{i=1}^N >_N \{R_i(\theta)\}_{i=1}^N\}$ where $>_N$ is the dominance relation in $\mathcal{R}^N$. In practice, we only need to retain one policy for each possible value vector, i.e., a Pareto coverage set (PCS). We now introduce the key Hypothesis 2.

Working Hypothesis 2 (Pareto optimality)**.** *The set* $\{\sum_i \lambda_i \cdot \theta_i | \{\lambda_i\}_i \in \Delta_N\}$ *is a PCS of* $\{R_i\}_i$.

Hypothesis 2 holds if the rewarded soups solutions, uncovered by interpolation, are Pareto-optimal. Overall, we empirically validate Hypotheses 1 and 2 in Section 3, yet also report a few limitations in Appendix (Figures 9(a) and 10) and research directions to fix them. Moreover, we theoretically prove in Appendix B.2 they approximately hold when rewards are replaced by their second-order Taylor expansion with co-diagonalizable Hessians, a simplified setup justifiable when weights remain close.

Remark 1. *Hypotheses 1 and 2 rely on a good pre-trained initialization, making RS particularly well-suited to fine-tune foundation models. This is because pre-training prevents the weights from diverging during training [61]. When the weights remain close, we can theoretically justify Hypotheses 1 and 2 (see Appendix B.2) and, more broadly, demonstrate that WI approximates ensembling [72, 73] (see Lemma 4). In contrast, the LMC does not hold when training from scratch [61]. Neuron permutations strategies [74, 75] tried to enforce connectivity by aligning the weights, though (so far) with moderate empirical results: their complementarity with RS is a promising research avenue.*

Remark 2. *Pareto-optimality in Hypothesis 2 is defined w.r.t. a set of possible weights* Θ*. Yet, in full generality, improvements in initialization, RL algorithms, data, or specific hyperparameters could enhance performances. In other words, for real-world applications, the true PF is unknown and needs to be defined w.r.t. a training procedure. In this case,* Θ *represents the set of weights attainable by fine-tuning within a shared procedure. As such, in Section 3 we analyze Hypothesis 2 by comparing the fronts obtained by RS and scalarized MORL while keeping everything else constant.*

2.2.3 Consequences of Pareto optimality if the user's reward is linear in the proxy rewards

Lemma 1 (Reduced reward misspecification in the linear case)**.** *If Hypothesis 2 holds, and for linear reward* $\hat{R} = \sum_i \hat{\mu}_i R_i$ *with* $\{\hat{\mu}_i\}_i \in \Delta_N$*, then* $\exists \{\lambda_i\}_i \in \Delta_N$ *such that* $\sum_i \lambda_i \cdot \theta_i$ *is optimal for* $\hat{R}$.

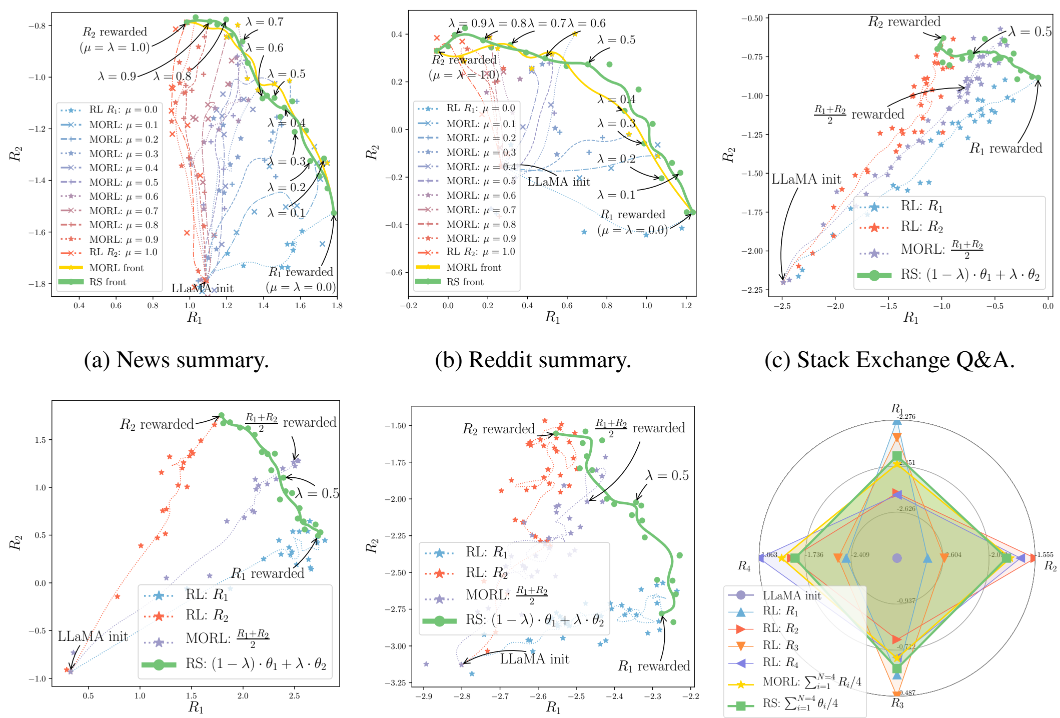
The proof outlined in Appendix B.1 directly follows the definition of Pareto optimality. In simpler terms, Lemma 1 implies that if Hypothesis 2 is true, then RS can mitigate reward misspecification. For any preference $\hat{\mu}$, there exists a λ such that the λ-interpolation over weights maximizes the $\hat{\mu}$-interpolation over rewards. In practice, as we will see in Figure 4(a), we can set $\lambda = \hat{\mu}$, or cross-validate λ on other samples. Yet, this theoretically holds only for $\hat{R}$ linear over the proxy rewards. This follows the *linear utility functions* setup from the MORL literature [57], whose limitations [71] are discussed in Section 5. This motivates having sufficiently rich and diverse proxy rewards to capture the essential aspects of all possible users' rewards. Despite the lack of theoretical guarantees, we will show in Figures 4(b) and 8 that weight interpolation improves results even for non-linear $\hat{R}$.

3 Experiments

In this section we implement RS across a variety of standard learning tasks: text-to-text generation, image captioning, image generation, visual grounding, and locomotion. We use either model or statistical rewards. We follow a systematic procedure. First, we independently optimize diverse rewards on training samples. For all tasks, we employ the default architecture, hyperparameters and RL algorithm; the only variation being the reward used across runs. Second, we evaluate the rewards on the test samples: the results are visually represented in series of plots. Third, we verify Hypothesis 1 by examining whether RS's rewards exceed the interpolated rewards. Lastly, as the true Pareto front is unknown in real-world applications, we present empirical support for Hypothesis 2 by comparing the front defined by RS (sliding λ between 0 and 1) to the MORL's solutions optimizing the μ-weighted rewards for $0 \leq \mu \leq 1$ (sometimes only $\mu = 0.5$ for computational reasons).

3.1 Text-to-text: LLaMA with diverse RLHFs

Given the significance of RLHF to train LLMs, we begin our experiments with text-to-text generation tasks. Our pre-trained network is LLaMA-7b [45], instruction fine-tuned [20, 77] on Alpaca [22]. For RL training with PPO [78], we employ the trl package [79] and the setup from [80] with low-rank adapters (LoRA) [81] for efficiency. We consider the following tasks: summarization [12, 17] on two datasets (Reuter news [82] in Figures 1(b) and 2(a) and Reddit posts [83] in Figure 2(b)), answering Stack Exchange questions [84] in Figure 2(c), movie review generation in Figure 2(d), and helpfulness as a conversational assistant [41] in Figures 2(e) and 2(f). To evaluate the generation in the absence of supervision, we utilized $N = 2$ different reward models (RMs) for each task, except in Figure 2(f)



(a) News summary.

(b) Reddit summary.

(c) Stack Exchange Q&A.

(d) Useful movie review.

(e) Helpful assistant.

(f) Helpful assistant: spider map.

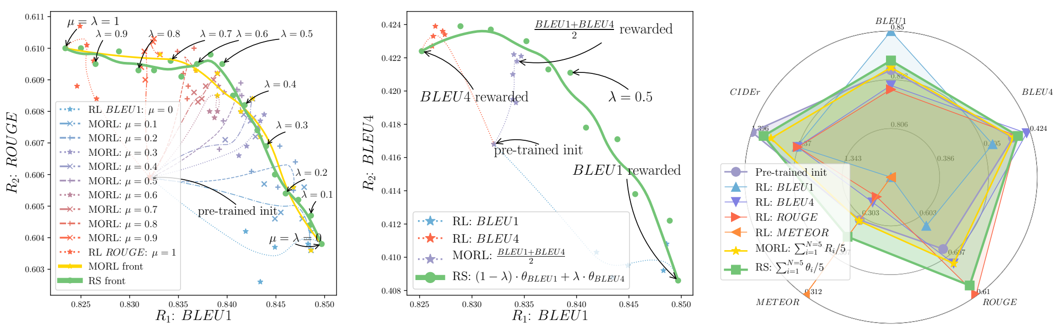
Figure 2: RLHF results in NLP with LLaMA-7b [45] and reward models R_i from HuggingFace [76]. The blue line reports checkpoints' results along the training trajectory of θ_1 rewarding R_1, the red line θ_2 rewarding R_2, and the purple line the MORL rewarding $\frac{R_1+R_2}{2}$. Our rewarded soup (RS) linearly interpolates between the weights θ_1 and θ_2; sliding the interpolation coefficient λ from 0 to 1 reveals the green solid front of rewarded soups solutions. In Figures 2(a) and 2(b), we additionally show the multiple MORL runs rewarding $(1 - \mu) \times R_1 + \mu \times R_2$ with preferences $0 \leq \mu \leq 1$. It reveals a similar yellow front, yet more costly. In Figure 2(f), we uniformly ($\lambda_i = \frac{1}{4}$) average the weights fine-tuned for the assistant task on $N = 4$ reward models.

where $N = 4$. These RMs were trained on human preferences datasets [15] and all open-sourced on HuggingFace [76]. For example in summarization, R_1 follows the "Summarize from Human Feedback" paper [12], while R_2 leverages "contrast candidate generation" [85]. For other tasks, we rely on diverse RMs from OpenAssistant [86]; though they all assess if the answer is adequate, they differ by their architectures and procedures. Table 1 further details the experiments.

The results are reported in Figure 2. The green front, defined by RS between the two weights specialized on R_1 and R_2, is above the straight line connecting those two points, validating Hypothesis 1. Second, the front passes through the point obtained by MORL fine-tuning on the average of the two rewards, supporting Hypothesis 2. Moreover, when comparing both full fronts, they have qualitatively the same shape; quantitatively in hypervolume [87] (lower is better, the area over the curve w.r.t. an optimal point), RS's hypervolume is 0.367 vs. 0.340 for MORL in Figure 2(a), while it is 1.176 vs. 1.186 in Figure 2(b). Finally, in Figure 2(f), we use $N = 4$ RMs for the assistant task and uniformly average the $N = 4$ weights, confirming that RS can scale and trade-off between more rewards.

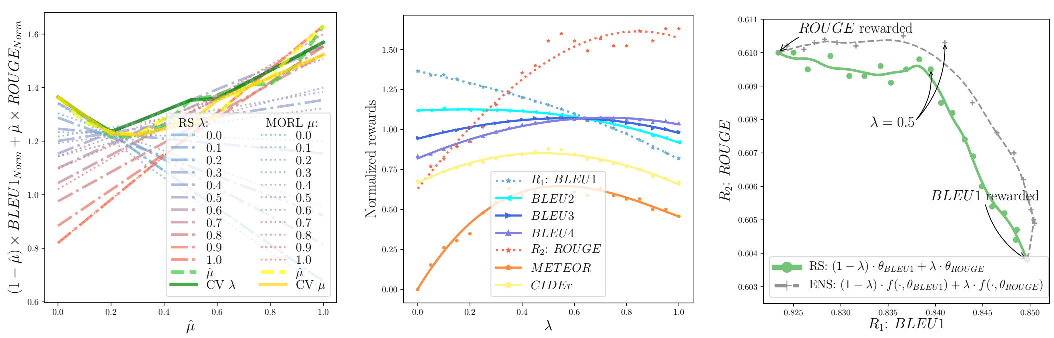
3.2 Image-to-text: captioning with diverse statistical rewards

RL training is also effective for multimodal tasks [14], for example in image captioning [24] where the task is to generate textual descriptions of images. Precisely evaluating the quality of a prediction w.r.t. a set of human-written captions is a challenging task, thus the literature relies on various hand-engineered, non-differentiable metrics: e.g., the precision-focused BLEU [29], the recall-focused



(a) BLEU1 and ROUGE. (b) BLEU1 and BLEU4. (c) Captioning: spider map.

Figure 3: Results in image captioning on COCO [88]. As rewards R_1 (blue stars every epoch) and R_2 (red stars), we consider standard statistical metrics: BLEU1 (1-gram overlap), BLEU4 (4-grams overlap), ROUGE, METEOR and CIDEr. Figure 3(a) include the MORL training trajectories optimizing $(1-\mu) \times BLEU1 + \mu \times ROUGE$, uncovering a yellow front similar to RS's green front. In Figure 3(c), RS uniformly averages the 5 weights (one for each reward), resulting in the largest area and the best trade-off between the 5 rewards.



(a) Analysis of Lemma 1. (b) Performances w.r.t. λ. (c) RS vs. prediction ensembling.

Figure 4: Refined results in captioning with $R_1 = BLEU1$ and $R_2 = ROUGE$. Figure 4(a) empirically validates Lemma 1 by reporting results of RS (for varying λ) and of MORL (for varying μ) for varying user's preference $\hat{\mu}$. In Figure 4(b), all rewards are used for evaluation as a function of the interpolating coefficient. In Figure 4(c), we report the front of the costly ensembling [72, 73] of predictions (rather than of weights).

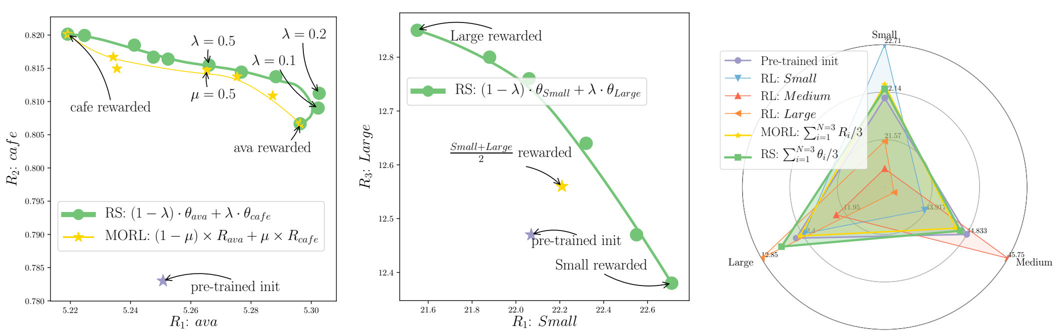
ROUGE [30], METEOR [89] handling synonyms and CIDEr [31] using TF-IDF. As these metrics are proxies for human preferences, good trade-offs are desirable. We conduct our experiments on COCO [88], with an ExpansionNetv2 [90] network and a Swin Transformer [91] visual encoder, initialized from the state-of-the-art weights of [90] optimized on CIDEr. We then utilize the code of [90] and their self-critical [24] procedure (a variant of REINFORCE [92]) to reward the network on BLEU1, BLEU4, ROUGE or METEOR. More details and results can be found in Appendix D.

We observe in Figure 3 that tuning solely BLEU1 sacrifices some points on ROUGE or BLEU4. Yet interpolating between θ_1 and θ_2 uncovers a convex set of solutions approximating the ones obtained through scalarization of the rewards in MORL. When comparing both full fronts in Figure 3(a), they qualitatively have the same shape, and quantitatively the same hypervolume [87] of 0.140. One of the strengths of RS is its ability to scale to any number of rewards. In Figure 3(c), we uniformly ($\lambda_i = \frac{1}{5}$) average $N = 5$ weights fine-tuned independently. It improves upon the initialization [90] and current state-of-the-art on all metrics, except for CIDEr, on which [90] was explicitly optimized.

Figure 4 refines our analysis of RS. In Figures 4(a) and 4(b), rewards are normalized to 1 for the initialization and 0 for the worst model. Figure 4(a) validates Lemma 1: for any linear preference $\hat{\mu}$ over the proxy rewards, there exists an optimal solution in the set described by RS. Two empirical strategies to set the value of λ are close to optimal: selecting $\lambda = \hat{\mu}$ if $\hat{\mu}$ is known, or cross-validating (CV) λ if a different data split [93] is available. Moreover, Figure 4(b) (and Figure 8 in Appendix D) investigate all metrics as evaluation. Excluding results' variance, we observe monotonicity in both training rewards, linear in BLEU1 and quadratic in ROUGE. For other evaluation rewards that **cannot be linearly expressed** over the training rewards, the curves' concavity shows that RS consistently improves the endpoints, thereby mitigating reward misspecification. The optimal λ depends on the similarity between the evaluation and training rewards: e.g., best BLEU2 are with small λ. Lastly, as per [94] and Lemma 4, Figure 4(c) suggests that RS succeeds because WI approximates *deep ensembling* [72, 73], interpolating the predictions rather than the weights. Actually, ensembling performs better, but it cannot be fairly compared as its inference cost is doubled.

3.3 Text-to-image: diffusion models with diverse RLHFs

Beyond text generation, we now apply RS to align text-to-image generation with human feedbacks [25, 26, 33]. Our network is a diffusion model [95] with 2.2B parameters, pre-trained on an internal dataset of 300M images; it reaches similar quality as Stable Diffusion [96], which was not used for copyright reasons. To represent the subjectivity of human aesthetics, we employ $N = 2$ open-source reward models: *ava*, trained on the AVA dataset [97], and *cafe*, trained on a mix of real-life and manga images. We first generate 10000 images; then, for each reward, we remove half of the images with the lowest reward's score, and fine-tune 10% of the parameters [98] on the reward-weighted negative log-likelihood [25]. More details and generations for qualitative visual inspection are in Appendix E.



(a) Image generation: *ava* and *cafe*.
(b) VG: Small and Large.
(c) VG: spider map.

Figure 5: Figure 5(a) reports our RLHF experiments on text-to-image generation with diffusion models. From the pre-trained initialization, we learn θ_{ava} and θ_{cafe} by optimizing the two reward models *ava* and *cafe*. Interpolation between them reveals the green Pareto-optimal front, above the yellow MORL front. Figures 5(b) and 5(c) report our results in visual grounding (VG) on RefCOCO+ [99], where we optimize to predict boxes with IoU>0.5 w.r.t. the ground-truth, for objects of either small, medium or large size.

The results displayed in Figure 5(a) validate Hypothesis 1, as the front described by RS when sliding λ from 0 and 1 is convex. Moreover, RS gives a better front than MORL, validating Hypothesis 2. Interestingly, the *ava* reward model seems to be more general-purpose than *cafe*, as RL training on *ava* also enhances the scores of *cafe*. In contrast, the model θ_{cafe} performs poorly in terms of *ava* in Figure 5(a). Nonetheless, RS with $(1-\lambda)\cdot\theta_{ava}+\lambda\cdot\theta_{cafe}$ outperforms θ_{ava} alone, not only in terms of *cafe*, but also of *ava* when $\lambda\in\{0.1,0.2\}$. These findings confirm that RS can better align text-to-image models with a variety of aesthetic preferences. This ability to adapt at test time paves the way for a new form of user interaction with text-to-image models, beyond prompt engineering.

3.4 Text-to-box: visual grounding of objects with diverse sizes

We now consider visual grounding (VG) [99]: the task is to predict the bounding box of the region described by an input text. We use a seq-to-seq unified model predicting the box auto-regressively as a sequence of location tokens [100]. This model is pre-trained on a large image-text dataset, then fine-tuned with cross-entropy for VG; finally, we use a weighted loss between the cross-entropy and REINFORCE in the RL stage. As the main evaluation metric for VG is the accuracy (i.e., intersection over union (IoU) >0.5), we consider 3 non-differentiable rewards: the accuracy on small, medium, and large objects. We design this experimental setup because improving results on all sizes simultaneously is challenging, as shown in Figure 5(c), where MORL performs similarly to the initialization. The results in Figure 5(b) confirm that optimizing for small objects degrades performance on large ones; fortunately, interpolating can trade-off. In conclusion, we can adapt to users' preferences at test time by adjusting λ, which in turn changes the object sizes that the model effectively handles. On the one hand, if focusing on distant and small objects, a large coefficient should be assigned to θ_{Small}. On the other hand, to perform well across all sizes, we can recover initialization's performances by averaging uniformly (in Figure 5(c)). More details are in Appendix F.

3.5 Locomotion with diverse engineered rewards

Teaching humanoids to walk in a human-like manner [101] serves as a benchmark to evaluate RL strategies [102] for continuous control. One of the main challenges is to shape a suitable proxy reward [103, 104], given the intricate coordination and balance involved in human locomotion. It is standard [105] to consider dense rewards of the form $R = velocity - \alpha\times\sum_t a_t^2$, controlling the agent's velocity while regularizing the actions $\{a_t\}_t$ taken over time. Yet, the penalty coefficient α is challenging to set. To address this, we devised two rewards in the Brax physics engine [106]: a risky R_1 with $\alpha=0$, and a more cautious R_2 with $\alpha=1$.

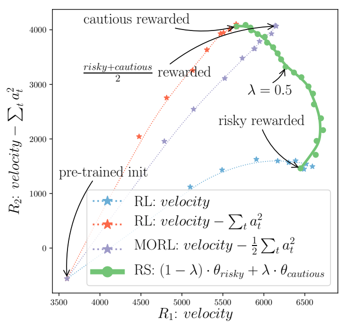


Figure 6: Locomotion results.

Like in all previous tasks, RS's front in Figure 6 exceeds the interpolated rewards, as per Hypothesis 1. Moreover, the front defined by RS indicates an effective balance between risk-taking and cautiousness, providing empirical support for Hypothesis 2, although MORL with $\mu=0.5$ (i.e., $\alpha=0.5$) slightly surpasses RS's front. For a more qualitative and intuitive assessment, we provide animations of our RL agent's locomotion on this anonymized url page. More details are in Appendix G.

4 Related work

Our RS approach leans on two key components from traditional DRL. The first is **proxy rewards**, whose design is challenging. Statistical metrics, the standard in captioning [24] or language translation [107], are not practical to measure human concepts [32] such as helpfulness [41, 43]. Reward models can be trained via inverse DRL [108, 109] when supervision from experts is available, otherwise from prediction comparison in recent RLHF works[12, 13, 15]. The latest [32, 110, 111, 112, 113] further reduce the labeling costs by using the in-context abilities of LLMs. Second, RS relies on existing **RL algorithms** to maximize the given rewards. RS succeeds with variants of two of the most common, REINFORCE [92] and PPO [78], suggesting it could be applied to others [114, 115]. Among the ensembling-like RL strategies [116, 117, 118] handling multiple policies, some [119, 120] aim to explicitly increase the diversity, yet never with foundation models nor weight interpolation. Moreover, pre-training could address stability and exploration issues [121, 122, 123]. When dealing with multiple objectives in deep learning, the common approach is to combine them into a single reward [56, 57]: [42] multiply the predictions of a preference RM (evaluating factfullness) and a rule RM (detecting rules breaking). The **multi-policy** alternatives [46, 47, 54, 55] are usually more costly. To reduce the cost, [124, 125] build experts and then train a new network to combine them; [126, 127, 128] share weights across experts; [129, 130, 131, 132] directly train a single model; the recent and more similar [133] learns one linear embedding per (locomotion) task that can be interpolated. Yet, these works are mostly for academic benchmarks [105, 134]; adapting them to larger tasks (e.g., RLHF for foundation models with PPO) is challenging as they modify the training procedure. Finally, we relate to **multitask learning** [135], where predictions are evaluated for multiple tasks; in contrast, we have a single prediction evaluated by multiple rewards.

Recent works extended the **linear mode connectivity** when fine-tuning on different tasks [65, 66, 67, 136] or with different losses [63, 137], while [138] highlighted some failures in NLP for classification. In contrast, we investigate the LMC in RL. The most similar works are for control system tasks: [139] averaging decision transformers and [140] explicitly enforcing connectivity in subspaces of policies trained from scratch on a single reward. When the LMC holds, combining networks in weights combines their abilities [141, 142]; e.g., averaging an English summarizer and an English-to-French translator can summarize in French [143]. In domain generalization, [62, 63, 144] showed that WI reduces model misspecification [145]; by analogy, we show that RS reduces reward misspecification.

5 Discussion: limitations and societal impacts

The recent and rapid scaling of networks presents both opportunities and major concerns [9, 146, 147]. Our approach is a step towards better **empirical alignment** [10, 11]. Yet, reward misspecification is only one of the many challenges inherited from the RL paradigm. First, proxy rewards may lack robustness [148] or be hacked [149] via adversarial exploitation, making them unreliable. Second, RL algorithms may cause overfitting, leading to poor generalization in test, with a risk of goal misgeneralization [150, 151]. Third, RLHF has drawbacks, such as harming calibration [18]. Our a posteriori multi-policy strategy could alleviate the impact of some badly shaped proxy rewards and some failed optimizations, as well as tackling Goodhart's law [152]. Yet, without constraint on the test distribution, complete alignment may be impossible [153], for example for LLMs with prompts of arbitrary (long) length. Therefore, new training paradigms [154, 155] beyond RL may be required.

Theoretical guarantees for alignment are also needed [156]. Yet, RS relies on an empirical finding: the LMC [60], which currently lacks full theoretical guarantees, even in the simplest case of moving averages [94]. The best existing explanation [63, 94] relies on the similarities between weight interpolation and functional ensembling [72, 73] when weights remain close, as recalled in Lemma 4. Moreover, assuming the LMC, Lemma 1 theoretically fixes issues only for $\hat{R}$ linear over the proxy rewards. Yet, such **linearization** cannot encapsulate all types of (human) preferences [53, 71]. Thus, considering more complex combinations [157, 158, 159, 160] is a promising direction. We may empirically overcome this limitation within RS by continually adjusting and adding new proxy rewards, such that their linear mixtures have increasingly good coverage. Indeed, RS is flexible and was shown to handle variable numbers of rewards, allowing for an iterative development process.

Finally, our a posteriori alignment with users facilitates **personalization** [161] of models. As discussed in Appendix A.1 and in [50], this could increase usefulness by providing tailored generation, notably to under-represented groups. Moreover, the distributed nature of RS makes it parallelizable thus practical in a federated learning setup [162] where data must remain private. Yet, this personalization comes with risks for individuals of "reinforcing their biases [...] and narrowing their information diet"[50]. This may worsen the polarization of the public sphere. Under these concerns, we concur with the notion of "personalization within bounds" [50], with these boundaries potentially set by weights fine-tuned on diverse and carefully inspected rewards.

Conclusion. As AI systems are increasingly applied to crucial real-world tasks, there is a pressing issue to align them to our specific and diverse needs, while making the process more transparent and limiting the cultural hegemony of a few individuals. In this paper, we proposed rewarded soup, a strategy that efficiently yields Pareto-optimal solutions through weight interpolation after training. Our experiments have consistently validated our working hypotheses for various significant large-scale learning tasks, demonstrating that rewarded soup can mitigate reward misspecification. We hope to inspire further research in exploring how the generalization literature in deep learning can help for alignment, to create AIs handling the diversity of opinions, and benefit society as a whole.

References

[1] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint*, 2021. (p. 1)

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. (p. 1)

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. (p. 1)

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. (p. 1)

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. (p. 1)

[6] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. (p. 1)

[7] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *NeuriPS*, 2014. (p. 1)

[8] Vladimir N Vapnik. An overview of statistical learning theory. In *TNN*, 1999. (p. 1)

[9] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint*, 2016. (pp. 1, 2, and 9)

[10] Jessica Taylor, Eliezer Yudkowsky, Patrick LaVictoire, and Andrew Critch. Alignment for advanced machine learning systems. *Ethics of AI*, 2016. (pp. 1 and 9)

[11] Richard Ngo. The alignment problem from a deep learning perspective. *arXiv preprint*, 2022. (pp. 1 and 9)

[12] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *NeurIPS*, 2020. (pp. 1, 2, 5, 6, 8, 26, and 27)

[13] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. (pp. 1, 2, and 8)

[14] André Susano Pinto, Alexander Kolesnikov, Yuge Shi, Lucas Beyer, and Xiaohua Zhai. Tuning computer vision models with task rewards. *arXiv preprint*, 2023. (pp. 1 and 6)

[15] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017. (pp. 1, 2, 6, and 8)

[16] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint*, 2019. (p. 1)

[17] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint*, 2021. (pp. 1 and 5)

[18] OpenAI. Gpt-4 technical report. *arXiv preprint*, 2023. (pp. 1 and 9)

[19] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. (p. 1)

[20] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022. (pp. 1 and 5)

[21] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *ACL*, 2022. (p. 1)

[22] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023. (pp. 1, 2, 5, 26, and 27)

[23] Yoav Goldberg. Reinforcement learning for language models. `https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81`, 2023. (p. 1)

[24] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017. (pp. 1, 6, 7, 8, and 28)

[25] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint*, 2023. (pp. 1, 7, and 30)

[26] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Better aligning text-to-image models with human preference. *arXiv preprint*, 2023. (pp. 1, 7, and 30)

[27] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. HIVE: Harnessing human feedback for instructional visual editing. *arXiv preprint*, 2023. (p. 1)

[28] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 2021. (p. 1)

[29] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. (pp. 1, 2, 6, and 28)

[30] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*, 2003. (pp. 1, 2, 7, and 28)

[31] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Consensus-based image description evaluation. In *ICCV*, 2015. (pp. 1, 7, and 28)

[32] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *ICLR*, 2023. (pp. 2 and 8)

[33] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. *arXiv preprint*, 2023. (pp. 2, 7, and 30)

[34] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *ICLR*, 2022. (p. 2)

[35] Eric J Michaud, Adam Gleave, and Stuart Russell. Understanding learned reward functions. *arXiv preprint*, 2020. (p. 2)

[36] Aaron Wildavsky. Choosing preferences by constructing institutions: A cultural theory of preference formation. *American political science review*, 1987. (p. 2)

[37] CA Coello. Handling preferences in evolutionary multiobjective optimization: A survey. In *CEC*, 2000. (p. 2)

[38] Shalom H Schwartz et al. An overview of the schwartz theory of basic values. *Online readings in Psychology and Culture*, 2012. (p. 2)

[39] Marcos Nadal and Anjan Chatterjee. Neuroaesthetics and art's diversity and universality. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2019. (p. 2)

[40] David Lopez-Paz, Diane Bouchacourt, Levent Sagun, and Nicolas Usunier. Measuring and signing fairness as performance under multiple stakeholder distributions. *arXiv preprint*, 2022. (p. 2)

[41] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint*, 2022. (pp. 2, 5, 8, and 27)

[42] Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint*, 2022. (pp. 2 and 9)

[43] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *arXiv preprint*, 2021. (pp. 2, 3, and 8)

[44] Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, Christie-Carol Beauchamp, et al. Rewarding chatbots for real-world engagement with millions of users. *arXiv preprint*, 2023. (p. 2)

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. (pp. 2, 5, 26, and 27)

[46] Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *ICML*, 2008. (pp. 2, 3, and 9)

[47] Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE-T-CYBERNETICS*, 2020. (pp. 2, 3, and 9)

[48] Michiel A. Bakker, Martin J Chadwick, Hannah Sheahan, Michael Henry Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matthew Botvinick, and Christopher Summerfield. Fine-tuning language models to find agreement among humans with diverse preferences. In *NeurIPS*, 2022. (p. 2)

[49] Aviv Ovadya. Generative CI through collective response systems. *arXiv preprint*, 2023. (p. 2)

[50] Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback. *arXiv preprint*, 2023. (pp. 2, 9, and 21)

[51] Alexander Pan, Chan Jun Shern, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Jonathan Ng, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the MACHIAVELLI benchmark. *arXiv preprint*, 2023. (p. 2)

[52] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *JAAMAS*, 2022. (pp. 2 and 21)

[53] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology*, 2018. (pp. 2, 9, and 21)

[54] Fumihide Tanaka and Masayuki Yamamura. Multitask reinforcement learning on the distribution of mdps. In *CIRA*, 2003. (pp. 2 and 9)

[55] Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *JMLR*, 2014. (pp. 2 and 9)

[56] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *JAIR*, 2013. (pp. 2 and 9)

[57] Roxana Rădulescu, Patrick Mannion, Diederik M Roijers, and Ann Nowé. Multi-objective multi-agent decision making: a utility-based analysis and survey. *AAMAS*, 2020. (pp. 2, 4, and 9)

[58] Vilfredo Pareto. *Cours d'économie politique*. Librairie Droz, 1964. (p. 2)

[59] Patrick Mannion, Fredrik Heintz, Thommen George Karimpanal, and Peter Vamplew. Multi-objective decision making for trustworthy ai. In *MODeM Workshop*, 2021. (p. 2)

[60] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *ICML*, 2020. (pp. 2, 4, and 9)

[61] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *NeurIPS*, 2020. (pp. 2 and 4)

[62] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022. (pp. 2, 4, 9, 26, and 29)

[63] Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022. (pp. 2, 9, 26, and 29)

[64] Michael Matena and Colin Raffel. Merging models with Fisher-weighted averaging. In *NeurIPS*, 2022. (pp. 2 and 24)

[65] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *NeurIPS*, 2022. (pp. 2 and 9)

[66] Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. ColD fusion: Collaborative descent for distributed multitask finetuning. *arXiv preprint*, 2022. (pp. 2 and 9)

[67] Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model Ratatouille: Recycling diverse models for out-of-distribution generalization. In *ICML*, 2023. (pp. 2, 4, 9, and 22)

[68] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. (p. 3)

[69] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint*, 2018. (p. 3)

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. (pp. 3 and 27)

[71] Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In *AJCAIA*, 2008. (pp. 3, 4, and 9)

[72] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *TPAMI*, 1990. (pp. 4, 6, 7, and 9)

[73] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. (pp. 4, 6, 7, and 9)

[74] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *ICLR*, 2022. (p. 4)

[75] Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint*, 2022. (p. 4)

[76] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020. (pp. 5, 6, and 26)

[77] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint*, 2022. (p. 5)

[78] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017. (pp. 5, 8, 27, and 32)

[79] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. TRL: Transformer reinforcement learning. `https://github.com/lvwerra/trl`, 2020. (pp. 5 and 27)

[80] Edward Beeching, Younes Belkada, Leandro von Werra, Sourab Mangrulkar, Lewis Tunstall, and Kashif Rasul. Fine-tuning 20B LLMs with RLHF on a 24GB consumer GPU. `https://huggingface.co/blog/trl-peft`, 2023. (pp. 5, 26, and 27)

[81] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. (pp. 5, 26, and 27)

[82] Hadeer Ahmed. *Detecting opinion spam and fake news using n-gram analysis and semantic similarity*. PhD thesis, 2017. (pp. 5 and 27)

[83] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *ACL Workshop*, 2017. (pp. 5 and 27)

[84] Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. Huggingface h4 stack exchange preference dataset, 2023. (pp. 5 and 27)

[85] Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. Improving Faithfulness in Abstractive Summarization with Contrast Candidate Generation and Selection. In *NAACL*, 2021. (pp. 6, 26, and 27)

[86] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint*, 2023. (pp. 6 and 26)

[87] Gary G Yen and Zhenan He. Performance metric ensemble for multiobjective evolutionary algorithms. *TEVC*, 2013. (pp. 6 and 7)

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. (pp. 6, 7, and 28)

[89] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. (pp. 7 and 28)

[90] Jia Cheng Hu, Roberto Cavicchioli, and Alessandro Capotondi. ExpansionNet v2: Block static expansion in fast end to end training for image captioning. *arXiv preprint*, 2022. (pp. 7 and 28)

[91] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin Transformer V2: Scaling up capacity and resolution. In *CVPR*, 2022. (pp. 7 and 28)

[92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 1992. (pp. 7, 8, 28, 31, and 32)

[93] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. (pp. 7 and 28)

[94] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. (pp. 7, 9, and 26)

[95] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. (p. 7)

[96] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. (pp. 7 and 30)

[97] Naila Murray, Luca Marchesotti, and Florent Perronnin. AVA: A large-scale database for aesthetic visual analysis. In *CVPR*, 2012. (pp. 7 and 30)

[98] Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. Difffit: Unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. *arXiv preprint arXiv:2304.06648*, 2023. (pp. 7 and 30)

[99] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. (pp. 7, 8, 31, and 32)

[100] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *CoRR*, 2022. (pp. 8 and 31)

[101] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *ICML*, 2016. (p. 8)

[102] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, 1999. (p. 8)

[103] Marco Dorigo and Marco Colombetti. Robot shaping: Developing autonomous agents through learning. *Artificial intelligence*, 1994. (p. 8)

[104] Dan Dewey. Reinforcement learning and the reward engineering principle. In *AAAI Spring Symposia*, 2014. (p. 8)

[105] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012. (pp. 8 and 9)

[106] C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax–a differentiable physics engine for large scale rigid body simulation. *arXiv preprint*, 2021. (pp. 8 and 32)

[107] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016. (p. 8)

[108] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, 2000. (p. 8)

[109] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004. (p. 8)

[110] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint*, 2022. (p. 8)

[111] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. *arXiv preprint*, 2023. (p. 8)

[112] Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale. *arXiv preprint*, 2023. (p. 8)

[113] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint*, 2023. (p. 8)

[114] Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint*, 2023. (p. 8)

[115] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: Rank responses to align language models with human feedback without tears. *arXiv preprint*, 2023. (p. 8)

[116] Jack M Wang, David J Fleet, and Aaron Hertzmann. Optimizing walking controllers for uncertain inputs and environments. *ACM*, 2010. (p. 8)

[117] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In *IROS*, 2015. (p. 8)

[118] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. EPOpt: Learning robust neural network policies using model ensembles. In *ICLR*, 2017. (p. 8)

[119] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *NeurIPS*, 2020. (p. 8)

[120] Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information. *Neural Networks*, 2022. (p. 8)

[121] Zhihui Xie, Zichuan Lin, Junyou Li, Shuai Li, and Deheng Ye. Pretraining in deep reinforcement learning: A survey. *arXiv preprint*, 2022. (p. 8)

[122] Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. Foundation models for decision making: Problems, methods, and opportunities. *arXiv preprint*, 2023. (p. 8)

[123] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, 2020. (p. 8)

[124] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *TOG*, 2020. (p. 9)

[125] Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 2020. (p. 9)

[126] Hossam Mossalam, Yannis M Assael, Diederik M Roijers, and Shimon Whiteson. Multi-objective deep reinforcement learning. *arXiv preprint*, 2016. (p. 9)

[127] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *ICML*, 2007. (p. 9)

[128] Thanh Thi Nguyen, Ngoc Duy Nguyen, Peter Vamplew, Saeid Nahavandi, Richard Dazeley, and Chee Peng Lim. A multi-objective deep reinforcement learning framework. *EAAI*, 2020. (p. 9)

[129] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research*, 2013. (p. 9)

[130] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *NeurIPS*, 2019. (p. 9)

[131] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *ICML*, 2019. (p. 9)

[132] Markus Peschl, Arkady Zgonnikov, Frans A Oliehoek, and Luciano C Siebert. Moral: Aligning ai with human norms through multi-objective reinforced active learning. *arXiv preprint*, 2021. (p. 9)

[133] Pu Hua, Yubei Chen, and Huazhe Xu. Simple emergent action representations from multi-task policy training. In *ICLR*, 2023. (p. 9)

[134] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Deakin University*, 2011. (p. 9)

[135] Rich Caruana. Multitask learning. *Machine learning*, 1997. (pp. 9 and 22)

[136] Chengyue Wu, Teng Wang, Yixiao Ge, Zeyu Lu, Ruisong Zhou, Ying Shan, and Ping Luo. π-tuning: Transferring multimodal foundation models with optimal multi-task interpolation. In *ICML*, 2023. (p. 9)

[137] Francesco Croce, Sylvestre-Alvise Rebuffi, Evan Shelhamer, and Sven Gowal. Seasoning model soups for robustness to adversarial and natural distribution shifts. *arXiv preprint*, 2023. (p. 9)

[138] Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies. In *ICLR*, 2023. (p. 9)

[139] Daniel Lawson and Ahmed H Qureshi. Merging decision transformers: Weight averaging for forming multi-task policies. In *ICLR RRL Workshop*, 2023. (p. 9)

[140] Jean-Baptiste Gaya, Laure Soulier, and Ludovic Denoyer. Learning a subspace of policies for online adaptation in reinforcement learning. In *ICLR*, 2022. (p. 9)

[141] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023. (pp. 9, 22, and 29)

[142] Nico Daheim, Nouha Dziri, Mrinmaya Sachan, Iryna Gurevych, and Edoardo M Ponti. Elastic weight removal for faithful and abstractive dialogue generation. *arXiv preprint*, 2023. (p. 9)

[143] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Exploring the benefits of training expert language models over instruction tuning. *arXiv preprint*, 2023. (p. 9)

[144] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. In *NeurIPS*, 2021. (p. 9)

[145] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *JMLR*, 2020. (p. 9)

[146] Dan Hendrycks and Mantas Mazeika. X-risk analysis for AI research. *arXiv preprint*, 2022. (p. 9)

[147] Dan Hendrycks. Natural selection favors AIs over humans. *arXiv preprint*, 2023. (p. 9)

[148] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *arXiv preprint*, 2022. (p. 9)

[149] Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *NeurIPS*, 2022. (p. 9)

[150] Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint*, 2022. (p. 9)

[151] Lauro Langosco Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *ICML*, 2022. (p. 9)

[152] Ben Smith. A brief review of the reasons multi-objective RL could be important in AI Safety Research. https://www.alignmentforum.org/posts/i5dLfi6m6FCexReK9/a-brief-review-of-the-reasons-multi-objective-rl-could-be, 2021. (p. 9)

[153] Yotam Wolf, Noam Wies, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models. *arXiv preprint*, 2023. (p. 9)

[154] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint*, 2019. (p. 9)

[155] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. Pretraining language models with human preferences. *arXiv preprint*, 2023. (p. 9)

[156] Manel Rodriguez-Soto, Maite Lopez-Sanchez, and Juan A Rodríguez-Aguilar. Guaranteeing the learning of ethical behaviour through multi-objective reinforcement learning. In *AAMAS*, 2021. (p. 9)

[157] Zoltán Gábor, Zsolt Kalmár, and Csaba Szepesvári. Multi-criteria reinforcement learning. In *ICML*, 1998. (p. 9)

[158] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, 2021. (p. 9)

[159] Kristof Van Moffaert, Madalina M Drugan, and Ann Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *ADPRL*, 2013. (p. 9)

[160] Benjamin J Smith, Robert Klassert, and Roland Pihlakas. Soft maximin approaches to multi-objective decision-making for encoding human intuitive values. In *Multi-Objective Decision Making Workshop*, 2021. (p. 9)

[161] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. *arXiv preprint*, 2023. (pp. 9 and 21)

[162] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017. (pp. 9 and 22)

[163] Philip E Tetlock. A value pluralism model of ideological reasoning. *JPSP*, 1986. (p. 21)

[164] Umer Siddique, Paul Weng, and Matthieu Zimmer. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *ICML*, 2020. (p. 21)

[165] Iason Gabriel and Vafa Ghazavi. The challenge of value alignment: From fairer algorithms to AI safety. *arXiv preprint*, 2021. (p. 21)

[166] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC*, 2016. (p. 21)

[167] Kristof Van Moffaert, Tim Brys, Arjun Chandra, Lukas Esterle, Peter R Lewis, and Ann Nowé. A novel adaptive weight selection algorithm for multi-objective multi-agent reinforcement learning. In *IJCNN*, 2014. (p. 22)

[168] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. (p. 22)

[169] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *NeurIPS*, 2021. (p. 22)

[170] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *ICML*, 2018. (p. 22)

[171] Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *NeurIPS*, 2017. (p. 22)

[172] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade learning environment: An evaluation platform for general agents. *JAIR*, 2013. (p. 22)

[173] Zafir Stojanovski, Karsten Roth, and Zeynep Akata. Momentum-based weight interpolation of strong zero-shot models for continual learning. In *NeurIPS Interpolate Workshop*, 2022. (p. 22)

[174] Steven Vander Eeckt et al. Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition. *arXiv preprint*, 2022. (p. 22)

[175] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. Branch-Train-Merge: Embarrassingly parallel training of expert language models. *arXiv preprint*, 2022. (p. 22)

[176] Colin Raffel. A Call to Build Models Like We Build Open-Source Software. `https://colinraffel.com/blog/a-call-to-build-models-like-we-build-open-source-software.html`, 2021. (p. 22)

[177] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks*. 2012. (p. 23)

[178] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. (pp. 23, 27, and 30)

[179] Yann LeCun, J. S. Denker, Sara A. Solla, R. E. Howard, and L.D. Jackel. Optimal brain damage. In *NeurIPS*, 1990. (p. 23)

[180] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *ICML*, 2022. (p. 23)

[181] Sue Becker and Yann Le Cun. Improving the convergence of back-propagation learning with second order methods. In *Connectionist models summer school*, 1988. (p. 23)

[182] Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London.*, 1922. (p. 24)

[183] Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. In *Neural computation*, 2002. (p. 24)

[184] Valentin Thomas, Fabian Pedregosa, Bart van Merriënboer, Pierre-Antoine Manzagol, Yoshua Bengio, and Nicolas Le Roux. On the interplay between noise and curvature and its effect on optimization and generalization. In *AISTATS*, 2020. (p. 24)

[185] Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In *NeurIPS*, 2019. (p. 24)

[186] Eric J. Wang. Alpaca-LoRA. https://github.com/tloen/alpaca-lora, 2023. (p. 27)

[187] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 2018. (p. 27)

[188] Edward Beeching, Younes Belkada, Kashif Rasul, Lewis Tunstall, Leandro von Werra, Nazneen Rajani, and Nathan Lambert. StackLLaMA: An RL Fine-tuned LLaMA Model for Stack Exchange Question and Answering, 2023. (p. 27)

[189] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, 2011. (p. 27)

[190] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. (p. 28)

[191] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *ICLR*, 2020. (p. 28)

[192] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Hanna Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *CVPR*, 2022. (p. 29)

[193] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint*, 2021. (p. 30)

[194] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint*, 2023. (p. 30)

[195] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint*, 2022. (p. 31)

[196] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020. (p. 31)

Rewarded soups: towards Pareto-optimality by averaging weights fine-tuned on diverse rewards

Supplementary material

This supplementary material is organized as follows:

- Appendix A further discusses the practical benefits of our RS strategy.
- Appendix B details some theoretical aspects of our RS strategy.
- Appendix C details our experiments in RLHF with LLaMA for text-to-text generation.
- Appendix D details and enriches our experiments in image captioning.
- Appendix E details and enriches our experiments in image generation.
- Appendix F details and enriches our experiments in visual grounding.
- Appendix G details and enriches our locomotion experiments.

The shareable code sections will be released on this anonymized url page.

A Discussion

In this section we discuss the benefits of our rewarded soup (RS) approach with respect to the two families of strategies: the **single-policy** and the **multi-policy** approaches.

A.1 Compared to single-policy approaches

The main reason why single-policy approaches are not suitable is because they optimize over a single set of preferences. In contrast, we build a coverage set of Pareto-optimal policies. This is important for the following reasons, mostly first discussed in Hayes *et al.* [52] and in Kirk *et al.* [50].

Indeed, the user's true reward is highly uncertain before training. This "semi-blind" [52] manual process forces a priori and uncertain decisions about the required trade-offs. It **shifts the responsibility** from the problem stakeholders to the system engineers, who need to anticipate the impact of their choices on the final performance. Critically, the RLHF process may cause the "tyranny of the crowdworker" [50], as models are "tailored to meet the expectations of [...] a small number of crowdworkers primarily based in the US, with little to no representation of broader human cultures, geographies or languages." [50]. Moreover, these "biases are exacerbated by a lack of […] documentation" [50]. Thus [50] argue that the **personalization should be explicit**—rather than implicitly caused by hidden and chaotic engineering choices. In contrast, our strategy could **support decision-making** to find a good balance between (potentially conflicting) parties' interests. This value pluralism [163] can lead to **fairer** and more equitable outcomes [53, 164]. Single-policy cannot adapt to test time requirements; in contrast, RS facilitates personalized assistances [161], with fewer prompts/inputs to the model, as we only need to adapt interpolating coefficients and not the full network. This is all the more important as human preferences change from time to time: in this **dynamic utility function** scenario, RS can quickly adapt by adjusting the λ to match new preferences. Finally, RS could also improve the **interpretability** and **explainability** of the decisions. Letting the users decide could make the process more **transparent** [165], which is essential to ensure that the development process is fair, unbiased, and inclusive [166].

A.2 Compared to multi-policy approaches

The main reason why other multi-policy approaches through multitasking are not suitable is because of their **computational costs** required to learn a dense set of policies. In contrast, RS only trains the proxy rewards independently, and enables the selection of the interpolating coefficient a posteriori. This is especially useful with large number of rewards and thus growing number of combinations. Second, multitask [135] is challenging; for example, even if the true reward is actually a linear weighted sum of some proxy rewards and those coefficients are known, using those preferences during training can lead to suboptimal results [167], because of conflicting gradients [168, 169] or different variance scales [170, 171]. This has been tackled in RL, but so far mostly for games such as ATARI [172]. Third, our strategy is compatible with the inherent **iterative engineering process** of alignment. Indeed, RS can continually include adjusted opinions while preventing forgetting of the old behaviours. This relates to the **continual learning** challenge, and the empirical observations that weight averaging can reduce catastrophic forgetting [173, 174]. Moreover, as shown in [141] and confirmed in Figure 9(c), negative editing by weight interpolation can fix and force the removal of some behaviours. Finally, RS is computationally effective, requiring **no communication across servers**, thus enabling "embarrassingly simple parallelization" [175]. This facilitates its use in **federated learning** scenario [162] where the data should remain private. Actually, RS follows the **updatable machine learning paradigm** [176], "allowing for the collaborative creation of increasingly sophisticated AI system" [67]. In the future, we may develop open-source personalized models, rewarded on decentralized private datasets, and combine them continuously.

B Theoretical insights

B.1 Proof of Lemma 1

Proof. Considering θ maximizing $\hat{R}$, we first show that θ is on the PF of $\{R_i\}_i$. Otherwise, considering $\theta' >_N \theta$ and as $\forall i, \hat{\mu}_i \geq 0$, we have $\sum_i \hat{\mu}_i R_i(\theta') > \sum_i \hat{\mu}_i R_i(\theta)$. This implies that θ' would produce a better policy than θ for $\hat{R} = \sum_i \hat{\mu}_i R_i$ and thus the contradiction. Finally, as θ is on the PF and by definition of a PCS, there exists λ s.t. $\forall k, R_k(\sum_i \lambda_i \cdot \theta_i) = R_k(\theta)$. □

B.2 Theoretical guarantees with quadratic rewards

In this section, we provide theoretical guarantees for the optimality or near-optimality of RS when considering quadratic rewards. This simplification this amounts to assuming that the rewards can be replaced by their second-order Taylor approximation, which is a realistic approach when the weights remain within a small neighborhood close to their maximum.

B.2.1 Same eigenvalues

For the first Lemma 2, we make the additional following Assumption 1.

Assumption 1 (Same eigenvalues)**.** *For every reward, the Hessian is constant and has only one positive eigenvalue.*

Lemma 2. *Let $\boldsymbol{\Theta} = \mathbb{R}^n$ be the parameter space, and let $r_1, ...r_d$ be d reward functions defined as:*

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad r_i(\boldsymbol{\theta}) = r_i(\boldsymbol{\theta}^{(i)}) - \lambda_i \|\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}\|^2$$

where $(\lambda_1, \cdots, \lambda_d) \in \mathbb{R}_+^d$ and $\boldsymbol{\theta}^{(i)}$ is the global maximum for reward r_i. Let $\mu = (\mu_1, \mu_2, ...\mu_d) \in \mathbb{R}_+^n$ such that $\sum_i \mu_i = 1$. Then, the optimum value for the reward function $r_\mu = \sum_i \mu_i r_i$ is in the convex hull of $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(d)}$.

Proof. The function r_μ is a quadratic function that has a unique global maximum that we can find analytically:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} r_\mu(\boldsymbol{\theta}^*) = 0 \implies & \sum_{i=1}^{d} \mu_i \lambda_i (\boldsymbol{\theta}^* - \boldsymbol{\theta}^{(i)}) = 0 \\
\implies & \boldsymbol{\theta}^* = \frac{\sum_{i=1}^d \mu_i \lambda_i \boldsymbol{\theta}^{(i)}}{\sum_{i=1}^d \mu_i \lambda_i}
\end{aligned}$$

Since all the $\mu_i \lambda_i$ are non-negative, $\boldsymbol{\theta}^*$ is indeed in the convex hull of $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(d)}$. □

In the case of quadratic rewards, it is reasonable to assume that the fine-tuning procedure will reach the global optimum. In this case, Lemma 2 tells us that the maximum value of an interpolation of rewards r_μ is always obtainable by weight interpolation.

B.2.2 Different eigenvalues

We now consider the more complex case with the relaxes assumption Assumption 2. For simplicity, we only consider two rewards.

Assumption 2 (Diagonal Hessian with different eigenvalues)**.** *The Hessian matrices of The reward functions have different eigenvalues.*

Remark 3. *This diagonal approximation of the Hessian is common: for example in optimization [177, 178], to prune networks [179] or in out-of-distribution generalization [180]. This strong assumption is supported by the empirical observation [181] that Hessians are diagonally dominant, in particular at the end of training. Also, we note that since the bound does not depend on the choice of base, our findings remain valid assuming only that the Hessians are co-diagonalizable.*

Lemma 3. *Let*

$$r_i = r_i(\boldsymbol{\theta}_i) - (\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)})^T H_i(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)})$$

for $i \in \{1, 2\}$, where H_i is the Hessian matrix of quadratic function r_i. Let $(\lambda_1^{(i)}, \cdots \lambda_n^{(i)})$ be the eigenvalues of Hessian H_i and suppose that the H_i are diagonal, so that

$$r_i(\theta_1, \cdots, \theta_n) = r_i(\theta_1^{(i)}, \cdots, \theta_n^{(i)}) - \sum_{j=1}^{n} \lambda_j^{(i)}(\theta_j - \theta_j^{(i)})^2$$

Considering the user's reward $r_\mu = \mu r_1 + (1-\mu)r_2$ with $\mu \in [0, 1]$, and

$$\Delta r = |\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} r_\mu(\boldsymbol{\theta}) - \max_{\nu\in[0,1]} r_\mu(\nu\boldsymbol{\theta}^{(1)} + (1-\nu)\boldsymbol{\theta}^{(2)})|.$$

Then the difference between xxx,

$$\frac{\Delta r}{\max\left(r_1(\boldsymbol{\theta}^{(1)}) - r_1(\boldsymbol{\theta}^{(2)}), r_2(\boldsymbol{\theta}^{(2)}) - r_2(\boldsymbol{\theta}^{(1)})\right)} \leq \frac{1}{4}\left(\frac{\mu(1-\mu)(M^2-1)}{\mu(1-\mu)(M-1)^2 + M}\right)^2$$

where $M = \max_{j\in\{1,\cdots,n\}} \max\left(\frac{\lambda_j^{(1)}}{\lambda_j^{(2)}}, \frac{\lambda_j^{(2)}}{\lambda_j^{(1)}}\right)$ is the maximum of eigenvalues ratio.

Proof. This novel proof is in three steps. First, we compute the maximum in R^n. Then we compute the maxium achievable by RS. Finally, we bound the differences between their rewards.

Let $R = \mu r_1(\theta_1^{(1)}, \cdots, \theta_n^{(1)}) + (1-\mu)r_2(\theta_1^{(2)}, \cdots, \theta_n^{(2)})$. Then

$$r_\mu(\theta_1, \cdots, \theta_n) = R - \sum_{j=1}^{n}\left(\mu\lambda_j^{(1)}(\theta_j - \theta_j^{(1)})^2 + (1-\mu)\lambda_j^{(2)}(\theta_j - \theta_j^{(2)})^2\right)$$

Let $(\nu_1, \cdots, \nu_n) \in \mathbb{R}^n$ and $\theta_j = \nu_j\theta_j^{(1)} + (1-\nu_j)\theta_j^{(2)}$. Then we can write

$$\begin{aligned} r_\mu(\theta_1, \cdots, \theta_n) = R - \sum_{j=1}^{n}\Bigg(&(\mu\lambda_j^{(1)} + (1-\mu)\lambda_j^{(2)})\left(\nu_j - \frac{\mu\lambda_j^{(1)}}{\mu\lambda_j^{(1)} + (1-\mu)\lambda_j^{(2)}}\right)^2 \\ &+ \frac{\mu(1-\mu)\lambda_j^{(1)}\lambda_j^{(2)}}{(\mu\lambda_j^{(1)} + (1-\mu)\lambda_j^{(2)})}\Bigg)\left(\theta_j^{(1)} - \theta_j^{(2)}\right)^2 \end{aligned} \tag{1}$$

In Equation 1, we can see that the global maximum for r_μ is reached for

$$\nu_j = \frac{\mu\lambda_j^{(1)}}{\mu\lambda_j^{(1)} + (1-\mu)\lambda_j^{(2)}} \in [0, 1]. \tag{2}$$

This serves as an upper-bound, available by weight interpolating only when having different interpolating coefficients for each dimension. In addition, the maximum reward for weight averaging is $\max_{\nu\in[0,1]} r_\mu(\nu\boldsymbol{\theta}^{(1)} + (1-\nu)\boldsymbol{\theta}^{(2)})$, which corresponds to optimizing equation 1 under the constraint that all ν_j are equal to a single value $\hat{\nu}$. Again, there is a unique value of $\hat{\nu}$ maximizing reward under that constraint, defined as

$$\hat{\nu} = \frac{\sum_{j=1}^{n} \mu\lambda_j^{(1)}\big(\theta_j^{(1)} - \theta_j^{(2)}\big)^2}{\sum_{j=1}^{n}\big(\mu\lambda_j^{(1)} + (1-\mu)\lambda_j^{(2)}\big)\big(\theta_j^{(1)} - \theta_j^{(2)}\big)^2}$$

Let $p_j = \big(\mu\lambda_j^{(1)} + (1-\mu)\lambda_j^{(2)}\big)\big(\theta_j^{(1)} - \theta_j^{(2)}\big)^2$. Then $\hat{\nu}$ writes

$$\hat{\nu} = \frac{\sum_{j=1}^{n} p_j\nu_j}{\sum_{j=1}^{n} p_j}$$

with ν_j defined by Equation 2. $\hat{\nu}$ is therefore the expectation of a random variable A such that $\mathbb{P}(A = \nu_i) = p_i / \sum_{j=1}^{n} p_j$.

By computing the value of r_μ with Equation 1 with θ corresponding to either the ν_j defined in 2 or with $\hat{\nu}$, the difference between the optimum value on the segment $[\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}]$ and the global optimum value writes

$$\Delta r = \Big(\sum_{i=1}^{n} \frac{p_i}{\sum_{j=1}^{n} p_i}(\nu_i - \hat{\nu})^2\Big)\Big(\sum_{j=1}^{n} p_j\Big)$$

In the first product term, we here recognize the variance of random variable A with $\mathbb{P}(A = \nu_i) = p_i / \sum_{j=1}^{n} p_j$. The variance is therefore smaller than the maximum value obtainable by a bounded random variable, and we apply here Popoviciu's inequality on variances:

$$\Delta r \leq \frac{1}{4}\big|\max_{1\leq j\leq n}\nu_j - \min_{1\leq j\leq n}\nu_j\big|^2 \sum_{j=1}^{n} p_j.$$

Now, we can bound the variables ν_i with M, since $1/M \leq \lambda_j^{(1)}/\lambda_j^{(2)} \leq M$ for all j we have

$$\frac{\mu}{\mu + (1-\mu)M} \leq \nu_j \leq \frac{M\mu}{M\mu + (1-\mu)}$$

So:

$$\Delta r \leq \frac{1}{4}\bigg(\frac{\mu(1-\mu)(M^2-1)}{\big(\mu M + (1-\mu)\big)\big(\mu + (1-\mu)M\big)}\bigg)^2\Big(\sum_{j=1}^{n} p_j\Big)$$

Furthermore, by computing $r_2(\boldsymbol{\theta}^{(1)})$ and $r_1(\boldsymbol{\theta}^{(2)})$, we have

$$\sum_{j=1}^{n} p_j = \mu\big(r_1(\boldsymbol{\theta}^{(1)}) - r_1(\boldsymbol{\theta}^{(2)})\big) + (1-\mu)\big(r_2(\boldsymbol{\theta}^{(2)}) - r_1(\boldsymbol{\theta}^{(1)})\big) \quad (3)$$

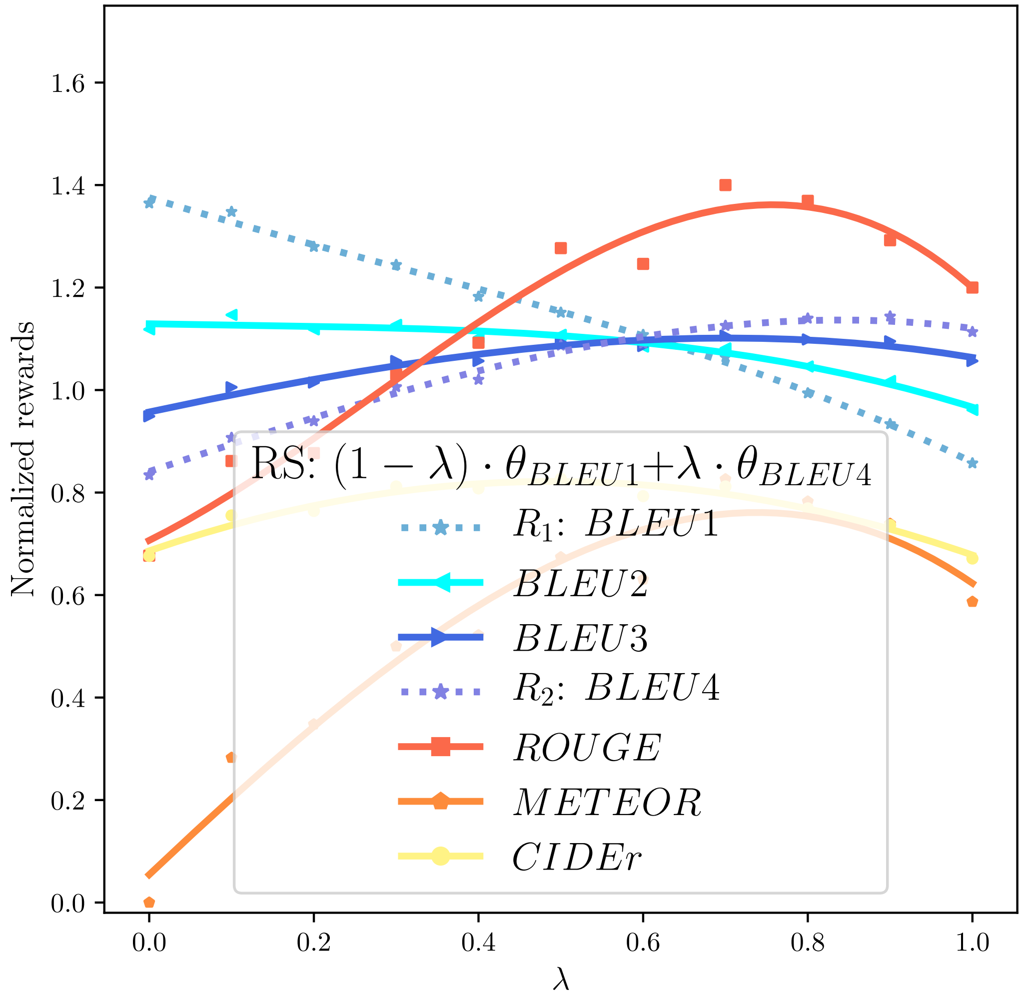
$$\leq \max\big(r_1(\boldsymbol{\theta}^{(1)}) - r_1(\boldsymbol{\theta}^{(2)}), r_2(\boldsymbol{\theta}^{(2)}) - r_2(\boldsymbol{\theta}^{(1)})\big) \quad (4)$$

which concludes the proof. Furthermore, the bound reaches its maximum for $\mu = 1/2$, for which we have the simpler bound

$$\frac{\Delta r}{\max\big(r_1(\boldsymbol{\theta}^{(1)}) - r_1(\boldsymbol{\theta}^{(2)}), r_2(\boldsymbol{\theta}^{(2)}) - r_2(\boldsymbol{\theta}^{(1)})\big)} \leq \frac{1}{4}\Big(\frac{M-1}{M+1}\Big)^2$$

□

Remark 4. *As a final remark, please note that subtleties in the proof above comes from the need of having one single interpolating coefficient for all n parameters of the network. Yet, the advanced merging operations in [64] propose exactly this, and actually their interpolating coefficients are inversely proportional to the diagonal of the Fisher matrices [182], which approximates the eigenvalues of the Hessian [183, 184]. Combining [64] and our rewarded soups is a promising research direction, the key issue being the computation of the Fisher information matrix [185] for networks with billions of parameters.*



Bound visualization

B.3 Similarity between weight interpolation and functional ensembling

Lemma 4 (λ-interpolation of weights approximates the λ-ensembling of predictions. Adapted from [62, 63, 94].)**.** *Given θ_1 and θ_2 optimized for R_1 and R_2 s.t. they remain close, i.e., $\|\theta_1 - \theta_2\|_2 \approx 0$. Denoting θ_λ the interpolated weights $\theta_\lambda = (1-\lambda)\cdot\theta_1 + \lambda\cdot\theta_2$ and f_λ the ensembling of predictions $f_\lambda(\cdot) = (1-\lambda)\cdot f(\cdot,\theta_1) + \lambda\cdot f(\cdot,\theta_2)$:*

$$f(\cdot,\theta_\lambda) \approx f_\lambda(\cdot)$$

and for $k \in \{1,2\}$:

$$R_k(f(\cdot,\theta_\lambda)) \approx R_k(f_\lambda(\cdot))$$

Proof. This proof follows [63] and has two components.

Functional approximation. First we perform a Taylor expansion at the first order of the models' predictions w.r.t. parameters θ for $x \in T$:

$$\begin{aligned} f(x,\theta_1) &= f(x,\theta_\lambda) + \nabla_\theta f(x,\theta_\lambda)^\intercal(\theta_1 - \theta_\lambda) + \mathcal{O}\Big(\|\theta_1-\theta_\lambda\|_2^2\Big) \\ &= f(x,\theta_\lambda) + \nabla_\theta f(x,\theta_\lambda)^\intercal(\lambda\cdot\theta_1 - \lambda\cdot\theta_2) + \mathcal{O}\Big(\|\theta_1-\theta_2\|_2^2\Big) \end{aligned}$$

and similarly:

$$f(x,\theta_2) = f(x,\theta_\lambda) + \nabla_\theta f(x,\theta_\lambda)^\intercal((\lambda-1)\cdot\theta_1 + (1-\lambda)\cdot\theta_2) + \mathcal{O}\Big(\|\theta_1-\theta_2\|_2^2\Big)$$

Then by λ-weighted sum over i, the term multiplying $\nabla_\theta f(x,\theta_\lambda)^\intercal$ cancels out and we obtain:

$$f_\lambda(x) = (1-\lambda)\cdot f(x,\theta_1) + \lambda\cdot f(x,\theta_2) = f(x,\theta_\lambda) + \mathcal{O}\Big(\|\theta_1-\theta_2\|_2^2\Big). \tag{5}$$

Reward approximation. Second, we obtain the reward approximation with a Taylor expansion at the zeroth order of the reward R_k for $k \in \{1,2\}$ and injecting Equation (5):

$$\begin{aligned} R_k(f_\lambda(x)) &= R_k(f(x,\theta_\lambda)(x)) + \mathcal{O}(\|f_\lambda(x) - f(x,\theta_\lambda)\|_2) \\ &= R_k(f(x,\theta_\lambda)(x)) + \mathcal{O}\Big(\|\theta_1-\theta_2\|_2^2\Big). \end{aligned}$$

We obtain the results when θ_1 and θ_2 remain close, i.e., when we can ignore the $\mathcal{O}$ term. □

C Text-to-text: LLaMA with diverse RLHFs

We summarize the key implementation details of our text-to-text generation experiments in Table 1. The pre-trained network is LLaMA-7b [45]; then low-rank adapters [81] were fine-tuned on Alpaca [22] to follow instructions. We eventually fine-tune via PPO on the different considered tasks. Our code is adapted from [80]; we kept most of their hyperparameter values, only dividing by 2 the batch size to fit in our GPU and extending the output length. For each considered task, we downloaded the reward models from HuggingFace [76]. For example in summarization tasks, R_1 was open-sourced in an effort to reproduce the Summarize from Human Feedback paper [12], while R_2 [85] aimed at improved "faithfulness in abstractive summarization with contrast candidate generation". For other dialog tasks, we mostly rely on different reward models from OpenAssistant [86]. Though they all aim at evaluating whether an answer is adequate given a question, they differ in their predictions due to differences in their architecture and training procedures. In practice, we simply leverage them as block-box classification pipelines, implemented in the transformers library [76].

Table 1: LLaMA with RLHF experiments: key implementation details.

	Model
Architecture	Transformer [70]
Pre-training	LLaMA-7b [45]
Instruction FT	Alpaca [22]
	RL procedure
Fine-tuning strategy	LoRA [81]
	following Alpaca-LoRA [186]
LoRA alpha	16
LoRA dropout	0.05
	following trl-peft [79, 80]
Optimizer	Adam [178]
Learning rate	1.41e-5
Batch size	128
Output length	Between 16 and 32
RL algorithm	PPO [78]
KL PPO	0.05 for summary tasks else 0.2
Epochs	2 for Reuter summary else 1
Hardware	NVIDIA RTX A6000 49 Go
Compute budget	4000 GPUh
Task name	**Reuter summary**
Description	Generate a concise and clear summary of newspaper articles from Reuters.
Prompt	"Generate a one-sentence summary of this post."
Dataset	Reuter news from [82, 187] from news-summary
R_1	gpt2-reward-summarization trained here.
R_2	bart-faithful-summary-detector [85]
Figure	Figures 1(b) and 2(a)
Task name	**Reddit summary**
Description	Generate a concise and clear summary of posts from Reddit across a variety of topics (subreddits).
Prompt	"Generate a one-sentence summary of this post."
Dataset	Reddit crawl from the TL;DR dataset [83] from summarize-from-feedback [12]
R_1	gpt2-reward-summarization trained here.
R_2	bart-faithful-summary-detector [85]
Figure	Figure 2(b)
Task name	**Stack Exchange**
Description	Answer accurately to technical questions from Stack Exchange.
Prompt	No prompt, only users' questions.
Dataset	Q&A from Stack Exchange [84, 188] from stack-exchange-preferences
R_1	reward-model-deberta-v3-base
R_2	reward-model-electra-large-discriminator
Figure	Figure 2(c)
Task name	**Movie review**
Description	Generate movie reviews that accurately describe a movie.
Prompt	"Generate a movie review."
Dataset	IMDB reviews [189] from IMDB
R_1	reward-model-deberta-v3-base
R_2	reward-model-electra-large-discriminator
Figure	Figure 2(d)
Task name	**Helpful assistant**
Description	Provide helpful and harmless answers to potentially complex and sensitive questions.
Prompt	No prompt, only users' questions.
Dataset	Helpfulness and Harmlessness datasets [41] from hh-rlhf
R_1	reward-model-deberta-v3-large-v2
R_2	reward-model-electra-large-discriminator
R_3	reward-model-deberta-v3-base-v2
R_4	reward-model-deberta-v3-base
Figure	Figures 2(e) and 2(f)

D Image-to-text: captioning with diverse statistical rewards

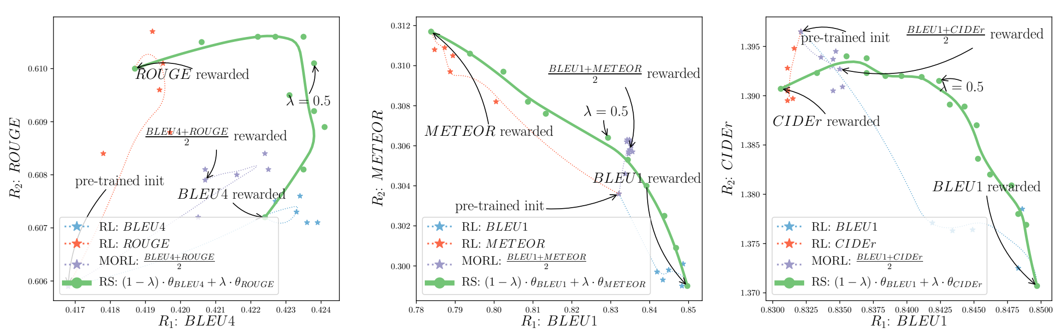
D.1 Experimental details

We summarize the key implementation details of our captioning experiments in Table 2. In short, we took the state-of-the-art network [90] for captioning on COCO, fine-tune with their code and only changing the reward. In more details, since the *self-critical* paper [24] (a variant of REINFORCE [92] with a specific estimation of the baseline score) it is now common in captioning to optimize the CIDEr reward [31] after a first step of supervised fine-training. The recent ExpansionNetv2 [90] follows this strategy to reach state-of-the-art results, with a Swin Transformer [91] visual encoder and a block static expansion for efficiency. We investigate whether additional RL trainings can help. We use their code and most of their hyperparameters, only reducing the batch size from 24 to 18 to fit in our GPUs and consequently adapt the learning rate.

Table 2: Captioning experiments: key implementation details.

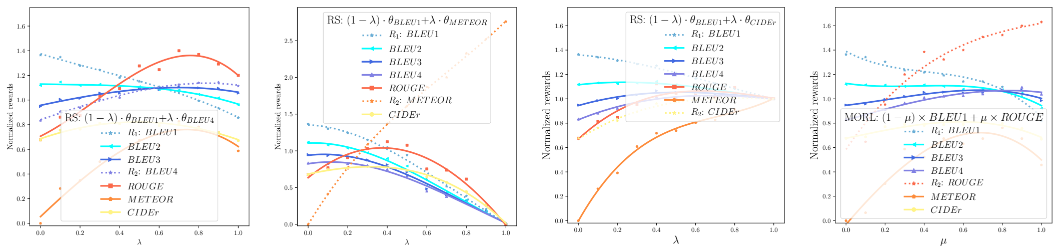
	Model
Architecture	ExpansionNetv2 [90]
Visual encoder	Swin Transformer [91]
Visual encoder pre-training	ImageNet 22k [190]
Fine-tuning	Cross-entropy then CIDEr RL [24] on COCO [88]
	RL procedure
Fine-tuning strategy	Usually frozen visual backbone, but end-to-end in Figure 9(d)
RL algorithm	self-critical [24], a variant of REINFORCE [92]
Optimizer	Radam [191]
Dataset	COCO [88] and Karpathy split [93]
Rewards	BLEU [29] (with 1-gram or 4-grams), ROUGE [30], METEOR [89], CIDEr [31]
Learning rate	1e-5
Batch size	18
Gradient accumulation	2
Warmup	Anneal 0.8 during 1 epoch
Epochs	6
Hardware	GPU V100 32G
Compute budget	1500 GPUh

D.2 Additional results



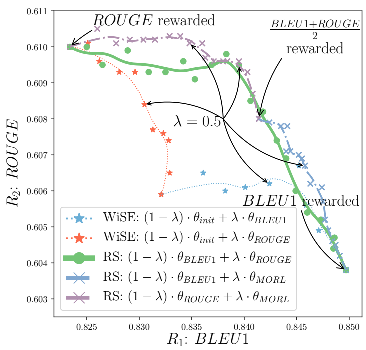
(a) BLEU4 and ROUGE. (b) BLEU1 and METEOR. (c) BLEU1 and CIDEr.

Figure 7: Additional results in captioning with more rewards, complementing Figure 3. Specifically, Figure 7(a) uses $R_1 = BLEU4$ and $R_2 = ROUGE$; then, with $R_1 = BLEU1$, Figure 7(b) uses $R_2 = METEOR$ and Figure 7(c) uses $R_2 = CIDEr$. In particular, the latter shows the failure when optimizing CIDEr; indeed, let's recall that the pre-trained initialization [90] has actually already been trained by optimizing CIDEr [24]. Thus optimizing CIDEr a second time does not help, nor in CIDEr neither in other rewards. That's why in Figure 3(c) we consider the initialization as the network parametrization optimized for CIDEr.

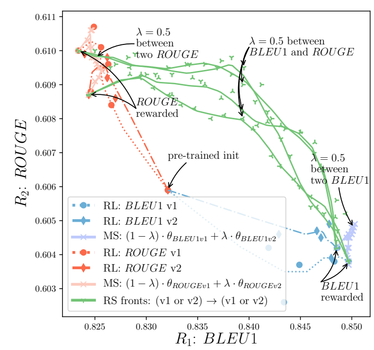


(a) $R_2 = BLEU4$. (b) $R_2 = METEOR$. (c) $R_2 = CIDEr$. (d) MORL.

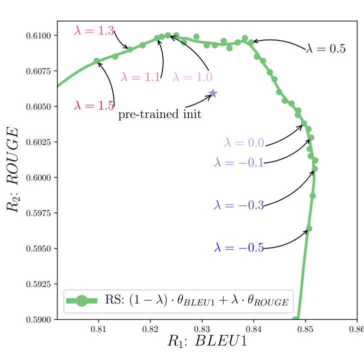
Figure 8: Additional results in captioning when measuring performances on all rewards and varying the interpolating coefficients, complementing Figure 4(b). In Figures 8(a) to 8(c), we extend the results for RS with $R_1 = BLEU1$ and for varying R_2; the optimal λ depends on the similarity between the evaluation metric and R_1 and R_2. We also see in Figure 8(c) that all rewards are normalized to 1 for the CIDEr-initialization. In Figure 8(d), we perform the same analysis for MORL while varying the weighting μ over the proxy rewards $R_1 = BLEU1$ and $R_2 = ROUGE$; we recover similar curves than in Figure 4(b) for RS.



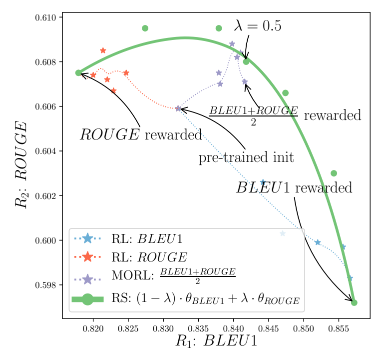
(a) Exploring new strategies.



(b) Results variances.



(c) λ outside of $[0, 1]$.



(d) End-to-end training.

Figure 9: Additional results in captioning with $R_1 = BLEU1$ and $R_2 = ROUGE$. In Figure 9(a), we investigate interpolating the fine-tuned networks with the pre-trained initialization as in WiSE [192]; this only reveals a small portion of the front. In contrast, the interpolation with θ_{MORL} ($\mu = 0.5$) solution improves RS's front: this highlights some limitations in Hypothesis 2 and strict Pareto optimality of RS. Adding the MORL solutions as *intermediate* weights may help interpolate between two weights too distant. This suggests some practical complementarity between RS and MORL; given a training budget larger than the number of rewards, one may learn a few MORL for varying $0 \leq \mu \leq 1$, and then interpolate the obtained solutions. Figure 9(b) shows results' variance with two RL trainings for BLEU, and two for ROUGE, each time with a different seed defining the data ordering and augmentations. Though we observe some randomness, the Hypothesis 1 is consistently validated. Moreover, it presents the fronts described when we interpolate weights fine-tuned on a shared reward, as in model soups (MS) [62, 63]. This also only reveals a small portion of the spectrum of preferences, validating the need of diverse rewards to reveal the full Pareto front. Figure 9(c) presents the extrapolation results when λ goes outside of $[0, 1]$. This suggests that we can artificially reduce a reward with negative coefficients, as shown in [141]. Finally, Figure 9(d) shows the results when the networks are trained end-to-end, rather than keeping the backbone frozen. This validates the efficiency of rewarded soups in a new more general setting where all layers are trainable.

E Text-to-image: diffusion models with diverse RLHFs

E.1 Experimental details

Several works have studied the problem of aligning the output of diffusion models with human feedbacks [25, 26, 33]. Models are expected to understand specific visual control signals like colors, counts, and backgrounds more accurately after alignment. Notably, diffusion models can be fine-tuned to match human aesthetics preferences. As for any subjective metric, there is a variety of reward models that capture different aspects of aesthetic preference. These models are trained in a supervised setting to match human quality ratings collected on large image datasets [33], like the AVA dataset [97]. As in the previous sections, RS allows to efficiently optimize multiple aesthetics reward models at test time, which allows adapting to the preferences of a single user.

We consider three metrics as rewards models: The cafe aesthetics model [3], trained on 3500 real-life and anime/manga images; An aesthetic score predictor based on CLIP features[4], trained on 250 000 images from the AVA dataset [97]; we also experiment with a CLIP-based NSFW detector that estimates the probability of an image being "safe" by computing the cosine similarity with the embeddings of a set of "unsafe" words. The last two reward models are used to filter the LAION dataset [193].

To fine-tune a diffusion model on a reward model R, we first generate 10000 images with the pre-trained diffusion model and compute the rewards for every generated image. Then, we fine-tune the diffusion model on the reward-weighted negative log-likelihood [25]:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}_0, Q) \in \mathcal{D}, \epsilon \sim \mathcal{N}(0,1), t \sim Uniform(0,T)} \quad r(\mathbf{x}_0) \| \epsilon_\theta(\mathbf{x}_t, t, Q) - \epsilon \|^2 \tag{6}$$

where ϵ_θ is the noise estimation network, T is the total number of training steps, $r(\mathbf{x}_0)$ is the reward of image $\mathbf{x}_0$ and Q is the text associated to image $\mathbf{x}_0$.

On-policy Reinforcement Learning would normally require to perform loops of image generation and model fine-tuning [194], but we only perform a single optimization loop for simplicity.

Implementation details. We use a 2.2B parameters diffusion model trained on an internal dataset of 300M images, which reaches similar generation quality as Stable Diffusion [96] in terms of CLIP alignment and FID scores. For efficient finetuning, we only fine-tune 10% of the diffusion model's weights [98] corresponding to the cross-attention layers and the bias/scaling parameters. We also remove the 50% images with the worse scores least and rescale rewards accordingly, see supplementary material for more details. All models are fine-tuned with Adam [178] for 4000 steps with a batch size of 32 and learning rate 5e-6. Fine-tuned checkpoints and checkpoints interpolated with RS are evaluated on 1000 images.

Table 3: Image generation experiments: key implementation details.

Model	
Architecture	TODO
Visual encoder	Swin Transformer
Fine-tuning strategy	
Optimizer	Radam
Dataset	CO
Rewards	BLEU
Learning rate	1e-5
Batch size	18
Epochs	6
Hardware	Single GPU V100 32G
Compute budget	1500 GPUh

E.2 Additional results

We show in Figure 10 the spider map when computing MORL and RS on all three metrics: *ava*, *cafe* and the *nsfw* detector. In this case, MORL has higher scores than RS on the *ava* and *cafe* scores. We

[3] available at https://huggingface.co/cafeai/cafe_aesthetic
[4] available at https://github.com/christophschuhmann/improved-aesthetic-predictor/

speculate that this is because the *nsfw* is very different from aesthetics preferences and that it can be inversely correlated with image quality: we have indeed noticed that lower quality images result in higher scores for the *nsfw* metric, being less often flagged as *unsafe*.

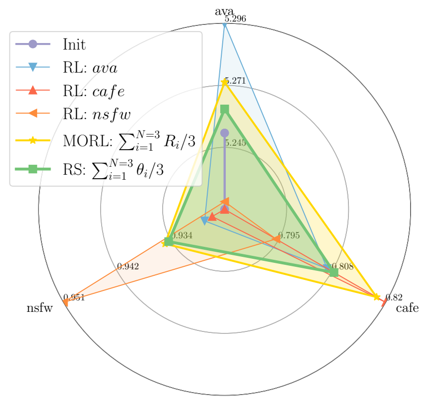


Figure 10: Image generation: spider map.

F Text-to-box: visual grounding of objects with diverse sizes

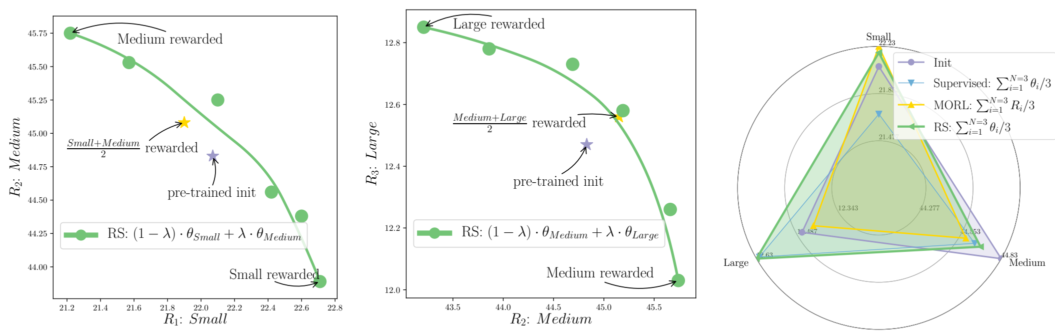
F.1 Experimental details

We show the implementation details in Table 4. We use a unified model [100, 195] pre-trained on public benchmarks, for VQA, visual grounding and image captioning. It is then fine-tuned on RefCOCO+ dataset for visual grounding. During the last fine-tuning phase, we complement the cross-entropy loss with an additional REINFORCE [92] term rewarding accuracy when the object is of the considered size. This means that the loss for θ_{Small} is $-\big(log(\hat{y}) + 5 \times 1_{\{\text{size}(\hat{y}) \text{ is small}\}} \times 1_{AUC(y,\hat{y})>0.5} \times log(y)\big)$ for an object with ground-truth box $\hat{y}$ and prediction y.

Table 4: Visual grounding experiments: key implementation details.

	Model
Architecture	Unified Model (ResNet-101+BART [196])
Visual encoder	ResNet-101
Pretraining	Cross-Entropy on Public datasets (VQA, VG, Captioning)
Finetuning	Cross-Entropy on RefCOCO+ [99]
	RL procedure
Fine-tuning strategy	end-to-end
Dataset	RefCOCO+ [99]
RL algorithm	Cross-entropy + $5\times$ REINFORCE
Optimizer	Adam
Learning rate	3e-5
Batch size	256
Epochs	10
Hardware	8 GPU 60GB
Compute budget	800 GPUh

F.2 Additional results



(a) Small and Medium. (b) Medium and Large. (c) CE vs. RL.

Figure 11: Results in visual grounding on RefCOCO+ [99]. We use REINFORCE [92] to improve directly the non-differentiable accuracy, i.e., predict boxes with IoU>0.5 w.r.t. the ground-truth. Trainings are specialized on either small, medium or large objects. These experiments complement Figures 5(b) and 5(c). Finally, Figure 11(c) compares between cross-entropy (CE) supervised fine-tuning (with Cross-entropy CE) and REINFORCE RL fine-tuning, using RS and MORL.

G Locomotion with diverse engineered rewards

G.1 Experimental details

Setup and task. This experiment consists in fine-tuning a policy that has already learned how to make an humanoid run on Brax physics engine [106].

Pre-training. We used the Brax implementation of PPO [78] algorithm to pre-train the base policy used fine-tuning (see Table 5). The goal task used for pre-training is to make a Humanoid run with the default dense reward implemented in Brax: $R = velocity - 0.5 \cdot a_t^T a_t$. This phase is also used to collected statistics about observations and normalize them before inputting to the model, which helps training a lot.

Fine-tuning. The pre-trained policy is saved while the value function is discarded. We use the normalization procedure inherited from the pre-training but freeze it. We keep the same environment. Two reward functions are designed: a *risky* one for $R_1(t) = velocity$ and a *cautious* one where $R_2(t) = velocity - \cdot a_t^T a_t$. We make a grid-search on a few hyperparameters over 3 seeds (see the values between brackets in Table 5).

Results. Overall, we can get reasonable Pareto fronts whatever parameter we choose, as shown in Figure 12.

G.2 Additional results

Table 5: Locomotion experiments: key implementation details.

PPO Pre-training	
Interactions	5e8
Reward Scaling	1.0
Episode Length	1000
Normalize observations	True
Unroll Length	10
Discounting	0.99
Learning Rate	5e-5
Entropy Cost	1e-3
Number of environments in parallel	4096
Batch Size	1024
Hardware	1GPU Tesla V100-SXM2-16GB
Runtime per experiment	80min
PPO Fine-tuning	
Interactions	1e8
Reward Scaling	1.
Normalize observations	True
Unroll Length	10
Discounting	{0.97, 0.99, 0.999}
Learning Rate	(1e-5, 3e-5, 1e-4)
Entropy Cost	1e-3, 3e-3, 1e-2
Number of environments in parallel	4096
Batch Size	1024
Hardware	1GPU Tesla V100-SXM2-16GB
Runtime per experiment	20min
Model architecture	
Policy	
Architecture	MLP
Nb of Layers	6
Hidden Size	512
Value	
Architecture	MLP
Nb of Layers	5
Hidden Size	256

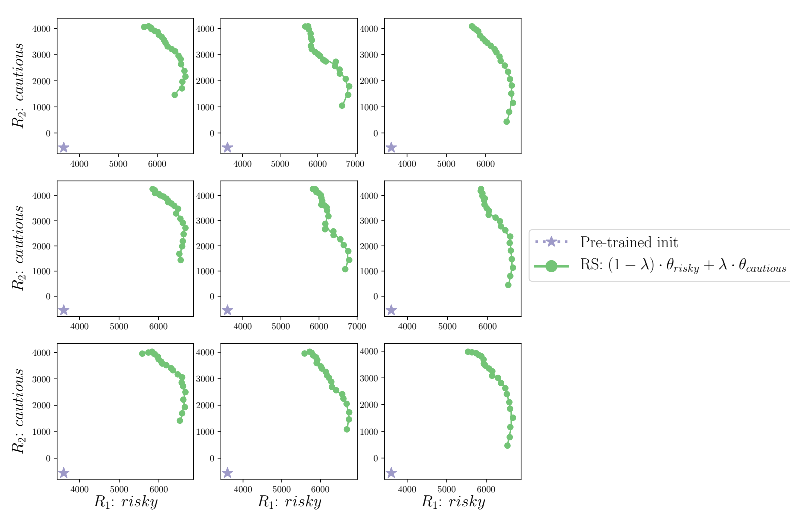


Figure 12: Some other runs for the locomotion task when varying the seed / hyperparameters.