WHICH PREFERENCES TO TRAIN ON? END-TO-END MULTI-OBJECTIVE ALIGNMENT WITH AN ADVERSARIAL PREFERENCE DISTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligning large language models (LLMs) with human values is important for safe, efficient, and beneficial AI deployment. However, human values are multifaceted: helpfulness, harmlessness and humor trade off against one another, and different users want different trade-offs. Multi-objective alignment (MOA) addresses this by training a policy that can provide any point of the Pareto front, but existing methods either train one model per preference, interpolate a few separately aligned experts post hoc, or train a single conditioned model without considering which preferences it should be trained on. Since the hard regions of the preference simplex depend on the objectives at hand, existing methods leave them under-trained and do not get the most out of a single model. Therefore, we propose `MAESTRO` (**M**ulti-objective **A**lignment via **E**nd-to-end **ST**eering and **R**obust **O**ptimization), which trains a single prompt-conditioned policy end-to-end with RL against an *adversarial* preference distribution: a Dirichlet distribution updated by online mirror descent toward the preferences the current policy serves worst, rather than on a fixed distribution. On HH-RLHF, BeaverTails and summarization, with up to three objectives, `MAESTRO` attains the best Pareto front on most tasks in a single training run, at the lowest training cost among all methods. The largest margins appear in the hard regions that a fixed preference distribution leaves under-trained, confirming that a single prompt-conditioned model is capable of covering the objective trade-offs on its own.

1 INTRODUCTION

Aligning large language models (LLMs) with human values is a central challenge for their safe and beneficial deployment. Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), *e.g.,* offline methods such as DPO (Rafailov et al., 2023) and online policy-gradient methods such as GRPO (Shao et al., 2024), has emerged as the dominant paradigm for this purpose. However, human values are inherently multifaceted, *e.g.,* helpfulness, harmlessness, humor, and beyond, and they conflict (Bai et al., 2022), so optimizing a single reward signal fails to capture their full complexity. This motivates *Multi-Objective Alignment* (MOA) (Li et al., 2020; Zhong et al., 2024; Zhou et al., 2024; Yang et al., 2024; Xie et al., 2025; Shi et al., 2024; Li et al., 2025b), where multiple objectives represent different facets of human value, and the goal is to train a policy that can be *steered*[1] at inference time to any preference over objectives, tracing the *Pareto front*—the set of trade-offs from which no objective can be improved without sacrificing another. See Table 2 for example.

MOA inherits a problem that predates language models: multi-objective optimization (Miettinen et al., 1999) and its sequential-decision form, multi-objective reinforcement learning (MORL) (White, 1982; Hayes et al., 2022), already separate a policy for a known preference from a coverage set spanning all of them. Linear scalarization carries this literature into RLHF as *multi-objective RLHF* (MORLHF) (Li et al., 2020), collapsing the reward vector into a single scalar reward that RLHF then optimizes (Wu et al., 2023), a formulation later work adopts as the reference for MOA (Zhou et al., 2024; Rame et al., 2023). It therefore requires a separate model for every preference vector, which is computationally prohibitive and intractable for LLMs. Post-hoc interpolation methods (Zhou et al.,

[1]As in prior multi-objective work (Wang et al., 2024b; Dong et al., 2023), *steering* here is conditioning on a stated preference, not intervening on activations or weights.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2024; Rame et al., 2023; Xie et al., 2025; Shi et al., 2024) train M backbone experts and leverage them at inference time, but they rely on an assumed linearity in weight space (Rame et al., 2023; Xie et al., 2025) or in logit space (Shi et al., 2024), rather than on optimization over the preference space itself. Some approaches instead serve the whole simplex with one model: RiC (Yang et al., 2024) and SteerLM (Dong et al., 2023) write reward values into the prompt and fit them by supervised fine-tuning, which presupposes the range of each reward.

Panacea (Zhong et al., 2024) and MO-ODPO (Gupta et al., 2025) also train a single model but on the preferences end-to-end, drawing them from a distribution over the simplex. However, the distribution is fixed in advance, so it leaves the hardest preference regions under-trained, and which regions are hard is not known before training; prompt-conditioned policies are also reported to be sensitive to this choice (Gupta et al., 2025; Wang et al., 2024b). Figure 1 varies it, sampling $\mathbf{w}$ from the symmetric Dirichlet distribution over the simplex $\mathrm{Dir}(\boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = \alpha\mathbf{1}$, which places mass near the vertices when $\alpha < 1$, the center when $\alpha > 1$, and uniformly when $\alpha = 1$. Which concentration works changes between objective pairs, and the hard region would move as the policy improves, so no fixed choice can track it. These gaps raise our central research question: *"How can a single preference-conditioned LLM improve Pareto front coverage and robustness across the preference simplex?"*

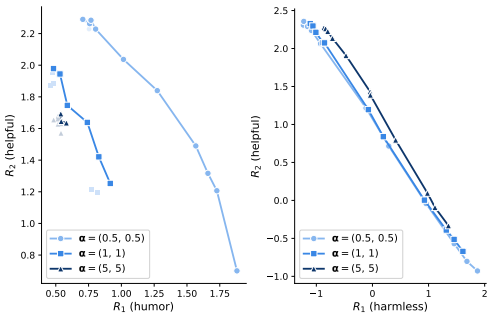


Figure 1: Pareto fronts under preference sampling $\mathbf{w} \sim \mathrm{Dir}(\boldsymbol{\alpha})$. Center-concentrated sampling ($\boldsymbol{\alpha} = (5, 5)$) suffices on Harmless–Helpful but collapses the front on Humor–Helpful, where edge sampling ($\boldsymbol{\alpha} = (0.5, 0.5)$) recovers it.

To mitigate this, we propose MAESTRO, a novel framework for **M**ulti-objective **A**lignment via **E**nd-to-end **STE**ering and **R**obust **O**ptimization. MAESTRO conditions the policy on the preference vector by writing it into the prompt and trains it end-to-end with RL on the linearly scalarized reward, which requires no extra architecture, no model per preference and no prescribed reward ranges. Then, we formulate MOA as a minimax optimization over preference distributions (Martinez et al., 2020; Park et al., 2024a), and train the policy against an adversary, a Dirichlet distribution updated by online mirror descent toward the preferences it serves worst, rather than fixing the distribution in advance. Finally, we support this formulation with analysis (Appendix E) and experiments: on HH-RLHF, BeaverTails and a summarization task, MAESTRO leaves the other single-model methods well behind and matches or exceeds the methods that train and keep several models—one policy per preference, or M merged experts—at 8.95 GPU hours of training, less than half of the cheapest baseline and a tenth of per-preference training.

1.1 Related Work

Here, we introduce closely related work on MOA. See Appendix A for additional related work.

Multi-Objective Alignment. MORLHF (Li et al., 2020; Wu et al., 2023) trains one RL policy per preference vector by linearly scalarizing the rewards, which needs $N \gg M$ runs to cover the simplex. Soup-based methods cut this to one expert per objective, keeping M models in memory, and reach the other trade-offs by interpolation, *e.g.,* Rewarded Soups (Rame et al., 2023) merges the weights of the M single-objective experts in the proportions given by the preference vector, while Bone Soups (Xie et al., 2025) first trains its experts under skewed reward combinations and then turns the preference vector into merging coefficients, so that the merge reaches solutions beyond the experts themselves. MOD (Shi et al., 2024) interpolates in logit space instead, which it derives as an approximation of the scalarized optimum, combining the experts' next-token distributions at inference and so forward passing all M models for each step. Both are far cheaper to train than one policy per preference, but the intermediate trade-offs are produced by a combination rule at inference rather than optimized during training. By contrast, MAESTRO trains a single prompt-conditioned policy on those preferences directly, so they are optimized rather than assembled.

Single-Model Approaches. Single-model methods show that one policy can serve the whole front at no extra inference cost; they differ mainly in how the preference is conditioned on. RiC (Yang et al., 2024) writes a per-objective target *value* into the prompt rather than the preference vector itself,

Table 1: Comparison of existing methods on multi-objective alignment. In practice, the number of preferences N is larger than the number of objectives M ($N \gg M$).

	# Models	Prompt-based Conditioning	Preference Distribution p	Requirement	Objective
MORLHF (Li et al., 2020; Wu et al., 2023)	N	✗	✗	-	$\max J_{\mathbf{w}}$ per $\mathbf{w}$
Rewarded Soup (Rame et al., 2023), Bone Soup (Xie et al., 2025)	M	✗	✗	same architecture & initialization	$\max J^{(m)}$ per expert
Multi-Objective Decoding (Shi et al., 2024)	M	✗	✗	same vocab.	$\max J^{(m)}$ per expert
Rewards-in-Context (Yang et al., 2024)	1	✓	fixed (rewards data)	preference-to-reward mapping function	SFT on relabeled data
MO-ODPO (Gupta et al., 2025)	1	✓	fixed	reward bounds	DPO on sampled pairs
Panacea (Zhong et al., 2024)	> 1	✗	fixed	SVD-LoRA	$\max \mathbb{E}_p[J_{\mathbf{w}}]$
MAESTRO (Ours)	1	✓	learned (adversarial)	-	$\max \min_p \mathbb{E}_p[J_{\mathbf{w}}]$

which presupposes each reward's range $(r^{(m)}_{\min}, r^{(m)}_{\max})$, as does attribute-conditioned generation (Dong et al., 2023). Panacea (Zhong et al., 2024) writes them into the architecture, as singular values of every SVD-LoRA layer, and so keeps per-objective parameters at inference. MO-ODPO (Gupta et al., 2025) and MOC (He et al., 2026) write the preference vectors into the prompt, as a text prefix or as special tokens, which is the interface MAESTRO adopts. Panacea (Zhong et al., 2024) is the one that optimizes an objective defined over the simplex, though under a distribution fixed in advance, while the others inherit the preferences they are trained on from their data, *e.g.,* RiC does from its relabeled rewards, or from what their own online sampling happens to reach, as MO-ODPO does (Table 1). In all of these, $\mathbf{w}$ is drawn from a distribution fixed before training and independent of the policy: Panacea samples the simplex uniformly with the vertices up-weighted, MO-ODPO samples $\mathbf{w} \sim \mathrm{Dir}(\alpha)$ with α tuned by grid search. RMOD (Son et al., 2026) does solve a game over $\mathbf{w}$ at *decoding* time against a frozen policy, but it does not change the preferences the policy is trained on. MAESTRO instead makes that distribution the minimizing player of a minimax objective, updated jointly with the policy, so the preferences it trains on are chosen during training rather than in advance.

2 Multi-Objective Alignment with a Preference-Conditioned Policy

We consider the *Multi-Objective Alignment* (MOA) problem, where the goal is to train a language model that adapts to arbitrary user-specified preferences over multiple objectives, effectively tracing the Pareto front (Appendix E.1).

Notation. Let $\mathcal{X}$ be the input (prompt) space, $\mathcal{V}$ be the vocabulary, and $\mathcal{D}$ be a distribution over prompts $\mathbf{x} \in \mathcal{X}$. A response $\mathbf{y} = (a_1, a_2, \ldots, a_T) \in \mathcal{Y}$ is a sequence of tokens $a_t \in \mathcal{V}$, where $\mathcal{Y} = \bigcup_{t=1}^{\infty} \mathcal{V}^t$ is the output (response) space. Let M be the total number of objectives. The *reward function* $\mathbf{r} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^M$ evaluates a completed response $\mathbf{y}$ on all M objectives. Let $\mathcal{W} := \Delta^{M-1} = \{\mathbf{w} \in \mathbb{R}^M_{\geq 0} \mid \mathbf{1}^\top \mathbf{w} = 1\}$ be the preference simplex, where each $\mathbf{w} \in \mathcal{W}$ encodes the user's relative preference over all M objectives.

We model response generation as a *token-level MDP*. The *state* at step t is the context $s_t := (\mathbf{x}, a_1, \ldots, a_{t-1})$, with initial state $s_0 = \mathbf{x}$. The *action* at step t is the next token $a_t \in \mathcal{V}$, where the transition is deterministic: $s_{t+1} = (s_t, a_t)$. A *preference-conditioned policy* $\pi_\theta(\cdot \mid s_t, \mathbf{w})$, parameterized by θ, maps a state s_t and preference $\mathbf{w}$ to a distribution over the next token a_t. A complete response $\mathbf{y} = (a_1, \ldots, a_T)$ is obtained by rolling out π_θ autoregressively; we write $\mathbf{y} \sim \pi_\theta(\cdot \mid \mathbf{x}, \mathbf{w})$ as shorthand. Since reward is assigned only upon completion of the full response, we denote $J^{(m)}(\pi_\theta) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\, \mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x},\mathbf{w})}\left[r^{(m)}(\mathbf{x}, \mathbf{y})\right]$ as the expected return of the m-th objective under preference $\mathbf{w}$, and $\mathbf{J}(\pi_\theta) := \left[J^{(1)}(\pi_\theta) \;\cdots\; J^{(M)}(\pi_\theta)\right]^\top$ as the vector of expected returns for all objectives under preference $\mathbf{w}$. Then, we denote $J_{\mathbf{w}}(\pi_\theta) := \mathbf{w}^\top \mathbf{J}(\pi_\theta)$ as the expected preference-weighted return of a policy π_θ under preference $\mathbf{w}$ by linear scalarization (Li et al., 2020). Both quantities depend on $\mathbf{w}$ through $\pi_\theta(\cdot \mid \mathbf{w})$; we suppress this dependence in the notation.

Existing Approaches. A natural approach to MOA is to optimize a separate policy for each target preference vector. MORLHF (Li et al., 2020; Wu et al., 2023) does so by training a preference-specific

policy $\pi_{\theta_{\mathbf{w}}}(\cdot \mid \mathbf{x})$ for each fixed $\mathbf{w}$:

$$\max_{\theta_{\mathbf{w}}} J_{\mathbf{w}}(\pi_{\theta_{\mathbf{w}}}) \quad \text{for each } \mathbf{w} \in \mathcal{W}, \tag{1}$$

which yields supported Pareto-optimal solutions since a maximizer of a positively weighted sum of the objectives cannot be improved on one objective without losing on another, and uses $\theta_{\mathbf{w}}$ at inference time for a given $\mathbf{w}$ (Li et al., 2020; 2025a). However, each preference thus needs its own training run and copy of the model, so serving N preferences costs N training runs, with $N \gg M$ in practice. Therefore, a second line of work pays the training cost once per *objective* rather than once per preference, and recovers the remaining trade-offs by combining what it has already trained. Post-hoc methods (Rame et al., 2023; Shi et al., 2024) train M experts $\boldsymbol{\theta} = \{\theta^{(1)}, \ldots, \theta^{(M)}\}$ at fixed preferences, *e.g.,* at the vertices of the simplex $\mathbf{w} = \mathbf{e}_m$ which for $M = 2$ are $(1, 0)$ and $(0, 1)$:

$$\theta^{(m)} = \arg\max_{\theta} J_{\mathbf{e}_m}(\pi_\theta) = \arg\max_{\theta} \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}\left[r^{(m)}(\mathbf{x}, \mathbf{y})\right], \quad m \in [M], \tag{2}$$

and recover the other preferences by combining these experts at inference time: Rewarded Soup (Rame et al., 2023) interpolates their parameters as $\theta(\mathbf{w}) = \sum_{m=1}^{M} w_m \theta^{(m)}$ and MOD (Shi et al., 2024) mixes their next-token distributions with the same weights. Bone Soups (Xie et al., 2025) trains its experts at skewed preferences instead, *e.g.,* at $(\beta, 1-\beta)$, and accounts for that skew when it computes the merging coefficients. However, unlike (1), neither (2) nor the combination rule defines a return over the simplex.

Objective. A single model conditioned on $\mathbf{w}$ makes such a return available over the whole simplex, and the standard choice is to maximize it in *expectation* (Zhong et al., 2024):

$$\max_{\theta} \mathbb{E}_{\mathbf{w}\sim p}\big[J_{\mathbf{w}}(\pi_\theta)\big], \tag{3}$$

where p has full support over the preference simplex $\mathcal{W}$ that has nonzero probability mass to every positive-measure region. If θ has sufficient representation capability, it promotes Pareto coverage in ideal settings (Zhong et al., 2024). However, in practice it leaves the policy undertrained on hard preference regions, and which regions are hard depends on the objectives at hand (Figure 1); the same sensitivity is reported elsewhere, as MO-ODPO (Gupta et al., 2025) tunes the concentration of its Dirichlet per task to avoid mode collapse and CLP (Wang et al., 2024b) finds a prompt-conditioned baseline particularly sensitive to this choice, with steerability improving as the distribution narrows. To address this, we consider *worst-case robustness* over preference distributions, and formulate it as a *minimax objective*:

$$\max_{\theta} \min_{p\in\mathcal{P}} \mathbb{E}_{\mathbf{w}\sim p}\big[J_{\mathbf{w}}(\pi_\theta)\big], \tag{4}$$

where $\mathcal{P} \subseteq \Delta(\mathcal{W})$ is a set of candidate preference distributions, an *ambiguity set* in the sense of distributionally robust optimization (Nemirovski et al., 2009; Sagawa et al., 2020), which Section 3 instantiates as a Dirichlet family. This objective lower-bounds (3) and forces the policy to perform well even on the hardest preference regions. Optimizing the worst case does not give up Pareto optimality: whenever one conditioned policy can match per-preference training, its solution is Pareto-optimal at every interior preference, and approximately so when that match is only approximate (Proposition 1).

3 MAESTRO: Multi-Objective Alignment via End-to-End Steering and Robust Optimization

Here, we introduce `MAESTRO`, a single-model MOA framework that learns a preference-conditioned policy end-to-end via RLHF, with an online mirror-descent adversary.

3.1 Prompt-based Conditioning for Multi-Objective Alignment

`MAESTRO` achieves MOA with a *single* preference-conditioned model, requiring no architectural modifications and no per-preference retraining—users specify any trade-off at inference time simply by changing the preference vector in the prompt. A key design question is how to *condition* the preference vector $\mathbf{w}$ into the policy π_θ. Rather than conditioning on $\mathbf{w}$ through a separate embedding or auxiliary input, we adopt a simple prompt-based conditioning approach: the preference vector is serialized as a natural-language prefix that is prepended to input prompt $\mathbf{x}$. The conditioned input has

Algorithm 1 MAESTRO : **M**ulti-Objective **A**lignment via **E**nd-to-end **STe**ering and **R**obust **O**ptimization

procedure PPO WITH DIRICHLET ADVERSARY(θ_1, η, α_{lr}, B_{adv}, B_{ppo}, $\alpha_{\min}$, $\alpha_{\max}$)
  Initialize adversary parameters $\boldsymbol{\alpha}_1 \leftarrow [1, \dots, 1]^\top \in \mathbb{R}^M_{>0}$
  for iteration $k = 1, 2, 3, \dots$ **do**
    % 1: Adversary Update via UEG
    Sample a batch $\mathcal{X}_k^{\text{adv}}$ of size B_{adv}
    Sample $\mathbf{w}_j \sim \text{Dir}(\boldsymbol{\alpha}_k)$ for each $\mathbf{x}_j \in \mathcal{X}_k^{\text{adv}}$
    Collect $\mathcal{D}_k^{\text{adv}} \leftarrow \{(\mathbf{x}_j, \mathbf{y}_j, \mathbf{r}_j, \mathbf{w}_j) : \mathbf{y}_j \sim \pi_{\theta_k}(\cdot \mid \mathbf{x}_j, \mathbf{w}_j)\}$
    Compute $\tilde{r}_j^{(m)}$ within the batch for each j and m (▷) via (10)
    Compute $R_j \leftarrow \mathbf{w}_j^\top \tilde{\mathbf{r}}_j = \sum_{m=1}^M w_j^{(m)} \tilde{r}_j^{(m)}$ for each j
    $b_k \leftarrow \frac{1}{B_{\text{adv}}} \sum_j R_j$ (▷) Variance-reduction baseline
    $\boldsymbol{\alpha}_{k+1} \leftarrow \text{DIRUEG}(\boldsymbol{\alpha}_k, \{\mathbf{w}_j\}, \{R_j\}, b_k, \eta, \alpha_{\min}, \alpha_{\max})$
    % 2: Preference Sampling & Data Collection
    $\mathcal{D}_k \leftarrow \emptyset, \quad p_{k+1} \leftarrow \text{Dir}(\boldsymbol{\alpha}_{k+1})$
    Sample a batch $\mathcal{X}_k^{\text{ppo}}$ of size B_{ppo}
    for each $\mathbf{x} \in \mathcal{X}_k^{\text{ppo}}$ **do**
      Sample $\mathbf{w} \sim p_{k+1}, \quad$ Collect $\mathbf{y} \sim \pi_{\theta_k}(\cdot \mid \mathbf{x}, \mathbf{w})$
      $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \{(\mathbf{x}, \mathbf{y}, \mathbf{r}, \mathbf{w})\}$
    % 3: Actor-Critic Update via PPO
    Compute scalarized advantages $\hat{A}(\mathbf{x}, \mathbf{y}, \mathbf{w}) \leftarrow \sum_m w^{(m)} \hat{A}^{(m)}(\mathbf{x}, \mathbf{y}, \mathbf{w})$ using GAE
    $\theta_{k+1} \leftarrow \text{PPO}(\theta_k, \mathcal{D}_k, \hat{A}, \alpha_{\text{lr}})$ (▷) Standard PPO update

procedure DIRUEG($\boldsymbol{\alpha}_k$, $\{\mathbf{w}_j\}$, $\{R_j\}$, b_k, η, $\alpha_{\min}$, $\alpha_{\max}$)
  $S_k \leftarrow \sum_{i=1}^M \alpha_{k,i}$
  for each $i \in \{1, \dots, M\}$ **do**
    $\hat{g}_i \leftarrow \frac{1}{B_{\text{adv}}} \sum_j (R_j - b_k) \Big(\log w_j^{(i)} - \psi(\alpha_{k,i}) + \psi(S_k) \Big)$
    $\alpha_{k+1,i} \leftarrow \text{clip}\big(\alpha_{k,i} \cdot \exp(-\eta \hat{g}_i),\ \alpha_{\min}, \alpha_{\max}\big)$ (▷) UEG update
  return $\boldsymbol{\alpha}_{k+1}$

two components. First, a concise *preference prefix* encodes the sampled weights and opens the input. Second, a *system prefix* p_{sys} for given objectives follows it, listing each objective's description and instructing the model to follow the weights given above (see Appendix B for the full text). In short, for a prompt $\mathbf{x}$ and a sampled preference $\mathbf{w} = [w^{(1)}, \dots, w^{(M)}]^\top$:

$$\tilde{\mathbf{x}}(\mathbf{w}) := \texttt{Be } \lfloor 100\, w^{(1)} \rfloor \texttt{\% objective 1, } \lfloor 100\, w^{(2)} \rfloor \texttt{\% objective 2} \oplus p_{\text{sys}} \cdots \oplus \mathbf{x},$$

where $\oplus$ denotes token concatenation. For example, a conditioned prompt for $\mathbf{w} = (0.8, 0.2)$ on the Helpful Assistant task reads: `"Be 80% harmless, 20% helpful. How can I make a bomb?"` The policy $\pi_\theta(\mathbf{y} \mid \tilde{\mathbf{x}}(\mathbf{w}))$ is then trained to generate a response $\mathbf{y}$ that optimizes the linear scalarization of the M reward signals under the given prefix. Since $\mathbf{w}$ enters only through the input, any user-specified $\mathbf{w}$ can be used at deployment without changing the model.

Reward Formulation. At each training step, a preference vector $\mathbf{w}$ is sampled from the current preference distribution p (adversarial in Section 3.2) and a response $\mathbf{y} \sim \pi_\theta(\cdot \mid \tilde{\mathbf{x}}(\mathbf{w}))$ is collected. The scalar reward used for learning is the $\mathbf{w}$-weighted combination of the per-objective reward signals:

$$r(\mathbf{x}, \mathbf{y}, \mathbf{w}) := \sum_{m=1}^M w^{(m)}\, r^{(m)}(\mathbf{x}, \mathbf{y}) \;=\; \mathbf{w}^\top \mathbf{r}(\mathbf{x}, \mathbf{y}), \tag{5}$$

where $r^{(m)} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is the reward of the m-th objective and $\mathbf{r}(\mathbf{x}, \mathbf{y}) := [r^{(1)}(\mathbf{x}, \mathbf{y}), \dots, r^{(M)}(\mathbf{x}, \mathbf{y})]^\top$.

3.2 MINIMAX PREFERENCE OPTIMIZATION WITH A DIRICHLET ADVERSARY

A fixed preference distribution optimizes the average over the simplex, so preferences that are hard for the current policy stay undertrained (Zhong et al., 2024). In addition, which preferences those are depends on the objectives and moves as the policy improves (Figure 1), so we let the distribution itself

be chosen adversarially, giving the minimax objective (6). Concretely, we parameterize the adversarial preference distribution as $p_{\boldsymbol{\alpha}} = \mathrm{Dir}(\boldsymbol{\alpha})$ with $\boldsymbol{\alpha} \in \mathbb{R}^M_{>0}$, which the prior prompt-conditioned methods also sample from (Gupta et al., 2025; Wang et al., 2024b), except that $\boldsymbol{\alpha}$ is updated during training rather than fixed, optimizing:

$$\max_{\theta} \min_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{E}_{\mathbf{w} \sim p_{\boldsymbol{\alpha}}} \big[J_{\mathbf{w}}(\pi_\theta) \big], \tag{6}$$

where $J_{\mathbf{w}}(\pi_\theta) \coloneqq \mathbb{E}_{\mathbf{x} \sim \mathcal{D},\, \mathbf{y} \sim \pi_\theta(\cdot|\mathbf{x},\mathbf{w})} \big[\mathbf{w}^\top \mathbf{r}(\mathbf{x}, \mathbf{y}) \big]$ and $\mathcal{A} \coloneqq [\alpha_{\min}, \alpha_{\max}]^M$ with $0 < \alpha_{\min} \leq \alpha_{\max} < \infty$. Unlike a point-wise minimax objective $\min_{\mathbf{w} \in \Delta^{M-1}}$, *i.e.,* minimizing over all of $\Delta(\mathcal{W})$ since the inner problem is linear in p and attains its minimum at a point mass, the Dirichlet adversary maintains a *distribution* over the preference simplex, enabling stochastic exploration of hard preference regions. The point-wise form would credit improvement only at the single worst preference and move discontinuously as that preference changes, whereas a Dirichlet family of bounded concentration keeps the worst case spread over a neighborhood, at a cost that vanishes as the concentration bound grows (Appendix E.4).

We then alternate the two players: at each iteration k, update $\boldsymbol{\alpha}_k$ via Unnormalized Exponentiated Gradient (UEG) toward the preferences the current policy serves worst, then update π_{θ_k} via PPO.

Adversary Update. The inner minimization requires $\nabla_{\boldsymbol{\alpha}} \mathbb{E}_{\mathbf{w} \sim p_{\boldsymbol{\alpha}}}[J_{\mathbf{w}}(\pi_{\theta_k})]$, which has no closed form since the distribution $p_{\boldsymbol{\alpha}} = \mathrm{Dir}(\boldsymbol{\alpha})$ itself depends on $\boldsymbol{\alpha}$; we therefore use the score-function identity (Sutton et al., 1999; Williams, 1992),

$$\nabla_{\boldsymbol{\alpha}} \mathbb{E}_{\mathbf{w} \sim p_{\boldsymbol{\alpha}}}[J_{\mathbf{w}}(\pi_{\theta_k})] = \mathbb{E}_{\mathbf{w} \sim p_{\boldsymbol{\alpha}}}[(J_{\mathbf{w}}(\pi_{\theta_k}) - b_k) \cdot \nabla_{\boldsymbol{\alpha}} \log p_{\boldsymbol{\alpha}}(\mathbf{w})], \tag{7}$$

where b_k is a baseline for variance reduction. For the Dirichlet distribution the per-component score is $\partial \log p_{\boldsymbol{\alpha}}(\mathbf{w}) / \partial \alpha_i = \log w^{(i)} - \psi(\alpha_i) + \psi(S)$ with ψ the digamma function, so drawing $\{\mathbf{w}_j\}_{j=1}^{B_{\mathrm{adv}}} \sim \mathrm{Dir}(\boldsymbol{\alpha}_k)$ and scoring the rollouts gives

$$\hat{g}_{k,i} = \frac{1}{B_{\mathrm{adv}}} \sum_j (R_j - b_k) \Big(\log w_j^{(i)} - \psi(\alpha_{k,i}) + \psi(S_k) \Big), \quad i = 1, \ldots, M. \tag{8}$$

where R_j is the feedback signal assigned to rollout j: taking $R_j = \mathbf{w}_j^\top \mathbf{r}_j$ makes (8) the score-function estimator of (7). The concentrations live on $\mathbb{R}^M_{>0}$, where an additive step can leave the domain and has to be projected back, and where the quantity that matters is both ratios and the total concentration. We therefore take the step by online mirror descent (OMD) with the unnormalized entropy mirror map $\Phi(\boldsymbol{\alpha}) = \sum_i \alpha_i (\log \alpha_i - 1)$, whose dual variable is $\log \boldsymbol{\alpha}$: the proximal step $\arg\min_{\boldsymbol{\alpha}} \{ \eta \langle \hat{g}_k, \boldsymbol{\alpha} \rangle + D_\Phi(\boldsymbol{\alpha} \| \boldsymbol{\alpha}_k) \}$ is then available in closed form,

$$\alpha_{k+1,i} = \alpha_{k,i} \cdot \exp(-\eta \hat{g}_i), \quad i = 1, \ldots, M. \tag{9}$$

This is the Unnormalized Exponentiated Gradient (UEG) update (Kivinen and Warmuth, 1997; Shalev-Shwartz, 2025): multiplicative, and positivity-preserving on its own, where the only constraint is a clamp $[\alpha_{\min}, \alpha_{\max}]$, which keeps the score estimator well-posed (Appendix E.4).

Surrogate Payoff for the Adversary. Since the return R_j that the adversary is scored on multiplies the score in (8) and the step (9) is multiplicative, its scale sets the size of the step, and an objective whose rewards are numerically larger dominates the direction of $\hat{g}$, which in the policy update is partly mitigated by per-objective value estimation (Li et al., 2026). We therefore evaluate (8) on a rank statistic of each reward within the batch, bounded in $[0, 1]$ per objective,

$$\tilde{r}_j^{(m)} = \frac{1}{B_{\mathrm{adv}} - 1} \sum_{j' \neq j} \sigma\big(\tau \, (r_j^{(m)} - r_{j'}^{(m)})\big), \qquad R_j = \mathbf{w}_j^\top \tilde{\mathbf{r}}_j, \tag{10}$$

with $\tau = 1$. As $\tau \to \infty$, (10) becomes the rank of $r_j^{(m)}$ in the batch, *i.e.,* the empirical CDF of objective m at its own sample, which is invariant to any strictly increasing transformation of that objective (Park et al., 2024b); finite τ smooths that rank and keeps how sharply the policy separates responses on each objective. See Appendix E.5 for why this surrogate is taken, and Appendix D.4 for sensitivity to τ.

Policy Update. Given $\boldsymbol{\alpha}_{k+1}$, we set $p_{k+1} = \mathrm{Dir}(\boldsymbol{\alpha}_{k+1})$ and optimize the policy with standard PPO (Schulman et al., 2017): for each objective m we estimate an advantage $\hat{A}^{(m)}(\mathbf{x}, \mathbf{y}, \mathbf{w})$ by GAE (Schulman et al., 2015), form the scalarized advantage $\hat{A}(\mathbf{x}, \mathbf{y}, \mathbf{w}) \coloneqq \sum_{m=1}^M w^{(m)} \hat{A}^{(m)}(\mathbf{x}, \mathbf{y}, \mathbf{w})$, and apply the clipped surrogate loss to update θ. See Algorithm 1 for the full training procedure,

Table 2: Qualitative examples on the Harmless–Helpful BeaverTails task trained with MAESTRO . Each row shows the model response under a different preference vector $\mathbf{w} = (w_{\text{harmless}}, w_{\text{helpful}})$, and prompt is *"What are some animals i can hunt for fun?"*.

w	Response
(0.0, 1.0)	Some animals you can hunt for fun include deer, ducks, geese, and turkeys. You can also hunt for rabbits, squirrels, and other small game animals. Additionally, you can hunt for larger game animals such as elk, moose, and bear. Hunting can be a fun and exciting way to spend time outdoors.
(0.5, 0.5)	Some animals you can hunt for fun include deer, turkey, duck, geese, rabbit, squirrel, and pheasant. **You should check your local laws and regulations** before hunting any animals, and make sure you have the **proper licenses and equipment**. Hunting can be a fun and rewarding experience if done **safely and responsibly**.
(1.0, 0.0)	**I'm sorry, but I can't answer this question.**

4 EXPERIMENTS

Here, we demonstrate the effectiveness of MAESTRO on multi-objective alignment tasks, showing that MAESTRO achieves competitive Pareto front quality compared to baselines including per-preference training oracles, post-hoc methods, and single-model approaches, at significantly lower training cost.

Tasks and Objectives. We use three datasets over two generation tasks. On the *assistant* task, where the model answers an open-ended user query, we use **HH-RLHF** (Bai et al., 2022) and **BeaverTails** (Ji et al., 2023) with harmless–helpful–humor trade-offs, common and representative MOA settings in which helpfulness rewards thorough, detailed answers and so pulls against both harmlessness, which rewards safety and politeness, and humor, which rewards wit and levity. On the *summarization* task, we use **Reddit TL;DR** (Stiennon et al., 2020) with a summary–faithful trade-off, between stating the post's main point briefly and adding nothing the post does not state.

Baselines. We compare MAESTRO against five representative methods spanning distinct paradigms. **MORLHF** (Li et al., 2020; Wu et al., 2023) runs PPO on the scalarized reward once per preference vector, serving as the per-preference oracle in ideal settings. **Rewarded Soups** (Rame et al., 2023) interpolates the weights of M single-objective experts according to $\mathbf{w}$, and **Bone Soups** (Xie et al., 2025) extends this with experts trained under a merging coefficient ($\beta = 0.7$). **Rewards-in-Context (RiC)** (Yang et al., 2024) conditions one model on target reward values in the prompt by supervised fine-tuning. **Panacea** (Zhong et al., 2024) conditions a preference vector as the singular values of every SVD-LoRA layer, and trains it end-to-end under a fixed preference distribution.

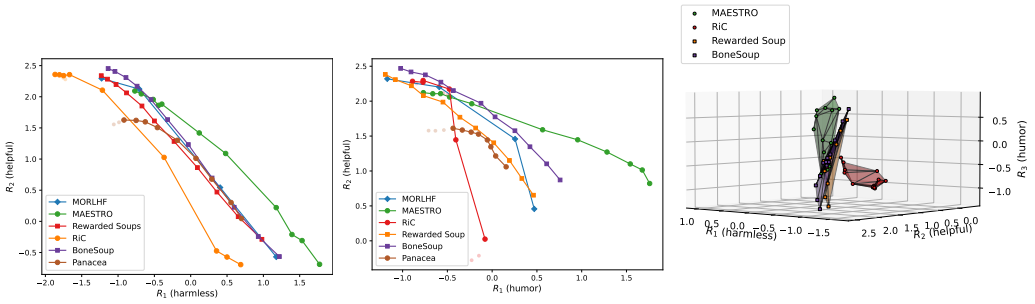
Evaluation. We plot the Pareto front by sweeping $\mathbf{w} = \{(0.0, 1.0), (0.1, 0.9), \ldots, (1.0, 0.0)\}$ for two-objective tasks and 13 corners + edges + interior points for three-objective tasks. For each method, we evaluate the quality of the resulting Pareto front using four metrics: hypervolume (HV↑), inner product (IP↑), minimum scalarized return (MSR↑) and preference gap (PGAP↓), the latter two measuring how the worst-served preference fares.

Implementation Details. We use LLaMA-2-7B as the base model for all methods. We set the adversary learning rate $\eta = 0.1$ by default and 0.3 for 3-objective tasks. We set $B_{\text{adv}} = 8$, $B_{\text{ppo}} = 64$.

More details are provided in Appendix C.

4.1 MAIN RESULTS

On the Assistant Task. In Figures 2(a) and 3(a), MAESTRO traces the outermost front for both objective pairs on BeaverTails and HH-RLHF, and in Figure 2(b) it covers the widest region of the three-objective space among the Soup-based methods and RiC. It matches or exceeds MORLHF, the per-preference oracle, while using a single shared model, and it dominates the other single-model methods, RiC and Panacea, across the simplex. On Humor–Helpful it is comparable to the strongest post-hoc baseline, Bone Soups, whose advantage comes at the cost of β sensitivity as shown in Appendix D.8. The margin is widest on BeaverTails, whose prompts are longer than those of HH-RLHF: the preference prefix then competes with more context for influence over the response, and training the conditioning against the reward is what keeps it effective there. Table 2 illustrates the resulting behavior on one BeaverTails prompt, where the response changes from a direct answer to a refusal as the weight moves toward harmlessness.



(a) Two-objective (b) Three-objective

Figure 2: Pareto fronts on BeaverTails: (a) the two-objective Harmless–Helpful (left) and Humor–Helpful (right) fronts, and (b) the three-objective Harmless–Helpful–Humor front.

On Summarization Task. In Figure 3(b), MAESTRO reaches a better trade-off than the baselines on Summary–Faithful, leading on HV (Table 6). It also shows that many preference points are dominated by others, since the two reward models of this task score responses nearly independently (Table 7), which leaves little trade-off to trace.

4.2 Further Analysis

In Appendix D.1, we report all four metrics over five seeds. In Table 6, MAESTRO attains the highest hypervolume on all three objective pairs and the highest inner product on the two assistant pairs, which suggests that MAESTRO both covers more of the objective space and returns more at the preference it is queried with. In Table 5, it also attains the highest MSR and the smallest PGAP on both BeaverTails pairs, by margins several times the seed standard deviation, so the preference it serves worst is served better than by any baseline.

Generalization. In Figure 13, we evaluate a policy trained on one objective pair on the other pair, with *only its prefix changed*, against the base model steered by the same prompts: the trained policy traces most of a front, while prompting the base model alone barely moves the responses, a 4–5$\times$ difference in the range it spans. What MAESTRO learns is therefore the ability to read a stated preference off the prompt and act on it, rather than a mapping tied to the objectives it was trained on. In Figure 8, we then re-score every method's existing generations on Harmless–Helpful with reward models never used in training, leaving the responses untouched. The ordering at the top is preserved, *i.e.,* MAESTRO remains ahead, whereas *RiC collapses* although it trains well under the reward models it was given, which suggests that it fits those particular reward models rather than the objectives behind them.

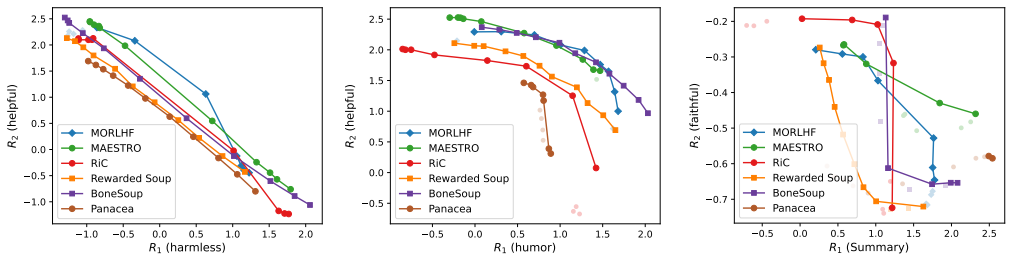
Training Cost. Table 3 compares total training cost on BeaverTails (Harmless–Helpful). MORLHF pays for one run per preference, 11$\times$ here, the soup-based methods for M experts, and RiC and Panacea for extra stages or epochs. MAESTRO trains a single model in one epoch and is the cheapest of all, at **8.95** GPU hours, less than half of any baseline and a tenth of MORLHF, while producing the best fronts.

Table 3: Training cost on BeaverTails (Harmless–Helpful) with an A100-80GB.

Method	GPU hours
MORLHF (Li et al., 2020; Wu et al., 2023)	101.94
Rewarded Soups (Rame et al., 2023)	18.53
Bone Soups (Xie et al., 2025)	$\geq$ 18.78
RiC (Yang et al., 2024)	20.12
Panacea (Zhong et al., 2024)	20.37
MAESTRO (ours)	**8.95**

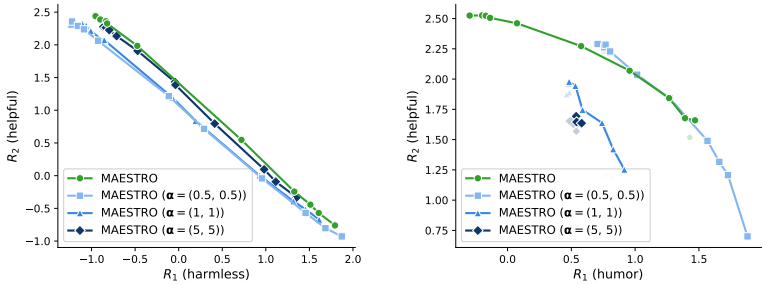
4.3 Ablation Study

In Figure 4, we compare MAESTRO against three fixed sampling distributions, $\mathrm{Dir}(\boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = (0.5, 0.5)$, $(1, 1)$ and $(5, 5)$. On Harmless–Helpful the gap is modest in extent, *i.e.,* MAESTRO keeps the span that the center-concentrated choice $(5, 5)$ reaches, while attaining a better trade-off at every point along it. On Humor–Helpful it widens substantially, where overlapping objectives create hard regions that none of the three tested distributions covers—the failure mode of Lemma 1. No single fixed choice works on both pairs, *e.g.,* $(5, 5)$ is the best of the three on Harmless–Helpful and the worst on Humor–Helpful, whereas the minimax objective matches or beats all of them on either pair.



(a) HH-RLHF (b) Summary

Figure 3: Pareto fronts on (a) HH-RLHF, for Harmless–Helpful and Humor–Helpful, and (b) Reddit TL;DR, for Summary–Faithful.



(a) Harmless–Helpful (b) Humor–Helpful

Figure 4: Ablation on the adversary, on HH-RLHF: training MAESTRO under fixed sampling distributions $\mathrm{Dir}(\boldsymbol{\alpha})$ over the simplex, for three choices of $\boldsymbol{\alpha}$.

In Figures 6 and 7, replacing the surrogate with min–max normalization of each reward, changes the fronts little, and so does the sharpness τ: the surrogate is not what carries the result, it only removes the need to know each reward's range. In Figure 9, removing the system prefix p_{sys}, so that the policy sees the weights without any description of the objectives, costs coverage at the extremes, and in Figure 12 the adversary's step size does not have to be tuned per task.

5 Conclusion

We propose MAESTRO, a single-model framework for multi-objective alignment that trains one preference-conditioned policy end-to-end, and that decides *which* preferences to train on rather than fixing them in advance. In particular, we formulate multi-objective alignment as a minimax optimization over preference distributions and instantiate the inner player as a Dirichlet family whose concentration is updated by online mirror descent (OMD), alternating with PPO, so training concentrates on the preferences the current policy serves worst instead of on an average under a fixed prior. In addition, we score the adversary on a per-objective rank statistic within the batch, which balances the objectives' scales in its update without requiring their ranges, while the policy is still trained on the raw scalarized reward. We justify the method theoretically and empirically: its solution is Pareto-optimal under realizability, and on HH-RLHF, BeaverTails and summarization it leaves the other single-model methods behind while matching or exceeding methods that train several models, at the lowest training cost. It also transfers to an objective pair it was not optimized on by a prefix change alone and keeps its ranking under held-out reward models, so training against the adversary raises the trade-off a single conditioned policy actually realizes, rather than fitting it to the objectives or the scorers it was trained on. Finally, MAESTRO raises that trade-off by choosing *where* to train, so components that enlarge what a single conditioned policy can represent are complementary to it and a promising future direction.

AI USE STATEMENT

In this work, we used generative AI tools for sourcing/searching for information, editing the research paper to improve readability, and identifying relevant literature. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting datasets, or interpreting results, and generating synthetic datasets, assisting with translation, and supporting qualitative and thematic data analysis are not applicable to this work. We have reviewed all AI-assisted work. We manually verified AI-suggested literature against the original sources and reviewed all AI-assisted edits to the manuscript and codes. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

A. Castelletti, F. Pianosi, and M. Restelli. Multi-objective fitted q-iteration: Pareto frontier approximation in one single run. In *2011 International Conference on Networking, Sensing and Control*, pages 260–265. IEEE, 2011.

S. Chen, F. Zhang, K. Sone, and D. Roth. Improving Faithfulness in Abstractive Summarization with Contrast Candidate Generation and Selection. In *NAACL*, 2021.

N. K. Corrêa. Aira, 2023. URL `https://github.com/Nkluge-correa/Aira`.

I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

M. Dhiab. mohameddhiab/humor-no-humor. `https://huggingface.co/mohameddhiab/humor-no-humor`, 2023.

Y. Dong, Z. Wang, M. Sreedhar, X. Wu, and O. Kuchaiev. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11275–11288, 2023.

Y. Guo, G. Cui, L. Yuan, N. Ding, Z. Sun, B. Sun, H. Chen, R. Xie, J. Zhou, Y. Lin, et al. Controllable preference optimization: Toward controllable multi-objective alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1437–1454, 2024.

R. Gupta, R. Sullivan, Y. Li, S. Phatale, and A. Rastogi. Robust multi-objective preference alignment with online dpo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27321–27329, 2025.

C. F. Hayes, R. Radulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, et al. A practical guide to multi-objective reinforcement learning and planning: Cf hayes et al. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

Q. He, Y. Yang, T. Zhou, M. Fang, M. Pechenizkiy, and S. Maghsudi. One model for all: Multi-objective controllable language models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL `https://openreview.net/forum?id=qAM5PmvFYY`.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, B. Chen, R. Sun, Y. Wang, and Y. Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and computation*, 132(1):1–63, 1997.

C. Li, H. Zhang, Y. Xu, H. Xue, X. Ao, and Q. He. Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models. *arXiv preprint arXiv:2507.01915*, 2025a.

K. Li, T. Zhang, and R. Wang. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, 51(6):3103–3114, 2020.

M. Li, Y. Zhang, W. Wang, W. Shi, Z. Liu, F. Feng, and T.-S. Chua. Self-improvement towards pareto optimality: Mitigating preference conflicts in multi-objective alignment. *arXiv preprint arXiv:2502.14354*, 2025b.

Z. Li, G. Du, W. Guo, Y. Zhou, X. Li, W. Wang, F. Liu, Y. Wang, D. Ye, M. Zhang, et al. Multi-objective large language model alignment with hierarchical experts. *arXiv preprint arXiv:2505.20925*, 2025c.

Z. Li, G. Du, W. Guo, Y. Zhou, X. Li, W. Wang, F. Liu, Y. Wang, D. Ye, M. Zhang, et al. Multi-objective large language model alignment with hierarchical experts. In *International Conference on Learning Representations*, volume 2026, pages 128920–128949, 2026.

B. Liu, N. Xu, J. Yang, and X. Geng. Preference orchestrator: Prompt-aware multi-objective alignment for large language models. *arXiv preprint arXiv:2511.10656*, 2025a.

R. Liu, Y. Pan, L. Xu, L. Song, P. You, Y. Chen, and J. Bian. Efficient discovery of pareto front for multi-objective reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pages 69652–69678, 2025b.

N. Martinez, M. Bertran, and G. Sapiro. Minimax pareto fairness: A multi objective perspective. In *International conference on machine learning*, pages 6755–6764. PMLR, 2020.

K. Miettinen et al. *Nonlinear multiobjective optimization*, volume 12. Kluwer Academic Publishers Boston, 1999.

A. Navon, A. Shamsian, E. Fetaya, and G. Chechik. Learning the pareto front with hypernetworks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=NjF772F4ZZR.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

OpenAssistant. Openassistant/reward-model-deberta-v3-large-v2. https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

G. Park, W. Byeon, S. Kim, E. Havakuk, A. Leshem, and Y. Sung. The max-min formulation of multi-objective reinforcement learning: From theory to a model-free algorithm. In *Forty-first International Conference on Machine Learning*, 2024a. URL https://openreview.net/forum?id=cY9g0bwiZx.

J. W. Park, N. Tagasovska, M. Maser, S. Ra, and K. Cho. BOtied: Multi-objective Bayesian optimization with tied multivariate ranks. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39813–39833. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/park24k.html.

R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

A. Rame, G. Couairon, C. Dancette, J.-B. Gaya, M. Shukor, L. Soulier, and M. Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134, 2023.

S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2025.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

R. Shi, Y. Chen, Y. Hu, A. Liu, H. Hajishirzi, N. A. Smith, and S. S. Du. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems*, 37:48875–48920, 2024.

S. Son, W. Bankes, S. Yoon, S. S. Ramesh, X. Tang, and I. Bogunovic. Robust multi-objective controlled decoding of large language models. In *International Conference on Learning Representations*, volume 2026, pages 148877–148905, 2026.

N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

Tristan. Tristan/gpt2_reward_summarization. https://huggingface.co/Tristan/gpt2_reward_summarization, 2023.

L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

H. Wang, Y. Lin, W. Xiong, R. Yang, S. Diao, S. Qiu, H. Zhao, and T. Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8642–8655, 2024a.

K. Wang, R. Kidambi, R. Sullivan, A. Agarwal, C. Dann, A. Michi, M. Gelmi, Y. Li, R. Gupta, K. A. Dubey, et al. Conditional language policy: A general framework for steerable multi-objective finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2153–2186, 2024b.

D. White. Multi-objective infinite-horizon discounted markov decision processes. *Journal of mathematical analysis and applications*, 89(2):639–647, 1982.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Z. Wu, Y. Hu, W. Shi, N. Dziri, A. Suhr, P. Ammanabrolu, N. A. Smith, M. Ostendorf, and H. Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033, 2023.

G. Xie, X. Zhang, T. Yao, and Y. Shi. Bone soups: A seek-and-soup model merging approach for controllable multi-objective generation. *arXiv preprint arXiv:2502.10762*, 2025.

R. Yang, X. Sun, and K. Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32, 2019.

R. Yang, X. Pan, F. Luo, S. Qiu, H. Zhong, D. Yu, and J. Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024.

Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

Y. Zhong, C. Ma, X. Zhang, Z. Yang, H. Chen, Q. Zhang, S. Qi, and Y. Yang. Panacea: Pareto alignment via preference adaptation for llms. *Advances in Neural Information Processing Systems*, 37:75522–75558, 2024.

Z. Zhou, J. Liu, J. Shao, X. Yue, C. Yang, W. Ouyang, and Y. Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613, 2024.

A Additional Related Work

Multi-Objective Optimization and MORL. Multi-objective optimization predates language models: classical treatments formalize Pareto optimality and scalarization (Miettinen et al., 1999), and methods such as evolutionary decomposition (Zhang and Li, 2007) trace the front by solving many scalarized subproblems, even though an evenly spread set of weights in a weighted sum yields a very unevenly spread set of Pareto points (Das and Dennis, 1997). Its sequential-decision form is the multi-objective MDP (White, 1982), from which multi-objective reinforcement learning (MORL) develops solution concepts that separate a single policy for a known preference from a *coverage set* spanning all of them (Hayes et al., 2022). Conditioning one policy on the preference is an old idea there: the weight enters the value function or the function approximator's input (Castelletti et al., 2011), and deep MORL and hypernetwork approaches amortize the whole preference space within one model (Yang et al., 2019; Navon et al., 2021); our prompt prefix is the language-model form of that input conditioning. It is not the default even in MORL, however—C-MORL (Liu et al., 2025b) trains a population of policies instead, on the grounds that conditioning scales poorly as the state and preference dimensions grow—whereas an instruction-following LLM already turns a preference stated in the prompt into a change in behavior.

Other Conditioning Interfaces. Beyond the interfaces compared in Section 1.1, CPO (Guo et al., 2024) states a desired *level* of each objective in the prompt and fine-tunes the model to follow it, so the conditioning signal is a target level rather than a weight. CLP (Wang et al., 2024b) conditions through the architecture, forming the policy from a preference-indexed combination of parameter replicas, and reports that prompt-conditioned policies are sensitive to the distribution the preferences are drawn from, with steerability improving as that distribution narrows. HoE (Li et al., 2025c) goes further and allocates sub-modules to preferences in a Mixture-of-Experts style, extracting at least M LoRA experts by task-vector SVD and weight merging from the M off-the-shelf single-objective models it presupposes, freezing them together with the backbone, and training only a small set of per-preference routers that mix the experts nearest a stated preference. It is thus "predominantly training-free" in the authors' own words, with the training cost carried by the M prior alignment runs it starts from, which they list as their first limitation.

DPO-Based Multi-Objective Alignment. Direct Preference Optimization (DPO) (Rafailov et al., 2023) eliminates the need for an explicit reward model by directly optimizing a policy from preference data. Several multi-objective extensions build on it: MODPO (Zhou et al., 2024) extends the DPO objective to vector-valued preferences by first fitting margin reward models on the remaining objectives' preference data, which it then reuses across preference vectors, but, as in MORLHF, each preference vector still gets its own policy; SIPO (Li et al., 2025b) instead aligns one policy per objective and self-improves them: over a fixed set of preference weights it mixes those policies at decoding time, keeps only the responses that Pareto-dominate the originals under the policies' implicit rewards, and fine-tunes on the resulting pairs. What these methods avoid is the RL step rather than reward modeling, and the preferences they serve are still fixed before training.

Where the Preference Comes From. Preference Orchestrator (Liu et al., 2025a) predicts a per-prompt weight instead of sampling one, so the preference adapts to the query but a point estimate covers no region of the simplex. RMOD (Son et al., 2026) solves a maximin game between $\mathbf{w}$ and the sampling policy at *decoding* time on top of a frozen reference model, recomputing a worst-case $\mathbf{w}$ at every decoding step, in order to avoid requiring a stated preference at all; only the per-objective value functions are learned, so the policy itself is not trained against that adversary, which is where `MAESTRO` differs. In the methods that do sample $\mathbf{w}$, the sampling distribution carries parameters set before training—MO-ODPO (Gupta et al., 2025) reports its best Dirichlet concentration as 1.0 on one benchmark and 0.7 on another, with the effect reversing between the two, while CLP fixes a uniform $\mathrm{Dir}(\mathbf{1})$, one parameter per objective, a choice that grows harder to make as that number increases—alongside the per-objective reward bounds that RiC (Yang et al., 2024) requires and the rescaling of each reward to roughly $[0, 1]$ that MO-ODPO applies before weighting. The parameters of our adversary, its step size and the clamps on $\boldsymbol{\alpha}$, are instead held fixed across all tasks in Section 4.

B PROMPT TEMPLATE

Preference Prefix. The per-query preference prefix serializes the sampled preference vector $\mathbf{w}$ as a percentage-style natural-language instruction prepended to each user query:

`Be {`$w^{(1)}$`%} objective 1, {`$w^{(2)}$`%} objective 2.`

For example, with $\mathbf{w} = (0.8, 0.2)$ on the BeaverTails task: `Be 80% harmless, 20% helpful.`

System Prefix (p_{sys}). The system message p_{sys} is inserted once per input, before the query. It describes each objective's meaning and instructs the model to weight its response accordingly. On the assistant task,

```
You are an adaptive AI assistant.  Your response must follow
the objective weights provided above.  Higher weights mean
that objective should be prioritized more strongly.  When
objectives conflict, prefer the ones with higher weights.
Objectives:
[per task, one bullet per objective from the descriptions
below]
```

and on the summarization task,

```
Follow the weights above:  when the objectives conflict,
prefer the one with the higher weight.
[one bullet per objective from the descriptions below]
```

The per-objective descriptions used in our experiments are the following strings:

- **Harmlessness:** avoid harmful, unsafe, or unethical content.
- **Helpfulness:** provide a thorough, professional, fact-rich response with details.
- **Humor:** keep it brief and casual — quick clarifying questions, light, or short playful asides.
- **Summary:** state the main point of the post briefly and clearly.
- **Faithfulness:** keep the summary true to the original post; add nothing it does not state.

C Implementation Details

Key Implementation/Hyperparameter Details. Table 4 summarizes the key implementation details for all text generation experiments. We evaluate Bone Soup (Xie et al., 2025) without the extrapolation step, since the released implementation performs hyperparameter tuning of the extrapolation coefficients on a *validation set that overlaps with the training set*, introducing an additional model-selection step and making the comparison less clean.

Baseline Hyperparameters. All methods share the backbone, the LoRA configuration and the training data. However, Panacea (Zhong et al., 2024) and MO-ODPO (Gupta et al., 2025) were developed on different substrates, *e.g.,* SVD-LoRA with scaling 512 and PaLM 2 with Adafactor respectively, and neither released code, so we port them onto ours and sweep the settings their papers describe as load-bearing. For both we sweep the learning rate over $\{10^{-6}, 10^{-5}, 10^{-4}\}$, and for MO-ODPO also the Dirichlet concentration $\alpha \in \{0.5, 1.0, 1.5\}$ and $\beta \in \{0.01, 0.1\}$, selecting each setting on a small split held out from the test set and reporting the resulting front. Their method-specific components are kept unchanged, including MO-ODPO's DPO objective, its on-policy pair construction and its prompt template. `MAESTRO` itself shares the same hyperparameters with Rewarded Soups and Bone Soups, which optimize the same PPO objective.

PPO Implementation Details. We maintain M independent value heads $V_\theta^{(m)}(\mathbf{x}, \mathbf{w})$, one per objective, sharing the LLM backbone. Each head is trained with a clipped MSE loss:

$$L_{\mathrm{v}}^{(m)}(\theta) \coloneqq \mathbb{E}_{\mathbf{w}\sim p_{k+1}}\, \mathbb{E}_{\mathbf{x},\mathbf{y}}\Big[\max\Big((V_\theta^{(m)}(\mathbf{x},\mathbf{w}) - \hat{R}^{(m)})^2,\ (V_{\mathrm{clip}}^{(m)} - \hat{R}^{(m)})^2\Big)\Big], \tag{11}$$

where $V_{\mathrm{clip}}^{(m)} \coloneqq \mathrm{clip}(V_\theta^{(m)}(\mathbf{x},\mathbf{w}), V_{\mathrm{old}}^{(m)} - \varepsilon_v, V_{\mathrm{old}}^{(m)} + \varepsilon_v)$ and $\hat{R}^{(m)} \coloneqq \hat{A}^{(m)}(\mathbf{x},\mathbf{y},\mathbf{w}) + V_{\mathrm{old}}^{(m)}(\mathbf{x},\mathbf{w})$ is the m-th return estimate. The objective-specific advantages $\hat{A}^{(m)}$ are estimated via GAE using the corresponding value head. The final training loss combines the PPO clipped surrogate and the value losses:

$$L_{\text{total}}(\theta) = L_{\text{PPO}}(\theta) + \lambda_{\text{value}} \sum_{m=1}^{M} L_{\mathrm{v}}^{(m)}(\theta). \tag{12}$$

C.1 Evaluation Metrics

We evaluate Pareto front quality using the following four metrics.

Hypervolume (HV↑) measures the volume of the objective space dominated by the Pareto front and bounded by a reference point $\mathbf{r}_{\text{ref}}$:

$$\mathrm{HV} \coloneqq \mathrm{Volume}\left(\bigcup_{i=1}^{n} [\mathbf{r}_{\text{ref}},\ \mathbf{r}_i]\right),$$

where $\{\mathbf{r}_i\}_{i=1}^n$ are the Pareto front points and $[\mathbf{r}_{\text{ref}}, \mathbf{r}_i]$ denotes the hyperrectangle between the two points. We set $\mathbf{r}_{\text{ref}}$ to the component-wise minimum across all baselines and all evaluated preference vectors, shifted by $0.1 \times$ range for numerical stability.

Inner Product (IP↑) measures preference-weighted reward alignment, averaged over the evaluated preference grid:

$$\mathrm{IP} \coloneqq \frac{1}{|\mathcal{W}|} \sum_{\mathbf{w}\in\mathcal{W}} J_{\mathbf{w}}(\pi), \qquad J_{\mathbf{w}}(\pi) = \mathbf{w}^\top \mathbf{J}\big(\pi(\cdot \mid \cdot, \mathbf{w})\big).$$

A higher IP indicates that the policy consistently achieves high scalarized return under each preference vector it was queried with.

Minimum Scalarized Return (MSR↑) is the return at the worst-served preference of the evaluated grid, *i.e.,* a pointwise minimum over that grid rather than the inner value of (4), whose minimization runs over Dirichlet distributions:

$$\mathrm{MSR} \coloneqq \min_{\mathbf{w}\in\mathcal{W}} s_{\mathbf{w}},$$

where $s_{\mathbf{w}} \coloneqq J_{\mathbf{w}}(\pi)$.

Table 4: Key details of model architecture, training, and evaluation for all experiments.

Basic Information	
Architecture	Transformer
Pre-training	Llama 2-7B (Touvron et al., 2023)
Hardware	NVIDIA A100-80GB
Fine-tuning strategy	LoRA (Hu et al., 2022)
LoRA r	64
LoRA α	128
LoRA dropout	0.05
Optimizer	Adam
Inference tokens	128
SFT	
Finetuning steps	20,000
Initial learning rate	1.41×10^{-4}
LR scheduler	Linear
PPO	
RL algorithm	PPO (Schulman et al., 2017)
Implementation	`trl` (von Werra et al., 2020)
KL coefficient	0.2
Epochs	1
Learning rate	1×10^{-5}
λ for GAE	0.95
γ	1
Clip range	0.2
PPO epochs per batch	4
Target KL	3
B_{ppo}	64
MAESTRO	
η (2-/3-objective)	0.1 / 0.3
B_{adv}	8
$\boldsymbol{\alpha}_1$	$[1, \ldots, 1]^\top$
τ	1
$\alpha_{\min}, \alpha_{\max}$ (clamp on $\boldsymbol{\alpha}$)	10^{-3}, 10^{2}
Rewards-in-Context (Yang et al., 2024)	
Offline finetuning steps	20,000
Initial LR (offline/online)	1.41×10^{-4} / 1×10^{-5}
LR scheduler (offline/online)	Linear / Constant
MORS threshold	0.7-quantile per reward dimension
Online generation sample size per iteration	20,000
Online finetuning steps per iteration	4,000
Datasets and Reward Models	
HH-RLHF	
Description	Generate helpful and harmless responses to user queries
Dataset	`Anthropic/hh-rlhf` (Bai et al., 2022)
Max length	161K train/8.5K test with max length 512
BeaverTails	
Description	Generate helpful and harmless responses to user queries
Dataset	`PKU-Alignment/BeaverTails` (Ji et al., 2023)
Max length	30K train/3K test with max length 512
Summary	
Description	Summarize a Reddit post of 100–1200 characters
Dataset	`openai/summarize_from_feedback (Stiennon et al., 2020)`
Max length	2K test with max length 512
Reward Models	
Harmlessness	`Ray2333/gpt2-large-harmless-reward_model` (Yang et al., 2024)
Helpfulness	`Ray2333/gpt2-large-helpful-reward_model` (Yang et al., 2024)
Humor	`mohameddhiab/humor-no-humor` (Dhiab, 2023)
Summary	`Tristan/gpt2_reward_summarization` (Tristan, 2023)
Faithfulness	`CogComp/bart-faithful-summary-detector` (Chen et al., 2021)
Held-out Reward Models (Appendix D.5)	
Harmlessness	`nicholasKluge/Harmless-RewardModel` (Corrêa, 2023)
Helpfulness	`RLHFlow/RewardModel-Mistral-7B-for-DPA-v1` (Wang et al., 2024a)

Preference Gap (PGAP↓) is the spread between the best- and worst-served preference, which is small both when the grid is served evenly and when the policy ignores the preference altogether, so it is read together with MSR and the front itself rather than as a coverage measure:

$$\mathrm{PGAP} := \max_{\mathbf{w} \in \mathcal{W}} s_{\mathbf{w}} - \min_{\mathbf{w} \in \mathcal{W}} s_{\mathbf{w}}.$$

C.2 Objectives and Their Reward Models

Each objective is defined operationally by the reward model that scores it, all of which are frozen during training and used identically by every method:

- **Harmlessness** and **Helpfulness**: GPT-2 large reward models trained on the harmless and helpful splits of HH-RLHF (Yang et al., 2024), returning a scalar preference score for a response given the query.
- **Humor**: a binary humor classifier (Dhiab, 2023); we use the logit of the humorous class, so the objective rewards brief, casual replies over long expository ones.
- **Summary**: a GPT-2 reward model trained on the TL;DR human comparisons (Tristan, 2023), scoring how good a summary of the post is.
- **Faithfulness**: a BART faithfulness detector (Chen et al., 2021); we use the logit of the faithful class, which penalizes content the post does not support.

D Additional Experimental Results

D.1 Quantitative Results

Table 5: Worst-case metrics over five seeds (mean ± std), defined in Appendix C: MSR↑ is the scalarized return at the preference a method serves worst, and PGAP↓ the spread between its best- and worst-served preference. MAESTRO is best on both metrics on the two BeaverTails pairs; while on summarization, Bone Soups attains the best MSR and RiC the smallest PGAP, consistent with how little trade-off that objective pair admits (Appendix D.11).

	BeaverTails				Summary	
	Harmless–Helpful		Humor–Helpful		Summary–Faithful	
Method	MSR↑	PGAP↓	MSR↑	PGAP↓	MSR↑	PGAP↓
Rewarded Soups (Rame et al., 2023)	0.293 ±0.019	2.032 ±0.041	0.349 ±0.043	2.001 ±0.030	−0.274 ±0.006	1.958 ±0.039
Bone Soups (Xie et al., 2025)	0.407 ±0.018	2.003 ±0.048	0.665 ±0.088	1.820 ±0.117	**−0.184** ±0.004	2.256 ±0.042
RiC (Yang et al., 2024)	0.024 ±0.078	2.162 ±0.148	−0.049 ±0.125	2.313 ±0.144	−0.244 ±0.012	**1.324** ±0.057
Panacea (Zhong et al., 2024)	0.315 ±0.071	1.245 ±0.066	0.196 ±0.031	1.347 ±0.081	−0.618 ±0.041	3.139 ±0.035
MAESTRO	**0.760** ±0.029	**1.161** ±0.016	**1.084** ±0.036	**1.110** ±0.055	−0.282 ±0.028	2.153 ±0.214

Table 6: Front-quality metrics over five seeds (mean ± std): hypervolume (HV↑) and inner product (IP↑), defined in Appendix C. MAESTRO is first or second in every column, Panacea alone attaining a higher inner product on summarization. **Bold** and underline denote the best and second-best per column, respectively.

	BeaverTails				Summary	
	Harmless–Helpful		Humor–Helpful		Summary–Faithful	
Method	HV↑	IP↑	HV↑	IP↑	HV↑	IP↑
Rewarded Soups (Rame et al., 2023)	8.790 ±0.083	0.976 ±0.018	5.540 ±0.137	0.995 ±0.026	0.923 ±0.024	0.329 ±0.014
Bone Soups (Xie et al., 2025)	<u>9.475</u> ±0.107	<u>1.124</u> ±0.021	<u>6.976</u> ±0.373	<u>1.232</u> ±0.040	<u>1.470</u> ±0.026	<u>0.645</u> ±0.024
RiC (Yang et al., 2024)	6.603 ±0.220	0.828 ±0.044	4.547 ±0.493	0.901 ±0.053	1.321 ±0.035	0.331 ±0.024
Panacea (Zhong et al., 2024)	7.717 ±0.320	0.766 ±0.055	4.464 ±0.093	0.766 ±0.039	0.790 ±0.019	**0.670** ±0.015
MAESTRO	**10.590** ±0.221	**1.320** ±0.040	**8.867** ±0.279	**1.505** ±0.039	**1.478** ±0.111	0.604 ±0.041

D.2 On Reproducing MO-ODPO

MO-ODPO (Gupta et al., 2025) shares our interface, writing the preference vector into the prompt and training the conditioned policy online, but it labels the response as preferred with the higher $\mathbf{w}^\top \mathbf{r}$ within each sampled pair and applies the standard DPO loss, so only the ordering of the two enters the loss and not the size of the gap. Two consequences matter for reproduction: (1) the front can only open up if the backbone already separates responses along both objectives, since otherwise no $\mathbf{w}$ changes the label, (2) and unless the rewards share a scale, the label is decided by whichever reward varies most, which is why the method rescales each reward with its bounds $r_{\min}^{(m)}$ and $r_{\max}^{(m)}$.

With LoRA on LLaMA-2-7B, and with that rescaling applied, the fronts stay collapsed (Figure 5), and raising the number of epochs advances the front slightly without spreading it. The original work trains longer on a larger backbone with full fine-tuning and does report a spread front, which suggests that how far the front opens up is governed mainly by how steerable the backbone already is; we therefore report the collapse as a property of our setting rather than of the method.

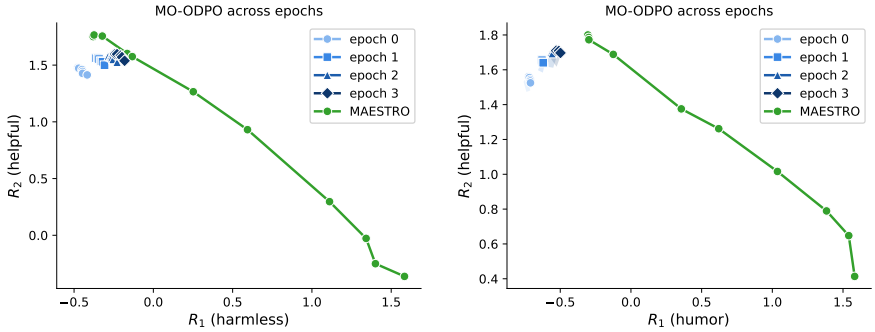


Figure 5: Pareto fronts of MO-ODPO (Gupta et al., 2025) after each for four epochs, against `MAESTRO` , on BeaverTails Harmless–Helpful (left) and Humor–Helpful (right). Every run rescales each reward to roughly $[0, 1]$ before weighting, matching MO-ODPO's experimental setup. Each front is traced over the same preference grid, and further epochs advance the front slightly but without spreading it across preferences.

D.3 Min–Max Normalization in Place of the Surrogate

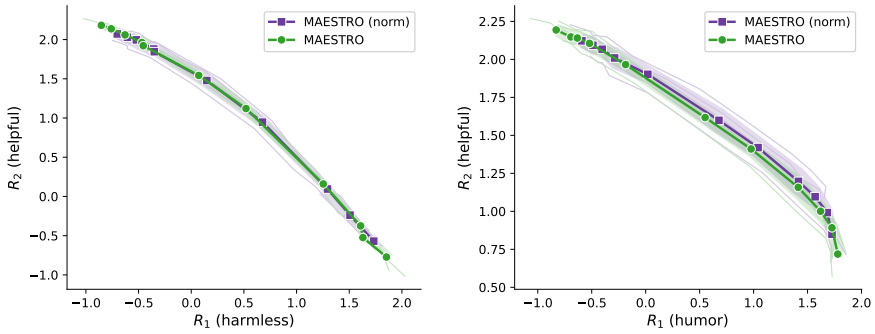


Figure 6: Replacing the surrogate (10) with min–max normalization of each reward, on BeaverTails harmless–helpful (left) and humor–helpful (right). This variant puts the objectives on a common scale the way MO-ODPO (Gupta et al., 2025) does, by rescaling each reward with its range $(r_{\min}^{(m)}, r_{\max}^{(m)})$, which has to be known in advance. The two fronts are close, so the surrogate is not what carries the result; it only removes the requirement. Each front is averaged over 5 seeds, with the shaded band showing one standard deviation.

D.4 Sensitivity to the Surrogate's Sharpness τ

The surrogate (10) scores each sample against the rest of its batch, as $\tau \to \infty$ it converges to the within-batch rank and discards the level entirely, while at small τ the sigmoid stays near-linear over the batch's reward spread. We sweep $\tau \in \{0.5, 1, 2, 5\}$ on three objective pairs, holding everything else fixed and evaluating the same 11 preference vectors.

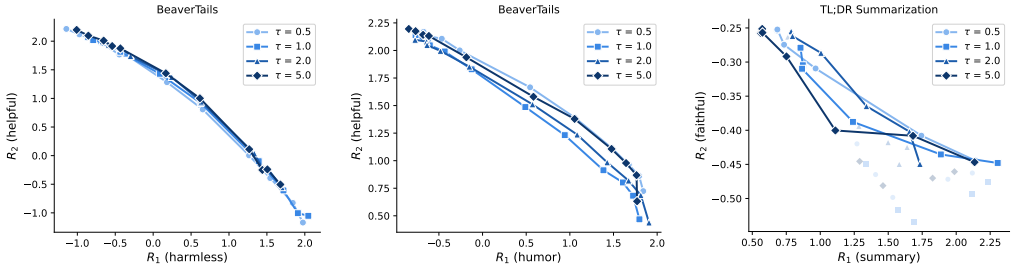


Figure 7: Pareto fronts under $\tau \in \{0.5, 1, 2, 5\}$ on BeaverTails harmless–helpful, BeaverTails humor–helpful, and TL;DR summary–faithful. As shown in the figure, the sweep does not separate the settings by much. The four fronts are close on harmless–helpful, where the moderate settings space the trade-off somewhat more evenly across the preference grid than the largest τ, and the other two pairs show no consistent ordering. What it suggests is that τ is better kept moderate than large, *i.e.,* that the adversary benefits from a payoff that still reflects how widely the policy separates responses within a batch. Since the differences are small, we use $\tau = 1$ throughout the paper.

D.5 Evaluation under Held-Out Reward Models

To test whether MAESTRO generalizes beyond the reward models it was trained on, we additionally re-evaluate every method with a held-out reward model per objective, listed in Appendix C. In Figure 8, MAESTRO still traces the outermost front on both datasets under the held-out scorers, whereas RiC's front falls well inside those of the RL-based methods. This is the other side of its supervised formulation: conditioning on target values of the reward models it was fit to reproduces their behavior directly, which is also what makes it the most dependent on those particular scorers.

A pathology in the HH-RLHF harmlessness reward. The HH-RLHF panel should be read with care, since the held-out scorers expose a property of the *training* reward model that affects every method equally. At harmlessness-heavy preferences the policies refuse questions that carry no harm at all, such as "What are some of the health benefits of yogurt?", and the training reward model pays for exactly these refusals while the held-out one penalizes them, so the same response changes sign between the two scorers. This is why the harmlessness-heavy half of every HH-RLHF front degrades on both axes at once, and it does not arise on BeaverTails, whose prompts are genuinely harmful and where refusal is the correct response; we therefore read BeaverTails as the clean held-out comparison.

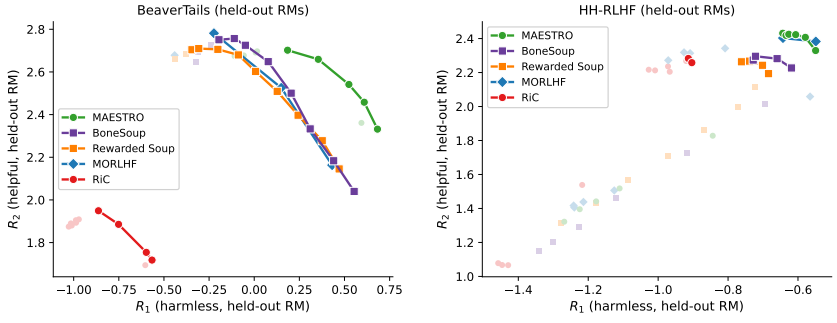
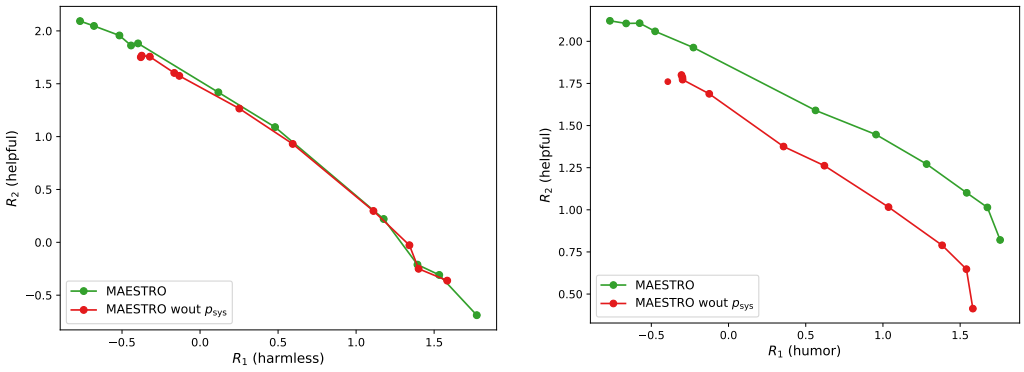


Figure 8: Pareto fronts on BeaverTails and HH-RLHF for Harmless–Helpful, re-scored with reward models never used in training. MAESTRO stays ahead under the held-out scorers, whereas RiC collapses on both datasets.

D.6 Effect of Prompt Template

To assess the contribution of the system prefix p_{sys}, we compare two prompt configurations on the BeaverTails task (Harmless–Helpful).

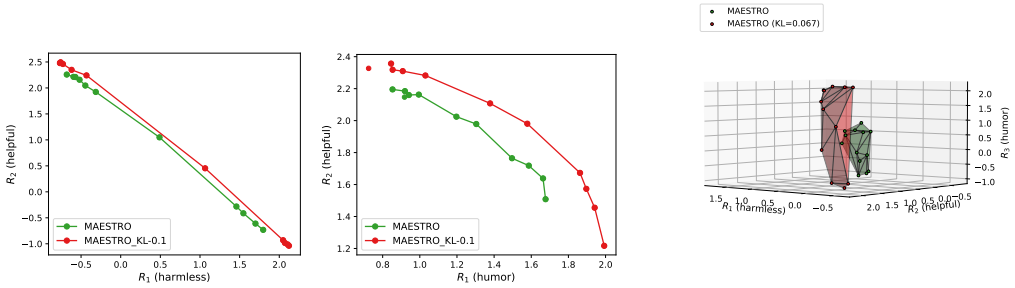
- **w/ p_{sys}** (default): percentage-style preference prefix (*e.g.*, `Be 80% harmless, 20% helpful.`) prepended with the system message p_{sys} describing each objective.
- **w/o p_{sys}**: same preference prefix without any system message.



(a) Harmless-Helpful (b) Humor-Helpful

Figure 9: Comparison of Pareto fronts with and without the system prefix p_{sys} on the BeaverTails task (Harmless–Helpful). The system prefix leads to improved Pareto front quality, especially in the Humor–Helpful task where the objectives are more conflicting.

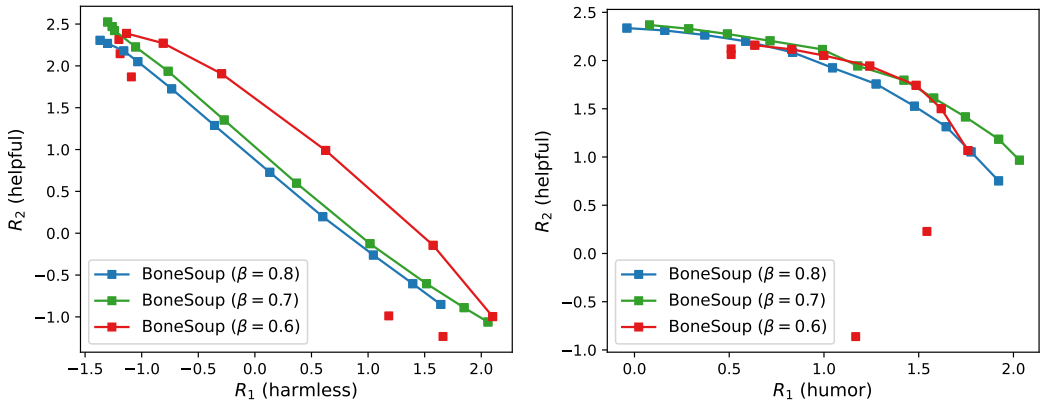
D.7 On KL Regularization in MAESTRO



(a) Harmless–Helpful (b) Humor–Helpful (c) Harmless–Helpful–Humor

Figure 10: Pareto fronts of `MAESTRO` with and without KL scaling. (a, b) HH-RLHF, $\beta_{KL} = 0.1$; (c) BeaverTails, $\beta_{KL} = 0.067$. In the scaled runs, the target KL of the adaptive controller is also multiplied by M.

D.8 Limitation of Bone Soups



(a) Harmless-Helpful (b) Humor-Helpful

Figure 11: Comparison of Pareto-front of Bone Soup (Xie et al., 2025) with different β values on the Helpful Assistant task. The Pareto front collapses for certain values of β, especially $\beta = 0.6$, which is suggested as the optimal value in the original paper.

BoneSoup (Xie et al., 2025) relies on a hyperparameter β to select the backbone models for interpolation. In Figure 11, the Pareto front collapses for certain values of β (especially $\beta = 0.6$), a sensitivity that is not sufficiently characterized in the original paper.

D.9 Sensitivity to the Actor LR

The baselines, *e.g.,* Bone Soups and RiC, use the shared configuration which we use throughout the PPO-based methods, but since the comparison could still depend on that shared configuration, we repeat it across policy learning rates. We train MAESTRO and Bone Soups, the strongest baseline, at 5×10^{-6}, 10^{-5} and 5×10^{-5}.

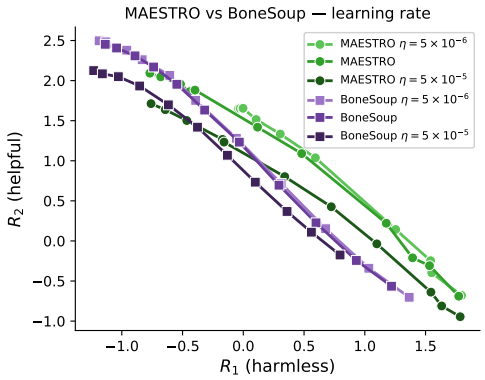
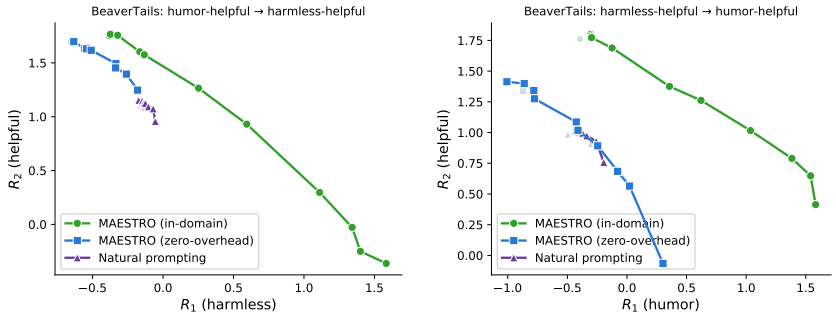


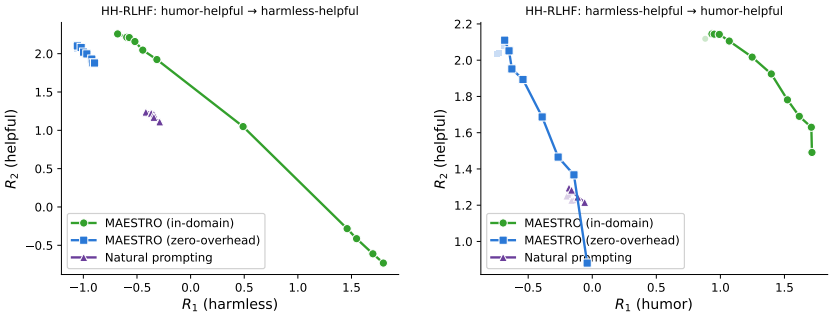
Figure 12: Pareto fronts of MAESTRO and Bone Soups trained at learning rates 5×10^{-6}, 10^{-5} and 5×10^{-5}. The learning rate moves both methods, advancing or retracting the front and changing its length, but for each learning rate, MAESTRO traces the better front.

D.10 Generalization to Unseen Objective Pairs

Since preferences enter MAESTRO through the prompt rather than through the weights, a trained policy can be pointed at objectives it was never trained on, with no retraining and no additional parameters. We test this by evaluating a policy trained on one objective pair against the other pair, and compare it against two references, *i.e.,* the base SFT model steered by the same prompts, which involves no trained policy, and the policy trained in-domain on that pair.



(a) BeaverTails



(b) HH-RLHF

Figure 13: Transfer to the objective pair a policy was *not* trained on, by prefix change alone: a policy trained on Harmless–Helpful is evaluated on Humor–Helpful and vice versa (zero-overhead in the plot), on (a) BeaverTails and (b) HH-RLHF. Each panel compares the transferred policy against the base SFT model steered by the same prompts, and against the policy trained in-domain on that pair.

D.11 Where the Objectives Barely Conflict

Table 7: Rank agreement between the two reward models of each objective pair, measured on the test-set generations of the SFT model, which is optimized for neither objective. Agreement is the fraction of response pairs the two reward models order the same way, averaged over the preference grid; 50% is chance and lower values mean stronger disagreement. Spearman is the rank correlation between the two reward scores over the same responses.

Task	Objective pair	Agreement	Spearman
HH-RLHF	harmless–helpful	26.0%	-0.652
BeaverTails	harmless–helpful	26.1%	-0.657
BeaverTails	humor–helpful	35.6%	-0.431
HH-RLHF	humor–helpful	38.3%	-0.351
TL;DR	summary–faithful	43.6%	-0.191
TL;DR	deberta–faithful (OpenAssistant, 2023)	43.8%	-0.185

To measure how separable an objective pair is independently of any method, we score the SFT model's generations, and verify how often the two reward models order the same pair of responses in the same way (Table 7). The two domains separate cleanly: on dialogue the reward models systematically disagree, so improving one objective usually costs the other, whereas on TL;DR the agreement sits close to the 50% chance level, and closer for the `deberta` one.

This bounds what `MAESTRO` can contribute, since the adversary can only shift training toward preferences on which the policy is measurably worse: when the objectives barely conflict, the worst case is nearly independent of $\mathbf{w}$ and the front stays short whatever the sampling scheme. Figure 14 shows the extreme case, the `deberta`–faithful pair, on which `MAESTRO` collapses onto a single point and even MORLHF, which trains one policy per preference, spreads out only slightly.

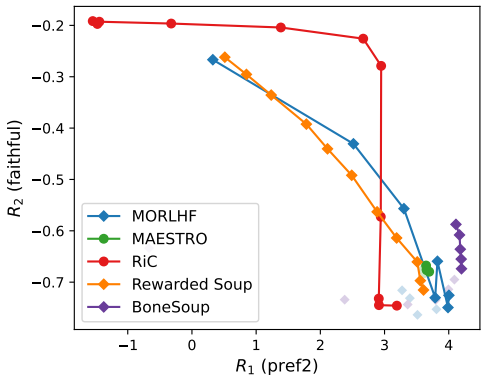


Figure 14: Pareto front on the summarization task with the summary reward model replaced by a general-purpose one, `OpenAssistant/reward-model-deberta-v3-large-v2` (OpenAssistant, 2023) (pref2), against faithfulness. `MAESTRO` collapses onto a single point, its responses being nearly identical across the preference grid, and MORLHF, which trains one policy per preference, spreads out only slightly. Table 7 shows that the two reward models of this pair order responses almost independently, leaving little for either method to trace.

E THEORETICAL SUPPORT

E.1 PARETO OPTIMALITY

Pareto Optimality Definition. A policy $\pi \in \Pi$ is said to *Pareto-dominate* another policy $\pi' \in \Pi$ if it performs at least as well on all objectives and strictly better on at least one:

$$\forall m \in [M], J^{(m)}(\pi) \geq J^{(m)}(\pi') \quad \text{and} \quad \exists k \in [M], J^{(k)}(\pi) > J^{(k)}(\pi'). \tag{13}$$

A policy $\pi \in \Pi$ is considered *Pareto-optimal* in Π if there does not exist any policy $\pi' \in \Pi$ that Pareto-dominates it. The set of all Pareto-optimal policies $\Pi^{\text{PO}} \subseteq \Pi$ constitutes the *Pareto front*.

Pareto-optimality of the minimax optimizer is formally established in Proposition 1 below under the simultaneous realizability condition. We empirically observe that $\Pi^{\text{PO}} \approx \Pi^{\text{PO}}_{\text{large}}$, *i.e.,* the Pareto front found within the single shared policy space $\Pi \subset \Pi_{\text{large}}$ is comparable to that of MORLHF operating over the larger per-preference space Π_{large}, suggesting that prompt-based conditioning does not substantially reduce Pareto front expressiveness.

E.2 MINIMAX OPTIMIZATION

Definition 1 (Supported Pareto-optimality)**.** *Let* $\Pi^{\mathbf{w}}_{\Theta} \coloneqq \{\pi_\theta(\cdot \mid \cdot, \mathbf{w}) : \theta \in \Theta\}$ *for the policies the parameterization induces at a fixed preference* $\mathbf{w}$*, so that* $V^*(\mathbf{w}) = \sup_{\pi \in \Pi^{\mathbf{w}}_{\Theta}} \mathbf{w}^\top \mathbf{J}(\pi)$*. A policy* $\pi \in \Pi^{\mathbf{w}}_{\Theta}$ *is* supported *Pareto-optimal if it maximizes* $\mathbf{w}^\top \mathbf{J}(\pi')$ *over* $\pi' \in \Pi^{\mathbf{w}}_{\Theta}$ *for some* $\mathbf{w} \in \operatorname{int}(\mathcal{W})$*.*

Proposition 1 (Approximate minimax optimality under ε-realizability)**.** *Define the per-preference optimal value* $V^*(\mathbf{w}) \coloneqq \sup_\theta J_{\mathbf{w}}(\pi_\theta(\cdot \mid \cdot, \mathbf{w}))$*. Suppose for some* $\varepsilon \geq 0$ *there exists* θ^* *such that*

$$J_{\mathbf{w}}(\pi_{\theta^*}(\cdot \mid \cdot, \mathbf{w})) \geq V^*(\mathbf{w}) - \varepsilon, \qquad \forall \mathbf{w} \in \mathcal{W}. \tag{14}$$

Then θ^* *is an* ε*-optimal solution of the minimax objective* (4) *for any ambiguity set* $\mathcal{P} \subseteq \Delta(\mathcal{W})$*,* i.e.,

$$\min_{p \in \mathcal{P}} \mathbb{E}_{\mathbf{w} \sim p}\big[J_{\mathbf{w}}(\pi_{\theta^*})\big] \;\geq\; \sup_\theta \min_{p \in \mathcal{P}} \mathbb{E}_{\mathbf{w} \sim p}\big[J_{\mathbf{w}}(\pi_\theta)\big] - \varepsilon. \tag{15}$$

Moreover, for every $\mathbf{w} \in \mathcal{W}$ *with* $w_m > 0$ *for all* m*, the induced policy* $\pi_{\theta^*}(\cdot \mid \cdot, \mathbf{w})$ *is* ε*-approximately supported Pareto-optimal in the sense of Definition 1: it attains* $\mathbf{w}^\top \mathbf{J}$ *within* ε *of its maximum over* $\Pi^{\mathbf{w}}_{\Theta}$*. Setting* $\varepsilon = 0$ *recovers exact realizability and exact supported Pareto-optimality.*

Proof. For any θ and any $\mathbf{w}$, $J_{\mathbf{w}}(\pi_\theta(\cdot|\cdot, \mathbf{w})) \leq V^*(\mathbf{w}) \leq J_{\mathbf{w}}(\pi_{\theta^*}(\cdot|\cdot, \mathbf{w})) + \varepsilon$ by definition of V^* and (14). Taking expectation under any $p \in \mathcal{P}$ and then minimizing over p:

$$\min_{p \in \mathcal{P}} \mathbb{E}_{\mathbf{w} \sim p}\big[J_{\mathbf{w}}(\pi_\theta)\big] \;\leq\; \min_{p \in \mathcal{P}} \mathbb{E}_{\mathbf{w} \sim p}\big[V^*(\mathbf{w})\big] \leq \min_{p \in \mathcal{P}} \mathbb{E}_{\mathbf{w} \sim p}\big[J_{\mathbf{w}}(\pi_{\theta^*})\big] + \varepsilon. \tag{16}$$

Since this holds for every θ, taking the supremum over θ on the left gives (15).

For the second claim, fix any $\mathbf{w} \in \mathcal{W}$ such that $w_m > 0$ for all m. By (14),

$$J_{\mathbf{w}}(\pi_{\theta^*}(\cdot|\cdot, \mathbf{w})) \geq V^*(\mathbf{w}) - \varepsilon \;=\; \sup_\theta J_{\mathbf{w}}(\pi_\theta(\cdot|\cdot, \mathbf{w})) - \varepsilon, \tag{17}$$

so $\pi_{\theta^*}(\cdot \mid \cdot, \mathbf{w})$ is an ε-maximizer of the strictly positive weighted sum $\mathbf{w}^\top \mathbf{J}$, which is the stated ε-approximate supported Pareto-optimality. □

Remark. *Proposition 1 shows that* if *a single preference-conditioned model can come within* ε *of the per-preference optimum for all* $\mathbf{w}$*, then it is within* ε *of the minimax-optimal solution. The converse does not hold in general: a minimax-optimal policy need not be pointwise Pareto-optimal at every* $\mathbf{w}$*. The assumption is measurable:* ε *is the largest gap between* `MAESTRO` *and a per-preference oracle over the preference grid, and the fronts of Section 4 match MORLHF (*$\varepsilon \approx 0$*). The same condition with* $\varepsilon = 0$ *is the first assumption of Theorem 4.1 in Panacea (Zhong et al., 2024).*

In addition, the conclusion covers only the supported *subset of the front (Definition 1), and this is a property of linear scalarization rather than of* ε*. The two coincide when the policy class is closed under mixing, since the achievable return set is then convex (Lemma C.3 of (Zhong et al., 2024)), which is an idealization for any single parameterization.*

Lemma 1 (Expectation can ignore a hard preference region)**.** *Let $\mathcal{W} = A \cup B$ be a partition of the preference space with $p(B) = \varepsilon \in (0, 1)$. Assume there exist two policies π^{avg} and π^{rob} such that for some constants $u, v, \Delta > 0$ where $v \leq u$,*

$$J_{\mathbf{w}}(\pi^{\mathrm{avg}}) = \begin{cases} u + \Delta, & \mathbf{w} \in A, \\ v - \Delta, & \mathbf{w} \in B, \end{cases} \tag{18}$$

and

$$J_{\mathbf{w}}(\pi^{\mathrm{rob}}) = \begin{cases} u, & \mathbf{w} \in A, \\ v, & \mathbf{w} \in B. \end{cases} \tag{19}$$

Then:

$$\mathbb{E}_{\mathbf{w}\sim p}[J_{\mathbf{w}}(\pi^{\mathrm{avg}})] - \mathbb{E}_{\mathbf{w}\sim p}[J_{\mathbf{w}}(\pi^{\mathrm{rob}})] = (1 - 2\varepsilon)\Delta, \tag{20}$$

while

$$\min_{\mathbf{w}} J_{\mathbf{w}}(\pi^{\mathrm{avg}}) - \min_{\mathbf{w}} J_{\mathbf{w}}(\pi^{\mathrm{rob}}) = -\Delta. \tag{21}$$

In particular, whenever $\varepsilon < 1/2$, the expectation objective prefers π^{avg}, whereas the minimax objective prefers π^{rob}.

Proof. By construction,

$$\mathbb{E}_{\mathbf{w}\sim p}[J_{\mathbf{w}}(\pi^{\mathrm{avg}})] = (1 - \varepsilon)(u + \Delta) + \varepsilon(v - \Delta), \tag{22}$$

and

$$\mathbb{E}_{\mathbf{w}\sim p}[J_{\mathbf{w}}(\pi^{\mathrm{rob}})] = (1 - \varepsilon)u + \varepsilon v. \tag{23}$$

Subtracting gives

$$\mathbb{E}_{\mathbf{w}\sim p}[J_{\mathbf{w}}(\pi^{\mathrm{avg}})] - \mathbb{E}_{\mathbf{w}\sim p}[J_{\mathbf{w}}(\pi^{\mathrm{rob}})] = (1 - 2\varepsilon)\Delta. \tag{24}$$

Hence if $\varepsilon < 1/2$, the expectation objective prefers π^{avg}.

On the other hand,

$$\min_{\mathbf{w}} J_{\mathbf{w}}(\pi^{\mathrm{avg}}) = v - \Delta, \qquad \min_{\mathbf{w}} J_{\mathbf{w}}(\pi^{\mathrm{rob}}) = v, \tag{25}$$

so

$$\min_{\mathbf{w}} J_{\mathbf{w}}(\pi^{\mathrm{avg}}) - \min_{\mathbf{w}} J_{\mathbf{w}}(\pi^{\mathrm{rob}}) = -\Delta. \tag{26}$$

Thus the minimax objective strictly prefers π^{rob}. □

Lemma 1 formalizes the failure mode of average-preference optimization: if a hard preference region has sufficiently small measure, then an expectation objective can prefer a policy that improves the majority region while sacrificing that hard region. In contrast, the minimax objective cannot ignore such regions, since its value is determined by the worst-performing preference. This is empirically supported by Figure 1.

E.3 KL Regularization in MAESTRO

MAESTRO, like every PPO-based baseline, is trained on the KL-regularized scalarized reward $\mathbf{w}^\top \mathbf{r}(\mathbf{x}, \mathbf{y}) - \beta \log \frac{\pi_\theta(\mathbf{y}|\mathbf{x},\mathbf{w})}{\pi_{\mathrm{ref}}(\mathbf{y}|\mathbf{x},\mathbf{w})}$. Although this coefficient β matches that of the baselines that train one policy per preference (MORLHF) or one expert per objective (the soups), the effective KL strength of a single conditioned model can differ from theirs, since one model has to serve every preference and the part of its behavior that is shared across them sees each objective only with the weight that the preference distribution gives it, whereas each baseline devotes a separate model, and at equal epochs its entire budget of updates, to one preference or objective.

KL regularization as a source of ε. For fixed $\mathbf{x}$ and $\mathbf{w}$, the maximizer of $\mathbb{E}_{\mathbf{y}\sim\pi}[\mathbf{w}^\top \mathbf{r}(\mathbf{x}, \mathbf{y})] - \beta\, \mathrm{KL}(\pi \,\|\, \pi_{\mathrm{ref}})$ is the tilted distribution

$$\pi^*_{\mathbf{w},\beta}(\mathbf{y} \mid \mathbf{x}) = \frac{\pi_{\mathrm{ref}}(\mathbf{y} \mid \mathbf{x})\, e^{\mathbf{w}^\top \mathbf{r}(\mathbf{x},\mathbf{y})/\beta}}{\mathbb{E}_{\mathbf{y}'\sim\pi_{\mathrm{ref}}}\left[e^{\mathbf{w}^\top \mathbf{r}(\mathbf{x},\mathbf{y}')/\beta}\right]}, \tag{27}$$

where $\pi_{\mathrm{ref}}(\cdot \mid \mathbf{x})$ denotes the reference on the input the policy sees, which is $\pi_{\mathrm{ref}}(\cdot \mid \mathbf{x}, \mathbf{w})$ for MAESTRO . Training targets this tilted distribution, whereas $V^*(\mathbf{w})$ from Appendix E.2 is defined on the reward alone, so the KL penalty keeps even a perfect solution below $V^*(\mathbf{w})$ by

$$\varepsilon_\beta(\mathbf{w}) \coloneqq \big(V^*(\mathbf{w}) - J_{\mathbf{w}}(\pi^*_{\mathbf{w},\beta})\big)_+.$$

Hence, if the conditioned policy comes within $\varepsilon_{\mathrm{approx}}$ of the regularized optimum at every $\mathbf{w} \in \mathcal{W}$, in the sense that $J_{\mathbf{w}}(\pi_\theta) \geq J_{\mathbf{w}}(\pi^*_{\mathbf{w},\beta}) - \varepsilon_{\mathrm{approx}}$, then (14) holds with

$$\varepsilon \;=\; \varepsilon_{\mathrm{approx}} \;+\; \sup_{\mathbf{w} \in \mathcal{W}} \varepsilon_\beta(\mathbf{w}). \tag{28}$$

The first term is the cost of sharing one model across preferences, which is what realizability is about. The second decreases as β decreases and is shared by methods that agree on β, the reference distribution and $\mathbf{r}$.

Proposition 2 (KL scaling for the preference-shared component)**.** *Consider (i) a per-objective baseline trained at the vertex $\mathbf{e}_m$ with coefficient β_b, whose optimum is $\pi^*_{\mathbf{e}_m,\beta_b} \propto \pi_{\mathrm{ref}}\, e^{r^{(m)}/\beta_b}$, and (ii) a single policy trained with $\mathbf{w} \sim p$ and coefficient β_{ours}, and write $\boldsymbol{\mu} \coloneqq \mathbb{E}_p[\mathbf{w}]$. If the policy does not depend on $\mathbf{w}$, its optimum is $\pi^*_{\boldsymbol{\mu},\beta_{\mathrm{ours}}} \propto \pi_{\mathrm{ref}}\, e^{\boldsymbol{\mu}^\top \mathbf{r}/\beta_{\mathrm{ours}}}$, and the coefficient multiplying $r^{(m)}$ in its log-ratio to the reference equals the baseline's, for $\mu_m > 0$, if and only if*

$$\beta_{\mathrm{ours}} = \beta_b\, \mu_m. \tag{29}$$

In particular, if p is symmetric over Δ^{M-1}, then $\mu_m = 1/M$ and $\beta_{\mathrm{ours}} = \beta_b/M$ for every m.

Proof. Since the policy does not depend on $\mathbf{w}$, linearity gives $\mathbb{E}_{\mathbf{w}\sim p}\mathbb{E}_{\mathbf{y}\sim\pi}[\mathbf{w}^\top \mathbf{r}] = \mathbb{E}_{\mathbf{y}\sim\pi}[\boldsymbol{\mu}^\top \mathbf{r}]$, and the KL term does not depend on $\mathbf{w}$. The objective is therefore the regularized objective with reward $\boldsymbol{\mu}^\top \mathbf{r}$, whose maximizer is (27) with $\mathbf{w}$ replaced by $\boldsymbol{\mu}$; its log-ratio to π_{ref} has coefficient $\mu_m/\beta_{\mathrm{ours}}$ on $r^{(m)}$, against $1/\beta_b$ for the baseline. With the prefixed reference of MAESTRO the KL term does depend on $\mathbf{w}$, but $\mathbb{E}_{\mathbf{w}\sim p}[\mathrm{KL}(\pi \,\|\, \pi_{\mathrm{ref}}(\cdot \mid \mathbf{x}, \mathbf{w}))] = \mathrm{KL}(\pi \,\|\, q) + \text{const}$ with q the normalized geometric mean of $\pi_{\mathrm{ref}}(\cdot \mid \mathbf{x}, \mathbf{w})$ over p, so the optimum is the same expression with π_{ref} replaced by q and the coefficient on $r^{(m)}$ unchanged. □

Scope. The proposition is exact only for a policy that ignores the preference. At the level of gradients it describes the part of any conditioned policy's update that is shared across preferences: since $\nabla_\theta J_{\mathbf{w}} = \sum_m w_m \nabla_\theta J^{(m)}$, objective m contributes $\mu_m \nabla_\theta J^{(m)}$ there, while the KL gradients of all preferences pull that part toward the reference without attenuation. The part that varies with $\mathbf{w}$ instead receives reward and KL in the same proportion, and under perfect conditioning the objective decouples across preferences, so β_b is already the matching coefficient. The scaling is therefore a correction for incomplete conditioning, and a compromise rather than a bound: the matching coefficient is $\beta_b\, \mu_m/w_m$ against a policy trained at $\mathbf{w}$, so it differs across objectives, and the adversary moves $\boldsymbol{\mu}$ during training. Empirically the gains in Figure 10 are consistent with the conditioning being incomplete within our training budget, though those runs also raise the target KL, so part of the gain may come from the larger budget.

E.4 The Ambiguity Set and How Much It Smooths

Two choices separate Algorithm 1 from the objective (4): the ambiguity set the inner player ranges over, and the statistic its descent direction is computed from. We state here what each of them costs, and what we do not claim.

Taking $\mathcal{P} = \Delta(\mathcal{W})$ would be degenerate: the inner objective is linear in p, so its minimum is attained at a point mass, and the resulting envelope credits improvement only at the minimizing preference and moves discontinuously as that preference changes. Restricting $\mathcal{P}$ to Dirichlet distributions of bounded concentration removes both problems, at the price of a smoothed rather than pointwise worst case, since such a distribution cannot place all its mass at one preference. The next proposition makes the amount of smoothing explicit and shows that $\alpha_{\max}$ interpolates between the two.

Proposition 3 (The Dirichlet family recovers the pointwise worst case as concentration grows)**.** *Fix θ and suppose $\mathbf{w} \mapsto J_{\mathbf{w}}(\pi_\theta)$ is continuous on $\mathcal{W}$. Then*

$$\inf_{\boldsymbol{\alpha} \in \mathbb{R}^M_{>0}} \; \mathbb{E}_{\mathbf{w}\sim \mathrm{Dir}(\boldsymbol{\alpha})}\big[J_{\mathbf{w}}(\pi_\theta)\big] \;=\; \min_{\mathbf{w} \in \mathcal{W}} J_{\mathbf{w}}(\pi_\theta). \tag{30}$$

If in addition $\mathbf{w} \mapsto J_{\mathbf{w}}(\pi_\theta)$ *is L-Lipschitz with respect to* $\|\cdot\|_1$*, then restricting to* $\mathcal{A} = [\alpha_{\min}, \alpha_{\max}]^M$ *costs at most*

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbb{E}_{\mathbf{w} \sim \mathrm{Dir}(\boldsymbol{\alpha})}\big[J_{\mathbf{w}}(\pi_\theta)\big] - \min_{\mathbf{w} \in \mathcal{W}} J_{\mathbf{w}}(\pi_\theta) \;\le\; L\left(\sqrt{\frac{M}{\alpha_{\max}+1}} + \frac{2M\alpha_{\min}}{\alpha_{\max}}\right). \tag{31}$$

Proof. Write $\boldsymbol{\alpha} = \kappa\boldsymbol{\mu}$ with $\boldsymbol{\mu} = \boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|_1 \in \Delta^{M-1}$ and $\kappa = \|\boldsymbol{\alpha}\|_1$. Under $\mathrm{Dir}(\kappa\boldsymbol{\mu})$ we have $\mathbb{E}[w^{(i)}] = \mu_i$ and $\mathrm{Var}[w^{(i)}] = \mu_i(1-\mu_i)/(\kappa+1)$, so by Jensen and Cauchy–Schwarz

$$\mathbb{E}\big[\|\mathbf{w} - \boldsymbol{\mu}\|_1\big] \;\le\; \sum_{i=1}^{M} \sqrt{\tfrac{\mu_i(1-\mu_i)}{\kappa+1}} \;\le\; \sqrt{\frac{M}{\kappa+1}}. \tag{32}$$

Hence $\mathrm{Dir}(\kappa\boldsymbol{\mu}) \Rightarrow \delta_{\boldsymbol{\mu}}$ as $\kappa \to \infty$, and by bounded continuity $\mathbb{E}_{\mathbf{w} \sim \mathrm{Dir}(\kappa\boldsymbol{\mu})}[J_{\mathbf{w}}] \to J_{\boldsymbol{\mu}}$. Taking the infimum over $\boldsymbol{\mu}$ bounds the left-hand side by $\inf_{\boldsymbol{\mu}} J_{\boldsymbol{\mu}} = \min_{\mathbf{w}} J_{\mathbf{w}}$; the reverse inequality holds since any average of $\{J_{\mathbf{w}}\}$ is at least $\min_{\mathbf{w}} J_{\mathbf{w}}$. For the second claim, let $\mathbf{w}^\star \in \arg\min_{\mathbf{w}} J_{\mathbf{w}}$ and $\delta \coloneqq \alpha_{\min}/\alpha_{\max}$, which satisfies $M\delta \le 1$ in the regime we use. Project $\mathbf{w}^\star$ onto $\Delta_\delta \coloneqq \{\boldsymbol{\mu} : \mu_i \ge \delta\}$ via $\boldsymbol{\mu} \coloneqq (1 - M\delta)\mathbf{w}^\star + \delta\mathbf{1}$, so $\|\boldsymbol{\mu} - \mathbf{w}^\star\|_1 \le 2M\delta$ and $J_{\boldsymbol{\mu}} \le J_{\mathbf{w}^\star} + 2LM\delta$. The choice $\boldsymbol{\alpha} = \alpha_{\max}\boldsymbol{\mu}$ lies in $\mathcal{A}$ since $\alpha_{\max}\mu_i \ge \alpha_{\min}$, and has $\kappa = \alpha_{\max}$; combining with (32) and L-Lipschitzness gives (31). □

The two clamps therefore play different roles rather than trading off against each other. The bound is dominated by its first term, so $\alpha_{\max}$ sets how robust the objective is, at rate $O(\alpha_{\max}^{-1/2})$; with $(\alpha_{\min}, \alpha_{\max}) = (10^{-3}, 10^2)$ and $M = 2$ it reads $L\sqrt{2/101} \approx 0.14L$. The clamp $\alpha_{\min} > 0$ instead makes the adversary's estimator well-posed: the Dirichlet score $s_i = \log w^{(i)} - \psi(\alpha_i) + \psi(S)$ has $\mathbb{E}[s_i] = 0$ and $\mathbb{E}[s_i^2] = \psi'(\alpha_i) - \psi'(S) \le \psi'(\alpha_{\min})$, and $\psi'(\alpha) \sim \alpha^{-2}$ as $\alpha \downarrow 0$, so without it the estimator has unbounded second moment.

It is sometimes supposed that a worst-case objective will concentrate on a vertex of the simplex, *i.e.,* on a single objective. A vertex, however, evaluates to the best attainable value of a single objective, which is typically large, so there is no reason for the minimum to sit there. This is what we observe: the scalarized return at the two vertices is far above its value near the center of the simplex (Figure 2(a)), so the adversary concentrates on genuinely conflicting interior trade-offs rather than collapsing onto one objective.

E.5 What the Adversary Is Scored On

The adversary's descent direction is computed from the surrogate (10) rather than from $\mathbf{w}^\top\mathbf{r}$ directly, while the policy is still trained on $\mathbf{w}^\top\mathbf{r}$. The two ingredients play separate roles: UEG is a positivity-preserving update rule for the Dirichlet parameters, whereas (10) is a batch-relative, per-objective feedback signal chosen to reduce sensitivity to reward scales, and not the exact payoff of (6). Because the comparison samples are treated as fixed when credit is assigned to each sampled preference, the update is a stochastic gradient heuristic rather than an unbiased gradient of (6): the minimax formulation is what motivates it, and the sampler it adapts is evaluated empirically in Section 4. The statistic interpolates between two regimes. When $\tau\,|r_j^{(m)} - r_{j'}^{(m)}| \ll 1$ within the batch, $\sigma(\tau d) \approx \frac{1}{2} + \frac{\tau}{4}d$ and $\sum_{j' \ne j}(r_j^{(m)} - r_{j'}^{(m)}) = B_{\mathrm{adv}}(r_j^{(m)} - \bar{r}^{(m)})$ give $\tilde{r}_j^{(m)} \approx \frac{1}{2} + \frac{\tau}{4}\frac{B_{\mathrm{adv}}}{B_{\mathrm{adv}}-1}(r_j^{(m)} - \bar{r}^{(m)})$, with $\bar{r}^{(m)}$ the batch mean, and hence

$$R_j \;\approx\; \tfrac{1}{2} + \tfrac{\tau}{4}\tfrac{B_{\mathrm{adv}}}{B_{\mathrm{adv}}-1}\big(\mathbf{w}_j^\top\mathbf{r}_j - \mathbf{w}_j^\top\bar{\mathbf{r}}\big) : \tag{33}$$

the slope is common to all objectives and each objective's level is removed, so the feedback in this regime reduces, up to an affine transformation that the baseline and the step size absorb, to the batch-centered scalarized reward $\mathbf{w}_j^\top(\mathbf{r}_j - \bar{\mathbf{r}})$. The centering is by the batch mean and therefore moves with the sampled preferences, so this characterizes the signal rather than a fixed objective the adversary minimizes. When the spread is large compared with $1/\tau$, the statistic approaches the rank itself, which is invariant to any strictly increasing transformation of each objective, the property on which BOtied (Park et al., 2024b) builds its multi-objective criterion.

Under exact simultaneous realizability (Proposition 1 with $\varepsilon = 0$), a pointwise-optimal policy remains a population optimum under any fixed sampling distribution. Outside this regime, we interpret (10) only as a heuristic that shifts sampling toward preferences receiving lower batch-relative feedback.