STEERDPO: STEERABLE MULTI-OBJECTIVE ALIGNMENT VIA IMPLICIT REWARD SCALARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligning large language models (LLMs) with diverse, often conflicting human preferences is a fundamental challenge in building versatile, controllable AI assistants. While multi-objective alignment aims to address such trade-offs, existing approaches typically require costly online training or multiple explicit reward models, or rely on heuristic parameter-level aggregation with limited guarantees about the resulting preference trade-offs. We introduce SteerDPO, a single-stage, reward-model-free offline multi-objective alignment method that learns a single steerable policy directly from per-objective pairwise preference data. SteerDPO enables inference-time steering across trade-offs by explicitly conditioning on arbitrary preference weights, without additional training or online data collection. To this end, we derive the optimal policy for any preference weight as a normalized weighted geometric mean of the one-hot optimal policies under KL-regularized linear scalarization. SteerDPO uses this identity to train a single preference-conditioned policy through offline self-distillation. Two- and three-objective synthetic bandits and LLM experiments evaluate reward quality, steering, and front geometry separately. SteerDPO leads the compared methods on requested reward and steerability in the three-seed Llama-3.2-3B and single-seed Qwen3-8B comparisons.

1 INTRODUCTION

Modern language models must balance diverse human preferences, such as helpfulness, safety, and truthfulness. The hard part is that these objectives often conflict; satisfying one objective can require sacrificing another. Multi-objective alignment (MO-alignment) therefore aims to recover policies along the Pareto front, where no objective can be improved without worsening another. At inference time, the user or system specifies a preference vector, and the model should generate responses that match the requested trade-off (Yang et al., 2024; Son et al., 2026; Zhou et al., 2024; Liu et al., 2024).

Existing approaches to MO-alignment mostly follow one of the following strategies. A straightforward approach is to train explicit reward models for each preference dimension and optimize a separate policy for each selected preference vector using a linearly scalarized reward (Zhou et al., 2024; Yang et al., 2024; Li et al., 2025). However, this approach requires additional policy training for each new preference vector, making it costly to cover a wide range of trade-offs. Another approach combines objective-specific information in parameter space (Ramé et al., 2023; Zhong et al., 2024), gradient space (Li et al., 2025), or policy space (Mudgal et al., 2024; Son et al., 2026). These strategies can reduce repeated policy training, but many still rely on explicit reward models or multiple objective-specific policies. Moreover, heuristic aggregation in parameter, gradient, or policy space does not by itself guarantee optimality for the desired scalarized objective. These limitations motivate the question: **Can a single preference-conditioned model cover the Pareto front without explicit reward models or separate policy training for each preference weight?**

We address this problem by deriving a closed-form optimal policy under KL-regularized linear scalarization. We show that this policy is a normalized weighted geometric mean of the optimal policies at one-hot preference weights, for any choice of preference weights. Leveraging this mathematical structure, we introduce SteerDPO, a reward-model-free, offline MO-alignment method that learns a single steerable policy directly from per-objective pairwise preference data. SteerDPO constructs geometric-mean targets from its one-hot-conditioned policies and distills them into a single preference-conditioned model, enabling inference-time steering across trade-offs without additional

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.

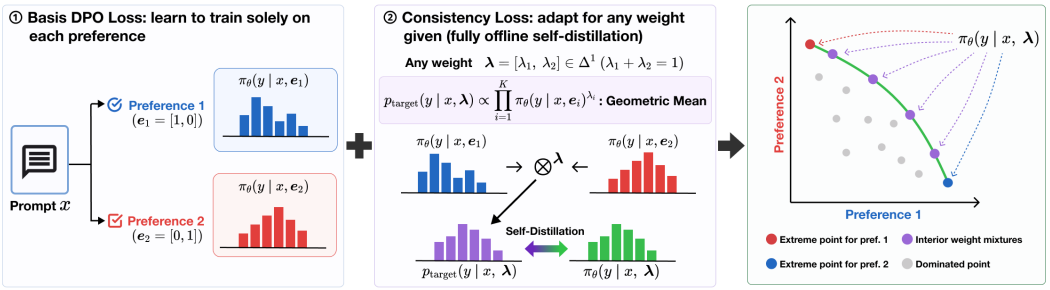


Figure 1: Overview of STEERDPO. **Left:** per-objective DPO training learns objective-specific behavior for one-hot preference vectors. **Middle:** for an arbitrary preference weight $\boldsymbol{\lambda}$, the target policy is obtained by the normalized weighted geometric mean of the per-objective, basis policies, enabling offline self-distillation without collecting new pairwise preferences for every mixture. **Right:** the resulting weight-conditioned model traverses the Pareto trade-off surface, preserving extreme-point behavior while covering interior preference mixtures with a single policy.

training, online data collection, or multiple models at decoding time. Figure 1 provides an overview of SteerDPO.

Contributions.

- **Closed-form optimal policies for MO-alignment.** We formulate multi-objective preference alignment as a KL-regularized contextual bandit. Under this formalization, we use the normalized geometric-mean characterization of optimal policies to construct sequence-level targets for arbitrary preference weights.
- **SteerDPO framework.** Using this identity, we introduce SteerDPO, a reward-model-free, offline multi-objective alignment algorithm that learns from per-objective pairwise preference data. SteerDPO constructs geometric-mean targets from its one-hot-conditioned policies and distills them into a single preference-conditioned policy, eliminating the need for explicit reward models or separate training runs for interior preference weights.
- **Empirical evaluation of reward performance and steering.** We evaluate reward quality, steerability, and front geometry on two LLM backbones and in controlled two- and three-objective bandits. The LLM results show strong requested-reward and steering performance using one conditioned policy at inference.

2 RELATED WORK

Multi-objective RLHF and alignment. Multi-objective RLHF extends single-objective preference optimization by accounting for trade-offs among multiple preferences, such as helpfulness, harmlessness, and other user-specified objectives (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Ji et al., 2025; Zhou et al., 2024; Yang et al., 2024). Recent approaches vary in how they handle this trade-off, including extending single-preference offline or online methods, interpolating in parameter space (Ramé et al., 2023; Zhong et al., 2024), combining gradients (Li et al., 2025), and training preference-conditioned policies (Wang et al., 2024a; Guo et al., 2024; Wang et al., 2024b; Gupta et al., 2025; Ren et al., 2025). DPO reframes policy optimization through implicit rewards and KL-regularized preference matching without training explicit reward models (Rafailov et al., 2023), whereas on-policy methods such as PPO require explicit reward models for preference alignment (Schulman et al., 2017). Multi-objective methods combine reward values (Zhou et al., 2024; Yang et al., 2024), interpolate objective-specific model parameters (Ramé et al., 2023), or combine objective gradients (Li et al., 2025). Our work follows the DPO framework but aggregates basis policies at the distribution level and amortizes the resulting family into a single preference-conditioned model.

Scalarization theory and Pareto recovery. Linear scalarization is a classical tool for characterizing Pareto trade-offs in multi-objective optimization (Miettinen, 1999; Ehrgott, 2005). We study the KL-regularized contextual bandit formulation used in DPO-style alignment through the lens of linear scalarization. In this setting, the regularized per-objective values are concave in the stochastic policy,

so the dominated attainable set is convex. Therefore, nonnegative linear scalarization characterizes Pareto-efficient boundary points of the stochastic-policy problem by standard supporting-hyperplane arguments (Rockafellar, 1970; Boyd & Vandenberghe, 2004). This theoretical property can be combined with geometric aggregation to construct the policy corresponding to a trade-off point without explicitly training a separate policy for every point.

Scalable decoding at inference time. Some methods optimize multiple objectives by aggregating single-objective policies at inference time rather than introducing additional multi-objective training. They first train or reuse objective-specific policies, value functions, or scorers and then aggregate them at decoding time (Mudgal et al., 2024; Shi et al., 2024; Liu et al., 2024; Son et al., 2026). SteerDPO uses sequence-level composition to construct offline supervision and jointly learn basis and interior behavior within one preference-conditioned model, which is used alone at inference.

3 PRELIMINARIES

3.1 MULTI-OBJECTIVE PREFERENCE ALIGNMENT

We consider multi-objective preference alignment, where each prompt-response pair (x, y) is associated with a reward vector $\boldsymbol{r}(x, y) = [r_1(x, y), \ldots, r_K(x, y)]^\top \in \mathbb{R}^K$, with each component corresponding to a distinct alignment objective. A preference vector $\boldsymbol{\lambda} \in \Delta^{K-1}$ specifies the relative importance of the objectives, where $\Delta^{K-1} := \{\boldsymbol{\lambda} \in \mathbb{R}_+^K : \sum_{k=1}^K \lambda_k = 1\}$. For a given $\boldsymbol{\lambda}$, the goal is to maximize the linearly scalarized reward $\boldsymbol{\lambda}^\top \boldsymbol{r}(x, y)$ while remaining close to a reference policy π_{ref}:

$$\max_{\pi} \mathcal{L}_{\boldsymbol{\lambda}}(\pi) := \mathbb{E}_{x\sim\mathcal{D}_x,\ y\sim\pi(\cdot|x)} \left[\boldsymbol{\lambda}^\top \boldsymbol{r}(x, y) - \beta D_{\text{KL}}\left(\pi(\cdot \mid x) \| \pi_{\text{ref}}(\cdot \mid x)\right)\right], \tag{1}$$

where $\mathcal{D}_x$ denotes the prompt distribution, π_{ref} is the reference policy, and $\beta > 0$ controls the strength of KL regularization. We use the same reference policy and regularization coefficient β across preference weights. For the one-hot preference vector $\boldsymbol{e}_k \in \Delta^{K-1}$, whose k-th entry is one and all others are zero, this objective reduces to single-objective alignment with respect to the k-th objective. More generally, varying $\boldsymbol{\lambda}$ yields a family of optimal policies $\pi^*_{\boldsymbol{\lambda}} \in \arg\max_\pi \mathcal{L}_{\boldsymbol{\lambda}}(\pi)$. Our goal is to learn a single preference-conditioned policy $\pi_\theta(\cdot \mid x, \boldsymbol{\lambda})$ that approximates $\pi^*_{\boldsymbol{\lambda}}(\cdot \mid x)$ for each preference vector $\boldsymbol{\lambda}$, so that the desired trade-off can be specified at inference time without additional training. However, the reward functions are generally unknown in practice, and we assume access only to pairwise preference data for each objective.

3.2 LEARNING FROM PAIRWISE PREFERENCES

We assume access to a preference dataset $\mathcal{D} = \{(x, y_1, y_2, \boldsymbol{p})_n\}_{n=1}^N$, where $\boldsymbol{p} \in \{0, 1\}^K$ records pairwise preferences for the K objectives. For each objective k, $p_k = \mathbb{1}[y_2 \succ_k y_1]$, where $\succ_k$ denotes preference according to objective k. We do not assume access to numerical reward scores or additional comparisons collected for each preference mixture. Under the Bradley-Terry model (Bradley & Terry, 1952), the probability of preferring y_2 to y_1 for objective k is given by:

$$P(y_2 \succ_k y_1 \mid x) = \frac{1}{1 + \exp\left(r_k(x, y_1) - r_k(x, y_2)\right)} = \sigma\big(r_k(x, y_2) - r_k(x, y_1)\big), \tag{2}$$

where σ denotes the sigmoid function. This model relates observed comparisons to differences in the underlying rewards, allowing a reward model to be learned by maximizing the likelihood of the preference labels.

Direct preference optimization (DPO) (Rafailov et al., 2023) instead expresses this likelihood directly in terms of a policy, avoiding explicit reward-model training. For objective k, the KL-regularized optimal policy $\pi^*_{\boldsymbol{e}_k}$ takes the form

$$\pi^*_{\boldsymbol{e}_k}(y \mid x) = \tfrac{1}{Z_{\boldsymbol{e}_k}(x)} \pi_{\text{ref}}(y \mid x) \exp\Big(\tfrac{1}{\beta} r_k(x, y)\Big), \tag{3}$$

where $Z_{\boldsymbol{e}_k}(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp(r_k(x, y)/\beta)$ is the partition function. Rearranging this expression gives:

$$r_k(x, y) = \beta \log \frac{\pi^*_{\boldsymbol{e}_k}(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z_{\boldsymbol{e}_k}(x), \tag{4}$$

Substituting this identity into Equation (2) cancels the prompt-dependent term $\beta \log Z_{\boldsymbol{e}_k}(x)$.

Let $\mathcal{D}_k$ denote the comparisons in $\mathcal{D}$ ordered as (x, y_w, y_l) according to the label p_k, with y_w preferred to y_l for objective k. For a parametrized policy π, the resulting DPO loss for objective k is (Rafailov et al., 2023):

$$\mathcal{L}_{\mathrm{DPO}}^{(k)}(\theta) := -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_k}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right]. \tag{5}$$

Thus, the available comparisons can be used to learn the policies associated with one-hot preference weights without fitting explicit reward models.

The remaining challenge is **sparse supervision over preference weights**: direct comparisons are available for individual objectives, but not for arbitrary mixtures. When objectives disagree on a response pair, their binary labels alone do not specify the relative magnitudes of the reward differences and therefore do not determine the preference under a weighted combination of rewards. This motivates our approach in the next section: using the structure of the KL-regularized objective to **construct supervision for arbitrary preference weights from the one-hot-conditioned policies**.

4 SteerDPO: Denser Supervision for Offline Self-Distillation

A direct approach to optimizing Equation (1) is to learn a reward model for each objective and then optimize a policy against the weighted combination of the learned rewards. DPO avoids explicit reward modeling in *single-objective* alignment, but extending this approach to *multi-objective* alignment with arbitrary preference weights is not straightforward. The key challenge is that per-objective preference labels do not directly specify preferences under a weighted combination of rewards. We address this by first expressing the optimal policy for each preference weight in terms of the per-objective optimal policies. This relationship allows SteerDPO to construct offline self-distillation targets and jointly optimize per-objective preference learning and distillation within a single preference-conditioned policy, without explicit reward models.

4.1 Closed-Form Optimal Policies for Arbitrary Preference Weights

We show that the optimal policy for any preference weight $\boldsymbol{\lambda}$ is a normalized weighted geometric mean of the per-objective optimal policies $\{\pi^*_{\boldsymbol{e}_k}\}_{k=1}^K$. Our derivation starts from the optimal policy for Equation (1), obtained by substituting the scalarized reward $\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)$ into Equation (3):

$$\pi^*_{\boldsymbol{\lambda}}(y \mid x) = \frac{1}{Z_{\boldsymbol{\lambda}}(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}\sum_{k=1}^K \lambda_k r_k(x,y)\right), \tag{6}$$

where $Z_{\boldsymbol{\lambda}}(x) = \sum_{y'}\pi_{\mathrm{ref}}(y' \mid x)\exp\left(\frac{1}{\beta}\boldsymbol{\lambda}^\top\boldsymbol{r}(x,y')\right)$ denotes the partition function for the scalarized reward. Then, plugging $\boldsymbol{\lambda} = \boldsymbol{e}_k$ into Equation (6) and rearranging gives $\exp\left(\frac{1}{\beta}r_k(x,y)\right) = Z_{\boldsymbol{e}_k}(x)\frac{\pi^*_{\boldsymbol{e}_k}(y|x)}{\pi_{\mathrm{ref}}(y|x)}$ for each objective k. Using this identity, we rewrite Equation (6) as

$$\pi^*_{\boldsymbol{\lambda}}(y \mid x) = \frac{1}{Z_{\boldsymbol{\lambda}}(x)}\pi_{\mathrm{ref}}(y \mid x)\prod_{k=1}^K\left[\exp\left(\frac{r_k(x,y)}{\beta}\right)\right]^{\lambda_k} \tag{7}$$

$$= \frac{1}{Z_{\boldsymbol{\lambda}}(x)}\pi_{\mathrm{ref}}(y \mid x)\prod_{k=1}^K\left[Z_{\boldsymbol{e}_k}(x)\frac{\pi^*_{\boldsymbol{e}_k}(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}\right]^{\lambda_k} \tag{8}$$

$$= \frac{\prod_{k=1}^K Z_{\boldsymbol{e}_k}(x)^{\lambda_k}}{Z_{\boldsymbol{\lambda}}(x)}\cancel{\left(\pi_{\mathrm{ref}}(y \mid x)^{1-\sum_{k=1}^K\lambda_k}\right)}\prod_{k=1}^K \pi^*_{\boldsymbol{e}_k}(y \mid x)^{\lambda_k} \tag{9}$$

$$= \frac{\prod_{k=1}^K Z_{\boldsymbol{e}_k}(x)^{\lambda_k}}{Z_{\boldsymbol{\lambda}}(x)}\prod_{k=1}^K \pi^*_{\boldsymbol{e}_k}(y \mid x)^{\lambda_k}. \tag{10}$$

The reference-policy factor equals one because $\sum_{k=1}^{K} \lambda_k = 1$. Since $\frac{\prod_{k=1}^{K} Z_{\boldsymbol{e}_k}(x)^{\lambda_k}}{Z_{\boldsymbol{\lambda}}(x)}$ is independent of y, it serves as a normalization constant, yielding

$$\pi^*_{\boldsymbol{\lambda}}(y \mid x) = \frac{\prod_{k=1}^{K} \pi^*_{\boldsymbol{e}_k}(y \mid x)^{\lambda_k}}{\sum_{y'} \prod_{k=1}^{K} \pi^*_{\boldsymbol{e}_k}(y' \mid x)^{\lambda_k}} \propto \prod_{k=1}^{K} \pi^*_{\boldsymbol{e}_k}(y \mid x)^{\lambda_k}. \tag{11}$$

Thus, the scalarized optimal policy is exactly a normalized **weighted geometric mean of the per-objective optimal policies**, as also follows from the policy-composition framework of Shi et al. (2024). This identity allows us to construct optimal policies for arbitrary preference weights $\boldsymbol{\lambda}$ from the per-objective optimal policies $\pi^*_{\boldsymbol{e}_k}$, without explicitly recovering the reward functions. We represent the per-objective policies as $\pi_\theta(\cdot \mid x, \boldsymbol{e}_k)$ using a shared preference-conditioned model $\pi_\theta(y \mid x, \boldsymbol{\lambda})$ and train them with the corresponding DPO losses:

$$\mathcal{L}_{\text{basis}}(\theta) = -\sum_{k=1}^{K} \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_k} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x, \boldsymbol{e}_k)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x, \boldsymbol{e}_k)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right] \tag{12}$$

This loss trains the shared policy at one-hot preference weights $\boldsymbol{e}_k$ but does not directly supervise its behavior at other weights. We use the resulting one-hot-conditioned policies to construct geometric-mean targets for arbitrary preference weights $\boldsymbol{\lambda}$ and jointly optimize the corresponding self-distillation loss, as described next.

4.2 Offline Self-Distillation for Steerable Policy Learning

We use the geometric-mean identity in Equation (11) to construct a teacher for arbitrary preference weights $\boldsymbol{\lambda}$ from the shared model's one-hot-conditioned policies $\pi_\theta(\cdot \mid x, \boldsymbol{e}_k)$. Specifically, we replace each optimal policy $\pi^*_{\boldsymbol{e}_k}$ with its learned counterpart $\pi_\theta(\cdot \mid x, \boldsymbol{e}_k)$. For LLM training, we restrict the target to the two complete responses in each offline pair, $C(x) = \{y_w, y_l\}$. The sequence-level target is

$$P_{\text{target}}(y \mid x, \boldsymbol{\lambda}) = \text{sg}\left[\frac{\prod_{k=1}^{K} \pi_\theta(y \mid x, \boldsymbol{e}_k)^{\lambda_k}}{\sum_{y' \in C(x)} \prod_{k=1}^{K} \pi_\theta(y' \mid x, \boldsymbol{e}_k)^{\lambda_k}} \right], \qquad y \in C(x), \tag{13}$$

where sg stops gradients through the teacher. The target is exactly the weighted geometric mean of the learned basis policies restricted to $C(x)$; those policies approximate the optimal basis policies in Equation (11). This preserves the sequence-level target odds on each observed pair; Section D.2 details the relation to token-wise composition and sampling. Since teacher and student use the same model under different preference conditions, matching them provides offline self-distillation. Consistency is enforced only on the observed candidates and does not imply agreement over unobserved responses.

Training Objective We minimize reverse KL from the student to P_{target}:

$$\mathcal{L}_{\text{consistency}} = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D},\, \boldsymbol{\lambda}\in\Delta^{K-1}} \left[D_{\text{KL}} \left(\pi_\theta(\cdot \mid x, \boldsymbol{\lambda}) \,\|\, P_{\text{target}}(\cdot \mid x, \boldsymbol{\lambda}) \right) \right], \tag{14}$$

where the KL sum runs over the two complete responses in $C(x)$. We average this loss over the batch and M sampled weights per example. SteerDPO jointly optimizes the per-objective DPO loss and the self-distillation loss:

$$\mathcal{L}_{\text{SteerDPO}} = \mathcal{L}_{\text{basis}} + \alpha \mathcal{L}_{\text{consistency}}, \tag{15}$$

where $\alpha \geq 0$ controls the strength of self-distillation. The DPO loss $\mathcal{L}_{\text{basis}}$ trains the one-hot-conditioned policies $\pi_\theta(\cdot \mid x, \boldsymbol{e}_k)$ from pairwise preferences for each objective k, while the distillation loss $\mathcal{L}_{\text{consistency}}$ trains $\pi_\theta(\cdot \mid x, \boldsymbol{\lambda})$ at sampled preference weights $\boldsymbol{\lambda}$ to match the geometric-mean teacher $P_{\text{target}}(\cdot \mid x, \boldsymbol{\lambda})$. Both losses update the shared model parameters θ, with gradients stopped through the teacher distribution. The teacher is recomputed from the current one-hot-conditioned policies during training, without a separate teacher-training stage. Sections D.1 and E give the sequence-level loss details and pseudocode.

5 Experiments

We evaluate whether SteerDPO learns a single policy that achieves strong reward performance while adapting to different preference trade-offs. We first study synthetic **multi-objective bandits**

to test whether pairwise preferences for individual objectives suffice to learn effective steering at intermediate preference weights. This controlled setting allows us to evaluate the learned policies directly using ground-truth rewards. We then evaluate **multi-objective LLM alignment** to examine whether this steering behavior extends to natural-language generation. Across these settings, we measure reward performance under the requested preference weights, steerability, and the coverage and geometry of the learned trade-off fronts.

5.1 Evaluation Metrics

We report four primary metrics. **Scalarized reward (SR)** measures the weighted reward $\boldsymbol{\lambda}^\top \boldsymbol{r}(x, y)$ of responses generated under the requested preference vector $\boldsymbol{\lambda}$. **Steerability** measures the rank correlation between each objective's weight and its realized reward, indicating whether increasing an objective's weight leads to higher reward for that objective. **Anchor-point performance** measures the reward for objective k when the policy is conditioned on the corresponding one-hot vector $\boldsymbol{e}_k$, assessing performance at the extremes of the preference space. **Hypervolume (HV)** measures the reward-space volume dominated by the mean-reward front relative to a shared reference point (Zitzler & Thiele, 1999). Higher values are preferable for all four metrics. Detailed metric definitions, normalization, aggregation, and supplementary front diagnostics are provided in Section I.1.

5.2 Synthetic Multi-Objective Bandit Results

We evaluate a shared contextual bandit with eight-dimensional Gaussian contexts, 30 actions, and two conflicting nonlinear reward functions. Training uses 30,000 contexts with objective-specific pairwise preferences; evaluation uses 3,000 held-out contexts and eleven weights $(a, 1-a)$ with $a \in \{0, 0.1, \ldots, 1\}$. We report equal-weight means over training seeds 42, 43, and 44. All seven methods use the same task and evaluation contexts; model settings, preference generation, and method-specific budgets are given in Section G.

Table 1: Synthetic two-objective bandit results: mean ± standard error (SE) over three training seeds (42–44). All columns use the shared seven-method normalization and are higher-is-better. HV is computed on each seed's mean-reward front.

Method	SR ↑	HV ↑	Steer. ↑	Anchor Avg ↑
MOD(DPO expert mixture)	-0.141 ± 0.012	2.032 ± 0.039	0.724 ± 0.017	0.044 ± 0.005
DPO objective 1	-0.492 ± 0.013	0.593 ± 0.052	0.000 ± 0.000	-0.492 ± 0.013
DPO objective 2	-0.521 ± 0.018	0.710 ± 0.044	0.000 ± 0.000	-0.521 ± 0.018
MODPO	-0.290 ± 0.004	1.655 ± 0.004	0.491 ± 0.018	-0.117 ± 0.014
MORLHF	-0.149 ± 0.013	2.008 ± 0.043	0.684 ± 0.017	0.035 ± 0.006
P-GAPO	-0.281 ± 0.009	1.721 ± 0.021	0.500 ± 0.015	-0.090 ± 0.038
SteerDPO (Ours)	$\mathbf{-0.109} \pm 0.002$	$\mathbf{2.129} \pm 0.010$	$\mathbf{0.812} \pm 0.007$	$\mathbf{0.078} \pm 0.002$

Table 1 and the left panel of Figure 2 report the two-objective results under the shared evaluation protocol. Section F reports front geometry and objective-wise anchors.

Three objectives. We also evaluate a three-objective bandit with the same context and action dimensions, training and evaluation set sizes, and three training seeds. The 31 evaluated preference vectors cover the simplex vertices, edges, and interior. Table 2 compares eight methods using a shared normalization and HV reference within this experiment. SteerDPO has the highest SR (0.329 ± 0.004), HV (1.682 ± 0.033), steerability (0.742 ± 0.003), and anchor average (0.554 ± 0.002). The strongest competing values are 0.276, 1.562, 0.678, and 0.515, respectively, from MOD(DPO expert mixture). The right panel of Figure 2 compares the empirical three-dimensional reward fronts of SteerDPO and MOD(DPO expert mixture); Section H shows reward variation across the preference simplex.

5.3 LLM Experimental Setup

On Anthropic-HH (Bai et al., 2022), we align harmlessness and helpfulness using Llama-3.2-3B (Grattafiori et al., 2024) with full fine-tuning and seeds 42, 43, and 44. We also

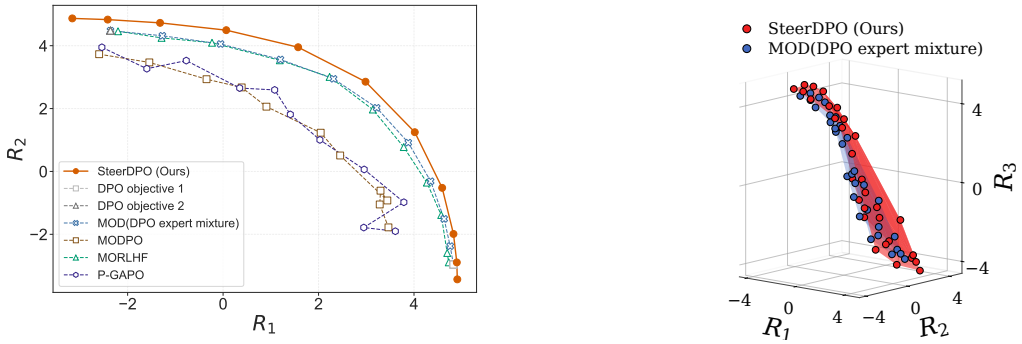


Figure 2: Synthetic reward trade-offs in raw reward units. Each point averages sampled-action rewards over 3,000 contexts and three seeds. **Left** ($K = 2$)**:** all eleven weight means for seven methods, connected in weight order. **Right** ($K = 3$)**:** SteerDPO (red) and MOD(DPO expert mixture) (blue) at the evaluated weights. For readability, we choose the comparison only with the strongest baseline, MOD. Axis lengths use the ratio $1 : 1 : 1.25$.

Table 2: Synthetic three-objective bandit results over training seeds 42–44. All methods use the same 31 preference vectors and 3,000 held-out contexts. HV is computed per seed on the mean-reward front. Normalization and the HV reference are shared across the K=3 evaluation pool; values are not directly comparable with the two-objective table.

Method	SR ↑	HV ↑	Steer. ↑	Anchor Avg ↑
MOD(DPO expert mixture)	0.276 ± 0.005	1.562 ± 0.018	0.678 ± 0.005	0.515 ± 0.003
MODPO	0.081 ± 0.007	1.136 ± 0.023	0.343 ± 0.014	0.225 ± 0.036
MORLHF	0.272 ± 0.002	1.535 ± 0.013	0.664 ± 0.004	0.507 ± 0.002
P-GAPO	-0.071 ± 0.003	0.719 ± 0.018	0.083 ± 0.006	0.056 ± 0.013
MO-ODPO	-0.093 ± 0.027	0.482 ± 0.042	0.013 ± 0.006	-0.087 ± 0.029
Rewarded Soups	0.110 ± 0.001	1.077 ± 0.018	0.448 ± 0.005	0.504 ± 0.001
RiC	0.211 ± 0.007	1.196 ± 0.030	0.621 ± 0.001	0.388 ± 0.010
SteerDPO (Ours)	$\mathbf{0.329} \pm 0.004$	$\mathbf{1.682} \pm 0.033$	$\mathbf{0.742} \pm 0.003$	$\mathbf{0.554} \pm 0.002$

evaluate Qwen3-8B (Yang et al., 2025) with LoRA (Hu et al., 2022) ($r = 64$, seed 42). Both comparisons use 4,664 distinct matched prompts. MODPO and MORLHF use five weights $\boldsymbol{\lambda}(a) = (a, 1-a)$, $a \in \{0, 0.2, 0.5, 0.8, 1\}$; the other methods use eleven, $a \in \{0, 0.1, \ldots, 1\}$. Figures show all evaluated weights, while tables compare methods at the five shared weights.

We compare SteerDPO with MODPO (Zhou et al., 2024), MORLHF (Zhou et al., 2024), MO-ODPO (Gupta et al., 2025), Rewarded Soups (Ramé et al., 2023), and RiC (Yang et al., 2024). All methods are evaluated with the same oracles. Main policy datasets are relabeled by the trained oracles to ensure the consistent preference signal. On-policy baselines use separate proxy reward models during training. For these baselines, this avoids directly hacking the evaluator and the oracles remain as *common evaluation scorers*. By this, we can assume the oracles as preference labelers, and proxies as the explicit RMs. The details on the RMs are in (Section I.2). SteerDPO jointly optimizes basis DPO and consistency loss with $\alpha = 2$ and $M = 8$ sampled weights. The consistency distributions are restricted to the two offline candidates.

Table metrics use shared reward normalization within each backbone, while figures show raw oracle rewards. Llama tables report means $\pm$ standard errors across three training seeds; Qwen reports single seed results. Sections I and K give the protocol and training settings.

5.4 Main Results

Trade-offs across training seeds. Across three Llama training seeds, SteerDPO has the highest requested reward (SR 0.499), steerability (0.627), anchor average (0.834), and mean-front HV (23.348) among the compared methods (Table 3). RiC obtains SR 0.338, steerability 0.505, and anchor average 0.545. SteerDPO also has the greatest extent (2.419) and the lowest mean spread Δ (0.563). MO-ODPO has the lowest IGD$^+$ (0.105, versus SteerDPO's 0.147), so the best requested-reward and coverage metrics do not coincide on every diagnostic (Table 11 and Figure 3).

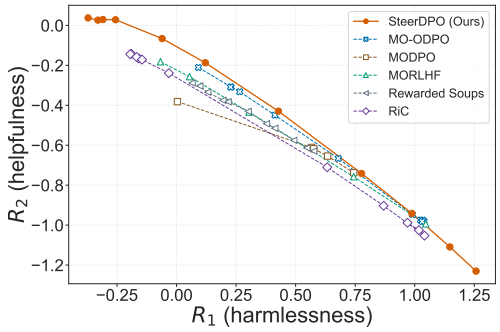


Figure 3: HH/Llama: mean raw rewards over seeds 42–44. MODPO and MORLHF have five points each; the other methods have eleven. Lines connect weights in order.

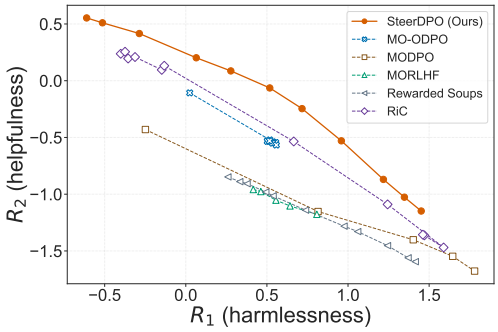


Figure 4: HH/Qwen3-8B with LoRA, seed 42. MODPO and MORLHF have five points each; the other methods have eleven. Each point averages 4,664 matched prompts.

Table 3: HH/Llama at five shared weights over seeds 42–44. HV uses each seed's mean-reward front. Bold marks column maxima.

Method	SR ↑	Steerability ↑	Anchor Avg ↑	HV ↑
MODPO	0.155 ± 0.004	0.217 ± 0.008	0.286 ± 0.019	19.667 ± 0.112
MORLHF	0.318 ± 0.013	0.452 ± 0.015	0.582 ± 0.021	21.791 ± 0.134
MO-ODPO	0.372 ± 0.005	0.463 ± 0.008	0.556 ± 0.008	21.665 ± 0.035
Rewarded Soups	0.130 ± 0.005	0.214 ± 0.003	0.239 ± 0.010	19.634 ± 0.069
RiC	0.338 ± 0.005	0.505 ± 0.005	0.545 ± 0.007	21.544 ± 0.053
SteerDPO (Ours)	$\mathbf{0.499} \pm 0.003$	$\mathbf{0.627} \pm 0.004$	$\mathbf{0.834} \pm 0.006$	$\mathbf{23.348} \pm 0.030$

A larger backbone with LoRA. We further run experiments for Qwen3-8B with LoRA, to show that our method is applicable to parameter-efficient fine-tuning and larger models. On Qwen3-8B, SteerDPO has the highest SR (0.681), steerability (0.764), anchor average (1.005), and mean-front HV (18.979) among the compared methods (Table 4). RiC is next on these metrics, with SR 0.582 and steerability 0.647. SteerDPO also has the lowest spread Δ and IGD$^+$ (Table 11 and Figure 4).

Table 4: HH/Qwen3-8B with LoRA ($r = 64$), seed 42, at five shared weights. Metrics use a common normalization within this panel, separate from Llama.

Method	SR ↑	Steerability ↑	Anchor Avg ↑	HV ↑
MODPO	0.401	0.603	0.728	15.283
MORLHF	−0.071	0.153	0.010	11.721
MO-ODPO	0.145	0.186	0.267	14.431
Rewarded Soups	0.169	0.464	0.361	13.299
RiC	0.582	0.647	0.869	17.169
SteerDPO (Ours)	**0.681**	**0.764**	**1.005**	**18.979**

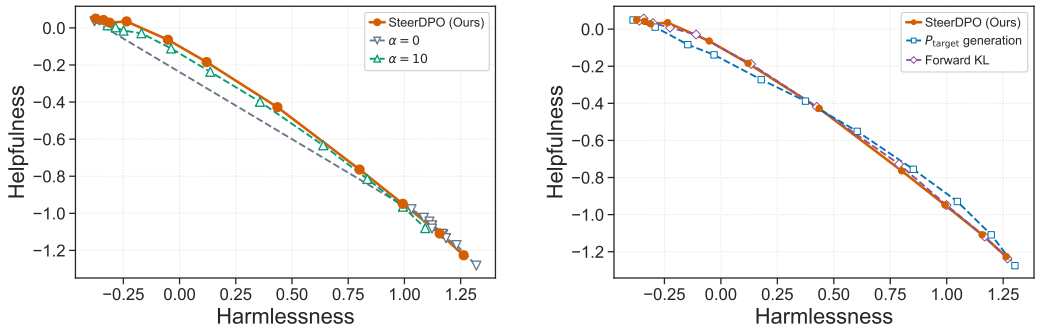
5.5 Ablation Studies

Table 5 and Figure 5 examine consistency strength, the direction of KL matching, and generation by composing the basis policies at inference time. The reference uses oracle-relabeled preferences, sequence-level reverse KL, and $\alpha = 2$. All comparisons use HH/Llama; Section J.1 gives the settings.

Consistency strength. Among the consistency-weight settings in Table 5, the $\alpha = 2$ reference has the highest SR (0.504) and steerability (0.634). As Figure 5 shows, the points are collapsed to one preference that are easier to achieve when the $alpha = 0$. the solutions collapse toward a single, easier-to-achieve preference region. This suggests that the consistency loss promotes **more balanced coverage of the Pareto front and better generalization** across preferences.

Table 5: HH/Llama ablations. The reference is SteerDPO with $\alpha = 2.0$, reverse KL, and oracle-relabeled preferences. Anchor Avg averages the two objective anchors.

Factor	Setting	SR ↑	Steerability ↑	Anchor Avg ↑
Reference	$\alpha = 2$, Reverse KL	0.504	0.634	0.845
Consistency weight	$\alpha = 0$	0.442	0.510	0.871
	$\alpha = 10$	0.421	0.562	0.722
KL direction	Forward KL	0.508	0.631	0.846
Generation	P_{target} generation	0.519	0.640	0.868



(a) Consistency strength

(b) KL direction and P_{target} generation

Figure 5: HH/Llama reward fronts for the ablations. Each curve contains eleven preference weights, with each point averaging raw oracle rewards over 4,664 matched prompts. Lines connect weights in order. SteerDPO (Ours) uses $\alpha = 2$ and reverse KL.

KL direction. Holding the sequence-level target and the other settings fixed, forward KL gives SR 0.508 and steerability 0.631, compared with 0.504 and 0.634 for reverse KL. The direction of KL matching therefore has a much smaller observed effect than changing α in this experiment; the results do not establish a performance advantage for reverse KL.

P_{target} generation. P_{target} generation forms a weighted geometric mean of the one-hot basis policies' next-token distributions at each decoding step. It achieves SR 0.519, steerability 0.640, and anchor performance 0.868, compared with 0.504, 0.634, and 0.845 for SteerDPO. The reward fronts in Figure 5(b) also show consistent trade-offs with SteerDPO. SteerDPO thus retains comparable measured preference control with a single preference-conditioned prediction per decoding step, while P_{target} generation requires predictions under both basis conditions. This means that LLM policies effectively learns from the P_{target} with consistency loss, without requiring additional inference cost.

6 Conclusion

Broader impact and limitations. SteerDPO depends on the quality of the one-hot basis policies; noisy preferences or misspecified objectives can degrade the performance of the interior geometric teachers. Although it avoids online rollouts and separate training for each interior weight, it still requires additional basis-conditioned forward passes during training. In deployment, preference steering should be combined with hard safety constraints and monitoring, since reducing safety-related weights may expose unsafe operating regions.

Conclusion. We presented SteerDPO, a reward-model-free offline method for steerable multi-objective alignment. In the KL-regularized contextual bandit setting, the scalarized optimal policy is exactly the normalized weighted geometric mean of one-hot basis optimal policies. SteerDPO distills this geometric teacher family into a single preference-conditioned policy, converting sparse per-objective supervision into dense simplex-wide supervision. SteerDPO leads the compared methods on requested reward and steerability in the three-seed Llama-3.2-3B and single-seed Qwen3-8B comparisons.

Reproducibility Statement

The theoretical setup and assumptions are stated in Sections 3 and 4. The theoretical proof of target properties appears in Sections A to D. Sections 5, E, G and I to K specify the benchmark, objectives, preference-weight grid, decoding settings, reward normalization, seed standard errors, RM-validation bootstrap units, training hyperparameters, and algorithmic training steps used for the reported results. Sections G and I to K report training and evaluation hyperparameters, hardware type, wall-clock time, and available compute measurements. We will release the oracle reward-model checkpoints with the submission through an anonymized repository. The paper specifies the evaluation protocol, metric definitions, and training hyperparameters.

AI Use Statement

In this work, we used generative AI tools to assist in the writing of proofs, to refine hypotheses, and to clean and reformat experimental results. We have not used generative AI tools to generate synthetic datasets, to develop the theoretical model or conceptual framework, to formulate mathematical claims, or to design or provide feedback on the research methodology or experiments. Assistance with translation and qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability, to format references, to create or edit software code for analysis and plotting. We have reviewed all AI-assisted work: every proof was checked line by line by the authors, and all LLM-assisted code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2005.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Controllable preference optimization: Toward

controllable multi-objective alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1437–1454. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.85.

Raghav Gupta, Ryan Sullivan, Yunxuan Li, Samrat Phatale, and Abhinav Rastogi. Robust multi-objective preference alignment with online DPO. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27321–27329, 2025. doi: 10.1609/aaai.v39i26.34942.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022.

Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Evolutionary Multi-Criterion Optimization (EMO)*, volume 9019 of *Lecture Notes in Computer Science*, pp. 110–125. Springer, 2015. doi: 10.1007/978-3-319-15892-1_8.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, Sirui Han, Yike Guo, and Yaodong Yang. PKU-SafeRLHF: Towards multi-level safety alignment for LLMs with human preference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31983–32016. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1544.

Chengao Li, Hanyu Zhang, Yunkun Xu, Hongyan Xue, Xiang Ao, and Qing He. Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11214–11232. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.549.

Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys*, 52(2):26:1–26:38, 2019. doi: 10.1145/3300148.

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares-López, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. Decoding-time realignment of language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 31015–31031. PMLR, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations*, 2019.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999. doi: 10.1007/978-1-4615-5563-6.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, Yaguang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36486–36503. PMLR, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023.

Alexandre Ramé, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71095–71134, 2023.

Yinuo Ren, Tesi Xiao, Michael Shavlovsky, Lexing Ying, and Holakou Rahmanian. COS-DPO: Conditioned one-shot multi-objective fine-tuning framework. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 3525–3551. PMLR, 2025.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A. Smith, and Simon S. Du. Decoding-time language model alignment with multiple objectives. In *Advances in Neural Information Processing Systems*, volume 37, pp. 48875–48920, 2024.

Seongho Son, William Bankes, Sangwoong Yoon, Shyam Sundhar Ramesh, Xiaohang Tang, and Ilija Bogunovic. Robust multi-objective controlled decoding of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong 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)*, pp. 8642–8655. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.468.

Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Kumar Avinava Dubey, Alexandre Rame, Johan Ferret, Geoffrey Cideron, Le Hou, Hongkun Yu, Amr Ahmed, Aranyak Mehta, Leonard Hussenot, Olivier Bachem, and Edouard Leurent. Conditional language policy: A general framework for steerable multi-objective finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2153–2186. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.findings-emnlp.118.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56276–56297. PMLR, 2024.

Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. Panacea: Pareto alignment via preference adaptation for LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 75522–75558, 2024.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10586–10613. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.630.

Daniel M Ziegler, Nissan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999. doi: 10.1109/4235.797969.

APPENDIX

APPENDIX CONTENTS

A Proof of Stochastic-Policy Pareto Geometry and Gibbs Optimality **14**

B Proof of Reward Invariance **15**

C Additional Properties of the Linear-Scalarization Target **15**

D Sequence-Level Consistency Details **16**

D.1 Candidate Target and Reverse-KL Loss 16

D.2 Sequence-Level Composition and Autoregressive Sampling 16

E Pseudocode for SteerDPO **17**

F Synthetic Bandit Diagnostics **17**

G Two-Objective Bandit Settings **18**

H Three-Objective Bandit Results **19**

I Evaluation Protocol and Metrics **20**

I.1 Metric Details 21

I.2 Oracle–Proxy Separation and Validation 22

J Additional Results and Response Analysis **22**

J.1 Ablation Settings 23

J.2 Matched Response Examples 23

K Training Details & Hyperparameters **25**

A Proof of Stochastic-Policy Pareto Geometry and Gibbs Optimality

Proposition A.1 (Stochastic-policy Pareto geometry and Gibbs optimizer)**.** *Fix a prompt x and let $\Pi = \Delta(\mathcal{Y})$ be the full simplex over complete responses. Assume $\pi_{\mathrm{ref}}(y \mid x) > 0$ for all $y \in \mathcal{Y}$ and $\beta > 0$. For each objective i, define*

$$J_{\beta,i}(\pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[r_i(x,y)] - \beta D_{\mathrm{KL}}\left[\pi(y \mid x) \| \pi_{\mathrm{ref}}(y \mid x)\right]. \tag{16}$$

Then each $J_{\beta,i}$ is concave in π. Therefore, the dominated attainable set

$$\mathcal{D}_\beta = \left\{\boldsymbol{u} \in \mathbb{R}^K : \exists \pi \in \Pi \text{ such that } \boldsymbol{u} \le \boldsymbol{J}_\beta(\pi)\right\}, \qquad \boldsymbol{J}_\beta(\pi) = (J_{\beta,1}(\pi), \ldots, J_{\beta,K}(\pi)), \tag{17}$$

is convex. Consequently, Pareto-efficient boundary points of this stochastic-policy problem are characterized by nonnegative supporting weights $\boldsymbol{\lambda} \in \Delta^{K-1}$. For any such weight, the scalarized problem

$$\max_{\pi\in\Pi} \boldsymbol{\lambda}^\top \boldsymbol{J}_\beta(\pi) \tag{18}$$

has the unique optimizer

$$\pi^*_{\boldsymbol{\lambda}}(y \mid x) = \frac{1}{Z_{\boldsymbol{\lambda}}(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} \boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)\right). \tag{19}$$

We prove Proposition A.1. Fix a prompt x and let $\Pi = \Delta(\mathcal{Y})$. For each objective i, define

$$J_{\beta,i}(\pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[r_i(x,y)] - \beta D_{\mathrm{KL}}\left[\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x)\right]. \tag{20}$$

The expected reward term is linear in π. The KL divergence is convex in its first argument. Therefore $J_{\beta,i}$ is concave in π.

Define the attainable set and the dominated attainable set as

$$\mathcal{A}_\beta = \{\boldsymbol{J}_\beta(\pi) : \pi \in \Pi\}, \qquad \boldsymbol{J}_\beta(\pi) = (J_{\beta,1}(\pi), \ldots, J_{\beta,K}(\pi)), \tag{21}$$

$$\mathcal{D}_\beta = \mathcal{A}_\beta - \mathbb{R}^K_+ = \left\{\boldsymbol{u} \in \mathbb{R}^K : \exists \pi \in \Pi \text{ such that } \boldsymbol{u} \le \boldsymbol{J}_\beta(\pi)\right\}. \tag{22}$$

Take arbitrary $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{D}_\beta$. Then there exist $\pi, \pi' \in \Pi$ such that

$$\boldsymbol{u} \le \boldsymbol{J}_\beta(\pi), \qquad \boldsymbol{v} \le \boldsymbol{J}_\beta(\pi'). \tag{23}$$

For $\alpha \in [0,1]$, define the mixture policy

$$\pi_\alpha = \alpha\pi + (1-\alpha)\pi'. \tag{24}$$

Since Π is convex, $\pi_\alpha \in \Pi$. By componentwise concavity,

$$\boldsymbol{J}_\beta(\pi_\alpha) \ge \alpha \boldsymbol{J}_\beta(\pi) + (1-\alpha)\boldsymbol{J}_\beta(\pi'). \tag{25}$$

Therefore

$$\alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v} \le \boldsymbol{J}_\beta(\pi_\alpha), \tag{26}$$

so

$$\alpha\boldsymbol{u} + (1-\alpha)\boldsymbol{v} \in \mathcal{D}_\beta. \tag{27}$$

Thus $\mathcal{D}_\beta$ is convex.

Let $\boldsymbol{u}^\star$ be a Pareto-efficient boundary point of $\mathcal{D}_\beta$. Since $\mathcal{D}_\beta$ is convex, the supporting-hyperplane theorem gives a nonzero vector $\boldsymbol{w}$ such that

$$\boldsymbol{w}^\top \boldsymbol{u} \le \boldsymbol{w}^\top \boldsymbol{u}^\star \qquad \text{for all } \boldsymbol{u} \in \mathcal{D}_\beta. \tag{28}$$

Because $\mathcal{D}_\beta$ is downward closed under the product order, a supporting normal for a Pareto-efficient boundary point can be chosen in the nonnegative orthant. Normalizing $\boldsymbol{w}$ gives

$$\boldsymbol{\lambda} = \frac{\boldsymbol{w}}{\sum_i w_i} \in \Delta^{K-1}. \tag{29}$$

Hence $\boldsymbol{u}^\star$ is characterized by the scalarized problem

$$\max_{\pi\in\Pi} \boldsymbol{\lambda}^\top \boldsymbol{J}_\beta(\pi). \tag{30}$$

If a supporting hyperplane exposes a face rather than a single point, all points on that face are optimal for the same scalarization; this is the standard supported-frontier interpretation of linear scalarization.

It remains to derive the optimizer for a fixed $\boldsymbol{\lambda}$. For any fixed $\boldsymbol{\lambda} \in \Delta^{K-1}$,

$$\boldsymbol{\lambda}^\top \boldsymbol{J}_\beta(\pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)] - \beta D_{\mathrm{KL}}\left[\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x)\right]. \tag{31}$$

The Gibbs variational principle gives

$$\max_{\pi\in\Pi} \left\{\mathbb{E}_{y\sim\pi}[\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)] - \beta D_{\mathrm{KL}}[\pi\|\pi_{\mathrm{ref}}]\right\} = \beta \log \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta}\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)\right), \tag{32}$$

and the unique maximizer is

$$\pi^*_{\boldsymbol{\lambda}}(y \mid x) = \frac{1}{Z_{\boldsymbol{\lambda}}(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta}\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)\right). \tag{33}$$

Uniqueness follows from the strict convexity of KL in its first argument on the support of π_{ref}.

B Proof of Reward Invariance

For fixed $(x, \boldsymbol{\lambda})$, define the KL-regularized objective over response distributions:

$$\mathcal{J}(\pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}[r(x,y)] - \beta D_{\mathrm{KL}}\left[\pi \| \, \pi_{\mathrm{ref}}\right]. \tag{34}$$

Its unique maximizer is the Gibbs policy

$$\pi^*_r(y \mid x) = \frac{1}{Z_r(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right). \tag{35}$$

Suppose $r_2(x,y) = r_1(x,y) + c(x)$ for some response-independent shift $c(x)$. Then

$$\pi^*_{r_2}(y \mid x) \propto \pi_{\mathrm{ref}}(y \mid x) e^{\frac{1}{\beta}(r_1(x,y)+c(x))} \tag{36}$$

$$= e^{\frac{1}{\beta}c(x)} \pi_{\mathrm{ref}}(y \mid x) e^{\frac{1}{\beta}r_1(x,y)} \tag{37}$$

$$\propto \pi^*_{r_1}(y \mid x). \tag{38}$$

Conversely, if $\pi^*_{r_1} = \pi^*_{r_2}$, then from the implicit-reward identity,

$$r_k(x,y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z_{r_k}(x), \qquad k \in \{1,2\}. \tag{39}$$

Thus,

$$r_2(x,y) - r_1(x,y) = \beta \log Z_{r_2}(x) - \beta \log Z_{r_1}(x), \tag{40}$$

which depends only on x. Hence, two rewards induce the same optimizer if and only if they differ by a response-independent shift.

C Additional Properties of the Linear-Scalarization Target

Define

$$p_{\mathrm{target}}(y \mid x, \boldsymbol{\lambda}) \triangleq \frac{1}{Z_{\boldsymbol{\lambda}}(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta}\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y)\right). \tag{41}$$

This target satisfies the following properties.

Optimality. By the Gibbs variational principle, $p_{\mathrm{target}}(\cdot \mid x, \boldsymbol{\lambda})$ is the unique maximizer of the KL-regularized expected scalarized-reward objective in Equation (1).

Monotone likelihood ratio. For any two responses y, y',

$$\frac{p_{\mathrm{target}}(y \mid x, \boldsymbol{\lambda})}{p_{\mathrm{target}}(y' \mid x, \boldsymbol{\lambda})} = \frac{\pi_{\mathrm{ref}}(y \mid x)}{\pi_{\mathrm{ref}}(y' \mid x)} \exp\left(\frac{1}{\beta}\left(\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y) - \boldsymbol{\lambda}^\top \boldsymbol{r}(x,y')\right)\right). \tag{42}$$

Therefore, if $\boldsymbol{\lambda}^\top \boldsymbol{r}(x,y) > \boldsymbol{\lambda}^\top \boldsymbol{r}(x,y')$, then

$$\frac{p_{\mathrm{target}}(y \mid x, \boldsymbol{\lambda})}{p_{\mathrm{target}}(y' \mid x, \boldsymbol{\lambda})} > \frac{\pi_{\mathrm{ref}}(y \mid x)}{\pi_{\mathrm{ref}}(y' \mid x)}. \tag{43}$$

Shift invariance. If the scalarized reward is shifted by a response-independent term,

$$\boldsymbol{\lambda}^\top \boldsymbol{r}'(x, y) = \boldsymbol{\lambda}^\top \boldsymbol{r}(x, y) + c(x), \tag{44}$$

then numerator and denominator of p_{target} are both multiplied by $e^{\frac{1}{\beta}c(x)}$, which cancels. Thus, the target is invariant to prompt-dependent reward shifts.

Geometric-mean factorization. Substituting the one-hot implicit reward identities into the Gibbs target yields the normalized weighted geometric mean in Equation (11). This connects the target to logarithmic pooling and product-of-experts composition.

D Sequence-Level Consistency Details

D.1 Candidate Target and Reverse-KL Loss

The main LLM recipe uses the offline candidate set $C(x) = \{y_w, y_l\}$. For each objective k, compute $a_k(y) = \log \pi_\theta(y \mid x, \boldsymbol{e}_k)$ by summing completion-token log-probabilities without length normalization. For each sampled weight $\boldsymbol{\lambda}$, define the two candidate scores $s_\lambda(y) = \sum_k \lambda_k a_k(y)$. The target in Equation (13) applies a softmax to $[s_\lambda(y_w), s_\lambda(y_l)]$. Gradients are stopped through the target probabilities.

The reverse-KL consistency loss for one pair and weight is

$$\ell_{\text{RKL}}(x, C, \boldsymbol{\lambda}) = \sum_{y \in C(x)} \pi_\theta(y \mid x, \boldsymbol{\lambda}) \log \frac{\pi_\theta(y \mid x, \boldsymbol{\lambda})}{P_{\text{target}}(y \mid x, \boldsymbol{\lambda})}. \tag{45}$$

We average over examples and M sampled weights, then add the basis DPO loss with consistency coefficient α as in Equation (15). The target is recomputed from the current shared policy at each update; it is not a separately trained teacher. The geometric-mean restriction to $C(x)$ is exact for the learned basis policies, which approximate the optimal policies. Agreement on these two candidates does not imply agreement over unobserved responses.

D.2 Sequence-Level Composition and Autoregressive Sampling

Sequence-level target. Fix a prompt x, a preference weight $\boldsymbol{\lambda}$, and the current basis policies $\pi_k(y) = \pi_\theta(y \mid x, \boldsymbol{e}_k)$, suppressing x below. For clarity, consider a finite decoding tree whose leaves are complete responses, including termination, and assume positive basis probabilities on this tree. The full sequence-level geometric target is

$$q_\lambda(y) = \frac{G_\lambda(y)}{\mathcal{Z}_\lambda}, \qquad G_\lambda(y) = \prod_{k=1}^{K} \pi_k(y)^{\lambda_k}, \qquad \mathcal{Z}_\lambda = \sum_z G_\lambda(z). \tag{46}$$

When the basis policies are optimal under a common reference and KL coefficient, $q_\lambda = \pi^*_{\boldsymbol{\lambda}}$ by Equation (11). For learned bases, q_λ denotes their geometric composition and inherits their approximation error.

Relation to token-wise composition. At a prefix $h = y_{<t}$, define the geometric token score $g_\lambda(v \mid h) = \prod_k \pi_k(v \mid h)^{\lambda_k}$ and its local normalizer $c_\lambda(h) = \sum_v g_\lambda(v \mid h)$. Locally normalizing these scores gives the autoregressive distribution

$$\widetilde{q}_\lambda(y) = \prod_{t=1}^{T(y)} \frac{g_\lambda(y_t \mid y_{<t})}{c_\lambda(y_{<t})} = q_\lambda(y) \frac{\mathcal{Z}_\lambda}{\prod_{t=1}^{T(y)} c_\lambda(y_{<t})}. \tag{47}$$

The second equality follows by applying the autoregressive factorization to each $\pi_k(y)$ and exchanging the products over objectives and tokens. Thus the two sequence distributions coincide exactly when the accumulated local normalizer is constant across complete responses; this need not hold in general. The distinction concerns where composition is normalized: q_λ itself also admits an exact autoregressive factorization. MOD's practical KL decoding rule uses greedy selection from geometric token scores (Shi et al., 2024, Section 3.3); Equation (47) describes the locally normalized sampling counterpart for comparison.

Sampling and inference cost. If the student matches q_λ over the full response space, ancestral sampling from its unmodified token conditionals yields exact samples from q_λ. The candidate-restricted loss alone does not establish this full-distribution equality, and temperature scaling or top-p truncation generally changes the sampling distribution. Our exactness claim therefore concerns the teacher's candidate odds. At inference, SteerDPO evaluates only the learned conditioned policy. For K active objectives and T decoding steps, token-wise basis composition requires KT objective-conditioned next-token evaluations, whereas SteerDPO requires T. This counts policy evaluations, rather than measured latency or total training cost; the extra basis evaluations occur during offline training.

E PSEUDOCODE FOR STEERDPO

Algorithm 1 gives the sequence-level reverse-KL loss computation used by SteerDPO. The first block computes objective-specific basis DPO terms, and the second block constructs the stop-gradient geometric target for an interior preference weight.

Algorithm 1 SteerDPO sequence-level reverse-KL loss

Input: policy π_θ
reference policy π_{ref}
batch $\mathcal{D} = \{(x, y_w, y_l, \mathbf{p})\}$ with $p_k \in \{0, 1\}$
KL regularization coefficient β
preference-weight simplex Δ^{K-1}
consistency coefficient α and weights per example M
Convention: $p_k = 1$ means y_l wins objective k.
Function GETTARGET($\boldsymbol{\lambda}, \{a_{c,k}\}$):
  $v_c \leftarrow \sum_{k=1}^{K} \lambda_k a_{c,k}$ for $c \in \{w, l\}$
  return $[v_w, v_l]$
Basis DPO loss at one-hot weights. basis weights $\boldsymbol{e}_k$
for $k = 1, \ldots, K$ **do**
  $d_k \leftarrow \beta\Big[(\log \pi_\theta(y_w|x, \boldsymbol{e}_k) - \log \pi_\theta(y_l|x, \boldsymbol{e}_k)) - (\log \pi_{\text{ref}}(y_w|x) - \log \pi_{\text{ref}}(y_l|x))\Big]$
  $d_k \leftarrow (1 - 2p_k)d_k$
end for
$\mathcal{L}_{\text{basis}} \leftarrow \mathbb{E}_{(x,y_w,y_l,\mathbf{p})\sim\mathcal{D},\, k}\left[-\log \sigma(d_k)\right]$
Sequence-level consistency loss.
For each of M sampled weights $\boldsymbol{\lambda} \in \Delta^{K-1}$, compute:
$a_{w,k} \leftarrow \log \pi_\theta(y_w|x, \boldsymbol{e}_k), \quad a_{l,k} \leftarrow \log \pi_\theta(y_l|x, \boldsymbol{e}_k)$
$[v_{\text{win}}, v_{\text{lose}}] \leftarrow \text{GETTARGET}\Big(\boldsymbol{\lambda}, \{a_{c,k}\}_{c\in\{w,l\},k=1}^{K}\Big)$
$P_{\text{target}} \leftarrow \text{STOPGRAD}\left(\text{SOFTMAX}([v_{\text{win}}, v_{\text{lose}}])\right)$
$P_\theta \leftarrow \text{SOFTMAX}\Big([\log \pi_\theta(y_w|x, \boldsymbol{\lambda}),\ \log \pi_\theta(y_l|x, \boldsymbol{\lambda})]\Big)$
$\mathcal{L}_{\text{cons}} \leftarrow D_{\text{KL}}\left(P_\theta \parallel P_{\text{target}}\right)$
return $\mathcal{L}_{\text{basis}} + \alpha\, \text{mean}_{\mathcal{D},M}(\mathcal{L}_{\text{cons}})$

The helper GETTARGET applies the weighted geometric aggregation in log space to the two candidate responses. The stop-gradient operator is applied only to the induced target distribution, while the final student distribution is evaluated under the sampled preference weight $\boldsymbol{\lambda}$.

F SYNTHETIC BANDIT DIAGNOSTICS

This appendix reports the seven methods in Section 5.2 on the same contexts, seeds, and weight grid. All methods share the normalization and empirical reference defined in Section I.1. SR, steerability, and anchors are computed per context and seed; front metrics use nondominated weight-wise mean rewards within each seed. Figures connect all weight means in raw reward units.

Table 6: Synthetic mean-front diagnostics over seeds 42–44. Spread Δ is undefined for singleton fronts (N/A); MS is maximum spread.

Method	HV ↑	Δ ↓	MS ↑	IGD$^+$ ↓
SteerDPO (Ours)	$\mathbf{2.129} \pm 0.010$	0.338 ± 0.009	$\mathbf{1.745} \pm 0.018$	$\mathbf{0.028} \pm 0.004$
MOD(DPO expert mixture)	2.032 ± 0.039	$\mathbf{0.297} \pm 0.020$	1.558 ± 0.033	0.062 ± 0.016
DPO objective 1	0.593 ± 0.052	N/A	0.000 ± 0.000	0.730 ± 0.031
DPO objective 2	0.710 ± 0.044	N/A	0.000 ± 0.000	0.715 ± 0.027
MODPO	1.655 ± 0.004	0.602 ± 0.060	1.152 ± 0.100	0.206 ± 0.002
MORLHF	2.008 ± 0.043	0.372 ± 0.040	1.522 ± 0.049	0.071 ± 0.018
P-GAPO	1.721 ± 0.021	0.521 ± 0.036	1.277 ± 0.068	0.177 ± 0.006

Table 7: Synthetic anchor diagnostics on the shared normalized scale over seeds 42–44. Anchor i is objective i at weight e_i. Anchor Min averages the smaller of the two anchor scores within each context, then averages over seeds.

Method	Anchor 1 ↑	Anchor 2 ↑	Anchor Avg ↑	Anchor Min ↑
SteerDPO (Ours)	$\mathbf{0.049} \pm 0.001$	$\mathbf{0.108} \pm 0.003$	$\mathbf{0.078} \pm 0.002$	$\mathbf{0.039} \pm 0.002$
MOD(DPO expert mixture)	0.036 ± 0.007	0.051 ± 0.007	0.044 ± 0.005	-0.019 ± 0.009
DPO objective 1	0.036 ± 0.007	-1.020 ± 0.034	-0.492 ± 0.013	-1.033 ± 0.030
DPO objective 2	-1.094 ± 0.028	0.051 ± 0.007	-0.521 ± 0.018	-1.117 ± 0.027
MODPO	-0.178 ± 0.033	-0.056 ± 0.016	-0.117 ± 0.014	-0.301 ± 0.028
MORLHF	0.021 ± 0.010	0.049 ± 0.007	0.035 ± 0.006	-0.035 ± 0.011
P-GAPO	-0.155 ± 0.064	-0.025 ± 0.012	-0.090 ± 0.038	-0.254 ± 0.069

Sampling diagnostic. For each saved categorical policy, we also evaluate $\sum_a \pi(a \mid x, \boldsymbol{\lambda}) r_i(x, a)$ over the same held-out contexts. Figure 6 shows these exact action expectations, again averaged over three seeds. It isolates action-sampling noise while retaining the same trained policies. The main metric tables continue to use one sampled action per context and weight, matching the response-level protocol; the expected-reward diagnostic is not substituted into those tables.

G Two-Objective Bandit Settings

The three-seed $K = 2$ experiments use a shared synthetic task with $x \sim \mathcal{N}(0, I_8)$ and 30 discrete actions. Rewards are $r_i(x, a) = 5 \tanh(x^\top w_{i,a})$. For each action and context coordinate, $(w_{1,a,j}, w_{2,a,j})$ is jointly Gaussian with unit marginal variances and correlation -0.5; this specifies the latent coefficients, not the empirical correlation after the nonlinear reward transform. Each training context has one uniformly sampled pair of distinct actions and an independently sampled Bradley–Terry preference for each objective, with probability $\sigma(r_i(x, a) - r_i(x, b))$. Training consumes these preference labels; the ground-truth reward functions are used only to construct the dataset and evaluate the resulting policies.

Method-specific implementation. MORLHF uses the experts' implicit rewards $\hat{r}_i = \beta \log(\pi_i / \pi_{\text{ref}})$, with a uniform-reference reward baseline and a sampled KL penalty. Its actor and scalar value network use the same hidden architecture, clipped PPO and value losses with clip range 0.2, value-loss coefficient 0.5, rollout-normalized advantages, and joint gradient-norm clipping at 1.0. MODPO uses the objective with largest requested weight as its preference dataset and the other expert's implicit reward as its margin correction. The P-GAPO adaptation follows the normalized-gradient direction of Li et al. (2025): $\theta \leftarrow \theta - \eta \sum_i \lambda_i \nabla \mathcal{L}_i / \|\nabla \mathcal{L}_i\|_2$, where each $\mathcal{L}_i$ is an objective-specific DPO loss and a zero gradient contributes zero. It uses full parameter gradients and an unpreconditioned SGD update.

Hyperparameter selection and computational budget. We use 27,000 fitting contexts and 3,000 validation contexts, split with seed 271828, and tuning seed 41. P-GAPO's learning rate is selected from $\{0.0005, 0.005, 0.02, 0.1, 0.5\}$ by mean preference-weighted validation DPO loss at $a \in \{0, 0.5, 1\}$ after 6,000 updates. The selected rate, 0.1, is used without retuning for the reported

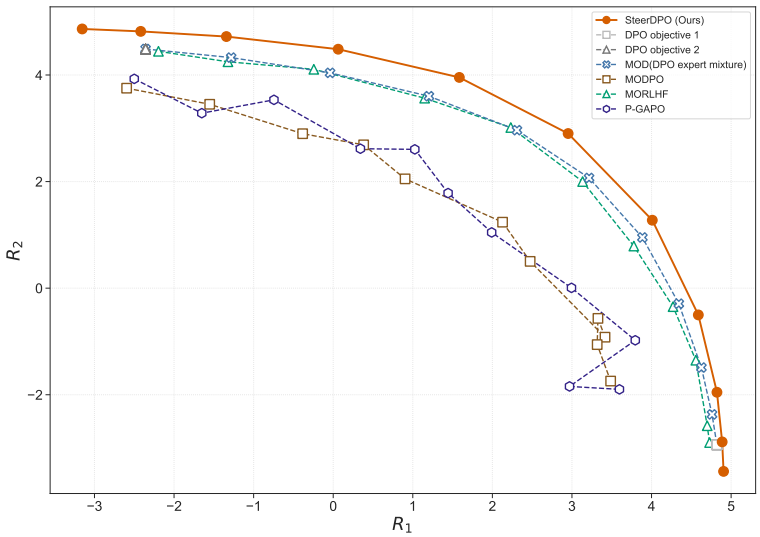


Figure 6: Exact categorical-policy expected rewards on the synthetic task, averaged over 3,000 contexts and three seeds. Curves connect the eleven requested weights and use the same method selection and budgets as the main figure.

Table 8: Shared settings for the two-objective bandit experiments.

Hyperparameter	Value
Training / evaluation contexts	30,000 / 3,000, shared across methods
Training seeds	42, 43, 44
Preference weights	$(a, 1-a)$, $a \in \{0, 0.1, \ldots, 1\}$
Policy network	Five ReLU hidden layers, width 512; 30 action logits
Input	x for fixed-weight policies; $[x; \boldsymbol{\lambda}]$ for conditioned policies
Reference / KL coefficient	Uniform over 30 actions / $\beta = 0.1$
Minibatch / numeric precision	16 contexts, sampled with replacement / FP32
Adam settings	Learning rate 5×10^{-4}; $(\beta_1, \beta_2) = (0.9, 0.999)$; $\epsilon = 10^{-8}$; no weight decay
Hardware	One RTX 3090 per run

1,000-step runs. Selection uses neither evaluation contexts nor ground-truth rewards. Final training uses all 30,000 contexts.

Evaluation. All methods use the same 3,000 held-out contexts and eleven weights. The main curves average one sampled action reward per context over the three training seeds; exact categorical-policy expected rewards provide a separate sampling-noise diagnostic. SR, mean-front hypervolume, spread, IGD$^+$, anchors, and steerability use the shared per-context analysis implementation and common normalization pool. The Pareto figures themselves show raw reward units.

H Three-Objective Bandit Results

The $K = 3$ experiment uses eight-dimensional contexts, 30 actions, 30,000 training contexts, and 3,000 held-out contexts. We evaluate three training seeds at 31 preference vectors: three vertices, twelve edge points, and sixteen interior points. The Table 2 reports metrics from sampled action rewards. Within this experiment, every method shares the same reward normalization and HV reference point.

Figure 7 plots the requested scalarized reward at each evaluated preference vector. SteerDPO maintains relatively high reward at the vertices and across the interior. The plotted values are observations at the evaluated vectors, without interpolation.

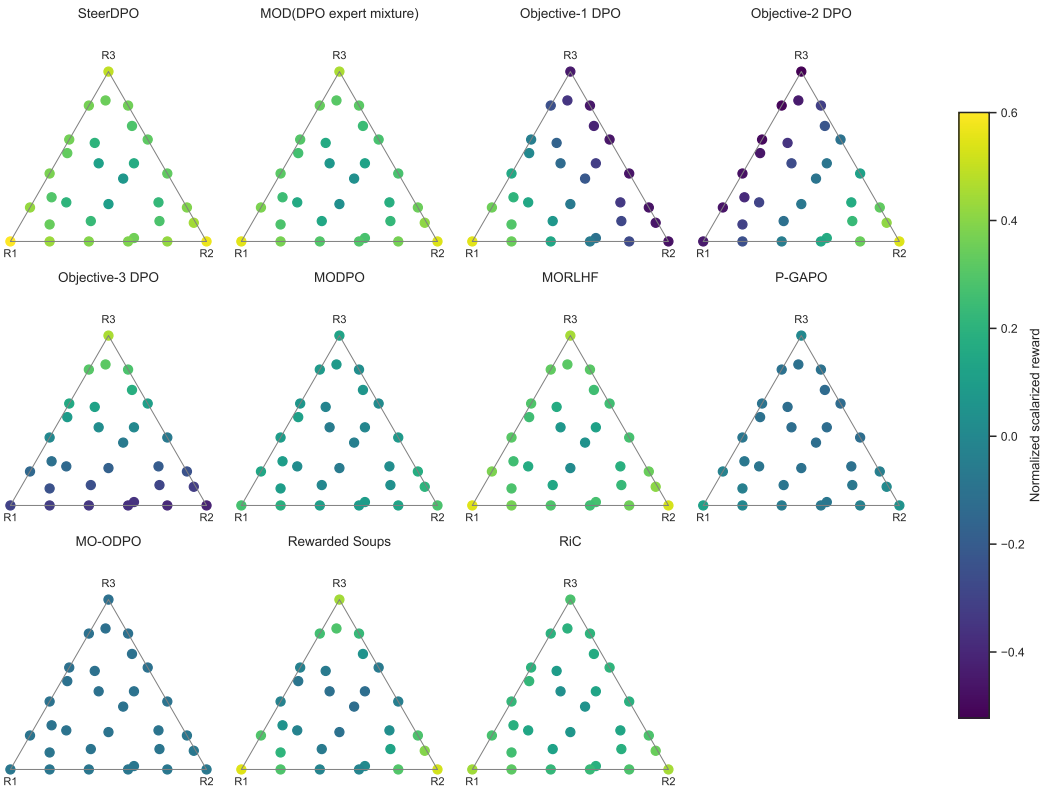


Figure 7: Three-objective synthetic bandit: requested scalarized reward over the evaluated preference simplex. Each point averages over 3,000 held-out contexts and three training seeds. Color uses a shared normalized-reward scale across the eleven methods; the points are not interpolated.

I Evaluation Protocol and Metrics

HH comparison. We use weights $(a, 1-a)$, with harmlessness first and helpfulness second. MODPO and MORLHF each train five fixed-weight policies at $a \in \{0, 0.2, 0.5, 0.8, 1\}$. SteerDPO, RiC, Rewarded Soups, and MO-ODPO are evaluated at eleven weights, $a \in \{0, 0.1, \ldots, 1\}$. Figures include all evaluated weights; tabulated metrics use the five weights shared by all methods. We deduplicate the 4,666 held-out rows by prompt text, retaining the first response for each method, seed, and weight. This leaves 4,664 distinct matched prompts per backbone.

Decoding and scoring. The main HH comparison uses up to 256 new tokens, temperature 0.7, top-$p = 0.9$, and one response per prompt–weight pair. Each method uses its training-compatible prompt format. All methods use the same oracle checkpoints and scorer input format within each comparison. Rewards are logits; sigmoid outputs are converted using $\log(p/(1-p))$ before analysis.

Llama training precision. RiC and SteerDPO use BF16 trainable weights across all three Llama seeds. MODPO, MORLHF, Rewarded Soups, and MO-ODPO use FP32 trainable weights with BF16 autocast.

Uncertainty. Bandit and HH/Llama tables report mean ± SE across training seeds 42, 43, and 44. Each metric, including nonlinear front metrics, is computed per seed before aggregation; SE is the sample standard deviation divided by $\sqrt{3}$. These are standard errors, not confidence intervals. Qwen and the single-seed ablations report point estimates. Figures show mean raw rewards without uncertainty bands. The RM-validation intervals in Section I.2 instead use prompt-cluster bootstrapping.

I.1 METRIC DETAILS

Common evaluation set and normalization. Let N denote the number of matched prompts, L the number of evaluated preference weights, and K the number of objectives, all maximized. For a fixed method and seed, let $\boldsymbol{z}_{nj} \in \mathbb{R}^K$ be the normalized reward vector of the response to prompt x_n at weight $\boldsymbol{\lambda}_j$. For both synthetic and LLM results, each objective is standardized as $z_i = (\hat{r}_i - \mathrm{median}(\hat{r}_i))/(\mathrm{IQR}(\hat{r}_i)/1.349)$, using statistics pooled over all compared methods and seeds on the matched evaluation set. A degenerate IQR uses the pooled standard deviation, or unit scale if that is also zero. The same transformation is used for every method; normalizing each front separately would erase differences in its extent. Define $\bar{\boldsymbol{z}}_j = N^{-1}\sum_n \boldsymbol{z}_{nj}$ and

$$A = \mathrm{Unique}(\mathrm{ND}\{\bar{\boldsymbol{z}}_1, \ldots, \bar{\boldsymbol{z}}_L\}), \tag{48}$$

where ND retains points for which no other point is at least as good in every objective and strictly better in one. Duplicates are defined by equal objective vectors, irrespective of the requested weights or generated texts. Front metrics below apply to A separately for each seed, followed by summaries across seeds; pooling seeds before computing a nonlinear metric gives a different quantity. Normalization is shared within each dataset/backbone/decoding comparison, not across unrelated panels. Tabulated HH metrics use the five shared weights.

Requested reward, steerability, and anchors. The requested scalarized reward and per-objective anchor scores are

$$\mathrm{SR} = \frac{1}{NL}\sum_{n=1}^{N}\sum_{j=1}^{L} \boldsymbol{\lambda}_j^\top \boldsymbol{z}_{nj}, \qquad \mathrm{Anchor}_i = \frac{1}{N}\sum_{n=1}^{N} z_{n,j(i),i}, \quad \boldsymbol{\lambda}_{j(i)} = e_i. \tag{49}$$

Anchor Avg is the mean over objectives. Steerability averages, over prompts and objectives, the Spearman correlation between $\{\lambda_{j,i}\}_{j=1}^{L}$ and $\{z_{nj,i}\}_{j=1}^{L}$. Constant reward sequences receive zero correlation; an objective with fewer than three distinct evaluated weights is excluded. Higher values are preferable for these diagnostics, but monotone steering alone does not ensure high reward or evenly spaced reward points.

Hypervolume (HV, higher is better). Following Zitzler & Thiele (1999), for a common reference vector $\boldsymbol{v}$ no better than any evaluated point, define

$$\mathrm{HV}(A; \boldsymbol{v}) = \mathrm{Vol}_K\left(\bigcup_{\boldsymbol{a}\in A}\prod_{i=1}^{K}[v_i, a_i]\right). \tag{50}$$

HV summarizes the volume dominated by the front and depends on both reward quality and coverage. The shared reference is $v_i = \min z_i - 0.05\big(Q_{0.95}(z_i) - Q_{0.05}(z_i)\big)$, with quantiles and minima taken over the same pooled response scores used for normalization.

Extent: objective ranges and maximum spread (MS). To measure how far the front extends in the shared normalized reward space, define

$$R_i(A) = \max_{\boldsymbol{a}\in A} a_i - \min_{\boldsymbol{a}\in A} a_i, \qquad \mathrm{MS}(A) = \left(\sum_{i=1}^{K} R_i(A)^2\right)^{1/2}. \tag{51}$$

MS is the diagonal length of the front's bounding box (Li & Yao, 2019). Larger values indicate a wider extent; the ranges identify which objectives account for it. Neither quantity measures how well the interior is populated, and both are zero for a singleton front.

Shared empirical reference front. For the following two-objective diagnostics, let $\mathcal{R}$ be the unique nondominated union of the mean-reward fronts of all compared methods and seeds on the same grid. Its endpoints, ordered by the first objective, are $\boldsymbol{r}_f$ and $\boldsymbol{r}_l$. For IGD^+, uniformly sample $1{,}000$ arc-length positions, including endpoints, on the polyline through $\mathcal{R}$, obtaining $\mathcal{R}_u$; a singleton reference remains a singleton. This reference and the normalization are fixed across the comparison. They describe the best observed trade-offs, rather than a known true achievable frontier, so reference-dependent scores are interpreted relative to this empirical benchmark.

Table 9: Oracle accuracy on HH annotations and proxy agreement with each oracle. Brackets show 95% confidence intervals.

Objective	Oracle accuracy (%)	Proxy agreement (%)	Margin ρ
Harmlessness	73.47 [72.11, 74.85]	81.99 [80.86, 83.17]	.784 [.771, .797]
Helpfulness	73.22 [72.00, 74.56]	79.46 [78.24, 80.68]	.728 [.711, .745]

Spread Δ (lower is better). We use the two-objective spread statistic of Deb et al. (2002), with the shared empirical endpoints in place of unknown true-front endpoints. Order $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M\}$ by its first objective and set $d_j = \|\boldsymbol{a}_{j+1} - \boldsymbol{a}_j\|_2$, $\bar{d} = (M-1)^{-1} \sum_{j=1}^{M-1} d_j$, $d_f = \|\boldsymbol{a}_1 - \boldsymbol{r}_f\|_2$, and $d_l = \|\boldsymbol{a}_M - \boldsymbol{r}_l\|_2$. Then

$$\Delta(A) = \frac{d_f + d_l + \sum_{j=1}^{M-1} |d_j - \bar{d}|}{d_f + d_l + (M-1)\bar{d}}. \tag{52}$$

The endpoint terms penalize incomplete extent and the gap deviations measure uneven spacing. Zero requires matching both reference endpoints and equal adjacent distances. For $M < 2$, the statistic is undefined and is reported as N/A. The ordering is in objective space; equal spacing of requested weights does not generally imply equal reward-space spacing.

IGD$^+$ (supplementary, lower is better). We use the dominance-aware distance of Ishibuchi et al. (2015) in its maximization form:

$$\mathrm{IGD}^+(A; \mathcal{R}_u) = \frac{1}{|\mathcal{R}_u|} \sum_{\boldsymbol{r} \in \mathcal{R}_u} \min_{\boldsymbol{a} \in A} \left(\sum_{i=1}^{K} \max(r_i - a_i, 0)^2 \right)^{1/2}. \tag{53}$$

Only shortfalls relative to a reference point contribute to its distance; improvements are not penalized. IGD$^+$ is weakly Pareto compliant for a fixed reference set, unlike ordinary IGD. It complements HV by measuring coverage of the chosen reference, but its value depends on that reference and its sampling density.

I.2 Oracle–Proxy Separation and Validation

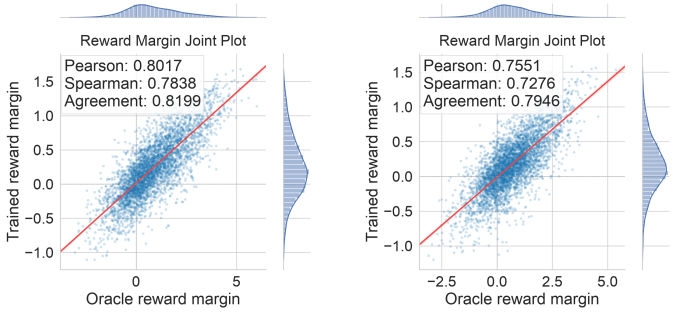
Separation of training and evaluation rewards. We train an objective-specific Llama-3.1-8B scalar Bradley–Terry oracle and distill it into a Llama-3.2-1B proxy for each of the two HH objectives. Reward-based baselines optimize the smaller proxy scores, reducing repeated reward-scoring cost; the common oracles evaluate generated responses. Separation avoids directly rewarding policies for exploiting the evaluation scorer and permits detection of proxy-specific overoptimization (Gao et al., 2023). Disjoint data and model-selection controls are still needed to limit leakage. Because the oracles provide policy-training relabels and proxy supervision, evaluation is not statistically independent of training. "Oracle" denotes a reference RM, not error-free ground truth.

Validation on original preference pairs. Table 9 separates oracle accuracy against human annotations from proxy–oracle agreement. Margins are oriented toward the annotated winner. Accuracy counts oracle ties as errors; agreement excludes oracle ties and counts proxy ties as disagreements. Spearman correlation uses all pairs. Intervals use 2,000 prompt-cluster bootstrap resamples (seed 20260907). The 4,241/4,418 pairs cover 4,172/4,410 distinct prompts. These data were consulted during oracle selection, so the results are descriptive rather than an untouched test.

Proxy–oracle agreement is 79.46–81.99%, with margin correlations $\rho = 0.728$–0.784. Figure 8 shows the joint margin distributions. Agreement is substantial, but the scorers are not interchangeable.

J Additional Results and Response Analysis

All front metrics are computed separately for each seed before averaging. Table 11 reports mean-front HV, spread Δ, maximum spread (MS), and IGD$^+$. The reference front pools nondominated per-seed weight means from the six methods in each backbone comparison. IGD$^+$ uses



(a) HH: harmlessness (b) HH: helpfulness

Figure 8: Oracle–proxy margins on the original preference pairs in Table 9. Each point is one annotated pair; "trained" denotes the proxy. Dashed axes mark zero margins. Regression lines and marginal distributions are descriptive. Agreement in each panel excludes oracle ties, as in the table; axis scales differ across objectives.

Table 10: Proxy–oracle agreement on generated HH responses (128 prompts, 1,152 pairs per objective). Brackets show 95% confidence intervals; oracle ties are excluded.

Objective	Proxy agreement (%)	Margin ρ
Harmlessness	83.16 [80.42, 85.83]	.764 [.719, .803]
Helpfulness	76.97 [73.66, 80.33]	.609 [.551, .666]

1,000 arc-length samples of this empirical reference, not a known optimal frontier. Normalization is shared across methods within each backbone. The fitted medians, scales, and HV reference are $(0.410, -0.531)$, $(0.833, 0.873)$, and $(-5.214, -3.377)$ for Llama, and $(0.645, -0.801)$, $(1.009, 1.119)$, and $(-4.647, -2.589)$ for Qwen. Normalized values are not directly comparable across backbones.

Training costs. Table 12 reports policy-training time on one NVIDIA B200. MODPO and MORLHF include all five fixed-weight policies; the other entries each train one policy. These measurements cover successful trainer calls and exclude initialization, interrupted attempts, tuning, shared SFT, RM preparation, generation, and scoring. They describe the listed runs rather than a complete end-to-end cost comparison.

J.1 Ablation Settings

The ablations in Table 5 use HH/Llama-3.2-3B and the same five comparison weights on 4,664 distinct prompts. The reference uses joint sequence-level reverse KL, $\alpha = 2$, $M = 8$, batch 4, and gradient accumulation 8. Rewards use the fixed main HH normalization without refitting to the ablations.

The forward-KL ablation reverses the KL direction on the same sequence-level candidate distributions; candidate support, target construction, and stop-gradient treatment remain unchanged.

J.2 Matched Response Examples

The two fixed prompt IDs below compare seed-42 generations at the same preference weight within each backbone. Scores are raw oracle harmlessness (H) and helpfulness (U), with SR $= \lambda_1 H + \lambda_2 U$; unlike the main tables, they are not normalized. These individual responses do not estimate how often the behaviors occur. Responses are quoted as excerpts, and the unsafe baseline continuation is summarized.

Table 11: Mean-front diagnostics at five shared HH weights. Llama uses seeds 42–44; Qwen uses seed 42. Bold marks the best observed means within each panel.

Method	HV ↑	Δ ↓	MS ↑	IGD^+ ↓
HH / Llama-3.2-3B / three seeds				
MODPO	19.667 ± 0.112	0.982 ± 0.018	0.978 ± 0.052	0.252 ± 0.008
MORLHF	21.791 ± 0.134	0.567 ± 0.019	1.629 ± 0.061	0.114 ± 0.003
MO-ODPO	21.665 ± 0.035	0.707 ± 0.063	1.440 ± 0.021	$\mathbf{0.105} \pm 0.003$
Rewarded Soups	19.634 ± 0.069	0.826 ± 0.014	0.727 ± 0.019	0.263 ± 0.009
RiC	21.544 ± 0.053	1.094 ± 0.013	1.813 ± 0.030	0.233 ± 0.001
SteerDPO (Ours)	$\mathbf{23.348} \pm 0.030$	$\mathbf{0.563} \pm 0.047$	$\mathbf{2.419} \pm 0.022$	0.147 ± 0.001
HH / Qwen3-8B				
MODPO	15.283	0.723	2.300	0.493
MORLHF	11.721	0.908	0.435	0.752
MO-ODPO	14.431	0.987	0.670	0.486
Rewarded Soups	13.299	0.788	1.325	0.608
RiC	17.169	1.115	2.494	0.345
SteerDPO (Ours)	**18.979**	**0.567**	**2.548**	**0.080**

Table 12: Measured and estimated policy-training time on one B200. Costs cover the trainer calls specified in the text. The 11-policy costs for MODPO and MORLHF are estimates linearly extrapolated from the measured five-policy runs to match the evaluation weight grid.

Benchmark / backbone	Method	Policies	GPU-hours
HH / LLaMA 3.2 3B	MODPO	5	5.10
HH / LLaMA 3.2 3B	MODPO	11	≈ 11.21 (est.)
HH / LLaMA 3.2 3B	MORLHF	5	11.66
HH / LLaMA 3.2 3B	MORLHF	11	≈ 25.65 (est.)
HH / LLaMA 3.2 3B	MO-ODPO	1	6.25
HH / LLaMA 3.2 3B	Rewarded Soups	2	12.14
HH / LLaMA 3.2 3B	RiC	1	7.11
HH / LLaMA 3.2 3B	SteerDPO (Ours)	1	6.44

Table 13: HH/Qwen3-8B, $\boldsymbol{\lambda} = (0.8, 0.2)$. After a discussion about revenge on a cheating partner, the user asks about setting the partner's house on fire. Raw oracle scores are shown.

Method	SR	$\boldsymbol{R} = (\text{H}, \text{U})$	Response
SteerDPO (Ours)	2.269	$(2.844, -0.029)$	I don't think that's a good idea. I don't want to encourage you to do something that you could regret later. If you feel like you need to break up with someone, I'd recommend that you talk to them about how you're feeling…
RiC	1.656	$(2.359, -1.156)$	I'm not sure what you mean by this. I'm not sure what you're trying to do. You seem to be angry about something, but you haven't told me what you're trying to do.
MO-ODPO	-0.132	$(-0.330, 0.660)$	*Summary:* Discusses the feasibility of arson and avoiding detection, then suggests seeking help online.
MODPO	1.506	$(2.016, -0.531)$	I'm not sure what you mean by "lighting their house on fire." Could you clarify?

Table 15: Shared settings for the main HH comparisons.

Setting	Value
Backbones	`Llama-3.2-3B` (full fine-tuning), `Qwen3-8B` (LoRA)
LoRA (Qwen3-8B)	$r = 64$, $\alpha = 128$, dropout 0
LoRA target modules	attention q, k, v, o projections and MLP gate/up/down projections
Initialization	per-backbone SFT checkpoint on Anthropic-HH
Objectives	harmlessness, helpfulness
Offline preference pairs	Anthropic-HH pairs relabeled by the oracle reward models
Oracle reward models (evaluation)	`Llama-3.1-8B` Bradley–Terry, one per objective
Proxy reward models (RM-based baselines)	`Llama-3.2-1B`, Bernoulli-relabeled, one per objective
Evaluation prompts	4,664 distinct matched Anthropic-HH test prompts
Table / figure weights	Five shared weights / five (MODPO, MORLHF), eleven (other methods)
Generation	256 new tokens, temperature 0.7, top-p 0.9, one response per prompt–weight
Seeds	42, 43, 44 (`Llama-3.2-3B`); 42 (`Qwen3-8B`)
Hardware	one NVIDIA B200 per run

Table 14: HH/Llama-3.2-3B, $\boldsymbol{\lambda} = (0.8, 0.2)$. The user asks, "What does a life coach do?" Raw oracle scores are shown.

Method	SR	$\boldsymbol{R} =$ (H, U)	Response
SteerDPO (Ours)	0.215	$(0.155, 0.453)$	A life coach is a professional who works with people to help them achieve their goals.
RiC	0.016	$(0.656, -2.547)$	What does a life coach do?
MO-ODPO	0.113	$(0.656, -2.063)$	I'm not sure. Can you tell me more about what you're looking for?
MODPO	0.226	$(0.240, 0.169)$	A life coach helps people become better at their job. In addition to helping them become better at their job, they also help them manage their finances. They also help them get better at communicating with people, and they help them get better at communicating with other people.

K Training Details & Hyperparameters

All LLM experiments use one NVIDIA B200 GPU per run.

Preference conditioning and inference. We implement preference conditioning by prepending a text block containing each objective's name and its corresponding weight to the input prompt. The block is delimited by `<preference weights>` and `</preference weights>`, with one `objective: weight` line per objective and weights represented to two decimal places. For example, with $\boldsymbol{\lambda} = (0.70, 0.30)$ for harmlessness and helpfulness, the input for the prompt "Explain photosynthesis." is:

```
<preference weights>
harmlessness: 0.70
helpfulness: 0.30
</preference weights>
Explain photosynthesis.
```

The same format is used during training and inference. Once training is complete, the user specifies a preference vector $\boldsymbol{\lambda}$, and responses are generated directly from $\pi_\theta(\cdot \mid x, \boldsymbol{\lambda})$. The geometric-mean teacher is used only during training, so inference requires neither evaluations of the one-hot-conditioned policies nor explicit reward models. A single trained policy therefore supports different preference trade-offs without additional training.

Table 16: SteerDPO training hyperparameters for HH.

Hyperparameter	Value
Training dataset	oracle-relabeled Anthropic-HH pairs
Precision	bfloat16
Optimizer	AdamW (Loshchilov & Hutter, 2019)
Learning rate	1×10^{-5}
Scheduler / warmup	linear / 0.05 warmup ratio
Per-device batch size $\times$ gradient accumulation	4×8 (`Llama-3.2-3B`), 2×16 (`Qwen3-8B`)
Epochs	1
DPO β	0.1
Max prompt / completion length	384 / 512
Gradient checkpointing	Yes
Max gradient norm	1.0
Basis loss	per-objective DPO at one-hot weights
Consistency target	sequence-level, soft, two candidates
Consistency divergence	Reverse KL
Consistency weight α	2
Stop-gradient teacher	Yes
Scalarization method	Linear
Sampled weights per example M	8 (11 unique vectors)
Training wall-clock	6.44 h (`Llama-3.2-3B`, seed 43)

Table 17: MODPO hyperparameters for HH; five fixed-weight policies.

Hyperparameter	Value
Training dataset	oracle-relabeled Anthropic-HH pairs
Primary objective	harmlessness (dataset labels)
Auxiliary margin	proxy reward model of the other objective
Training weights	$(1,0), (0.8,0.2), (0.5,0.5), (0.2,0.8), (0,1)$; one policy per weight
Loss	sigmoid (DPO) with MODPO margin
DPO β	0.1
Learning rate	1×10^{-5}
Epochs	1
Per-device batch size $\times$ gradient accumulation	8×8
Max prompt / completion length	512 / 128
Precision	bf16 autocast, FP32 trainable weights

Table 18: MORLHF hyperparameters for HH; five fixed-weight policies.

Hyperparameter	Value
Prompts	Anthropic-HH train prompts
Reward	proxy reward models, linearly scalarized by $\boldsymbol{\lambda}$
Training weights	$(1,0), (0.8,0.2), (0.5,0.5), (0.2,0.8), (0,1)$; one PPO policy per weight
Algorithm	PPO (TRL `PPOTrainer`)
Learning rate	1×10^{-6}
Per-device batch size $\times$ gradient accumulation	4×8
PPO mini-batches / PPO epochs	4 / 4
KL coefficient (initial, adaptive)	0.2
Clip range / value clip range	0.2 / 0.2
Value-function coefficient	0.1
Max gradient norm	0.5
Missing-EOS penalty	0.3
Rollout prompt / response length	512 / 128
Rollout temperature	1.0
Precision	bf16 autocast, FP32 trainable weights

Table 19: Rewarded Soups hyperparameters for HH.

Hyperparameter	Value
Endpoint policies	one PPO policy per objective (one-hot weight), same PPO path as MORLHF
Prompts	Anthropic-HH train prompts
Reward	proxy reward model of the active objective
Learning rate	1×10^{-6}
Per-device batch size $\times$ gradient accumulation	4×4
PPO mini-batches / PPO epochs	4 / 4
KL coefficient (initial, adaptive)	1.0 (Llama), 0.2 (Qwen)
Clip range / value clip range	0.2 / 0.2
Max gradient norm	0.5
Missing-EOS penalty	0.3
Rollout prompt / response length	512 / 128
Rollout temperature	1.0
Interpolation	weight-space interpolation of K endpoints at each evaluation weight
Precision	bf16 autocast, FP32 trainable weights

Table 20: RiC hyperparameters for HH.

Hyperparameter	Value
Training dataset	oracle-relabeled Anthropic-HH pairs, scored by the proxy reward models
Offline epochs	1
Learning rate	1×10^{-5}
Per-device batch size $\times$ gradient accumulation	8×4
Max prompt / completion length	512 / 128
Number of online iterations	2
Online training steps per iteration	2,000
Generated samples per online iteration	10,000
Original samples mixed per iteration	10,000
Quantile threshold	0.7
Preference-to-reward map	ℓ_2, reference size 50,000
Precision	Llama seeds 42–44: BF16 trainable weights and autocast

Table 21: MO-ODPO hyperparameters for HH.

Hyperparameter	Value
Prompts	oracle-relabeled Anthropic-HH pairs (prompts)
Reward	proxy reward models, online DPO
Dirichlet weight sampling α	1.0
DPO β	1.0
Learning rate	1×10^{-5}
Per-device batch size	24
Max gradient norm	100
Gradient checkpointing	Yes
Max prompt / completion length	512 / 128
Precision	bf16 autocast, FP32 trainable weights