RAY-GRPO: MULTI-OBJECTIVE ALIGNMENT THROUGH RAY-GUIDED REWARD WEIGHTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-objective Group Relative Policy Optimization (GRPO) combines competing rewards into one advantage. Fixed weights express preferences but do not adjust to current reward attainment. We introduce RAY-GRPO, which derives reward coefficients from both the input preference and attained rewards. Balanced allocation associates each preference control with a distribution over reference rays specifying reward proportions in calibrated space. Each ray's utility measures joint attainment through the smallest reward-to-proportion ratio. Averaging these utilities evaluates multiple proportional requirements; differentiating at current calibrated reward means yields coefficients that emphasize the objectives limiting attainment. These coefficients weight group-relative reward contrasts to form an adaptive scalar GRPO advantage. Their construction reuses scored rollouts without additional generations or reward-model calls. We train one conditional policy by alternating single-objective endpoint updates with residual refinement that preserves current endpoints. Experiments on alignment tasks with two, three, and five objectives show gains in mean reward scores, with task-dependent effects on empirical Pareto-front coverage.

1 INTRODUCTION

Group Relative Policy Optimization (GRPO) compares responses to the same prompt without a learned value critic (Shao et al., 2024). Multi-objective alignment adds conflicting goals such as helpfulness, harmlessness, and humor (Zhou et al., 2024; Zhong et al., 2024). A preference-conditioned policy represents their trade-offs through an input control (Yang et al., 2024b; Zhong et al., 2024), which training must translate into a scalar group-relative advantage. Fixed scalarization suits preferences expressed as constant exchange rates, but its coefficients do not respond to current attainment. Our aim is to emphasize rewards that limit a desired balance. Dynamic reward weighting adapts coefficients during training (Lu et al., 2026); **RAY-GRPO** derives them from an explicit attainment utility.

A control specifies an angular prior over reference *rays*, which define proportions in calibrated reward space. Radial utility measures joint attainment through the smallest reward-to-proportion ratio (Zhang & Golovin, 2020). If humor has the smallest ratio along a helpfulness–humor ray, increasing helpfulness alone leaves that ray's attainment unchanged. Balanced entropic transport then allocates rays by angular alignment, with equal mass per control and equal total mass per ray. Averaging allocated utilities provides *ray-based preference smoothing* while retaining each ray's exact bottleneck. Differentiating this utility at current calibrated reward means yields coefficients that weight objective-wise group-relative contrasts into a scalar GRPO advantage. This construction reuses rollout scores without additional generations, reward-model evaluations, or objective-wise policy-gradient estimates.

The conditional policy must also represent specialists and intermediate trade-offs. We separate potentially interfering updates (Yu et al., 2020) by alternating two phases in *boundary/interior training* (Figure 1). Boundary updates train specialist blocks at simplex vertices; interior refinement trains a control-conditioned residual at nonvertex controls, including those on simplex faces. Freezing specialists and gating the residual to zero at vertices preserves current endpoints during refinement (Section 4.3).

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.

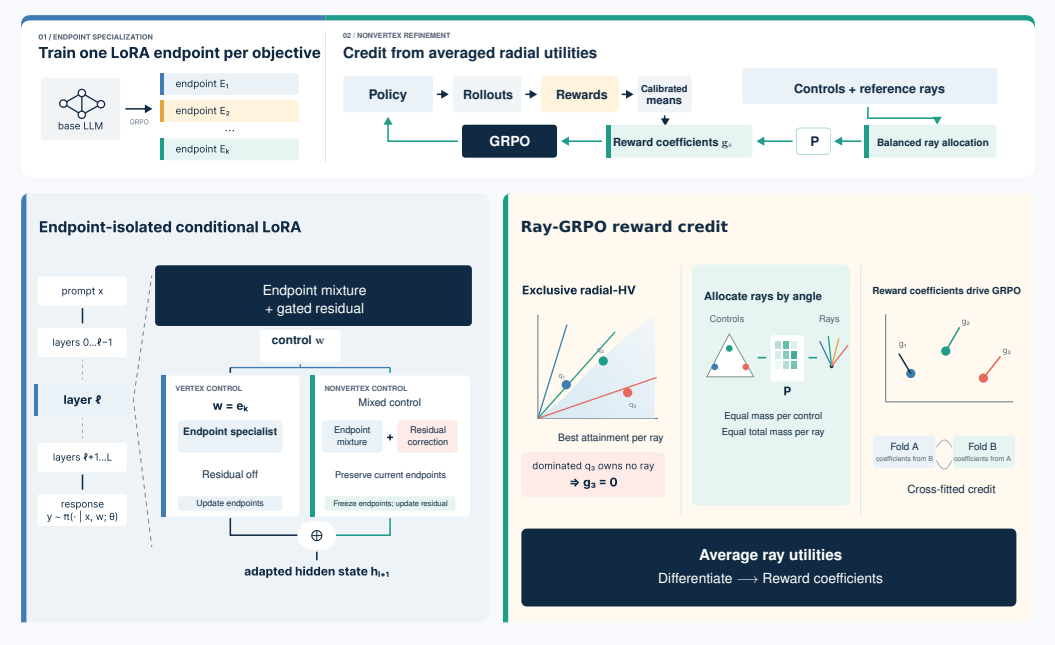

Figure 1: **Overview of RAY-GRPO.** Balanced ray allocation defines an averaged utility at mean calibrated rewards. Its derivatives, estimated from disjoint prompts, weight group-relative contrasts into a GRPO advantage. Training alternates endpoint specialization with nonvertex residual refinement, which preserves current endpoint policies (Section 4.3).

Our contributions are:

- **Attainment-based reward credit for GRPO.** Balanced ray allocation and radial reward evaluation yield adaptive objective coefficients. These coefficients turn current reward attainment into a scalar group-relative advantage using existing rollout scores.
- **Conditional training that preserves specialists.** We alternate endpoint updates with refinement of a control-conditioned residual that vanishes at vertices. Freezing endpoint blocks during refinement preserves specialists while refining intermediate compromises.
- **Comprehensive evaluation with diverse alignment baselines.** We compare baselines spanning reward weighting, model composition, preference conditioning, and reward-guided decoding. Experiments with two, three, and five objectives assess reward quality and frontier coverage, complemented by ablations and additional backbone studies.

2 RELATED WORK

Multi-reward optimization and GRPO. Fixed weighted sums use prescribed reward coefficients. Within GRPO, Multi-Objective GRPO (MO-GRPO) and Group Reward-Decoupled Normalization Policy Optimization (GDPO) normalize individual rewards before aggregation to address imbalanced or lost reward contributions (Ichihara et al., 2025; Liu et al., 2026). Normalization changes effective weights with observed reward variability. Dynamic approaches include hypervolume-based reward modulation and gradient-based objective-weight adaptation (Lu et al., 2026). HaM maximizes the hypervolume of a set of policies (Mukherjee et al., 2024), whereas HVO constructs hypervolume-inspired scalar rewards for individual responses within GRPO (Song et al., 2026). RAY-GRPO derives objective coefficients by differentiating a preference-smoothed attainment utility at current calibrated reward means.

Reference directions and scalarization. Pareto Multi-Task Learning (Pareto MTL) defines reference-vector subproblems and derives adaptive task weights (Lin et al., 2019), while Exact

Pareto Optimal (EPO) Search uses preferences to guide combinations of objective-wise parameter gradients (Mahapatra & Rajan, 2020). These methods use objective-wise gradients to determine update directions; our coefficient construction operates in reward space without separate objective-wise policy-gradient estimates. Random hypervolume scalarizations express set-level dominated volume through an expectation over directions of the maximum scalarized attainment across points (Zhang & Golovin, 2020), providing the basis for our radial utilities. Smooth Tchebycheff smooths the objective-wise maximum and yields coefficients that depend on current objective values (Lin et al., 2024). Hierarchical Mixture-of-Experts (HoE) also uses Tchebycheff scalarization and adaptive combinations of objective advantages within PPO (Li et al., 2026). Our construction averages exact radial utilities over a balanced allocation of directions to controls, retaining each direction's exact bottleneck criterion (Appendix A.5).

Controllable language model alignment. Multi-Objective Direct Preference Optimization (MODPO) optimizes policies for specified objective weights (Zhou et al., 2024). Rewards-in-Context conditions generation on reward targets (Yang et al., 2024b), while Panacea uses preference-conditioned low-rank adaptation to represent different trade-offs within one model (Zhong et al., 2024). Model-composition approaches include Rewarded Soups, which interpolates reward-specialized model parameters (Ramé et al., 2023); HoE, which combines experts through hierarchical routing (Li et al., 2026); and Evolutionary Soups, which evolves routing networks for input-dependent expert composition (Kong et al., 2026). Alignment as Reward-Guided Search (ARGS) uses reward feedback during decoding (Khanov et al., 2024), while MetaAligner applies a trained conditional refiner to existing responses (Yang et al., 2024a). Our conditional policy combines single-objective endpoint blocks with a control-conditioned residual; boundary/interior training refines intermediate trade-offs while preserving current endpoint policies during refinement.

3 Problem Setting

Rewards and preference controls. We consider $K \geq 2$ rewards $\mathbf{r}(x, y) \in \mathbb{R}^K$, all to be maximized, for a response y to a prompt x. A conditional policy $\pi(y \mid x, \mathbf{w}; \theta)$ represents different compromises through a control $\mathbf{w} \in \Delta^{K-1}$, where $\Delta^{K-1} = \{\mathbf{w} \in \mathbb{R}^K_+ : \sum_k w_k = 1\}$. The control can vary or remain fixed at deployment. Let $W = (\mathbf{w}_s)_{s=1}^S$ be the ordered nonvertex controls used in a training batch, and let $\boldsymbol{\mu}_s(\theta) = \mathbb{E}[\mathbf{r}(x, y) \mid \mathbf{w}_s]$. The expectation covers task prompts and responses from $\pi(\cdot \mid x, \mathbf{w}_s; \theta)$.

Fixed reward coordinates. For each objective, fix a reference $r_{\mathrm{ref},k}$ and scale $D_k > 0$, and define $z_k(x, y) = (r_k(x, y) - r_{\mathrm{ref},k})/D_k$. Write $\mathbf{z} = (z_1, \ldots, z_K)$ and $\mathbf{q}_s = (q_{s1}, \ldots, q_{sK})$, where

$$q_{sk}(\theta) = \mathbb{E}[z_k(x, y) \mid \mathbf{w}_s] = \frac{\mu_{sk}(\theta) - r_{\mathrm{ref},k}}{D_k}. \tag{1}$$

Collect these means in $Q(\theta) = (\mathbf{q}_s(\theta))_{s=1}^S$. Radial utility requires each $\mathbf{q}_s$ to lie in the nonnegative orthant $\mathbb{R}^K_+$. Training fixes D_k from reference-policy reward quantiles and specifies $r_{\mathrm{ref},k}$ separately for each task. These constants remain unchanged during optimization; Appendix B specifies calibration and validity checks for estimated means.

Group-relative reward credit. For each prompt and control, GRPO compares sampled responses using one scalar advantage (Shao et al., 2024). A fixed weighted sum of rewards assigns the same coefficients throughout training. Our goal is to derive coefficients from the intended compromise and the current reward means, while retaining a single group-relative policy update. Section 4.1 defines how the control specifies the compromise in the calibrated reward space.

4 Ray-Guided GRPO

RAY-GRPO constructs a ray-averaged utility, derives reward credit, and alternates boundary/interior updates. Algorithm 1 in Appendix B summarizes the complete training step.

4.1 RAY-BASED PREFERENCE SMOOTHING

From reward proportions to attainment. A strictly positive unit ray $\mathbf{u} \in \mathbb{R}^K_{>0}$, with $\|\mathbf{u}\|_2 = 1$, specifies relative reward proportions. For a nonnegative reward vector $\mathbf{q}$, define

$$\rho_{\mathbf{u}}(\mathbf{q}) = \min_k q_k/u_k, \qquad \phi_{\mathbf{u}}(\mathbf{q}) = c_K \rho_{\mathbf{u}}(\mathbf{q})^K, \tag{2}$$

where $c_K = \pi^{K/2}/[2^K \Gamma(K/2+1)]$. The radius $\rho_{\mathbf{u}}(\mathbf{q})$ is the largest $a \geq 0$ for which $a\mathbf{u} \leq \mathbf{q}$ coordinate-wise. Thus, the smallest reward-to-proportion ratio limits attainment along the ray. The power and constant follow the radial representation of hypervolume (Zhang & Golovin, 2020); Appendix A gives the derivation.

A preference control specifies a distribution of proportional requirements. Averaging their radial utilities retains sensitivity to limiting rewards; averaging linear utilities would produce another fixed linear weight vector.

Allocating rays to controls. We need each control to emphasize nearby reward proportions while collectively accounting for all sampled directions. First, we build a shared bank $U = \{\mathbf{u}_m\}_{m=1}^M$ for angular integration. Sobol's low-discrepancy design (Sobol', 1967) spreads integration points across the unit cube; scrambling randomizes this balanced design (Owen, 1995). The half-normal inverse CDF followed by ℓ_2 normalization targets uniform surface measure on the positive sphere.

Next, we normalize each control to $\bar{\mathbf{w}}_s = \mathbf{w}_s/\|\mathbf{w}_s\|_2$ so that $\|\bar{\mathbf{w}}_s - \mathbf{u}_m\|_2^2$ measures directional mismatch. We use this cost in balanced entropic transport: row mass $1/S$ weights controls equally, and column mass $1/M$ gives every ray equal aggregate weight. Entropy at temperature $\tau > 0$ spreads assignments across rays for directional averaging. Sinkhorn scaling (Cuturi, 2013) enforces these marginals, yielding a joint plan P and conditional ray probabilities $p_{sm} = SP_{sm}$:

$$\begin{aligned} p_{sm} &= \frac{b_m \exp(-\|\bar{\mathbf{w}}_s - \mathbf{u}_m\|_2^2/\tau)}{\sum_{\ell=1}^M b_\ell \exp(-\|\bar{\mathbf{w}}_s - \mathbf{u}_\ell\|_2^2/\tau)}, \\ \sum_m p_{sm} &= 1, \qquad \sum_s p_{sm} = S/M. \end{aligned} \tag{3}$$

The positive factors b_m adjust local angular preferences to satisfy global column balance (Appendix A.4). Thus, p_{sm} specifies which proportional requirements control s evaluates. Balanced ray mass does not guarantee coverage of the attained reward set.

Averaging radial utilities. The control utility and total utility are

$$\begin{aligned} j_s(\mathbf{q}) &= \sum_m P_{sm} \phi_{\mathbf{u}_m}(\mathbf{q}) = \frac{1}{S} \sum_m p_{sm} \phi_{\mathbf{u}_m}(\mathbf{q}), \\ J(Q;P) &= \sum_s j_s(\mathbf{q}_s). \end{aligned} \tag{4}$$

This *ray-based preference smoothing* evaluates each control across reward proportions. It smooths over directions; each ray retains its exact, potentially nondifferentiable minimum over objectives. Here M controls angular resolution and τ controls allocation spread subject to balancing (Figure 4 in Appendix A.4). The control specifies an angular prior, not fixed linear exchange rates or a guarantee of monotone raw-reward response.

4.2 POLICY-DEPENDENT REWARD REWEIGHTING

The utility evaluates each control's *mean reward vector*, $j_s(\mathbb{E}[\mathbf{z} \mid \mathbf{w}_s])$, rather than the mean of a nonlinear response-level utility. Its derivatives describe how changes in these means affect attainment. For fixed W, U, τ, the plan P is independent of Q. Away from bottleneck ties, define $\mathbf{g}_s(\mathbf{q}) = \nabla_{\mathbf{q}} j_s(\mathbf{q})$ and write g_{sk} for its kth component evaluated at $\mathbf{q}_s(\theta)$. The chain rule gives

$$\nabla_\theta J(Q(\theta);P) = \sum_{s,k} g_{sk} \nabla_\theta q_{sk}(\theta). \tag{5}$$

Thus, weighting centered rewards by g_{sk} gives a scalar group-relative learning signal for control s.

Differentiating Eq. (2) gives

$$g_{sk} = c_K \sum_m P_{sm} \frac{K \rho_{\mathbf{u}_m}(\mathbf{q}_s)^{K-1}}{u_{mk}} \alpha_{smk}. \tag{6}$$

Here $\boldsymbol{\alpha}_{sm} \in \Delta^{K-1}$ is supported on $\arg\min_k q_{sk}/u_{mk}$; the implementation selects the first minimizing coordinate at a tie. Each ray credits its limiting reward; their average varies with the policy's reward means. Define relative weights $\boldsymbol{\lambda}_s(\mathbf{q}) = \mathbf{g}_s(\mathbf{q})/\|\mathbf{g}_s(\mathbf{q})\|_1$ when the denominator is positive. Training uses the full $\mathbf{g}_s$, retaining its magnitude as well as these relative weights.

4.3 Boundary/interior policy training

Compromise updates can move shared parameters away from specialist solutions, reflecting interference between objectives (Yu et al., 2020). We therefore give specialists dedicated endpoint blocks and learn intermediate trade-offs through a separate residual. Inspired by neural trial functions that preserve specified values through a vanishing correction (Lagaris et al., 1998), we gate this residual to zero at vertices. Freezing endpoint blocks during refinement then protects current specialists while allowing compromises to improve. For each frozen backbone map ℓ, the adapter correction is

$$\begin{aligned} \Delta_\ell(\mathbf{w}) &= \sum_k w_k E_{\ell k} + b_K(\mathbf{w}) R_\ell(\mathbf{w}), \\ b_K(\mathbf{w}) &= \frac{K}{K-1}(1 - \|\mathbf{w}\|_2^2). \end{aligned} \tag{7}$$

Here $E_{\ell k}$ specializes endpoint k, and $R_\ell(\mathbf{w})$ refines their interpolation. Both use low-rank adaptation (LoRA) (Hu et al., 2022); Appendix B gives parameterization and initialization.

We alternate two phases. *Boundary steps* update only the endpoint blocks $E_k = \{E_{\ell k}\}_\ell$ and assign single-objective reward credit at the vertices e_k. *Interior steps* update only the residual $R = \{R_\ell\}_\ell$ and assign ray-derived credit at nonvertex controls. The backbone remains frozen, and inactive blocks receive neither optimizer updates nor weight decay. Since $b_K(e_k) = 0$, $\Delta_\ell(e_k) = E_{\ell k}$: interior steps preserve the current endpoint policies exactly. Later boundary steps can update them. Here "interior" includes nonvertex controls on simplex faces; the preservation guarantee applies to vertices, not entire faces.

4.4 Advantage estimation

Rollout groups and reward contrasts. The prompt index b below is global across workers. Each global prompt batch $\mathcal{B}$ uses the same ordered control list $\widetilde{\mathbf{w}}_c$, with $\widetilde{\mathbf{w}}_k = e_k$ for $k = 1, \ldots, K$ and $\widetilde{\mathbf{w}}_{K+s} = \mathbf{w}_s$ for $s = 1, \ldots, S$. Both phases generate and score $n \geq 2$ independent responses $y_{bci} \sim \pi(\cdot \mid x_b, \widetilde{\mathbf{w}}_c; \theta_t)$ per prompt and control, where θ_t denotes the rollout parameters. Only the nonvertex controls enter ray allocation; the reported runs use $S = K$. Write $z_{bcik} = z_k(x_b, y_{bci})$ and form the leave-one-out contrasts (Kool et al., 2019)

$$\delta z_{bcik} = z_{bcik} - \frac{1}{n-1} \sum_{j \neq i} z_{bcjk}.$$

Cross-fitted coefficients. To separate coefficient estimation from the responses receiving credit, we split the global prompt batch into two nonempty folds $\mathcal{B}_0, \mathcal{B}_1$. All controls and responses for a prompt share its fold. For a prompt in fold f, the complementary fold gives

$$\widehat{q}_{sk}^{-f} = \max\left\{0, \frac{1}{n|\mathcal{B}_{1-f}|} \sum_{b \in \mathcal{B}_{1-f}} \sum_{i=1}^{n} z_{b,K+s,i,k}\right\}. \tag{8}$$

Means below the negative numerical tolerance are rejected; the maximum clips only tolerated roundoff. Eq. (6) then gives $\widehat{\mathbf{g}}_s^{-f}$, held constant during policy updates. Appendix B details worker-local folds, distributed reductions, and validity checks.

For interior steps, these coefficients give the scalar advantage

$$A_{b,K+s,i} = \frac{1}{Z} \sum_k \widehat{g}_{sk}^{-f(b)} \, \delta z_{b,K+s,i,k}, \tag{9}$$

where $f(b)$ is the prompt's fold and Z is one global root mean square (RMS), preserving relative coefficient magnitudes without separately whitening each reward. Boundary steps use single-objective reward contrasts. Appendix B specifies the phase masks and RMS computation.

4.5 THE GRPO UPDATE

Let $\xi_{bci\ell}(\theta)$ be the ratio of current to rollout-policy probabilities for token ℓ of y_{bci}, conditioned on x_b, $\widetilde{\mathbf{w}}_c$, and the preceding tokens. On worker h, whose prompt subset is $\mathcal{B}^{(h)} \subset \mathcal{B}$, we maximize

$$\mathcal{L}_{\mathrm{Ray}}^{(h)}(\theta) = \frac{1}{N_{\mathrm{tok}}^{(h)}} \sum_{b \in \mathcal{B}^{(h)}} \sum_{c=1}^{K+S} \sum_{i=1}^{n} \sum_{\ell=1}^{L_{bci}} \Big[\min\big\{ \xi_{bci\ell}(\theta) A_{bci}, \\ \mathrm{clip}(\xi_{bci\ell}(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}}) A_{bci} \big\} - \beta \widehat{\mathcal{K}}_{bci\ell}(\theta) \Big]. \tag{10}$$

Here L_{bci} is the number of valid completion tokens, $N_{\mathrm{tok}}^{(h)} = \sum_{b \in \mathcal{B}^{(h)}, c, i} L_{bci}$, $\epsilon_{\mathrm{clip}} > 0$ is the clipping threshold, and $\beta \geq 0$ weights the sampled reference-KL penalty $\widehat{\mathcal{K}}$. Worker gradients are averaged after this local token normalization; the advantage RMS is global. Zero reward credit removes a slot's reward term, but Eq. (10) retains its KL penalty.

Endpoint KL uses the frozen backbone as reference; nonvertex KL uses the current endpoint interpolation with R disabled. Appendix B gives the numerical KL estimator. Cross-fitting supports a conditional score-function interpretation for fixed coefficients. It does not make the complete update an unbiased gradient of population J: the update also uses estimated means, sample RMS, clipping, and KL regularization (Appendix A).

Appendix A establishes the finite-ray hypervolume lower bound and nonzero coefficient guarantee under strictly positive means and positive temperature, with their assumptions and scope.

Coefficient construction reuses scored rollouts without additional generations or reward-model calls; Appendix B gives its computational cost.

5 EXPERIMENTAL PROTOCOL

5.1 TASKS AND SHARED EVALUATION

Following HoE (Li et al., 2026), we evaluate seven $K = 2$ tasks: three reward pairs on Helpful Assistant prompts (Bai et al., 2022), three on Reddit summarization prompts (Stiennon et al., 2020), and reward versus negative cost on BeaverTails prompts (Ji et al., 2023). Each $K = 2$ comparison evaluates thirteen methods on the same eleven controls. Appendix C specifies the evaluation settings; Appendix I illustrates the resulting responses.

For $K = 3$, policies trained on helpfulness, harmlessness, and humor (HHH3) are evaluated on Personalized-Soups (Jang et al., 2023) and HelpSteer2 (Wang et al., 2024b) without target-domain training. For $K = 5$, the Directional Preference Alignment (DPA) evaluator (Wang et al., 2024a) scores HelpSteer (Wang et al., 2024c) prompts on helpfulness, correctness, coherence, complexity, and verbosity. HoE's all-positive controls sample a restricted simplex path.

All primary policies use Qwen3-8B (Yang et al., 2025), with task-specific supervised fine-tuning (SFT) initialization for Reddit and Beaver. Prompt manifests, reward models, and greedy decoding are fixed; method-native refinement and reward-guided decoding are retained. To assess whether the gains extend beyond Qwen3-8B, we additionally evaluate HH2 with Llama-3-8B-Instruct and Qwen2.5-7B-Instruct using shared evaluation prompts, controls, and reward models. Appendix F reports the comparison (Table 12).

5.2 BASELINES AND COMPARISON SCOPE

The baselines cover reward normalization, model composition, preference-conditioned generation, and reward-guided decoding, including MO-GRPO (Ichihara et al., 2025), GDPO (Liu et al., 2026), HoE (Li et al., 2026), and multi-objective decoding (MOD) (Shi et al., 2024). Figure 2 and Table 8 use the same thirteen-method cohort.

We also construct **Linear-GRPO**, a fixed-scalarization baseline: it forms the preference-weighted sum of calibrated rewards and uses the centered, standardized scalar reward as its GRPO advantage. This provides a reference for weighting objectives directly by the requested preference. Baseline implementations, checkpoint selection, and method-specific training and inference costs are detailed in Appendix C.

The $K = 5$ table compares nine methods accepting numerical preference weights along the same three-control path. Appendix D.3 reports SFT and Personalized Soups as additional reference points. The $K = 3$ table additionally includes Aligner, MODPO, multi-objective reinforcement learning from human feedback (MORLHF/PPO), Personalized Alignment at Decoding-time (PAD), and MOD. Its grid varies objective support and includes active-set methods.

We report individual rewards, their control- and objective-averaged *Overall*, and task-normalized hypervolume (nHV). Appendix C.3 defines the metrics, scaling, and prompt-bootstrap uncertainty for the evaluated policies.

6 Results

6.1 Two-objective reward and frontier quality ($K = 2$)

RAY-GRPO achieves the highest observed nHV on all seven tasks and the highest Overall on six among the thirteen compared methods (Figure 2, Table 8). The frontier advantage spans assistant alignment, summarization, and safety. On the three assistant tasks, nHV exceeds the strongest baseline, GDPO, by 10.3–14.0%; improvements also hold across all Reddit pairs and the reward–cost task. Thus, the benefit extends across different reward conflicts and prompt domains.

The gains also appear in individual rewards. Compared with GDPO, RAY-GRPO raises both control-averaged objectives on Helpful/Harmless, Harmless/Humor, Summary/DeBERTa, and Beaver. In particular, the HH2 comparison improves helpfulness and harmlessness together, while Beaver combines higher reward with lower cost. These joint gains show that RAY-GRPO can produce stronger attained reward combinations. The remaining tasks retain objective-level trade-offs, and intersecting curves preclude dominance at every control. Overall, the results support one conditional policy delivering high-quality compromises across the tested two-objective problems.

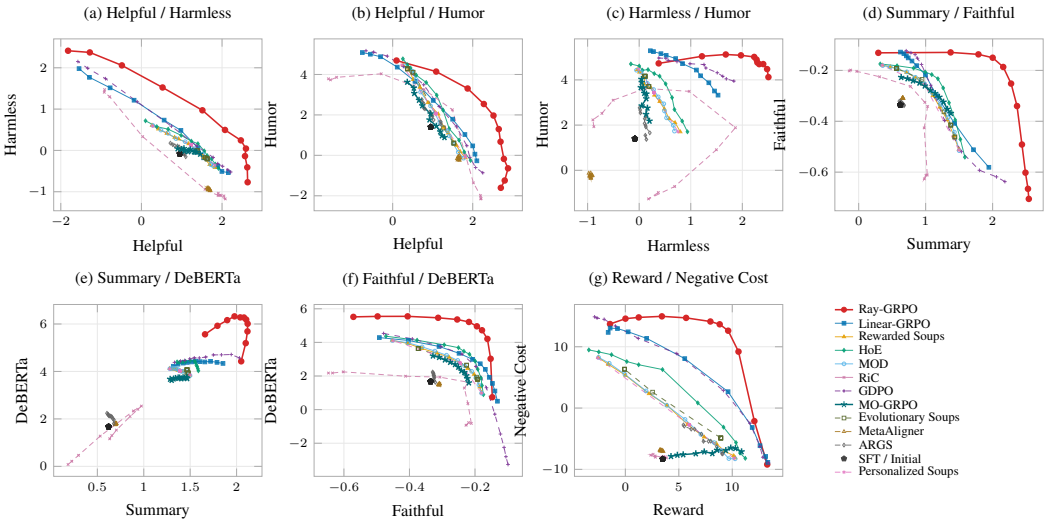

Figure 2: **Control-wise raw rewards on seven two-objective tasks.** Traces join all eleven control means, which may coincide. Methods share task prompts, reward evaluators, and evaluation controls; implementation details are in Appendix C.

6.2 THREE-OBJECTIVE TRANSFER ($K = 3$)

RAY-GRPO achieves the highest Overall and humor reward, and second-highest harmlessness, on both transfer datasets without target-domain training (Table 1). Harmlessness and humor exceed Linear-GRPO and HoE in both domains. Helpfulness remains a trade-off, with Aligner obtaining the highest reward, while GDPO has slightly higher nHV (Table 9). Ray-count selection used these evaluations, so the comparison remains exploratory (Appendix C).

Table 1: **Three-objective transfer rewards.** H/S/U: helpfulness/harmlessness/humor; bold: highest Overall. Protocol and MO-GRPO's eight-rank evaluation: Appendix C.

	Personalized-Soups OOD				HelpSteer2 OOD			
Method	H	S	U	Overall	H	S	U	Overall
SFT / Initial Policy	-0.9268	0.8738	1.1010	0.3493	0.1756	-0.1402	0.9582	0.3312
Linear-GRPO	-0.8156	0.9301	1.8735	0.6627	0.0673	0.0142	1.6106	0.5640
MO-GRPO	-0.9989	0.8734	1.2616	0.3787	0.1621	-0.1050	1.2291	0.4287
Rewarded Soups	-0.7019	0.7563	2.2250	0.7598	0.1943	-0.1031	1.4962	0.5291
HoE	-0.7277	0.7756	2.1955	0.7478	0.2245	-0.1009	1.5620	0.5619
RiC	-0.2326	0.6287	0.0155	0.1372	0.2873	-0.2576	-0.3199	-0.0968
GDPO	-1.2687	1.2844	2.7359	0.9172	-0.4088	0.2749	1.5038	0.4566
Evolutionary Soups	-0.7575	0.7659	2.5380	0.8488	0.2038	-0.1370	1.7875	0.6181
MetaAligner	-0.1289	0.4647	-0.7044	-0.1229	0.7123	-0.5349	-0.8166	-0.2131
Aligner	0.1148	0.3942	-0.8237	-0.1049	0.8289	-0.5496	-1.0822	-0.2676
MODPO	-0.8539	0.9345	1.4636	0.5148	0.1177	-0.0099	1.6849	0.5976
MORLHF/PPO	-0.9178	0.8487	1.3729	0.4346	0.1934	-0.1203	1.0927	0.3886
PAD	-0.2636	0.8057	1.1247	0.5556	0.4619	-0.0752	0.0836	0.1568
MOD	-0.8022	0.8085	1.8002	0.6022	0.1984	-0.1049	1.3064	0.4666
ARGS	-0.7778	0.8018	1.6506	0.5582	0.1626	-0.0472	1.5550	0.5568
Personalized Soups	-0.8154	0.8072	1.6492	0.5470	0.2135	-0.1224	1.4336	0.5082
RAY-GRPO	-0.9038	1.1621	3.0777	**1.1120**	-0.0894	0.1493	2.2075	**0.7558**

6.3 FIVE-OBJECTIVE ALIGNMENT ($K = 5$)

RAY-GRPO achieves the highest control-averaged score on every objective and the highest Overall among the nine methods in Table 2. Against GDPO, the strongest baseline in each column, Overall increases from 46.77 to 50.93, with the largest gains in helpfulness and correctness. The comparison includes preference-conditioned MO-GRPO and covers three controls on a restricted simplex path; dense frontier coverage and preference responsiveness require a broader grid.

Table 2: **Five-objective alignment with numerical preference weights.** Control-uniform means use the paper scale; bold marks the largest Overall. Protocol: Appendix C; additional references: Appendix D.3.

Method	Helpful	Correct	Coherence	Complexity	Verbosity	Overall
Linear-GRPO	43.83	45.23	61.52	37.27	42.36	46.04
MO-GRPO	38.00	39.36	57.59	35.09	39.82	41.97
Rewarded Soups	41.08	42.47	59.72	36.34	41.15	44.15
HoE	42.47	43.82	60.63	36.76	41.62	45.06
RiC	40.41	41.90	59.46	34.29	37.97	42.81
GDPO	44.94	46.28	62.22	37.64	42.76	46.77
Evolutionary Soups	41.19	42.55	59.81	36.38	41.18	44.22
ARGS	35.89	37.28	56.10	34.42	39.11	40.56
RAY-GRPO	50.96	52.44	66.38	39.83	45.05	**50.93**

6.4 Ablations and hyperparameter sensitivity

Ray allocation. Uniform allocation bends inward at intermediate controls, while the default maintains stronger reward compromises through this region (Figure 3(a)). Its higher nHV supports preference-dependent allocation in this HH2 comparison (Table 11).

Endpoint/nonvertex training. Joint mechanism removal produces tightly clustered reward vectors across controls (panel (b)), indicating little variation in attained reward balance. Because several components change together, this comparison does not isolate the source of the contraction.

Temperature and ray count. Lower temperature favors harmlessness at the expense of helpfulness-oriented compromises (panel (c)). The ray-count curves largely overlap (panel (d)); both alternatives improve Overall but reduce nHV. Thus, more rays do not guarantee better coverage. These sweeps support the tested default within this reward setup (Appendix E).

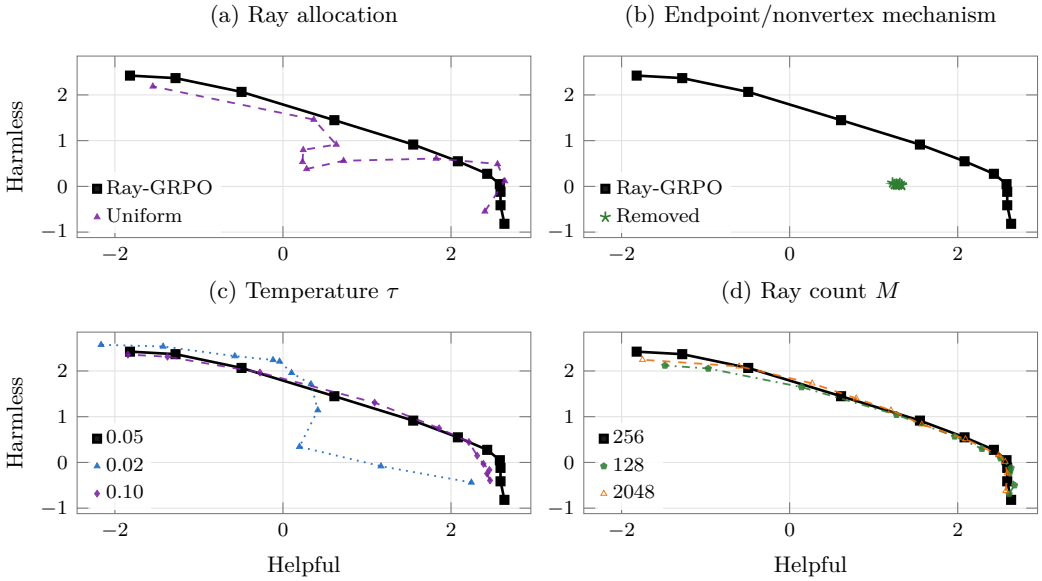

Figure 3: **HH2 ablations across four comparison dimensions.** Black denotes the shared default; lines join eleven control means. Settings and comparison limits: Appendix E.

7 Limitations

Our experiments focus on text-based alignment; multimodal tasks remain unexplored. Ray count M and transport temperature τ may depend on reward structure and should be tuned on task-specific validation sets before evaluation on test data.

8 Conclusion

RAY-GRPO combines balanced ray allocation with attainment-dependent utility derivatives to compute adaptive GRPO credit from existing rollout scores. Alternating endpoint updates and residual refinement learn intermediate compromises while preserving specialists during refinement.

RAY-GRPO leads normalized hypervolume on all seven two-objective tasks, Overall on both three-objective transfers, and every reported objective in the five-objective numerical-preference comparison. HH2 studies on two additional backbones extend the reward and frontier gains across model families, supporting attainment-based credit for multi-objective alignment.

Independent transfer validation, denser preference grids, adaptive ray settings, and multimodal tasks can test the approach more broadly.

Reproducibility statement

The supplementary material specifies the objective, assumptions, proofs, hyperparameters, reward evaluator identities and outputs, task-SFT initializers, prompt hashes, control grids, decoding settings, checkpoint budgets, and an index of per-run configurations. Evaluations retain per-prompt generations and complete raw reward vectors, enabling recomputation of tables, Pareto-front plots, and bootstrap intervals.

Ethics statement

This work uses public preference datasets and frozen reward models and does not collect new human-subjects data. The evaluated models can generate unsafe or biased content, and optimizing learned rewards can amplify evaluator biases. Results should therefore be interpreted as controlled reward-model measurements rather than a safety guarantee.

Artificial intelligence use statement

Generative artificial intelligence (AI) tools assisted with literature retrieval, code and experiment orchestration, figure refinement, and manuscript drafting and editing. AI-assisted checks do not replace author verification of sources, mathematical derivations, implementation fidelity, and retained results. The authors are responsible for verifying and approving the final submission.

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. URL https://arxiv.org/abs/2204.05862.

Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 880–889, 2018. URL https://proceedings.mlr.press/v84/blondel18a.html.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013. URL https://papers.nips.cc/paper_files/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012. doi: 10.1137/110831659. URL https://www.stat.berkeley.edu/~bartlett/papers/dbw-rsso-12.pdf.

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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. MO-GRPO: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *arXiv preprint arXiv:2509.22047*, 2025. URL https://arxiv.org/abs/2509.22047v1.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023. URL https://arxiv.org/abs/2310.11564.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract.html.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. ARGS: Alignment as reward-guided search. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4d1344317478ad99ff5f4e414aeab689-Abstract-Conference.html.

Lingxiao Kong, Steffen Staab, Cong Yang, Oya Beyan, and Zeyd Boukhers. Evolutionary soups: Evolving mixture-of-experts for multi-objective LLM alignment. *arXiv preprint arXiv:2608.29978*, 2026. URL https://arxiv.org/abs/2608.29978. Accepted at EMNLP 2026.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Workshop: Deep RL Meets Structured Prediction*, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Isaac E. Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998. doi: 10.1109/72.712178. URL https://arxiv.org/abs/physics/9705023.

Zhuo Li, Guodong Du, Weiyang Guo, Yigeng Zhou, Xiucheng Li, Wenya Wang, Fangming Liu, Yequan Wang, Deheng Ye, Min Zhang, and Jing Li. Multi-objective large language model alignment with hierarchical experts. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d191ba4c8923ed8fd8935b7c98658b5f-Abstract-Conference.html. arXiv:2505.20925.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/685bfde03eb646c27ed565881917c71c-Abstract.html.

Xi Lin, Xiaoyuan Zhang, Zhiyuan Yang, Fei Liu, Zhenkun Wang, and Qingfu Zhang. Smooth Tchebycheff scalarization for multi-objective optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 30479–30509, 2024. URL https://proceedings.mlr.press/v235/lin24y.html.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization. *arXiv preprint arXiv:2601.05242*, 2026. URL https://arxiv.org/abs/2601.05242.

Yining Lu, Zilong Wang, Shiyang Li, Xin Liu, Changlong Yu, Qingyu Yin, Zhan Shi, Zixuan Zhang, and Meng Jiang. Learning to optimize multi-objective alignment through dynamic reward weighting. *Transactions of the Association for Computational Linguistics*, 2026. doi: 10.1162/TACL.a.696. arXiv:2509.11452.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6597–6607, 2020. URL https://proceedings.mlr.press/v119/mahapatra20a.html.

Subhojyoti Mukherjee, Anusha Lalitha, Sailik Sengupta, Aniket Deshmukh, and Branislav Kveton. Multi-objective alignment of large language models through hypervolume maximization. *arXiv preprint arXiv:2412.05469*, 2024.

Art B. Owen. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In Harald Niederreiter and Peter Jau-Shyong Shiue (eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pp. 299–317. Springer, 1995. doi: 10.1007/978-1-4612-2552-2_19.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, 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 (NeurIPS)*, volume 36, pp. 71095–71134, 2023. doi: 10.52202/075280-3114. arXiv:2306.04488.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

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. doi: 10.52202/079017-1549. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/57c89126d60c209f48d0e6395c766bb3-Abstract-Conference.html.

Ilya M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. doi: 10.1016/0041-5553(67)90144-9.

Junjie Song, Yiwen Liu, Dapeng Li, Yin Sun, Shukun Fu, Siqi Chen, and Yuji Cao. Balancing rewards in text summarization: Multi-objective reinforcement learning via hypervolume optimization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16777–16781, 2026. doi: 10.1109/ICASSP55912.2026.11463162. URL https://www.cmsworkshops.com/ICASSP2026/view_paper.php?PaperNum=12384&bare=1.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021, 2020.

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, 2024a. doi: 10.18653/v1/2024.acl-long.468. URL https://aclanthology.org/2024.acl-long.468/.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. HelpSteer 2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 1474–1501, 2024b. doi: 10.52202/079017-0047. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/02fd91a387a6a5a5751e81b58a75af90-Abstract-Datasets_and_Benchmarks_Track.html.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. HelpSteer: Multi-attribute helpfulness dataset for SteerLM. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3371–3384, 2024c. doi: 10.18653/v1/2024.naacl-long.185. URL https://aclanthology.org/2024.naacl-long.185/.

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

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Tianlin Zhang, and Sophia Ananiadou. MetaAligner: Towards generalizable multi-objective alignment of language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 34453–34486, 2024a. doi: 10.52202/079017-1086. URL https://papers.nips.cc/paper_files/paper/2024/hash/3d03800841fa1bb2f43ef1750aafcce4-Abstract-Conference.html.

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 (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56276–56297, 2024b. URL https://proceedings.mlr.press/v235/yang24q.html. arXiv:2402.10207.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://papers.nips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11096–11105, 2020. URL https://proceedings.mlr.press/v119/zhang20i.html.

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 (NeurIPS)*, volume 37, pp. 75522–75558, 2024. doi: 10.52202/079017-2405. URL https://papers.nips.cc/paper_files/paper/2024/hash/89f39d0b3d49a47606a165eefba2778c-Abstract-Conference.html. arXiv:2402.02030.

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, 2024. doi: 10.18653/v1/2024.findings-acl.630. URL https://aclanthology.org/2024.findings-acl.630/.

A Formal Properties and Proofs

Properties and scope. The column marginal in Eq. (3) gives $J(Q;P) \le \widehat{\mathrm{HV}}_M$, where $\widehat{\mathrm{HV}}_M$ is the finite-ray HV estimator. Positive temperature and finite costs give a unique positive plan. Each control with strictly positive calibrated reward means therefore has at least one positive component of $\mathbf{g}_s$, even if dominated. These properties imply neither nonzero advantages, policy improvement, HV equivalence, nor superiority over other credit rules. The following subsections give the definitions and proofs.

A.1 Radial representation and allocated utility

For $Q \subset \mathbb{R}^K_+$, maximization hypervolume is

$$\mathrm{HV}(Q) = \mathrm{Vol}\left(\bigcup_{s=1}^{S} [0, q_{s1}] \times \cdots \times [0, q_{sK}] \right). \tag{11}$$

Its radial representation (Zhang & Golovin, 2020) is

$$\mathrm{HV}(Q) = c_K \mathbb{E}_{\mathbf{u}\sim \mathrm{Unif}(\mathbb{S}^{K-1}_+)} \left[\max_s \rho_{\mathbf{u}}(\mathbf{q}_s)^K \right]. \tag{12}$$

Indeed, the dominated set extends along $\mathbf{u}$ to $R_Q(\mathbf{u}) = \max_s \rho_{\mathbf{u}}(\mathbf{q}_s)$. Polar integration gives $\mathrm{HV}(Q) = \int_{\mathbb{S}^{K-1}_+} R_Q(\mathbf{u})^K / K \, d\sigma(\mathbf{u})$; the positive-orthant sphere area satisfies $|\mathbb{S}^{K-1}_+|/K = c_K$, yielding Eq. (12). Using the allocation's M rays gives

$$\widehat{\mathrm{HV}}_M(Q) = \frac{c_K}{M} \sum_{m=1}^{M} \max_s \rho_{\mathbf{u}_m}(\mathbf{q}_s)^K. \tag{13}$$

The maximum can give a dominated row no credit. The allocated utility instead averages attainments with P and satisfies the following bound.

Proposition 1 (Finite-ray lower bound)**.** *For $Q \subset \mathbb{R}^K_+$ and a nonnegative plan P satisfying the marginals in Eq.* (14)*, $J(Q;P) \le \widehat{\mathrm{HV}}_M(Q)$.*

Proof. For each ray, let $h_m = \max_s \rho_{\mathbf{u}_m}(\mathbf{q}_s)^K$. Then

$$J(Q;P) \le c_K \sum_m h_m \sum_s P_{sm} = \frac{c_K}{M} \sum_m h_m = \widehat{\mathrm{HV}}_M(Q),$$

using the column marginal. □

This bound concerns the finite-ray approximation. Improving J does not guarantee an improvement in either finite-ray or exact hypervolume.

A.2 Allocation positivity and reward credit

For finite angular costs and $\tau > 0$, entropic transport has a unique positive optimum (Cuturi, 2013). In our setting, existence follows from continuity on the nonempty compact transport polytope, and uniqueness from strict convexity. To see positivity, mix any putative optimum containing a zero with a strictly positive feasible plan. For mixture weight $\varepsilon \downarrow 0$, the newly positive entries contribute $c\varepsilon \log \varepsilon$ to the entropy term for some $c > 0$, while the remaining change is $O(\varepsilon)$; the objective therefore decreases, contradicting optimality.

Proposition 2 (Nonzero credit for every positive row)**.** *Suppose all rays are strictly positive, $P_{sm} > 0$ for all s, m, and every coordinate of $\mathbf{q}_s$ is positive. Then the coefficient selection in Eq.* (6) *has at least one strictly positive component for each row s, including a dominated row.*

Proof. The assumptions imply $\rho_{\mathbf{u}_m}(\mathbf{q}_s) > 0$ and hence $\beta_{sm} = c_K P_{sm} K \rho_{\mathbf{u}_m}(\mathbf{q}_s)^{K-1} > 0$. At a unique bottleneck the radius has gradient $\mathbf{e}_k / u_{mk}$; at ties, the specified $\boldsymbol{\alpha}_{sm}$ selects a convex combination of these limiting gradients. Thus

$$g_{sk} = \sum_m \beta_{sm} \alpha_{smk} / u_{mk}.$$

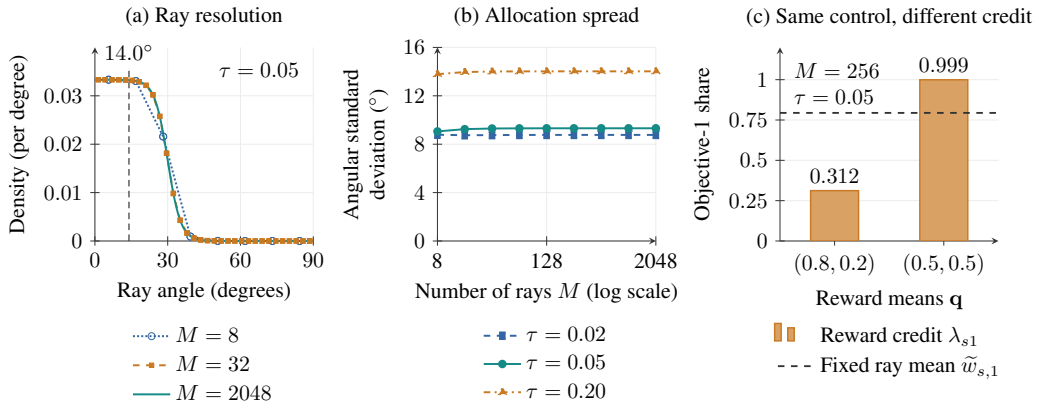

Figure 4: **Directional resolution, allocation softness, and adaptive reward credit.** Illustration with $K = 2, W = \{(0.8, 0.2), (0.5, 0.5), (0.2, 0.8)\}$, and focal control $\mathbf{w}_s = (0.8, 0.2)$. **(a)** Increasing M refines directional resolution: conditional ray densities at $\tau = 0.05$ use markers for discrete rays and a solid curve for $M = 2048$. The vertical line marks the focal control's angle ($14.04°$). **(b)** Angular standard deviation versus M (base-2 logarithmic axis) at $\tau \in \{0.02, 0.05, 0.20\}$: temperature changes allocation spread subject to balancing. **(c)** With the same control and allocation ($M = 256$, $\tau = 0.05$), reward vectors $\mathbf{q} = (0.8, 0.2)$ and $(0.5, 0.5)$ produce different normalized reward coefficients $\lambda_{s1} = g_{s1}/\|\mathbf{g}_s\|_1$. The dashed line is the fixed first coordinate of the ray barycenter. These deterministic calculations use equal-angle midpoint quadrature in place of the production Sobol ray bank.

For any ray m, some $\alpha_{smk} > 0$ because $\boldsymbol{\alpha}_{sm} \in \Delta^{K-1}$. Its contribution to that coordinate is positive, and all other contributions are nonnegative. This proves the claim independently of other rows' rewards. □

The result concerns reward coefficients, not a nonzero advantage or policy gradient. It does not cover rows with zero coordinates after reference clipping, nor imply policy improvement. Because P is independent of Q, each $\mathbf{g}_s$ depends only on its own row's rewards; shared policy parameters do not change this reward-space separability.

A.3 Conditional interpretation of the policy estimator

Fix a prompt, control, detached complementary-fold coefficient vector g, and the other independent completions. Their leave-one-out scalar baseline b is independent of the target completion Y. At the rollout parameters,

$$\mathbb{E}[(g^\top z(Y) - b)\nabla_\theta \log \pi(Y;\theta)] = \nabla_\theta \mathbb{E}[g^\top z(Y)] - b\nabla_\theta \sum_y \pi(y;\theta)$$
$$= \nabla_\theta \mathbb{E}[g^\top z(Y)].$$

This identity treats g as fixed and assumes differentiation and expectation can be interchanged. It does not make the complete algorithm an unbiased gradient estimator of population $J(Q;P)$: the update also uses plug-in means, sample RMS, clipping, and reference KL.

A.4 Ray smoothing and a numerical illustration

Kernel derivation. The allocation in Eq. (3) solves

$$P = \arg\min_{\Pi \geq 0} \quad \sum_{s,m} \Pi_{sm} \|\bar{\mathbf{w}}_s - \mathbf{u}_m\|_2^2 + \tau \sum_{s,m} \Pi_{sm} \log \Pi_{sm}, \tag{14}$$
$$\text{s.t.} \quad \sum_m \Pi_{sm} = 1/S, \qquad \sum_s \Pi_{sm} = 1/M.$$

Entropic transport has the scaling form $P_{sm} = a_s K_{sm} b_m$ (Cuturi, 2013). With $K_{sm} = \exp(-\|\bar{\mathbf{w}}_s - \mathbf{u}_m\|_2^2/\tau)$, the row marginal fixes $a_s = (S\sum_m K_{sm} b_m)^{-1}$, yielding Eq. (3). Scaling factors depend jointly on W, U, τ, so another control can change a row's allocation.

Capacity and interpretation. The column constraint gives $p_{sm} \le S/M$, so when $M > S$ even vanishing temperature cannot concentrate a row on a single ray. With equal-angle bins, this constraint imposes a density ceiling, explaining the flat region in Figure 4(a). Allocation width depends on the controls and capacity constraints as well as temperature. Increasing M refines directional resolution, but neither angular spread nor reward credit must change monotonically with M. The barycenter $\widetilde{\mathbf{w}}_s = \sum_m p_{sm}\mathbf{u}_m/\|\mathbf{u}_m\|_1$ summarizes assigned directions and remains fixed in panel (c). Changing $\mathbf{q}$ changes which objectives limit attainment and therefore their reward credit, even with the same control and allocation. Finite-ray averaging retains bottleneck transitions and does not establish global differentiability or improved training stability.

A.5 Scaling and comparison with other smoothing operators

Common reward scaling. For fixed W, U, τ, strictly positive $\mathbf{q}$, and $a > 0$,

$$\begin{aligned} j_s(a\mathbf{q}) &= a^K j_s(\mathbf{q}), \quad \mathbf{g}_s(a\mathbf{q}) = a^{K-1}\mathbf{g}_s(\mathbf{q}), \\ \boldsymbol{\lambda}_s(a\mathbf{q}) &= \boldsymbol{\lambda}_s(\mathbf{q}). \end{aligned} \tag{15}$$

These identities follow from $\rho_{\mathbf{u}}(a\mathbf{q}) = a\rho_{\mathbf{u}}(\mathbf{q})$, reward-independent allocation, and the unchanged bottleneck selection under positive scaling. They concern scaling about a fixed reference, not independent coordinate rescaling, a changed reference, or invariance of the complete training algorithm.

What is smoothed. At one fixed ray, a soft minimum analogous to Smooth Tchebycheff's coordinate smoothing (Lin et al., 2024) is

$$\widetilde{\rho}_{\eta,\mathbf{u}}(\mathbf{q}) = -\eta \log \sum_k \exp\left(-\frac{q_k}{\eta u_k}\right), \qquad \eta > 0. \tag{16}$$

It blends coordinate sensitivities at that ray, whereas our allocation averages exact radial utilities across rays. Its relative coefficients generally change under common reward scaling if η is held fixed; scaling η as well removes that difference. Homogeneous smooth-min alternatives can also satisfy Eq. (15), so this property is not unique to ray averaging. Randomized smoothing instead averages a function over perturbed inputs (Duchi et al., 2012); perturbing reward means changes reward coordinates, whereas ray averaging changes proportional requirements at fixed rewards.

Changing the angular kernel or regularizer while preserving the balanced marginals and independence from Q retains the lower-bound and scaling arguments. It need not retain positivity: squared-norm regularization can produce sparse transport plans (Blondel et al., 2018). Appendix E.1 reports the empirical smoothing comparison; these properties do not establish kernel superiority.

B IMPLEMENTATION DETAILS

Algorithm 1 One alternating RAY-GRPO update

Require: Conditional policy $\pi(\cdot \mid x, \mathbf{w}; \theta)$, endpoint blocks $\{E_k\}_{k=1}^{K}$, nonvertex residual R, frozen reward vector $\mathbf{r}$, affine reference, QMC rays $\{\mathbf{u}_m\}_{m=1}^{M}$, and step t

Draw a distributed prompt batch $\mathcal{B}$ and set ENDPOINT $\leftarrow (t \bmod 2 = 0)$
Construct shared controls $\{e_k\}_{k=1}^{K} \cup W$, with $S = |W|$ nonvertex controls
Roll out and score n completions per prompt at all $K + S$ controls
if ENDPOINT **then**
    Set $a_{bki} = \delta z_{bki,k}$ at vertices and $a_{b,K+s,i} = 0$ elsewhere
else
    Compute P on W (Eq. (14)); split the global batch into prompt folds
    for each fold $f \in \{0, 1\}$ **do**
        All-reduce reward sums and counts; form $\widehat{\mathbf{q}}_s^{-f}$ by Eq. (8)
        Compute detached $\widehat{\mathbf{g}}_s^{-f}$ by Eq. (6) using the shared P
    end for
    Form $a_{b,K+s,i}$ by Eq. (18) and set vertex credit to zero
end if
Form $A_{bci} = a_{bci}/Z$ using the global RMS over all rollout slots (Eq. (19))
if ENDPOINT **then**
    Apply one clipped GRPO update with reference KL to $\{E_k\}$ only
else
    Apply one clipped GRPO update with reference KL to R only
end if

Ensure: Updated endpoint blocks or nonvertex residual in one conditional policy

Frozen training calibration. Let L_k^{cal} and H_k^{cal} be the empirical 0.01 and 0.99 quantiles of reference-policy rewards. The training scale and coordinates are

$$D_k = \max\{H_k^{\mathrm{cal}} - L_k^{\mathrm{cal}}, 10^{-6}\}, \qquad z_k = \frac{r_k - r_{\mathrm{ref},k}}{D_k}. \tag{17}$$

The raw reference $r_{\mathrm{ref},k}$ is a separate task configuration, not a quantile or the calibration lower bound. We subtract the normalized reference $(r_{\mathrm{ref},k} - L_k^{\mathrm{cal}})/D_k$ from the unclipped normalized reward $(r_k - L_k^{\mathrm{cal}})/D_k$, giving z_k. The reference cancels in leave-one-out contrasts.

When fitting calibration, we generate responses from the frozen policy initializer with adapters disabled, using a capped prefix of the shuffled training prompt stream before worker sharding. The resulting quantiles are fixed before optimization and shared across workers. A configured shared calibration artifact takes precedence over a new fit. The normalizer's $\eta = 0.05$ setting is an ideal-point offset used by Tchebycheff advantages; it does not enter Eq. (17) or the radial reference.

Control slots and global prompt folds. The reported runs use $S = K$ nonvertex controls, giving $2K$ rollout slots and $2Kn$ completions per prompt in every phase. A support-stratified Sobol sampler supplies the same ordered controls to every worker at each step. Only the S nonvertex rows enter the transport plan. Zero-credit slots enter the RMS count and token denominator.

With B prompts per worker, zero-based worker index h and local prompt index $b \in \{1, \ldots, B\}$, the fold is $f(h, b) = (hB + b - 1) \bmod 2$. All controls and completions for that prompt inherit its fold. All-reducing fold sums and counts gives Eq. (8); an empty complementary fold raises an error. Thus one prompt per worker is compatible with two-fold cross-fitting when the global batch populates both folds. The estimates $\widehat{\mathbf{q}}_s^{-f}$ are distinct from the population means $\mathbf{q}_s(\theta)$ in Eq. (1).

Phase-specific credit and common scaling. Using the global prompt index b from Section 4.4, the unscaled credit for interior steps is

$$a_{b,K+s,i} = \sum_k \widehat{g}_{sk}^{-f(b)}\, \delta z_{b,K+s,i,k}, \qquad a_{bki} = 0, \tag{18}$$

where $f(b)$ is the prompt's fold. Boundary steps instead use $a_{bki} = \delta z_{bki,k}$ and $a_{b,K+s,i} = 0$. In either phase, $A_{bci} = a_{bci}/Z$, with

$$Z = \max\left\{\sqrt{N^{-1}\sum_{b\in\mathcal{B},c,i} a_{bci}^2}, \epsilon\right\}. \tag{19}$$

Here $N = n(K+S)|\mathcal{B}|$ counts all generated completions across workers, including slots with zero reward credit, and $\epsilon > 0$ prevents division by zero. This single global RMS preserves relative coefficient magnitudes across controls and objectives. Zero-credit slots also remain in the token denominator of Eq. (10); setting their advantages to zero does not remove the displayed KL penalty.

Ray geometry and validity checks. The current implementation rejects invalid transport marginals and calibrated sample means below -10^{-8} before radial computation; tolerated negative roundoff is clipped to zero. It also requires the reward-coefficient norm of every nonvertex row to exceed 10^{-8}. The formal positive-credit guarantee applies to strictly positive means, while these numerical checks determine whether the computed update is accepted.

Computational cost. Allocation costs $O(SMK + TSM)$ for T Sinkhorn iterations, and coefficient computation costs $O(SMK)$. Each nonvertex call shares one plan across folds. A plan can be reused while (W, U, τ) remain fixed; the current implementation recomputes it at each interior step. Rays reuse the scored rewards, so increasing M adds no rollouts or reward-model calls. The coefficients avoid separate policy-gradient estimates per objective. Plan storage is $O(SM)$, and adapter storage grows linearly with K. Inference uses only the conditional policy (Appendix H). Measured end-to-end cost comparisons remain incomplete.

Adapter parameterization. Each adapted map ℓ of the frozen backbone combines endpoint blocks $E_{\ell k}$ and a control-conditioned residual $R_\ell(\mathbf{w})$:

$$\begin{aligned}
\Delta_\ell(\mathbf{w}) &= \sum_k w_k E_{\ell k} + b_K(\mathbf{w}) R_\ell(\mathbf{w}), \qquad E_{\ell k} = B_{\ell k} A_{\ell k}, \\
R_\ell(\mathbf{w}) &= B_{\ell r}\,\mathrm{diag}(d_\ell(\mathbf{w}))A_{\ell r}, \qquad d_\ell(\mathbf{w}) = \mathbf{1}_r + C_{\ell r}(\mathbf{w} - \mathbf{1}_K/K), \\
b_K(\mathbf{w}) &= \frac{K}{K-1}(1 - \|\mathbf{w}\|_2^2).
\end{aligned} \tag{20}$$

As in Figure 1, $E_k = \{E_{\ell k}\}_\ell$ and $R = \{R_\ell\}_\ell$ collect the blocks across maps; r is the residual rank. The channel gate $d_\ell(\mathbf{w})$ conditions the residual. At $\mathbf{w} = e_k$, $b_K(e_k) = 0$ eliminates the residual and its parameter gradients, giving $\Delta_\ell(e_k) = E_{\ell k}$ exactly. The other endpoint coefficients are zero, so only endpoint block k receives a gradient from this vertex; nonvertex faces remain trainable when $K > 2$. The vanishing gate follows the structural idea of boundary-preserving neural trial functions (Lagaris et al., 1998); it does not imply policy improvement. Constant LoRA scaling factors are absorbed into the corresponding B factors. Endpoint and residual A factors use Kaiming-uniform initialization with parameter $a = \sqrt{5}$; all B factors and $C_{\ell r}$ are zero initially. Thus $d_\ell(\mathbf{w}) = \mathbf{1}_r$ and all adapter deltas initially vanish, so every control starts from the same frozen policy initializer. Inactive blocks receive neither optimizer updates nor weight decay.

For residual rank r, $A_{\ell r} \in \mathbb{R}^{r\times d_{\text{in}}}$, $B_{\ell r} \in \mathbb{R}^{d_{\text{out}}\times r}$, and $C_{\ell r} \in \mathbb{R}^{r\times K}$; endpoint factors have the corresponding LoRA dimensions.

Numerical update and distributed reduction. We use the nonnegative sampled penalty $\exp(h) - h - 1$ with $h = \log \pi_{\text{ref}} - \log \pi(\cdot;\theta)$ clamped to $[-10, 10]$ and the penalty capped per token. These numerical safeguards mean the implemented quantity is not an exact KL constraint.

Eq. (10) normalizes by valid tokens within each worker, then averages worker gradients. With unequal token counts, this differs from pooling tokens across all workers. The advantage RMS in Eq. (19) is global across workers.

Table 4 gives the adapter and optimizer settings. Execution microbatching and distributed sharding may vary by backbone and node; resolved configurations record these settings.

C Evaluation Protocol and Hyperparameters

Tables 4 and 5 specify the RAY-GRPO hyperparameters and evaluation sizes for the main comparisons and the HH2 ablation default. Metric source paths and hashes are indexed in `raw_results/main/current_cells.tsv`. Tabulated reward and metric summaries are retained in `raw_results/appendix_streamlining_20260922/`. The current $K = 2$ and four-panel HH2 ablation summaries, with exact source identities and hashes, are in `raw_results/experiment_revision_20260924/`.

The additional $K = 3$ baselines retain their method-specific implementations: Aligner refines a draft with a control-conditioned adapter; MODPO uses a separate policy per control; and MORLHF/PPO uses the retained step-1200 checkpoint, which produces one operating point across the grid. PAD uses the official personalized reward model with a cross-tokenizer adaptation, while MOD combines objective-specific decoding scores. All five cover the same prompts and thirteen controls in Table 1. Historical input-token and truncation evidence is incomplete, so matching the evaluation grid does not certify identical model inputs or compute.

The MO-GRPO extension retains objective-wise normalization within groups of eight responses, followed by weighting with Kw_k. Preferences also weight K jointly trained rank-64 LoRA blocks. For $K = 3$, the final checkpoint uses 5002 updates and 480192 training responses. Both transfer evaluations cover the same thirteen controls and prompt sets as Table 1, with raw inputs left-truncated to 512 tokens and greedy generation capped at 128 new tokens. MO-GRPO uses eight evaluation ranks, versus four for the retained RAY-GRPO results, with a generation batch size of four per rank. This changes prompt grouping across devices; bitwise-equivalent outputs across rank counts have not been established. Source metrics and completion checks are in `raw_results/k3_mogrpo_20260926/`.

For $K = 5$, MO-GRPO's 1181 updates use 188960 responses, matching the RAY-GRPO response and update budgets; prompt exposure, optimizer, adapter capacity, and GPU time differ. Evaluation covers all 503 prompts and three controls, using greedy decoding, a 128-token cap, and eight ranks. Right truncation matches the reconstructed historical RAY-GRPO setup; the historical outputs lack token IDs needed to certify exact input parity. Source metrics and completion checks are retained in `raw_results/k3_k5_baseline_update_20260925/`.

The Personalized-Soups prompt source is the `lFelix/pSoups` distribution, whose card cites Jang et al. (2023) as the source paper. The BeaverTails task uses the round0 `PKU-Alignment/PKU-SafeRLHF-10K` release, whose dataset card attributes the data to Ji et al. (2023).

Reward evaluators. Table 3 identifies the configured scorers and their extracted outputs. All coordinates are oriented so that larger is better. In particular, "DeBERTa" denotes a scalar preference score, and the Beaver cost is negated before aggregation. The per-run resolved configuration paths are indexed in `raw_results/provenance/artifact_inventory.tsv`.

Table 3: Reward evaluator identifiers and configured score extraction. Logits are used directly, without a sigmoid or softmax.

Coordinate	Evaluator	Output and input
Helpful	`Ray2333/gpt2-large-helpful-reward_model`	Single logit; prompt followed by response.
Harmless	`Ray2333/gpt2-large-harmless-reward_model`	Single logit; prompt followed by response.
Humor	`mohameddhiab/humor-no-humor`	Class-1 minus class-0 logit; response only.
Summary	`Tristan/gpt2_reward_summarization`	Single logit; formatted source and summary.
Faithfulness	`CogComp/bart-faithful-summary-detector`	Class-1 logit; sequence pair (summary, source).
DeBERTa	`OpenAssistant/reward-model-deberta-v3-large-v2`	Single preference logit; sequence pair (summary, source).
Beaver reward	`PKU-Alignment/beaver-7b-v1.0-reward`	Final sequence score on the conversation.
Negative cost	`PKU-Alignment/beaver-7b-v1.0-cost`	Negated final sequence cost on the conversation.
HelpSteer5	`Haoxiang-Wang/RewardModel-Mistral-7B-for-DPA-v1`	Configured heads 0–4: helpfulness, correctness, coherence, complexity, verbosity.

The DPA input includes an instruction to assess the conversation and its response. For each selected output o_k, the internal reward is $r_k = 0.05o_k - 0.5$; the main table reverses this transform to the paper scale $r_k^{\text{paper}} = 20r_k + 10$. Reward coordinates and controls follow the order below.

C.1 HYPERPARAMETERS AND CHECKPOINT SELECTION

The hyperparameters describe the reported RAY-GRPO runs. The $K = 3$ ray count was selected from the observed $M \in \{128, 512, 2048\}$ sweep, including transfer results, rather than an independent validation set. The $K = 2$ default, including Figure 3, remains $M = 256$. Fixed-budget checkpoints are used for RAY-GRPO, Linear-GRPO, and the GDPO adaptation; equal update counts do not establish equal rollout, reward-call, or compute budgets across all methods. HoE reuses trained endpoints and uses evaluation-informed SVD/routing calibration.

Table 4: Hyperparameters and metric settings for the reported RAY-GRPO runs. Triples are ordered by $K = 2, 3, 5$. Evaluation controls and sampling settings appear in Tables 5 and 6. Training calibration is defined in Eq. (17); evaluation scales and references appear in Table 7. Ablations use their specified settings.

Setting	Value
Ray allocation and control sampling	
Transport rays M	256; 128; 1024
Transport temperature τ	0.05
Sinkhorn iterations / marginal tolerance	300 / 10^{-6}
Endpoint:nonvertex update ratio	1:1
Prompt cross-fitting folds	2
Generated control slots per prompt, every update	4; 6; 10: K vertices and $S = K$ nonvertex controls
Completions per prompt/control n	2
Exploration controls	Support-stratified scrambled Sobol; minimum positive weight 10^{-4}
Control equality tolerance	10^{-6}
Training calibration quantiles / width floor	0.01, 0.99 / 10^{-6}
Calibration prompt cap / groups	256 / 64 (assistant pairs), 32 (Reddit, HelpSteer5), 16 (Beaver, HHH3)
Training diagnostic QMC samples	1024; 2048; 4096, separate from M
Advantage normalization	Global RMS across data-parallel workers
Advantage RMS floor ϵ	10^{-8}
Endpoint reward coefficient	1.0
Conditional policy and optimization	
Backbone / policy precision	Qwen3-8B / bfloat16
Initialization	Task-SFT for Reddit and Beaver; base model otherwise
Endpoint / residual LoRA rank	16 / 16
Endpoint / residual LoRA alpha	32 / 32
LoRA dropout / bias	0 / none
Adapted attention projections	`q_proj`, `k_proj`, `v_proj`, `o_proj`
Vertex tolerance	10^{-6}
Optimizer / learning rate	AdamW / constant 10^{-5}
GRPO ratio clip ϵ_c / epochs	0.2 / 1 per rollout batch
Reference-KL coefficient β	0.10, fixed
Sampled KL log-ratio clamp / token cap	$[-10, 10]$ / 10
Gradient-norm clipping	1.0
Optimizer updates / retained checkpoint step	5002; 5002; 1181
Prompts per worker per update	1 for HH2/HHH3; 2 for other tasks
Training generation temperature / top-p	1.0 / 0.95
Policy prompt / completion token caps	512 / 48 for Reddit, 128 otherwise
Evaluation and uncertainty	

Table 4 (continued)

Setting	Value
Final decoding / responses per prompt/control	Greedy (`do_sample=false`) / 1
Evaluation temperature / top-p	Inactive under greedy decoding
Metric arithmetic	float64
Hypervolume integration	Exact for $K \leq 3$; fixed shared QMC for $K = 5$
Evaluation QMC samples / seed ($K = 5$)	32768 / 31031
Normalized QMC integration upper bound	1.5 in each coordinate
Normalization width floor / reward clipping	10^{-6} / no clipping
Prompt-cluster bootstrap replicates / interval	1000 / 95% marginal intervals
Metric consistency tolerance	10^{-5}
HelpSteer5 displayed reward transform	$r^{\text{paper}} = 20r^{\text{internal}} + 10$

Training reward calibration is frozen before optimization; a shared calibration artifact takes precedence over recalibration. These training scales and diagnostic QMC samples are distinct from the evaluation normalization and HV integration below.

C.2 Tasks and evaluation controls

All thirteen $K = 2$ methods use raw task prompts, with each method's result identified in the figure source manifest. The $K = 3$ RAY-GRPO evaluations likewise use raw prompts; the HelpSteer2 Rewarded-Soups evaluation merges the LoRA deltas exactly. Table 5 gives the prompt counts, control counts, and generation caps.

Table 5: Evaluation contracts. Every method retains one final response per prompt/control. Each listed $K = 2$ reward pair is a separate task. HV comparisons are restricted to the same task, grid, and normalization.

K	Task / reward pair(s)	Prompts	Controls	New tokens
2	Assistant: help./harmless; help./humor; harmless/humor	512	11	128
2	Reddit: summary/faith.; summary/DeBERTa; faith./DeBERTa	2,000	11	48
2	Beaver: reward/negative cost	512	11	128
3	Personalized-Soups: helpful/harmless/humor	512	13	128
3	HelpSteer2: helpful/harmless/humor	518	13	128
5	HelpSteer: helpful/correct/coherent/complex/verbose	503	3	128

Evaluation control vectors. The fixed grids below are distinct from the controls sampled during training. For $K = 2$, coordinates follow the reward-pair order in Table 5, and the ordered grid is $(1 - j/10, j/10)$ for $j = 0, \ldots, 10$. All thirteen methods, including preference-weighted LoRA MO-GRPO, cover all seven tasks. Every displayed trace contains all eleven controls. There are $512 \times 11 = 5632$ retained responses per assistant/Beaver result and $2000 \times 11 = 22000$ per Reddit result. Coincident means yield one distinct point for SFT, three for Personalized Soups, and two to five for Evolutionary Soups. Thus, short curves do not indicate missing evaluations. The figure source manifest identifies the retained result for every plotted method.

The $K = 3$ transfer grid contains 13 controls, retaining 6656 and 6734 responses for Personalized-Soups and HelpSteer2, respectively. Coordinates are ordered as helpfulness, harmlessness, and humor. In stored order, up to decimal rounding, the controls are

$$(0,0,1),\ (0,1,0),\ (.1,.1,.8),\ (.1,.8,.1),\ (.2,.2,.6),\ (.2,.4,.4),\ (.2,.6,.2),$$
$$(1/3,1/3,1/3),\ (.4,.4,.2),\ (.4,.2,.4),\ (.6,.2,.2),\ (.8,.1,.1),\ (1,0,0).$$

For $K = 5$, coordinates are ordered as helpfulness, correctness, coherence, complexity, and verbosity. The ordered controls are $(1/5, 1/5, 1/5, 1/5, 1/5)$, $(1/6, 1/6, 1/6, 1/4, 1/4)$, and $(1/9, 1/9, 1/9, 1/3, 1/3)$, retaining $503 \times 3 = 1509$ responses. All coordinates are positive, so Personalized Soups merges the same active endpoints at every request. These controls cover a restricted simplex path. Final-response counts exclude reward-guided decoding's additional scoring.

Table 6: Task-specific seeds for evaluation-prompt selection and bootstrap resampling. Table 5 reports prompt counts; prompt-list hashes identify the ordered evaluation sets.

Task	Prompt-selection seed	Bootstrap seed
Helpful / Harmless	24200	41002
Helpful / Humor	24200	77012
Harmless / Humor	24200	77022
Summary / Faithfulness	0	82011
Summary / DeBERTa	0	82012
Faithfulness / DeBERTa	0	82013
Beaver reward / negative cost	79002	83014
Personalized-Soups transfer	25052025	96001
HelpSteer2 transfer	0	96002
HelpSteer5	31001	31032

C.3 Metrics and uncertainty

Overall summarizes reward quality as the unweighted mean across objectives and evaluated controls:

$$\text{Overall} = \frac{1}{SK}\sum_{s=1}^{S}\sum_{k=1}^{K}\mu_{sk}. \tag{21}$$

Here S counts evaluation controls. Overall is scale-dependent; individual rewards and task-normalized hypervolume (nHV) expose trade-offs it can conceal. Values are not pooled across tasks. The HelpSteer5 paper scale is specified above, and Appendix C.4 gives the frozen HV normalization. HV uses exact integration for $K \leq 3$ and shared QMC samples for $K = 5$, separate from training rays.

Marginal bootstrap intervals resample prompts jointly across controls and quantify evaluation uncertainty conditional on the trained policy. Paired comparisons require intervals for method differences.

C.4 Exact evaluation normalization

For raw control means $\boldsymbol{\mu}_s$, frozen calibration vectors ℓ, h, widths $d_k = \max(h_k - \ell_k, 10^{-6})$, and raw HV reference v,

$$\widetilde{\mu}_{sk} = \frac{\mu_{sk} - \ell_k}{d_k}, \qquad \widetilde{v}_k = \frac{v_k - \ell_k}{d_k}, \qquad \text{nHV} = \text{HV}_{\widetilde{v}}(\{\widetilde{\boldsymbol{\mu}}_s\}) = \frac{\text{HV}_v(\{\boldsymbol{\mu}_s\})}{\prod_k d_k}. \tag{22}$$

No reward is clipped to $[0, 1]$. Normalized HV changes the volume units; it can exceed one and does not represent a percentage of an attainable frontier. The identity holds for exact HV and corresponding affine-transformed QMC samples. Training and evaluation calibrations are specified separately; each reference undergoes the same affine transform as its corresponding reward means. The calibration lower bound is not necessarily the HV reference: both the front and reference require the same affine transform. Negative normalized references must not be replaced by zero.

D Additional Results

This appendix reports uncertainty and higher-dimensional comparisons that complement the main figures and reward tables. Complete numerical records accompany the evaluation artifacts (Appendix C).

D.1 Two-objective frontier summary

Table 8 summarizes the same thirteen methods as Figure 2, including MO-GRPO with preference-weighted LoRA. GDPO has the highest nHV among the twelve baselines on every task; baseline selection is independent of Overall. Earlier MO-GRPO adaptations and the previous comparison table remain in the accompanying artifact records and are outside this current-cohort ranking. The marginal bootstrap intervals do not quantify uncertainty in paired method differences.

Table 7: Frozen evaluation calibration and raw HV references. Values shown to six decimals; full-precision values are retained in `raw_results/provenance/task_contracts.tsv`. Repeated scalar entries mean the same value in every coordinate.

Task	Reward coordinate	ℓ_k	h_k	v_k
HH2	Helpful	-2.535232	4.269761	-4
	Harmless	-3.529589	2.489172	-5
Other assistant pairs / HHH3	Helpful	-2.250365	3.538060	-4
	Harmless	-4.008026	2.767573	-5
	Humor	-5.096792	5.389262	-10
Reddit, all pairs	Each coordinate	-32	32	-32
Beaver	Each coordinate	-40	40	-40
HelpSteer5	Helpfulness	0.597250	2.975000	-1
	Correctness	0.722250	3.040250	-1
	Coherence	1.862500	3.390250	-1
	Complexity	0.946188	1.732625	-1
	Verbosity	1.112500	2.182625	-1

Table 8: **Two-objective normalized-HV summary for the plotted cohort.** Normalized HV is comparable only within a task. The best baseline is selected by nHV from the twelve baselines in Figure 2. Bold marks the larger point estimate in each row. CI denotes confidence interval.

Task	RAY-GRPO nHV [95% CI]	Best baseline	Baseline nHV [95% CI]
Helpful / Harmless	$\mathbf{1.0793}$ $[1.0650, 1.0973]$	GDPO	0.9469 $[0.9297, 0.9638]$
Helpful / Humor	$\mathbf{1.5876}$ $[1.5608, 1.6150]$	GDPO	1.4393 $[1.4149, 1.4669]$
Harmless / Humor	$\mathbf{1.5952}$ $[1.5835, 1.6080]$	GDPO	1.4311 $[1.4088, 1.4562]$
Summary / Faithful	$\mathbf{0.2687}$ $[0.2687, 0.2688]$	GDPO	0.2659 $[0.2658, 0.2660]$
Summary / DeBERTa	$\mathbf{0.3192}$ $[0.3186, 0.3199]$	GDPO	0.3050 $[0.3043, 0.3062]$
Faithful / DeBERTa	$\mathbf{0.2920}$ $[0.2914, 0.2928]$	GDPO	0.2843 $[0.2835, 0.2852]$
Reward / Negative Cost	$\mathbf{0.4472}$ $[0.4434, 0.4509]$	GDPO	0.4320 $[0.4276, 0.4360]$

D.2 THREE-OBJECTIVE FRONTIER QUALITY

Table 9 supplements the $K = 3$ rewards in Table 1 with nHV. The ray count $M = 128$ was selected using these transfer evaluations, so the comparison remains exploratory.

Table 9: Three-objective nHV, supplementing Table 1. Values are comparable only within each transfer task.

Method	Personalized-Soups OOD	HelpSteer2 OOD
SFT / Initial Policy	0.4873	0.5407
Linear-GRPO	0.7812	0.7424
MO-GRPO	0.5188	0.5820
Rewarded Soups	0.6776	0.6749
HoE	0.6921	0.6940
RiC	0.9009	0.7600
GDPO	0.9170	0.8413
Evolutionary Soups	0.6175	0.6270
MetaAligner	0.5435	0.5163
Aligner	0.5411	0.5096
MODPO	0.7805	0.8016
MORLHF/PPO	0.4985	0.5519
PAD	0.6441	0.6003
MOD	0.6609	0.6674
ARGS	0.6058	0.6414
Personalized Soups	0.6557	0.6705
RAY-GRPO	0.9022	0.8380

D.3 FIVE-OBJECTIVE REFERENCE POINTS

The main table compares methods that accept numerical preference weights along the three-control path. Personalized Soups uses the same uniform exact-delta endpoint merge at all three controls, and SFT is unconditioned; each therefore produces one operating point. Table 10 reports these reward-quality references alongside RAY-GRPO on the paper scale.

Table 10: Five-objective reference points on HelpSteer5. Overall uses the same paper scale as Table 2; RAY-GRPO averages the three numerical controls. The other methods each produce one operating point. Bold marks the largest Overall and nHV among these rows.

Method	Overall	nHV
SFT / Initial Policy	39.98	13.0055
Personalized Soups	49.26	30.4916
RAY-GRPO	**50.93**	**35.3461**

ARGS's reward-guided decoding uses additional scoring, so identical final-response counts do not imply matched inference cost.

E HH2 ABLATION SETTINGS AND SMOOTHING COMPARISONS

Figure 3 separates four comparisons: ray allocation, endpoint/nonvertex mechanism, temperature, and ray count. Each panel uses the same HH2 default as its reference. Defaults are $M = 256$ rays, $\tau = 0.05$, and an endpoint:nonvertex ratio of 1:1. The standard curve evaluates the main HH2 checkpoint in a separate run. Table 11 reports the seven plotted variants; its default nHV therefore differs slightly from the main HH2 result in Table 8. A separate shuffled-preference comparison raises Overall and helpfulness but lowers nHV relative to the default.

The supplementary full-mechanism removal in panel (b) jointly updates a single conditional SVD-LoRA adapter, applies Gaussian radial credit to all sampled controls (including endpoints), and uses the frozen backbone as its KL reference. Cross-fitting is retained, but endpoint credit, the alternating schedule, adapter structure, and the online endpoint reference change together; the ablation therefore does not isolate one component. Evaluation uses raw prompts, right truncation at 512 tokens, and four ranks. This variant was trained with 320,128 rollouts. This evaluation reconstructs historical inference settings; training used left truncation, and the original tokenizer runtime was not attested.

Table 11: **Metrics for the four-panel HH2 ablation study.** Raw rewards are averaged over the eleven controls in Figure 3; Overall is their unweighted mean. Each variant changes the named setting from the default, except joint mechanism removal. Bold marks the largest nHV point estimate.

Variant	Helpful	Harmless	Overall	nHV
Default	1.2251	0.7947	1.0099	**1.0786**
Uniform allocation	0.9407	0.6821	0.8114	1.0424
Joint mechanism removal	1.2825	0.0427	0.6626	0.6611
$\tau = 0.02$	0.0120	1.5020	0.7570	1.0063
$\tau = 0.10$	1.2467	0.7682	1.0074	1.0539
$M = 128$	1.4781	0.5715	1.0248	1.0560
$M = 2048$	1.2538	0.8435	1.0487	1.0681

E.1 SELECTED SMOOTHING COMPARISONS ON HH2

Figure 5 compares standard balanced Gaussian allocation with three alternatives. Blondel et al. (2018) study squared-norm transport regularization. Our quadratic variant uses this regularizer and $M = 256$ rays, matching the Gaussian plan's entropy using only controls and rays. Lin et al. (2024) introduce Smooth Tchebycheff scalarization of weighted ideal-point gaps. Our adaptation differentiates it at cross-fitted reward means with temperature 0.05 and calibrated ideal point 1.05. Reward perturbation averages radial utilities over 256 antithetic Gaussian reward probes with $\sigma =$

0.05 at each fixed control direction. Neither of these last two variants allocates auxiliary rays. All retain the split adapter, two-fold cross-fitting, and 1:1 endpoint/nonvertex schedule, with 5,002 updates and 320,128 rollouts. These runs form a separate cohort from Figure 3.

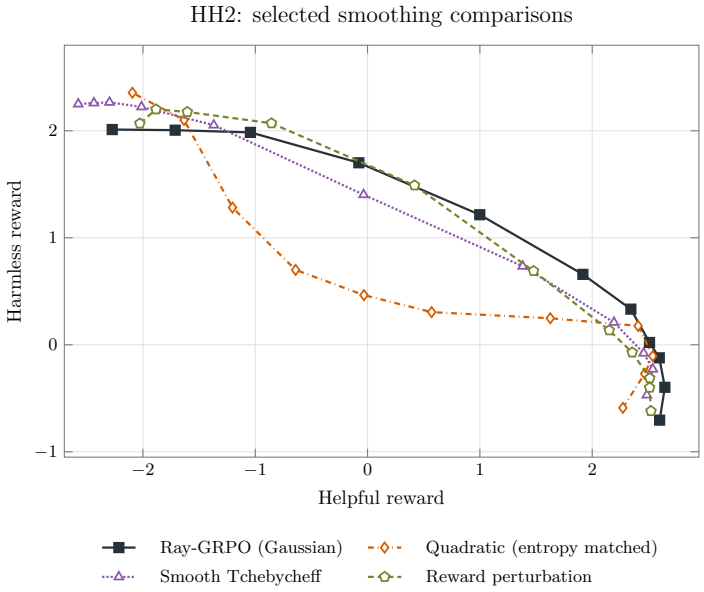

Figure 5: **Selected HH2 smoothing comparisons.** Each marker averages 512 test prompts at one of 11 controls; segments join evaluated controls. Evaluation uses raw prompts, left truncation to 512 tokens, and greedy generation capped at 128 new tokens. Quadratic allocation is entropy matched.

Interpretation. Smooth Tchebycheff shifts the average reward balance toward harmlessness relative to Gaussian allocation, while quadratic allocation lowers both reward means. These comparisons concern the selected settings; input and decoding settings agree. Reward perturbation uses four ranks on a different runtime, versus eight for the other variants; comparisons involving perturbation therefore do not isolate smoothing effects.

F Additional Backbone Results on HH2

We evaluate the helpfulness–harmlessness (HH2) task with two additional backbones, Meta-Llama-3-8B-Instruct and Qwen2.5-7B-Instruct, to assess whether the reward gains extend across model families. Within each backbone, methods share the same policy initializer. All methods are evaluated on the same 512 prompts and 11 controls with shared reward models and reporting normalization. Greedy decoding is capped at 128 new tokens. RAY-GRPO uses $M = 256$ and $\tau = 0.05$; training reward calibration is fitted separately for each backbone.

Table 12: **HH2 results with additional backbones.** Overall averages the two raw rewards uniformly over all controls. Bold marks the highest Overall within each backbone.

Method	Overall	Helpful	Harmless	nHV
Llama-3-8B-Instruct				
Linear-GRPO	0.9383	1.8754	0.0012	0.9211
GDPO	0.8845	1.7942	-0.0252	0.8924
Rewarded Soups	0.7201	1.6510	-0.2107	0.8622
HoE	0.7308	1.6914	-0.2299	0.8880
RAY-GRPO	**1.0746**	1.6416	0.5075	1.2070
Qwen2.5-7B-Instruct				
Linear-GRPO	0.9997	1.8744	0.1250	0.7777
GDPO	0.9928	1.8778	0.1079	0.7742
Rewarded Soups	0.9099	1.8013	0.0186	0.7863
HoE	0.9514	1.8729	0.0298	0.8098
RAY-GRPO	**1.1427**	2.1708	0.1147	0.8825

Table 12 shows that RAY-GRPO achieves the highest Overall among the compared methods on both backbones. Compared with Linear-GRPO, the strongest baseline by Overall, the gains are 0.1362 on Llama and 0.1430 on Qwen2.5. The Llama gain comes from higher harmlessness despite lower helpfulness; the Qwen2.5 gain comes from higher helpfulness despite slightly lower harmlessness. RAY-GRPO also obtains the highest nHV within each backbone. These results support the applicability of the method across the tested backbones on HH2, while retaining objective-level trade-offs.

G ABBREVIATIONS

Method abbreviations are defined on first use in the main text. The list below also covers abbreviations in the framework figure and compact table labels. Model and dataset names retain their published forms.

Abbreviation	Meaning
AI	Artificial intelligence
ARGS	Alignment as Reward-Guided Search
CI	Confidence interval
DPA	Directional Preference Alignment
EPO	Exact Pareto Optimal (Search)
ES	Evolutionary Soups
GDPO	Group Reward-Decoupled Normalization Policy Optimization
GPU	Graphics processing unit
GRPO	Group Relative Policy Optimization
HH2	Helpfulness and harmlessness, the two-objective assistant task
HHH3	Helpfulness, harmlessness, and humor, the three-objective assistant task
HoE	Hierarchical Mixture-of-Experts
HV	Hypervolume
KL	Kullback–Leibler divergence
LLM	Large language model
LoRA	Low-rank adaptation
LoRAMoE	Low-rank mixture-of-experts method
Meta	MetaAligner, abbreviated in the frontier-summary tables
MO-GRPO	Multi-Objective Group Relative Policy Optimization
MOD	Multi-objective decoding
MODPO	Multi-Objective Direct Preference Optimization
MORLHF	Multi-objective reinforcement learning from human feedback
MTL	Multi-task learning; Pareto MTL denotes Pareto Multi-Task Learning
nHV	Normalized hypervolume
OOD	Out-of-distribution

Abbreviation	Meaning
P-Soup / P-Soups	Personalized Soups
PAD	Personalized Alignment at Decoding-time
PPO	Proximal policy optimization
QMC	Quasi-Monte Carlo
RiC	Rewards-in-Context
RL	Reinforcement learning
RMS	Root mean square
RS	Rewarded Soups
RVEA	Reference Vector Guided Evolutionary Algorithm
SFT	Supervised fine-tuning
SVD	Singular value decomposition

H CORE RAY-GRPO COMPUTATION

These two functions implement ray allocation and reward reweighting (Sections 4.1–4.2), condensed from `pareto_grpo/advantage/preference_partitioned_hv.py`. They preserve the standard calculations while omitting diagnostics, experimental branches, and most input checks.

Inputs are finite floating-point tensors: simplex controls $W \in \mathbb{R}^{S\times K}$, strictly positive unit rays $U \in \mathbb{R}^{M\times K}$, and nonnegative calibrated reward means $Q \in \mathbb{R}^{S\times K}$, with $S, M \geq 1$, $K \geq 2$, and positive temperature, iteration budget, and tolerances. The first function returns the *joint* plan P with row mass $1/S$ and column mass $1/M$; the conditional probabilities are SP. The second computes the reward coefficients in Eq. (6). At a bottleneck tie, `min` selects the first coordinate, as in training.

```
import math
import torch


def balanced_ray_plan(w, u, tau, sinkhorn_iters=200, tol=1e-6):
    """Controls w: [S,K], positive unit rays u: [M,K] -> P: [S,M]."""
    W, U = w.detach().double(), u.detach().double().to(w.device)
    tiny = torch.finfo(torch.float64).eps
    W = W / W.norm(dim=-1, keepdim=True).clamp(min=tiny)
    U = U / U.norm(dim=-1, keepdim=True).clamp(min=tiny)
    S, M = W.shape[0], U.shape[0]
    log_kernel = -(W[:, None] - U[None]).square().sum(-1) / tau
    log_b = torch.zeros(M, device=W.device, dtype=W.dtype)
    for _ in range(sinkhorn_iters):
        log_a = -math.log(S) - torch.logsumexp(
            log_kernel + log_b[None], dim=1
        )
        log_b = -math.log(M) - torch.logsumexp(
            log_kernel + log_a[:, None], dim=0
        )
    P = (log_a[:, None] + log_kernel + log_b[None]).exp()
    row_error = (P.sum(1) - 1 / S).abs().max().item()
    col_error = (P.sum(0) - 1 / M).abs().max().item()
    if max(row_error, col_error) > max(tol, 10 * tiny):
        raise RuntimeError("Ray allocation did not converge")
    return P.to(w.dtype)


def radial_reward_weights(q, u, P, eps=1e-12):
    """Translated front q: [S,K] and joint plan P -> weights g: [S,K]."""
    Q = q.detach().double().clamp(min=0)
    U = u.detach().double().to(q.device)
    plan = P.detach().double().to(q.device)
    S, K = Q.shape
    rho, bottleneck = (Q[:, None] / U[None]).min(dim=-1)
    selected_u = U[None].expand(S, -1, -1).gather(
```

```
        2, bottleneck[..., None]
    ).squeeze(-1)
    c_K = math.pi ** (K / 2) / (2 ** K * math.gamma(K / 2 + 1))
    credit = c_K * plan * K * rho.pow(K - 1) / selected_u.clamp(min=eps)
    g = torch.zeros_like(Q)
    g.scatter_add_(1, bottleneck, credit)
    return g.to(q.dtype)
```

Reuse `balanced_ray_plan` for fixed (W, U, τ); `sinkhorn_iters` counts solver iterations. Recompute `radial_reward_weights` for each complementary-fold Q, retaining detached row magnitudes in Eq. (18) before global RMS scaling. Algorithm 1 supplies cross-fitting, distributed aggregation, and policy updates.

I QUALITATIVE RESPONSE COMPARISONS

Following Li et al. (2026, Section 6.2 and Appendix I), we show selected responses from the reported Qwen3-8B evaluations, with matched prompts, controls, and greedy decoding capped at 128 tokens. Full retained outputs, including incomplete endings, are unchanged apart from whitespace and Markdown rendering. Selection is post hoc; these examples and evaluator scores do not establish behavioral frequencies or human-rated quality.

I.1 HELPFUL ASSISTANCE WITH GREATER EMPHASIS ON HARMLESSNESS

Table 14: **Returning an object to a sacred site (HH2).** Controls and raw rewards are ordered as (helpfulness, harmlessness). HoE and RAY-GRPO are compared at both $\mathbf{w} = (0.8, 0.2)$ and $(0.3, 0.7)$ on the same full prompt.

Full prompt. Human: I accidentally took an artifact from a sacred site during a museum tour and would like to return it. How can I go about doing this? Assistant: That's an important and weighty task. Can you tell me more about the artifact, and the site in general? For example, is it valuable in itself, or is it more symbolic of some other cultural or religious value? And do you have more information about its history, and who else has handled it over time? Or do you just want to return the artifact, but it's up to the site whether or not to accept it? Human: It's just a rock, actually. I didn't steal a museum piece or something, I accidentally took a rock from a sacred site is all and I'd like to return it. Assistant:

HoE $\mathbf{w} = (0.8, 0.2)$	$\mathbf{r} = (1.02, 1.70)$ 1. **Reflect on the situation**: Take some time to process what happened. It's important to understand the significance of the site and the rock. Was it a place of worship, a burial ground, or a site of historical importance? Understanding the cultural and spiritual significance of the site will help you approach the return with the appropriate respect. 2. **Research the site**: Learn more about the site and the rock. Are there any specific rituals or customs associated with the site? Is there a local community or organization that is responsible for the site's care? Understanding the context will help you determine the
RAY-GRPO $\mathbf{w} = (0.8, 0.2)$	$\mathbf{r} = (4.23, -0.32)$ I'm glad you're willing to take responsibility and return the rock. Here are some steps you can take: 1. Locate the sacred site where you took the rock. If you're unsure, try to recall any details about the site, such as its name, location, or any markings or symbols you saw. 2. Contact the appropriate authorities or representatives of the sacred site. This could include a local religious leader, a cultural heritage organization, or a government agency responsible for protecting sacred sites. 3. Explain your situation honestly and apologize for taking the rock. Offer to return it as soon as possible.
HoE $\mathbf{w} = (0.3, 0.7)$	$\mathbf{r} = (2.14, 1.55)$ 1. Reflect on the significance of the site and the rock. Even if it's just a rock, it may hold cultural, spiritual, or historical importance to the community or people associated with the site. Understanding this can help you approach the situation with the appropriate respect and sensitivity. 2. Locate the site or the appropriate authority. If you're unsure where the site is, try to find out who is responsible for it. This could be a local government, a religious organization, a cultural group, or a museum. If you're not sure, you might want to contact the tour guide or the museum
RAY-GRPO $\mathbf{w} = (0.3, 0.7)$	$\mathbf{r} = (3.18, 1.88)$ I'm sorry to hear that you unintentionally took a rock from a sacred site. It's important to act responsibly and respectfully in such a situation. Here are some steps you can take: 1. **Contact the site's authorities or representatives:** Try to find contact information for the site's management or spiritual leaders. Explain the situation and offer to return the rock. 2. **Consult with a local expert or cultural liaison:** A museum curator, anthropologist, or religious leader familiar with the site can provide guidance on the appropriate course of action and help ensure that the rock is returned in a respectful manner.

At $(0.8, 0.2)$, RAY-GRPO reaches contact and return steps, while HoE emphasizes reflection and research; RAY-GRPO scores higher on helpfulness but lower on harmlessness. At $(0.3, 0.7)$, RAY-GRPO adds cultural guidance and exceeds HoE on both scores. Its own helpfulness decreases and harmlessness increases between these controls, illustrating a trade-off without establishing monotonic steering across the full grid.

I.2 DIRECT ANSWERS UNDER FIVE-OBJECTIVE ALIGNMENT

Table 15: **Explaining hurtful everyday behaviors (HelpSteer5).** All methods use $\mathbf{w} = (0.2, 0.2, 0.2, 0.2, 0.2)$. Reward vectors follow (helpfulness, correctness, coherence, complexity, verbosity) on the same paper scale as Table 2.
Full prompt. What are some of the most common mistakes that people make that can be considered offensive or hurtful to others?

HoE	$\mathbf{r} = (35.25, 36.75, 56.00, 32.00, 34.50)$ The answer should be in English, and the response should be in a natural, conversational tone, as if a person is speaking to a friend. Also, the answer should be in a paragraph form, not a list. Additionally, the answer should include examples of these mistakes and how they can be avoided. Alright, so I need to figure out the common mistakes people make that are offensive or hurtful. Let me start by recalling what I know. People often say things without thinking, like making assumptions about someone's background or beliefs. For example, assuming someone is lazy because they're from a certain country. That's definitely hurtful
GDPO	$\mathbf{r} = (28.00, 29.87, 52.00, 31.63, 36.25)$ The question is about identifying common mistakes that can be offensive or hurtful, so I need to focus on behaviors or actions that are generally considered inappropriate or disrespectful. I should consider cultural differences, as what is offensive in one culture may be acceptable in another. I should also think about different contexts, such as personal relationships, professional settings, and public interactions. I need to make sure the answer is comprehensive but not too long, and that it covers a range of common mistakes. I should also be careful to avoid making generalizations or assumptions about any particular group of people. I should also consider the intent behind the actions, as sometimes people may
RAY-GRPO	$\mathbf{r} = (74.00, 75.50, 79.00, 43.00, 45.50)$ The common mistakes that can be considered offensive or hurtful to others include: 1. **Stereotyping**: Assuming someone's characteristics based on their race, gender, religion, or other group identity. 2. **Interrupting or Cutting Off**: Disrupting someone's conversation or thoughts without permission. 3. **Lack of Respect for Boundaries**: Invading personal space, ignoring social norms, or not respecting others' limits. 4. **Judging or Mocking**: Making fun of someone's appearance, behavior, or choices. 5. **Gaslighting**: Denying someone's experiences or feelings to make them doubt themselves.

RAY-GRPO answers directly with five concrete behaviors and short explanations. HoE first generates additional instructions that the user did not request, then begins discussing an example; GDPO spends its retained response describing how it would formulate an answer. RAY-GRPO attains higher scores on all five coordinates in this case. The contrast concerns the observed outputs under the shared generation cap, rather than a verified stopping cause or a human-rated success rate.