GENERAL PREFERENCE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning (RL) with verifiable rewards drives emergent reasoning on math and code, but open-ended tasks have no programmatic verifier. A learned scalar reward model can be applied to these tasks in place of a verifier, yet a scalar is the wrong shape for the job. Quality is multi-dimensional, and any scalar score is an incomplete proxy that lets extended online RL collapse onto whichever axis the score is most sensitive to. Recent multi-reward methods counter this collapse by normalizing each reward separately, but they require every reward to be specified in advance. We instead learn latent quality dimensions from ordinary binary preference labels with the General Preference Model (GPM), and propose General Preference Reinforcement Learning (GPRL), which keeps these dimensions separate through the policy update. GPRL computes group-relative advantages per dimension, normalizes each on its own scale, and aggregates them with per-dimension weights. We characterize when this aggregate rejects responses that improve one dimension at the expense of the others, even when a scalar reward would favor them. The same structure yields a drift monitor that detects such exploitation during training and corrects it online. GPRL reaches length-controlled win rates of 56.51% from `Llama-3-8B-Instruct` and 74.31% from `Gemma-2-9B-it` on AlpacaEval 2.0 with the shortest responses of any method; it also outperforms offline, iterative, and scalar-reward online baselines on Arena-Hard, MT-Bench, and WildBench, and does not degrade under extended training.

1 INTRODUCTION

Post-training turns a pretrained model into a system that follows instructions, reasons through hard problems, and aligns with human values (Kumar et al., 2025). It has split into two largely disconnected tracks. The first optimizes the policy directly on preference data, through offline methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) and SimPO (Meng et al., 2024) or iterative game-theoretic variants such as SPPO (Wu et al., 2024b). Offline and iterative methods align well on open-ended tasks, but train on static or iteratively refreshed preference pairs and forgo continuous online exploration (Wu et al., 2024a; Gou & Nguyen, 2024). The second samples fresh responses from the current policy at every step, with Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as its standard algorithm, a critic-free variant of Proximal Policy Optimization (PPO) (Schulman et al., 2017) that estimates advantages within a group of sampled responses and accepts any scoring function as the reward. Paired with a programmatic verifier, this reinforcement learning with verifiable rewards (RLVR) recipe powers models such as DeepSeek-R1 (Guo et al., 2025) and elicits behaviors such as self-reflection that supervised fine-tuning alone does not produce, but open-ended tasks admit no such verifier.

A natural way to connect the two tracks is to replace the verifier with a learned scalar reward model (RM), as in RLMT (Bhaskar et al., 2025), which runs GRPO against a scalar Bradley-Terry (BT) RM (Bradley & Terry, 1952). However, such an approach already exhibits stylistic reward hacking within one epoch, and any further training only compounds it (Gao et al., 2023). The deeper issue is that any scalar score is an *incomplete proxy* for what humans want (Zhuang & Hadfield-Menell, 2020), since quality spans latent dimensions such as helpfulness, factuality, safety, and style, and folding them into one number discards how they trade off (Zhong et al., 2024). Optimizing the proxy then amplifies whichever dimension the RM is most sensitive to while silently degrading the rest.

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.

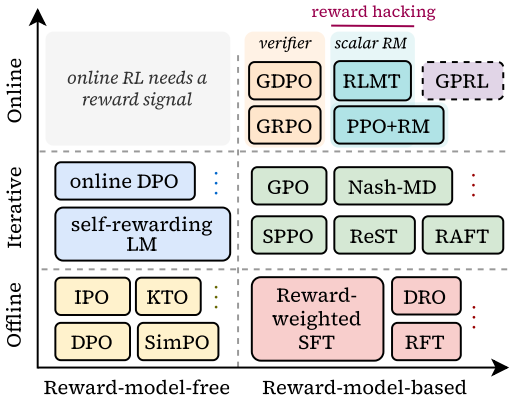


Figure 1: **Landscape of LLM post-training methods**, organized by supervision source and training regime.

Multi-reward variants of GRPO document related failures once rewards are summed, with the highest-variance reward dominating the advantage in MO-GRPO (Ichihara et al., 2026) and distinct reward combinations collapsing to identical advantages in GDPO (Liu et al., 2026). Both normalize each reward separately, but only over rewards specified in advance as rules or separately trained RMs, a limitation shared by multi-objective reinforcement learning from human feedback (RLHF) (Xiong & Singh, 2025; He & Maghsudi, 2025). Open-ended alignment typically offers only binary preference labels, so its dimensions must be learned rather than specified. The General Preference Model (GPM) (Zhang et al., 2025b) learns such dimensions directly from these labels, embedding each response into $\mathbb{R}^{2k}$ and scoring pairs with a block-diagonal skew-symmetric operator whose k two-dimensional subspaces can each capture a different facet of quality, at the same linear query complexity as a BT model.

We propose General Preference Reinforcement Learning (GPRL), which uses GPM as a structured reward for GRPO-style online RL and carries its k-way structure through to the policy update. GPRL computes group-relative advantages per subspace, normalizes each on its own scale as MO-GRPO and GDPO do for pre-specified rewards, and aggregates them with per-dimension weights into an advantage that replaces GRPO's scalar in an otherwise identical objective. A policy that gains on one axis at the expense of the others then sees its aggregate held in check by the rest. The same structure exposes a signal that scalar pipelines lack. A *drift monitor* tracks how score variance is distributed across dimensions, and when it concentrates on one axis, say verbosity, a controller downweights that axis and tightens the KL trust region, shifting the incentive back toward neglected axes such as factuality.

Contributions. **(i)** Building on the online preference-RM setting of RLMT, we propose GPRL, which trains a single online policy against a multi-dimensional reward whose latent axes are learned from binary preference labels rather than designed, and which recovers GRPO at $k{=}1$. **(ii)** We characterize when the aggregate advantage rejects a single-axis exploit that a scalar reward ranking it higher would reinforce, and find that this condition holds for about 80% of applicable response pairs throughout training. **(iii)** We design a closed-loop drift monitor that flags reward hacking from the shape of the per-dimension variance profile and corrects it online by reweighting dimensions and tightening the KL trust region, without any external evaluation signal. **(iv)** GPRL reaches 56.51% and 74.31% length-controlled win rate on AlpacaEval 2.0 from `Llama-3-8B-Instruct` and `Gemma-2-9B-it`, beating GRPO with a BT reward model by 14.59 and 12.56 points, and leads DPO, SimPO, SPPO, GPO, and GRPO+BT on all four benchmarks for both base policies. Ablations show that GPM's relational representation adds 5.67 points over a matched transitive multi-head control, and that the controller is worth 3.67 points at epoch 3 and 8.65 by epoch 5.

2 Background and Related Work

2.1 Online RL with group-relative advantages

Given a prompt x, GRPO (Shao et al., 2024) samples G responses $\{y_1, \ldots, y_G\}$ from $\pi_{\theta_{\text{old}}}(\cdot \mid x)$, scores each with a reward R, and normalizes within the group to form advantages $\hat{A}_i = \big(R(x, y_i) - \text{mean}_j\, R(x, y_j)\big)/\big(\text{std}_j\, R(x, y_j) + \epsilon\big)$, replacing PPO's learned value function with this group baseline. With importance ratio $r_i = \pi_\theta(y_i \mid x)/\pi_{\theta_{\text{old}}}(y_i \mid x)$ and its clipped counterpart $\bar{r}_i = \text{clip}(r_i, 1{-}\epsilon_{\text{clip}}, 1{+}\epsilon_{\text{clip}})$, GRPO minimizes

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{x,\{y_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(r_i\hat{A}_i, \bar{r}_i\hat{A}_i\right) - \beta\text{KL}\big(\pi_\theta \| \pi_{\text{ref}}\big)\right], \tag{1}$$

where β controls KL regularization toward a reference policy π_{ref}. In RLVR (Guo et al., 2025; Zhang et al., 2025a), R is a binary verifier $R(x, y) \in \{0, 1\}$ that checks correctness, which does not extend to open-ended tasks. RLMT (Bhaskar et al., 2025) applies Eq. 1 to open-ended chat with a scalar BT reward model as R; its contribution, reasoning before responding, is orthogonal to the reward shape that GPRL changes. A scalar reward model, however, reintroduces reward hacking (Gao et al., 2023; Fu et al., 2025; Bhaskar et al., 2025). Existing mitigations either debias the reward model against known biases such as response length (Chen et al., 2024; Park et al., 2024; Wang et al., 2025a) or cap the optimization budget, and neither changes the shape of the proxy (Appendix B).

Multiple rewards. When several rewards are summed before Eq. 1, they interfere. MO-GRPO (Ichihara et al., 2026) shows that the highest-variance reward dominates the advantage, and GDPO (Liu et al., 2026) shows that distinct reward combinations collapse to identical advantages. Both therefore normalize each reward within the group before summing, the same step GPRL adopts in Eq. 4. Their rewards, however, are specified in advance as rule-based checks or separately trained reward models. GPRL instead takes its dimensions from a GPM trained on binary preference labels and adapts their weights online, and Appendix D.8 shows that the same normalization and controller applied to independently trained scalar heads fall short of GPRL. Multi-objective RLHF (Wu et al., 2023; Rame et al., 2023; Zhou et al., 2024; Xiong & Singh, 2025; He & Maghsudi, 2025) likewise starts from pre-specified axes and combines separately learned rewards into a Pareto family of policies, with no per-step signal for detecting single-axis exploitation within a run.

2.2 General Preference Model and preference optimization

The BT model (Bradley & Terry, 1952) sets $\mathbb{P}(y_i \succ y_j \mid x) = \sigma(R(x, y_i) - R(x, y_j))$ with σ the logistic function and $R(\cdot) \in \mathbb{R}$ a scalar reward. This imposes a total ordering, so it cannot represent intransitive preferences such as $A \succ B \succ C \succ A$ that appear in human judgments (Fishburn, 1982; 1991). The classical fix is a *skew-symmetric bilinear* form, which admits such cycles by construction (Fishburn, 1981; Nakamura, 1998), and GPM (Zhang et al., 2025b) instantiates this form with learned response embeddings.

Preference embeddings. GPM maps each response y to an embedding $\mathbf{v}_{y|x} \in \mathbb{R}^{2k}$ with $\|\mathbf{v}_{y|x}\|_2 = 1$ and scores pairs through a fixed block-diagonal skew-symmetric operator $\mathbf{R}^{\succ} \in \mathbb{R}^{2k \times 2k}$ made of k blocks $\mathbf{R}_l = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$. Writing the embedding components as $(v_i^{(1)}, \ldots, v_i^{(2k)})$, the per-subspace score is

$$s_l(y_i, y_j \mid x) = v_i^{(2l)} v_j^{(2l-1)} - v_i^{(2l-1)} v_j^{(2l)}, \tag{2}$$

and the overall score aggregates the subspaces with prompt-dependent weights $\lambda_l(x) \geq 0$ produced by a learned gate on the prompt encoding (Zhang et al., 2025b),

$$s(y_i \succ y_j \mid x) = \sum_{l=1}^{k} \lambda_l(x) s_l(y_i, y_j \mid x), \quad \mathbb{P}(y_i \succ y_j \mid x) = \sigma\big(s(y_i \succ y_j \mid x)\big). \tag{3}$$

Since Eq. 2 is antisymmetric in (i, j), so is the aggregate score, and the resulting preference probabilities define a constant-sum two-player game (Wu et al., 2024b; Munos et al., 2024).

Preference optimization and GPO. DPO (Rafailov et al., 2023) reparameterizes the RLHF objective (Ouyang et al., 2022) as a closed-form classification loss, which led to offline and iterative variants such as IPO (Azar et al., 2024), KTO (Ethayarajh et al., 2024), and SimPO (Meng et al., 2024). SPPO (Wu et al., 2024b) and Nash-MD (Munos et al., 2024) go beyond BT by iteratively solving such constant-sum preference games, but their preference models still return a single scalar win probability per pair. Zhang et al. (2025b) pair GPM with an SPPO-style scheme called General Preference Optimization (GPO). Given an opponent policy μ, GPO samples K responses per prompt, estimates $\hat{s}(y_i \succ \mu \mid x) = \frac{1}{K} \sum_{m=1}^{K} s(y_i \succ y_m \mid x)$, and regresses $\log \pi_\theta / \pi_{\theta_t}$ onto these scores at each iteration t. Like SPPO and iterative DPO, GPO fixes π_{θ_t}, trains to convergence on the collected batch, and only then refreshes the rollout policy, so its improvement is limited by the quality of the collected data rather than by optimization compute (Song et al., 2024; Wang et al., 2025b). GPRL keeps GPM as the reward source but samples fresh responses from π_θ at every step, keeps the k subspaces separate through advantage estimation instead of collapsing them to $\hat{s}$, and reweights them online when the policy starts to exploit a single one.

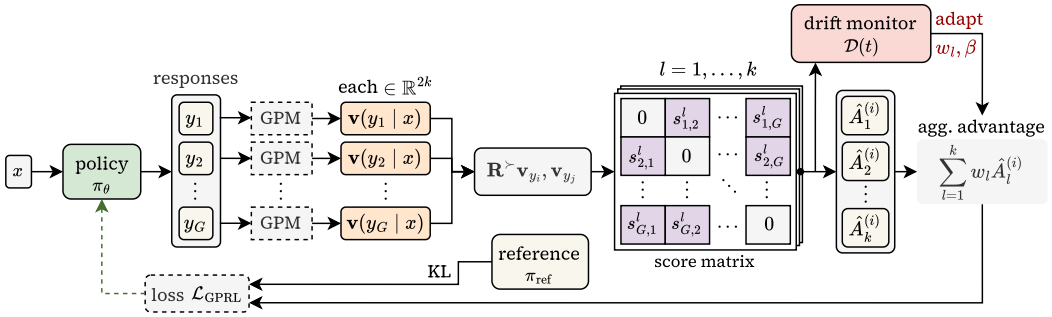


Figure 2: **Overview of GPRL.** The policy π_θ samples G responses per prompt, GPM embeds them, and $\mathbf{R}^{\succ}$ produces k pairwise score matrices that yield per-dimension advantages. The aggregate drives the GRPO-style clipped surrogate, while a drift monitor $\mathcal{D}(t)$ adapts the dimensional weights and β to suppress reward hacking.

3 General Preference Reinforcement Learning

GPRL replaces GRPO's scalar reward with GPM's multi-dimensional preference signal and carries the k-way structure through to the policy update. Collapsing the k subspace scores into a single number, as GPO does, would discard the very structure that makes GPM worth using, so GPRL keeps each subspace separate during advantage estimation, normalizes each on its own scale, and only then aggregates.

3.1 Per-dimension advantage estimation and GPRL objective

Given a prompt x, GPRL samples G responses $\{y_1, \ldots, y_G\}$ from $\pi_{\theta_{\text{old}}}$ and passes them through a frozen GPM, which returns embeddings $\mathbf{v}_{y_i|x} \in \mathbb{R}^{2k}$. We then build the advantage in three steps. First, we form the k pairwise score matrices using $s_l(y_i, y_j \mid x)$ from Eq. 2, as shown in Figure 2. Averaging each row against the rest of the group yields a per-dimension population score $\hat{s}_l^{(i)}(x) = \frac{1}{G-1} \sum_{j \neq i} s_l(y_i, y_j \mid x)$ that measures how much y_i beats the rest of the group along dimension l. Second, since the subspaces differ in how widely their scores spread across a group, we normalize within each dimension rather than across all of them,

$$\hat{A}_l^{(i)}(x) = \frac{\hat{s}_l^{(i)}(x) - \mu_l(x)}{\sigma_l(x) + \epsilon}, \quad \hat{A}_l^{(i)}(x) = 0 \text{ if } \sigma_l(x) = 0, \tag{4}$$

where $\mu_l(x)$ and $\sigma_l(x)$ are the mean and standard deviation of $\{\hat{s}_l^{(i)}(x)\}_{i=1}^{G}$. A single global normalization would let whichever subspace carries the largest spread drown out the rest, whereas per-dimension rescaling places every signal on a common unit-variance footing. Finally, we combine the rescaled advantages with per-dimension weights $w_l \geq 0$ to form the aggregate advantage,

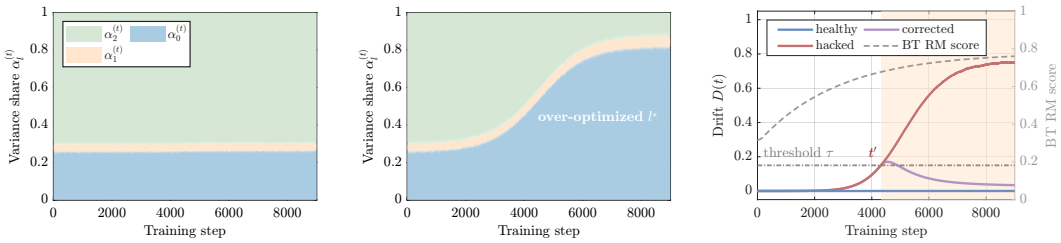
$$\hat{A}^{(i)}(x) = \sum_{l=1}^{k} w_l \, \hat{A}_l^{(i)}(x). \tag{5}$$

Plugging Eq. 5 into a GRPO-style clipped surrogate gives the GPRL objective,

$$\mathcal{L}_{\text{GPRL}}(\theta) = -\mathbb{E}_{x,\{y_i\}} \left[\frac{1}{G} \sum_{i=1}^{G} \min\left(r_i \hat{A}^{(i)}, \bar{r}_i \hat{A}^{(i)} \right) - \beta \mathrm{KL}\left(\pi_\theta \| \pi_{\text{ref}}\right) \right], \tag{6}$$

where r_i and $\bar{r}_i$ match those of Eq. 1. Since Eq. 6 differs from the GRPO objective only in how $\hat{A}^{(i)}$ is computed, GPRL requires no change to rollout infrastructure such as vLLM (Kwon et al., 2023) or to distributed training, and its only added cost is a frozen GPM forward pass per group, comparable to any learned RM call.

Proposition 1 (Zero-mean advantages). *For every prompt x and every dimension l, $\sum_{i=1}^{G} \hat{A}_l^{(i)}(x) = 0$, hence $\sum_{i=1}^{G} \hat{A}^{(i)}(x) = 0$ for any weights $w_l \in \mathbb{R}$.*



(a) Healthy run (b) Hacked run (c) Drift trajectory

Figure 3: **Illustration of dimensional drift under healthy training and reward hacking.** (a) The variance profile $\alpha^{(t)}$ holds its initial shape on a healthy GPRL run. (b) Under hacking, it collapses onto a single dimension $l^\star$. (c) $\mathcal{D}(t)$ stays near zero on the healthy run and crosses τ at t' on the hacked one, allowing the *corrected* trajectory to engage the controller at t' and pull back as the profile rebalances, while a BT reward model climbs monotonically and offers no comparable signal.

Proposition 1 implies that aggregating k subspace advantages preserves the zero-mean, group-relative baseline that makes GRPO well-behaved (Shao et al., 2024), so we are free to mix dimensions through any weights, including the controller's time-varying ones, without rederiving anything.[1] Proofs for this and what follows appear in Appendix C.

3.2 Drift control

The aggregate advantage of Eq. 5 acts as a lever the policy gradient pulls on, so understanding why a multi-dimensional reward resists reward hacking requires asking what it takes for the lever to flip the wrong way. Suppose a candidate policy $\pi^\dagger$ improves a single axis l^* relative to a balanced policy π^* while regressing on the rest. The gradient direction depends on which policy carries the larger aggregate advantage, and Proposition 2 pins down when GPRL prefers the balanced policy.

Proposition 2 (Multi-dimensional structure rejects single-axis hacking)**.** *Fix a prompt x and a group of responses, and let $\hat{A}_l^\pi(x)$ denote the dimension-l advantage of a response drawn from π, as in Eq. 4. Let π^* and $\pi^\dagger$ satisfy, for some axis l^*, $\hat{A}_{l^*}^{\pi^\dagger}(x) > \hat{A}_{l^*}^{\pi^*}(x)$ and $\hat{A}_l^{\pi^\dagger}(x) < \hat{A}_l^{\pi^*}(x)$ for all $l \neq l^*$. If*

$$\sum_{l \neq l^*} w_l \big(\hat{A}_l^{\pi^*}(x) - \hat{A}_l^{\pi^\dagger}(x)\big) > w_{l^*}\big(\hat{A}_{l^*}^{\pi^\dagger}(x) - \hat{A}_{l^*}^{\pi^*}(x)\big), \tag{7}$$

then $\hat{A}^{\pi^\dagger}(x) < \hat{A}^{\pi^*}(x)$.

Interpretation. $\hat{A}^{\pi^\dagger}(x) < \hat{A}^{\pi^*}(x)$ is the consequential half, since GPRL then gives the smaller aggregate advantage to the single-axis exploiter and the clipped surrogate in Eq. 6 pushes the policy gradient away from $\pi^\dagger$ and toward π^*. Under any scalar reward R that scores the response from $\pi^\dagger$ above the one from π^*, by contrast, the GRPO advantage is monotone in R within a fixed group, so $\pi^\dagger$ always wins and the gradient always rewards the exploit. The per-dimension normalization in Eq. 4 makes Eq. 7 more likely to hold, since rescaling every $\hat{A}_l^{(i)}$ to unit variance bounds how much any single dimension can grow its contribution to the aggregate. When the balance nonetheless tips toward one axis, the controller below intervenes on the aggregation weights that govern Eq. 7.

While Eq. 7 can be verified response by response, acting online requires a scalar that summarizes the balance across a training step. We use the variance profile of the per-dimension scores $\hat{s}_l^{(i)}$ before normalization, since Eq. 4 fixes the variance of every normalized advantage. When a policy starts reward hacking l^*, the between-response score variance concentrates on l^* and collapses on the rest,

[1] Fixing $k = 1$ and $w_1 = 1$ and defining $R_{\mathrm{GPM}}(x, y_i) := \hat{s}_1^{(i)}(x)$ collapses Eq. 5 to the GRPO advantage under R_{GPM} and Eq. 6 to Eq. 1, so GPRL strictly generalizes GRPO under a preference-model-based scalar reward, and any improvement we observe with $k > 1$ is attributable to the multi-dimensional structure rather than a different optimizer.

which is exactly the imbalance the controller has to push back on. Define

$$\alpha_l^{(t)} = \frac{\mathrm{Var}_{y_i \sim \pi_t}\left[\hat{s}_l^{(i)}(x)\right]}{\sum_{l'=1}^{k} \mathrm{Var}_{y_i \sim \pi_t}\left[\hat{s}_{l'}^{(i)}(x)\right]}, \quad l = 1, \ldots, k, \quad \mathcal{D}(t) = \mathrm{KL}\big(\alpha^{(t)} \,\|\, \alpha^{(0)}\big), \tag{8}$$

with variances averaged over the prompts in step t and $\alpha^{(t)} = \mathbf{1}/k$ in the degenerate all-zero case. The initial profile $\alpha^{(0)}$ serves as a change-detection reference rather than a quality target. In healthy training $\alpha^{(t)}$ holds close to its initial shape, but under hacking it spikes on the exploited axis, causing $\mathcal{D}(t)$ to rise accordingly. We feed $\mathcal{D}(t)$ back into a closed-loop controller that retunes the aggregation weights and the KL coefficient, and Figure 3 sketches the qualitative drift trajectories it acts on.

Controller. The controller acts through a per-dimension multiplier $m_l(t) \geq 0$, initialized to 1 and applied on top of the base weights, so the effective weight becomes $\tilde{w}_l(t) = m_l(t)\, w_l$. While $\mathcal{D}(t) > \tau$ for threshold τ, the controller tightens according to

$$m_l \leftarrow \mathcal{N}\left[m_l \cdot \left(\frac{\alpha_l^{(0)}}{\alpha_l^{(t)} + \varepsilon}\right)^{\gamma}\right], \quad \beta \leftarrow \min\left(\kappa \cdot \beta, \beta_{\max}\right), \tag{9}$$

where $\mathcal{N}[\cdot]$ renormalizes the multipliers to mean 1, $\gamma \in (0, 1]$ sets the redistribution strength, and $\kappa > 1$ tightens the trust region. The update thus targets only the dimensions that need adjusting, since over-grown axes receive $m_l < 1$ and stagnated axes receive $m_l > 1$, so the aggregate steers back toward balance while the increase in β slows further excursions in the meantime. Once $\mathcal{D}(t) \leq \tau$, the controller relaxes via $m_l \leftarrow \delta m_l + (1 - \delta)$ and $\beta \leftarrow \max(\beta_0, \beta \cdot \delta)$, where β_0 is the initial KL coefficient and $\delta \in (0, 1)$ is the recovery rate, allowing both quantities to decay toward their baselines as the profile rebalances.

4 Experiments

Setup. Every method starts from the same base policy, either `Llama-3-8B-Instruct` or `Gemma-2-9B-it` (Team et al., 2024), so any difference in scores within a base policy reflects the optimization signal rather than pretraining. To separate supervision structure from supervision content, we train a scalar BT model and a GPM with $2k = 6$ on the same `Skywork-Reward` corpus (Liu et al., 2024), each at two scales (`Gemma-2B-it` for 2B and `Llama-3.1-8B-Instruct` for 8B), and use the 8B models on `Gemma-2-9B-it`. The policy trains with online rollouts on `UltraFeedback` prompts (Cui et al., 2023) using $G = 8$ generations per prompt and $\beta = 0.01$ for three epochs, which matches the three-iteration protocol of SPPO and GPO in compute and reaches the regime where the drift controller engages, with all remaining hyperparameters listed in Table 11.

Benchmarks and baselines. AlpacaEval 2.0 (Dubois et al., 2024) reports length-controlled win rate (LC. WR), which explicitly debiases against verbosity, and raw WR against `gpt-4-turbo` on 805 prompts. Arena-Hard v2 (Li et al., 2024) draws 500 adversarial queries from Chatbot Arena traffic, WildBench (Lin et al., 2024) scores 1024 in-the-wild instructions with a calibrated WB-Score and a task-balanced WB-Reward, and MT-Bench (Zheng et al., 2023) covers 80 multi-turn prompts. All four are judged automatically, so the same length and stylistic biases apply across methods (Gu et al., 2024). Our baselines span both post-training tracks, namely DPO (Rafailov et al., 2023) and SimPO (Meng et al., 2024) on offline pairs, SPPO (Wu et al., 2024b) and GPO (Zhang et al., 2025b) as iterative game-theoretic methods reported at iteration 3, their strongest published setting, and GRPO (Shao et al., 2024) with a scalar BT reward model, which shares everything with GPRL except the multi-dimensional reward and is reported at the matched third epoch.

Main results. Tables 1, 2, and 3 show three patterns that repeat across all four benchmarks and both base policies. *(a) Static and iteratively refreshed methods plateau early.* DPO and SimPO sit at 40 to 45 LC. WR on AlpacaEval 2.0 with `Llama-3-8B-Instruct` and 62 to 67 with `Gemma-2-9B-it`, while SPPO and GPO at iteration 3 stay in or below the same band and cluster similarly on the other three benchmarks.[2] *(b) Online RL with a scalar BT reward does not close the gap.* Despite sampling fresh responses at every step, GRPO+BT trails SimPO on every benchmark for both base policies, reaching 41.92 against 44.70 LC. WR on `Llama-3-8B-Instruct` and 61.75

[2] AH2 absolute scores are low for both base policies, which are not strong on adversarial reasoning regardless of post-training, but the ordering we test is unchanged on `Gemma-2-9B-it` despite its higher absolute scores.

Table 1: **AlpacaEval 2.0.** LC. WR, WR, and average response length against the `gpt-4-turbo` reference with `gpt-4-turbo` as judge, on two base policies. SPPO and GPO follow the published three-iteration protocol, while GRPO and GPRL are reported at the matched third epoch so that all reward-model-based methods are compared at similar compute.

Group	Method	RM Size	RM Type	Iter./Ep.	LC. WR	WR	Avg. Len
Base policy: `Llama-3-8B-Instruct`							
No reward model	DPO (Rafailov et al.)	–	–	–	40.30	37.90	1837
	SimPO (Meng et al.)	–	–	–	44.70	40.50	1825
Reward model	SPPO (Wu et al.)	2B	BT	3	40.01	42.12	2136
		2B	GPM	3	36.06	45.61	2498
		8B	BT	3	42.55	40.92	1948
		8B	GPM	3	39.45	41.64	2385
	GPO (Zhang et al.)	2B	BT	3	42.21	44.20	2151
		2B	GPM	3	37.74	48.25	2582
		8B	BT	3	40.37	38.56	1969
		8B	GPM	3	38.98	41.54	3249
Online RL with reward model	GRPO (Shao et al.)	2B	BT	3	39.87	38.21	1925
		8B	BT	3	41.92	40.51	1893
	GPRL (ours)	2B	GPM	3	51.08	45.21	1699
		8B	GPM	3	**56.51**	**48.33**	**1600**
Base policy: `Gemma-2-9B-it`							
No reward model	DPO (Rafailov et al.)	–	–	–	62.10	60.80	1985
	SimPO (Meng et al.)	–	–	–	67.20	63.40	1845
Reward model	SPPO (Wu et al.)	8B	BT	3	60.84	61.32	2012
		8B	GPM	3	61.20	62.75	2329
	GPO (Zhang et al.)	8B	BT	3	59.45	60.18	2045
		8B	GPM	3	60.35	62.19	2452
Online RL with reward model	GRPO (Shao et al.)	8B	BT	3	61.75	58.20	1940
	GPRL (ours)	8B	GPM	3	**74.31**	**67.42**	**1791**

against 67.20 on `Gemma-2-9B-it`. This is consistent with a scalar proxy collapsing onto whichever axis it tracks most sensitively. *(c) Replacing the scalar reward with GPM under the same online paradigm produces the largest jump.* GPRL reaches 56.51 LC. WR on `Llama-3-8B-Instruct`, ahead of GRPO+BT by 14.59 points and of the strongest iterative baseline (SPPO with 8B BT) by 13.96. On `Gemma-2-9B-it` it reaches 74.31, ahead of GRPO+BT by 12.56 and of the strongest baseline, SimPO, by 7.11, and the same ordering holds on the other three benchmarks.

Response lengths also follow a consistent pattern in Table 1. Since the `gpt-4-turbo` judge favors longer responses (Dubois et al., 2024; Zheng et al., 2023; Shi et al., 2025), iterative GPM-driven methods inflate to 2300 to 3300 tokens by iteration 3 even as their LC win rates stall, whereas GPRL holds length at 1600 and 1791 tokens on the two base policies, the shortest of any method in either block. This is what per-dimension normalization (Eq. 4) is designed to produce, since after rescaling no single subspace can dominate the aggregate by growing in magnitude, so a length-correlated axis cannot pull the policy gradient on its own, and the drift controller reinforces this by reweighting whichever subspace's variance share grows past its initial profile. GPRL thus delivers the strongest LC. WR at the shortest length on both base policies, rather than trading one for the other.

4.1 Analysis and Ablations

Per-category gains and extended training. Though neither `Skywork-Reward` nor `UltraFeedback` targets structured reasoning, GPRL's largest single-category gains in Figure 4 land on structural categories, namely MT-Bench coding (+1.00 over the strongest iterative baseline) and WildBench math/data (+2.84 over GRPO+BT), with a comparable lead on extraction and a tie with SPPO on STEM. This matches reports that online RL with group-relative advantages lifts the categories where the reward discriminates well (Wang et al., 2025c; Guo et al., 2025), which GPM's subspaces appear to do on these prompts. Over five epochs (Figure 4a), the controller-on runs hold

Table 2: **Arena-Hard v2 and MT-Bench.** Win rate (WR) on Arena-Hard v2 hard prompts against the official `o3-mini-2025-01-31` baseline (AH2) and MT-Bench score.

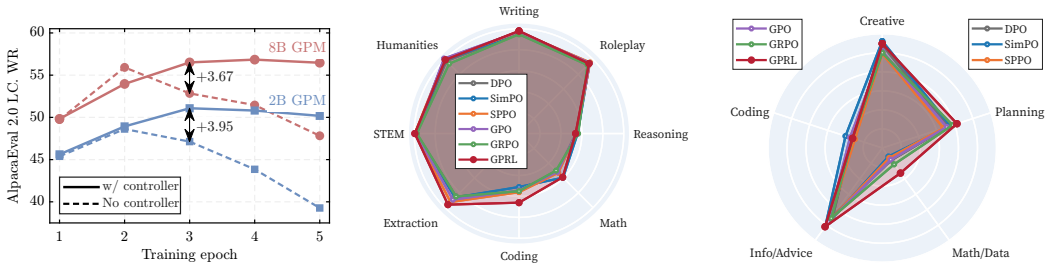
Group	Method	RM Size	RM Type	Iter./Ep.	AH2	MT-Bench
Base policy: `Llama-3-8B-Instruct`						
No reward model	DPO (Rafailov et al.)	–	–	–	0.9	7.90
	SimPO (Meng et al.)	–	–	–	1.0	8.03
Reward model	SPPO (Wu et al.)	8B	BT	3	0.9	8.02
		8B	GPM	3	0.9	8.12
	GPO (Zhang et al.)	8B	BT	3	0.8	7.87
		8B	GPM	3	0.9	8.03
Online RL with reward model	GRPO (Shao et al.)	8B	BT	3	0.8	7.69
	GPRL (ours)	8B	GPM	3	**1.3**	**8.33**
Base policy: `Gemma-2-9B-it`						
No reward model	DPO (Rafailov et al.)	–	–	–	2.2	8.82
	SimPO (Meng et al.)	–	–	–	2.4	8.89
Reward model	SPPO (Wu et al.)	8B	BT	3	2.2	8.84
		8B	GPM	3	2.0	8.90
	GPO (Zhang et al.)	8B	BT	3	2.1	8.80
		8B	GPM	3	2.2	8.88
Online RL with reward model	GRPO (Shao et al.)	8B	BT	3	2.2	8.79
	GPRL (ours)	8B	GPM	3	**2.7**	**9.08**

Table 3: **WildBench.** Calibrated WB-Score and task-balanced WB-Reward against the three official baseline references (`gpt-4-turbo`, `claude-3-haiku`, `llama-2-70b-chat`), same setup as Table 1.

Group	Method	RM Size	RM Type	Iter./Ep.	WB-Score	WB-Reward
Base policy: `Llama-3-8B-Instruct`						
No reward model	DPO (Rafailov et al.)	–	–	–	35.90	8.16
	SimPO (Meng et al.)	–	–	–	37.05	9.57
Reward model	SPPO (Wu et al.)	8B	BT	3	36.14	8.33
		8B	GPM	3	35.21	7.18
	GPO (Zhang et al.)	8B	BT	3	35.87	8.25
		8B	GPM	3	35.98	7.42
Online RL with reward model	GRPO (Shao et al.)	8B	BT	3	35.41	8.62
	GPRL (ours)	8B	GPM	3	**37.98**	**11.15**
Base policy: `Gemma-2-9B-it`						
No reward model	DPO (Rafailov et al.)	–	–	–	53.21	28.36
	SimPO (Meng et al.)	–	–	–	54.89	29.12
Reward model	SPPO (Wu et al.)	8B	BT	3	52.54	28.31
		8B	GPM	3	51.80	27.60
	GPO (Zhang et al.)	8B	BT	3	51.40	27.67
		8B	GPM	3	50.17	27.90
Online RL with reward model	GRPO (Shao et al.)	8B	BT	3	53.11	28.39
	GPRL (ours)	8B	GPM	3	**56.79**	**31.04**

within one point of their peak at both reward-model scales, while the controller-off runs degrade once drift develops and finish 8.65 and 10.91 points lower on the 8B and 2B GPMs (Appendix D.4).

Ablations. Appendix D provides the full results, and we summarize the main findings here. (i) *Number of subspaces k.* LC. WR climbs from 44.21 at $k = 1$, where GPRL recovers GRPO under a BT-style scalar reward derived from a single GPM block, to 56.51 at $k = 3$, before plateauing and slightly regressing by $k = 6$. Where exactly it plateaus is a property of the supervision rather



(a) Extended-training scaling (b) MT-Bench, per-category (c) WildBench, per-category

Figure 4: **Scaling and per-category breakdown** (`Llama-3-8B-Instruct`). (a) AlpacaEval 2.0 LC. WR over five epochs at both reward-model scales, with and without the drift controller. (b, c) Per-category scores on MT-Bench and WildBench.

than of GPRL, since a GPM trained on `HelpSteer2-Preference` (Wang et al., 2025d) plateaus at $k = 2$ instead, with one subspace tracking overall quality and the other tracking style through verbosity and complexity (Appendix D.9). (ii) *Per-dimension vs. global normalization.* Pooling (μ, σ) across subspaces instead of Eq. 4 costs roughly 4 LC. WR points and inflates length to 2104 tokens, recovering the verbose pattern of the iterative GPM methods. (iii) *Drift controller.* The controller changes LC. WR by at most 0.23 points in the first epoch, where the variance profile stays close to its initial shape, but becomes load-bearing as the profile starts leaning on a single axis, contributing 3.67 points on the 8B GPM and 3.95 on the 2B by epoch 3. (iv) *Drift threshold* τ. Any $\tau \in [0.1, 0.4]$ stays within 1.3 points of the default, whereas a very low τ underperforms even the disabled controller because it suppresses legitimate concentration of variance along with the spurious kind. (v) *Group size* G. Larger groups sharpen the variance estimates the drift monitor relies on, with gains saturating at our default of $G = 8$. (vi) *Transitive control.* Three orthogonality-regularized scalar BT heads, trained on the same corpus and passed through the same normalization and controller, reach 50.84 LC. WR, which is 8.92 points above GRPO+BT but 5.67 below GPRL and thus isolates the contribution of GPM's relational representation (Appendix D.8). Appendix E further shows that the frozen GPM loses only 2.9 points of on-policy pairwise accuracy over five epochs and that the condition of Proposition 2 holds for about 80% of applicable response pairs throughout training.

Limitations. GPRL inherits the assumptions of the GPM it consumes, so if the preference model is poorly calibrated on a quality axis or does not represent it as a subspace, GPRL cannot recover the missing signal. Its subspaces are also latent rather than identified, since with uniform weights the aggregate score is invariant to unitary mixing of the blocks, so the labels alone do not fix which basis per-dimension normalization acts on. Likewise, $\mathcal{D}(t)$ detects concentration of score variance rather than degradation of externally measured quality, which Appendix E.1 verifies only in aggregate, and all benchmarks rely on automatic judges. The controller is a feedback loop without convergence guarantees, so pathological choices of (γ, κ, τ) can produce oscillations in $\mathcal{D}(t)$. Finally, although we evaluate on two base policies and two reward corpora, all rollouts use `UltraFeedback` prompts, and the math and coding gains are best read as transfer rather than a substitute for RLVR.

5 Conclusion

We argued that the gap between online RL and open-ended alignment is a question of reward shape rather than reward strength. A scalar reward model is an incomplete proxy for multi-dimensional human quality, and online RL against it tends to collapse onto whichever axis the proxy is most sensitive to. GPRL instead carries GPM's k-subspace preference signal through to the policy update. This discourages single-axis exploitation whenever the remaining axes outweigh the exploited one, and it gives the controller a signal to correct exploitation when that balance starts to tip. Empirically, GPRL achieves the strongest length-controlled win rates at the shortest response lengths on both base policies, outperforms a matched control built from transitive reward heads, and holds near its peak under extended training where the uncontrolled variant degrades. More broadly, supervision structure can be treated as a design variable rather than a fixed property of the loss, which may prove useful wherever a learned proxy stands in for a target that cannot be measured directly.

AI USE STATEMENT

ChatGPT by OpenAI and Claude by Anthropic were used for minor editing, proofreading, and resolving LaTeX formatting issues in this document. Codex by OpenAI was used to assist with code writing in the implementation of gprl. The authors reviewed and verified all AI-assisted content and take full responsibility for the paper.

ETHICS STATEMENT

This work uses publicly available models and benchmarks and involves no human subjects, personally identifiable information, or new data collection.

REPRODUCIBILITY STATEMENT

GPRL is implemented as a modification of the GRPOTrainer in the publicly available trl library (von Werra et al., 2020), as described in Appendix F, and uses only publicly available models, datasets, and benchmarks. The method is specified in Section 3, with the per-dimension advantage estimation and training objective in Section 3.1 and the drift monitor and controller in Section 3.2, and complete proofs of all theoretical results are given in Appendix C. Hyperparameters, hardware, and training settings are documented in Appendix F and Table 11, the experimental setup and evaluation protocol in Section 4, and the configuration of every ablation in Appendix D. Within each base policy, all methods start from the same checkpoint, and all reward-model-based methods in the main comparison use reward models trained on the same corpus, as described in Section 4.

REFERENCES

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Adithya Bhaskar, Xi Ye, and Danqi Chen. Language models that think, chat better. *arXiv preprint arXiv:2509.20357*, 2025.

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

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Peter C Fishburn. An axiomatic characterization of skew-symmetric bilinear functionals, with applications to utility theory. *Economics Letters*, 8(4):311–313, 1981.

Peter C Fishburn. Nontransitive measurable utility. *Journal of Mathematical Psychology*, 26(1): 31–67, 1982.

Peter C Fishburn. Nontransitive preferences in decision theory. *Journal of risk and uncertainty*, 4(2): 113–134, 1991.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. Reward shaping to mitigate reward hacking in rlhf. *arXiv preprint arXiv:2502.18770*, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Qi Gou and Cam-Tu Nguyen. Mixed preference optimization: Reinforcement learning with data selection and better reference model. *arXiv preprint arXiv:2403.19443*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Qiang He and Setareh Maghsudi. Pareto multi-objective alignment for language models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 257–272. Springer, 2025.

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. *Transactions of the Association for Computational Linguistics*, 14:1867–1882, 2026.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025.

Thomas Kwa, Drake Thomas, and Adrià Garriga-Alonso. Catastrophic goodhart: regularizing rlhf with kl divergence does not mitigate heavy-tailed reward misspecification. *Advances in Neural Information Processing Systems*, 37:14608–14633, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*, 2024.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

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, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, et al. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024.

Yutaka Nakamura. Skew-symmetric additive representations of preferences. *Journal of Mathematical Economics*, 30(3):367–387, 1998.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4998–5017, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 292–314, 2025.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*, 2023.

Prasann Singhal, Nathan Lambert, Scott Niekum, Tanya Goyal, and Greg Durrett. D2po: Discriminator-guided dpo with response evaluation models. *arXiv preprint arXiv:2405.01511*, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Yuda Song, Gokul Swamy, Aarti Singh, J Bagnell, and Wen Sun. The importance of online data: Understanding preference fine-tuning via coverage. *Advances in Neural Information Processing Systems*, 37:12243–12270, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, et al. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620*, 2025a.

En Wang, Xingyu Lin, Chenfu Du Su, Zhonghou Lv Bao, Funing Yang, Yuanbo Xu, and Wenbin Liu. Indirect online preference optimization via reinforcement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 538–546, 2025b.

Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. Emergent hierarchical reasoning in llms through reinforcement learning. *arXiv preprint arXiv:2509.03646*, 2025c.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501, 2024.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. Helpsteer2-preference: Complementing ratings with preferences. In *International Conference on Learning Representations*, volume 2025, pp. 56548–56575, 2025d.

Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Alphadpo: Adaptive reward margin for direct preference optimization. *arXiv preprint arXiv:2410.10148*, 2024a.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024b.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033, 2023.

Nuoya Xiong and Aarti Singh. Projection optimization: A general framework for multi-objective and multi-group rlhf. *arXiv preprint arXiv:2502.15145*, 2025.

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

Mengyu Zhang, Siyu Ding, Weichong Yin, Yu Sun, and Hua Wu. Extending rlvr to open-ended tasks via verifiable multiple-choice reformulation. *arXiv preprint arXiv:2511.02463*, 2025a.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. Beyond bradley-terry models: a general preference model for language model alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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. *Advances in Neural Information Processing Systems*, 37:75522–75558, 2024.

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

Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned ai. *Advances in Neural Information Processing Systems*, 33:15763–15773, 2020.

APPENDIX

A MORE ON GENERAL PREFERENCE EMBEDDINGS

This appendix expands on the embedding construction that GPRL inherits from GPM (Zhang et al., 2025b), focusing on the structural properties that the per-dimension advantage estimation in Section 3.1 and the drift controller in Section 3.2 actually depend on, and we refer the reader to the original paper for the full derivation.

The choice of an even ambient dimension $\mathbb{R}^{2k}$ is forced by antisymmetry. A scalar reward $r(y \mid x)$ assumes preference is a difference of utilities, which imposes transitivity by construction, so to allow cycles such as $A \succ B \succ C \succ A$, the score must be antisymmetric in (y_i, y_j) without reducing to $r(y_i \mid x) - r(y_j \mid x)$ for any scalar r. Zhang et al. (2025b) show that any operator $\mathbf{R}$ which is both skew-symmetric ($\langle \mathbf{R}v, w\rangle = -\langle \mathbf{R}w, v\rangle$) and magnitude-preserving ($\|\mathbf{R}v\| = \|v\|$) satisfies $\mathbf{R}^2 = -\mathbf{I}$, forcing eigenvalues of $\pm i$ and an even ambient dimension. The block-diagonal canonical form then drops out from the real Schur decomposition, with each 2×2 block contributing one independent cycle direction, which explains both the $\mathbb{R}^{2k}$ ambient space and the rotation-by-90° shape of every $\mathbf{R}_l$ used in Eq. 2.

Within block l, the per-subspace score $s_l(y_i, y_j \mid x) = v_i^{(2l)} v_j^{(2l-1)} - v_i^{(2l-1)} v_j^{(2l)}$ is the signed area of the parallelogram spanned by the two response embeddings, restricted to coordinates $(2l-1, 2l)$. Equivalently, writing the embedding as a complex vector $\mathbf{v} \in \mathbb{C}^k$ with $z_l = v^{(2l-1)} + i\, v^{(2l)}$, the subspace score becomes $s_l(y_i, y_j \mid x) = \mathrm{Im}\big(z_l^{(i)} \bar{z}_l^{(j)}\big)$, so each block measures a relative phase between responses on a learned axis. Two responses with equal phases score zero against each other, and if every response in a group shares the same phase on block l, then $\sigma_l(x) = 0$ and the convention in Eq. 4 sets $\hat{A}_l^{(i)} = 0$. Because the blocks are independent, the cycles they express compose, which lets a single GPM represent intransitivity along several axes at once, although our experiments attribute GPRL's gains to the relational form of the score rather than to such cycles specifically (Appendix D.8).

The unit-norm constraint $\|\mathbf{v}_{y|x}\|_2 = 1$ combined with $\|\mathbf{R}_l\|_2 = 1$ forces $|s_l(y_i, y_j \mid x)| \le 1$ uniformly, so the per-subspace scores live on a fixed $[-1, 1]$ scale regardless of how the policy or rollout distribution shifts during training. This keeps the raw population scores $\hat{s}_l^{(i)}$, and hence the variance profile $\alpha^{(t)}$ computed from them, in fixed units throughout training, so that a change in $\alpha^{(t)}$ reflects how the policy's responses spread rather than a change of scale, and it gives the controller a meaningful reference profile $\alpha^{(0)}$ at step zero. Fine-tuning the GPM during policy training would move this anchor, which is the main practical reason we keep the GPM frozen throughout GPRL. Boundedness buys numerical stability rather than ranking accuracy, however, and Appendix E.1 measures how well the frozen GPM continues to rank on-policy responses as training proceeds.

The case $k = 1$ connects GPRL to BT and scalar GRPO. Setting $k = 1$ and, dropping the unit-norm constraint for this calculation, $\mathbf{v}_{y|x} = (c, r(y \mid x))^\top$ with $c \neq 0$ collapses the GPM score to $s(y_i \succ y_j \mid x) = c\,(r(y_i \mid x) - r(y_j \mid x))$, recovering the BT model up to a constant rescaling (Bradley & Terry, 1952). Under the unit-norm constraint the single block instead gives $s(y_i \succ y_j \mid x) = \sin(\theta_i - \theta_j)$ for $\mathbf{v}_{y|x} = (\cos\theta, \sin\theta)^\top$, which behaves like a BT utility difference for nearby responses but remains antisymmetric rather than transitive in general. Together with the calculation in Appendix C.2, this makes GPRL with $k = 1$ the single-block analogue of GRPO under a preference-model reward, while the $k > 1$ regime carries the additional structure that Sections 3 and 4 exploit. Increasing k does not change query complexity, since all k pairwise scores come from a single GPM forward pass per response followed by an $\mathcal{O}(k)$ inner product per pair, which matches the linear number of forward passes of BT and improves on the $\mathcal{O}(K^2)$ forward passes of supervised pair preference models such as PairRM (Jiang et al., 2023).

B MORE ON REWARD HACKING

This appendix locates GPRL within the existing reward-hacking literature, focusing on the specific failure modes that motivated the per-dimension structure and drift controller of Section 3.

Reward hacking refers to the phenomenon in which a policy increases a learned proxy reward $\hat{R}$ while the underlying quantity $R^\star$ that $\hat{R}$ was meant to estimate stays flat or degrades (Skalse et al., 2022; Fu et al., 2025). Gao et al. (2023) characterized this empirically in RLHF as reward over-optimization, showing that as the policy spends KL budget against a learned RM, the gold reward traces a hill-shaped curve that initially climbs and then falls, with the peak depending on RM size, KL coefficient, and amount of preference data. The same qualitative shape, namely a peak followed by sustained degradation, is what we observe in Appendix D.4 for GPRL with the controller disabled, which is the empirical signature the controller is designed to interrupt.

Three contributing factors explain the hill-shaped curve in scalar RLHF. First, Goodhart's law, which Gao et al. (2023) characterize for reward models, whereby any optimization pressure against an imperfect proxy eventually finds the gap between proxy and target, and the larger the optimization budget the wider the gap that gets exploited. Second, spurious correlation in the supervision corpus, which gives the RM surface features that correlate with quality on the training distribution but generalize poorly. Response length is the canonical example, since longer responses are systematically preferred in many preference datasets and judges, so the RM learns that long loosely implies good and the policy then exploits this by inflating length without improving content (Singhal et al., 2023; Chen et al., 2024; Park et al., 2024), while other documented spurious axes include sycophancy (Sharma et al., 2023), stylistic mimicry, and position or formatting biases in LLM-as-judge evaluation that propagate into RM training (Zheng et al., 2023; Dubois et al., 2024). Third, heavy-tailed misspecification, which compounds these effects because a scalar RM has no degree of freedom to express that several axes should improve simultaneously, so once the policy finds the dominant axis the gradient stays pointed at it for the rest of training (Kwa et al., 2024). None of these failure modes is detectable from the proxy alone, since the proxy by construction agrees with the policy that things are improving, which is precisely the asymmetry GPRL's variance profile $\alpha^{(t)}$ is built to expose.

Existing mitigations split into three families, each addressing one part of the problem. Reward model improvements such as ODIN (Chen et al., 2024), length-debiased preference data construction (Park et al., 2024), and causal regularization (Wang et al., 2025a) reduce the proxy gap on a spurious axis the designer anticipates, but they do not change the shape of the proxy. Optimization-side defenses such as KL regularization (Schulman et al., 2017), reward clipping, and early stopping cap the available gain rather than separating clean from dirty progress, since they cannot distinguish a policy that earned its reward from one that hacked it. Multi-objective approaches such as Fine-Grained RLHF (Wu et al., 2023), Rewarded Soups (Rame et al., 2023), and MODPO (Zhou et al., 2024) expose multi-dimensional structure but optimize a scalar combination per policy, so the same hacking dynamics apply within each combination, and while multi-reward GRPO variants such as MO-GRPO (Ichihara et al., 2026) and GDPO (Liu et al., 2026) prevent a high-variance reward from dominating by normalizing each reward separately, all of these methods start from rewards specified in advance and provide no per-step signal for monitoring single-axis exploitation during a run.

GPRL combines elements of all three families while addressing the underlying shape problem directly. The reward is structurally vector-valued through the k subspaces of GPM, the per-dimension normalization in Eq. 4 prevents any one axis from inflating its share of the aggregate by simply growing in magnitude, and Proposition 2 formalizes when the multi-dimensional structure tips the gradient against a single-axis exploiter. The drift controller of Section 3.2 then plays the optimization-side role by adjusting both the dimension weights and the KL coefficient β, but it does so based on the variance profile $\alpha^{(t)}$, which is the per-step signal that scalar pipelines cannot construct. This does not eliminate reward hacking in the broadest sense, since the GPM itself can be miscalibrated on axes outside its supervision and drifts modestly out of calibration over extended training (Appendix E.1), but it changes what the policy can hack from a scalar to a configuration of axes whose imbalance is observable. Tables 6 and 7 measure this effect quantitatively, with the controller acting on the dimensional fingerprint before the policy settles into any single-axis exploit, while Figure 3 illustrates the qualitative drift trajectories involved.

C PROOFS

We collect here the full proofs of the propositions stated in Section 3, together with the calculation that establishes GRPO as the $k = 1$ special case of GPRL.

C.1 ZERO-MEAN PROPERTY OF ADVANTAGES

Proof of Proposition 1. Fix a prompt x and a dimension $l \in \{1, \dots, k\}$, and consider two cases.

When $\sigma_l(x) = 0$. By the convention fixed in Eq. 4, $\hat{A}_l^{(i)}(x) = 0$ for all $i \in \{1, \dots, G\}$, so $\sum_{i=1}^{G} \hat{A}_l^{(i)}(x) = 0$ trivially.

When $\sigma_l(x) > 0$. By definition,

$$\mu_l(x) = \frac{1}{G} \sum_{i=1}^{G} \hat{s}_l^{(i)}(x), \quad \text{so} \quad \sum_{i=1}^{G} \hat{s}_l^{(i)}(x) = G\mu_l(x).$$

Applying Eq. 4 term by term and factoring the common denominator,

$$\begin{aligned} \sum_{i=1}^{G} \hat{A}_l^{(i)}(x) &= \sum_{i=1}^{G} \frac{\hat{s}_l^{(i)}(x) - \mu_l(x)}{\sigma_l(x) + \epsilon} = \frac{1}{\sigma_l(x) + \epsilon} \left(\sum_{i=1}^{G} \hat{s}_l^{(i)}(x) - \sum_{i=1}^{G} \mu_l(x) \right) \\ &= \frac{1}{\sigma_l(x) + \epsilon} \left(G\mu_l(x) - G\mu_l(x) \right) = 0. \end{aligned} \tag{10}$$

Combining both cases, $\sum_{i=1}^{G} \hat{A}_l^{(i)}(x) = 0$ for every l. For the aggregate advantage, linearity of summation in Eq. 5 gives

$$\sum_{i=1}^{G} \hat{A}^{(i)}(x) = \sum_{i=1}^{G} \sum_{l=1}^{k} w_l \, \hat{A}_l^{(i)}(x) = \sum_{l=1}^{k} w_l \sum_{i=1}^{G} \hat{A}_l^{(i)}(x) = \sum_{l=1}^{k} w_l \cdot 0 = 0,$$

where we used that w_l does not depend on i. The conclusion holds for any choice of weights $w_l \in \mathbb{R}$, including prompt-dependent or time-varying ones such as the controller's effective weights $\tilde{w}_l(t)$, since the inner sum vanishes regardless of sign. The aggregate therefore retains the zero-mean, group-relative baseline of GRPO. □

C.2 GRPO AS A SPECIAL CASE OF GPRL

The main text states without a formal proposition that fixing $k = 1$ and $w_1 = 1$ recovers GRPO under a preference-model-based scalar reward, which we verify here. Setting $k = 1$ and $w_1 = 1$, we define the induced scalar reward

$$R_{\mathrm{GPM}}(x, y_i) := \hat{s}_1^{(i)}(x) = \frac{1}{G - 1} \sum_{j \neq i} s_1(y_i, y_j \mid x),$$

which, unlike a standard reward model, is computed against the other responses in the sampled group, and show in turn for a fixed group that (i) the per-dimension normalization reduces to the group-relative normalization of GRPO, (ii) the aggregate advantage equals this normalized quantity, and (iii) the GPRL objective reduces to the GRPO objective.

(i) Normalization. With $k = 1$, Eq. 4 specializes to

$$\hat{A}_1^{(i)}(x) = \frac{\hat{s}_1^{(i)}(x) - \mu_1(x)}{\sigma_1(x) + \epsilon} = \frac{R_{\mathrm{GPM}}(x, y_i) - \mu_1(x)}{\sigma_1(x) + \epsilon},$$

where, by the definitions of $\mu_l(x)$ and $\sigma_l(x)$ in Section 3.1,

$$\mu_1(x) = \frac{1}{G} \sum_{j=1}^{G} \hat{s}_1^{(j)}(x) = \frac{1}{G} \sum_{j=1}^{G} R_{\mathrm{GPM}}(x, y_j) = \mathrm{mean}_j \, R_{\mathrm{GPM}}(x, y_j),$$

$$\sigma_1(x) = \mathrm{std}_j\left[\hat{s}_1^{(j)}(x)\right] = \mathrm{std}_j\left[R_{\mathrm{GPM}}(x, y_j)\right].$$

Substituting these into the expression for $\hat{A}_1^{(i)}(x)$ gives

$$\hat{A}_1^{(i)}(x) = \frac{R_{\mathrm{GPM}}(x, y_i) - \mathrm{mean}_j\, R_{\mathrm{GPM}}(x, y_j)}{\mathrm{std}_j\, R_{\mathrm{GPM}}(x, y_j) + \epsilon},$$

which is precisely the GRPO advantage from Eq. 1 computed under the reward R_{GPM}.

(ii) Aggregation. With $w_1 = 1$, Eq. 5 reduces to

$$\hat{A}^{(i)}(x) = \sum_{l=1}^{1} w_l\, \hat{A}_l^{(i)}(x) = 1 \cdot \hat{A}_1^{(i)}(x) = \hat{A}_1^{(i)}(x),$$

so the aggregate advantage equals the GRPO group-relative advantage of step (i).

(iii) Objective. Comparing Eq. 6 with Eq. 1, the two loss expressions share the same clipped-ratio structure, the same uniform average $\frac{1}{G}\sum_i$, and the same KL-regularization term, with their only formal difference being the symbol $\hat{A}^{(i)}$ versus $\hat{A}_i$ in the clipped surrogate. By step (ii), these two quantities coincide in the present limit. The drift controller also leaves both w_1 and β untouched, since with a single dimension the variance profile is trivially $\alpha^{(t)} = (1)$, so $\mathcal{D}(t) = 0 \le \tau$ at every step and the tightening rule of Eq. 9 never engages. Substitution therefore yields $\mathcal{L}_{\mathrm{GPRL}}(\theta) = \mathcal{L}_{\mathrm{GRPO}}(\theta)$ under the group-dependent reward R_{GPM}, which proves the claim.

C.3 Resistance to single-axis reward hacking

Proof of Proposition 2. We express the gap $\hat{A}^{\pi^\dagger}(x) - \hat{A}^{\pi^*}(x)$ in a form that isolates the contribution of the hacked axis l^*, then apply the hypothesized inequality. Applying Eq. 5 to each policy and using linearity in the per-dimension advantages,

$$\hat{A}^{\pi^\dagger}(x) - \hat{A}^{\pi^*}(x) = \sum_{l=1}^{k} w_l\, \hat{A}_l^{\pi^\dagger}(x) - \sum_{l=1}^{k} w_l\, \hat{A}_l^{\pi^*}(x) = \sum_{l=1}^{k} w_l \big(\hat{A}_l^{\pi^\dagger}(x) - \hat{A}_l^{\pi^*}(x)\big),$$

and splitting the sum into the hacked axis and its complement while factoring signs gives

$$\begin{aligned}\hat{A}^{\pi^\dagger}(x) - \hat{A}^{\pi^*}(x) &= w_{l^*}\big(\hat{A}_{l^*}^{\pi^\dagger}(x) - \hat{A}_{l^*}^{\pi^*}(x)\big) + \sum_{l \neq l^*} w_l \big(\hat{A}_l^{\pi^\dagger}(x) - \hat{A}_l^{\pi^*}(x)\big) \\ &= w_{l^*}\big(\hat{A}_{l^*}^{\pi^\dagger}(x) - \hat{A}_{l^*}^{\pi^*}(x)\big) - \sum_{l \neq l^*} w_l \big(\hat{A}_l^{\pi^*}(x) - \hat{A}_l^{\pi^\dagger}(x)\big). \qquad (*)\end{aligned}$$

Letting $H := w_{l^*}\big(\hat{A}_{l^*}^{\pi^\dagger}(x) - \hat{A}_{l^*}^{\pi^*}(x)\big)$ and $D := \sum_{l \neq l^*} w_l \big(\hat{A}_l^{\pi^*}(x) - \hat{A}_l^{\pi^\dagger}(x)\big)$ denote the weighted gain on the hacked axis and the weighted degradation on the other axes, Eq. $*$ reads

$$\hat{A}^{\pi^\dagger}(x) - \hat{A}^{\pi^*}(x) = H - D.$$

The hypothesis $\hat{A}_{l^*}^{\pi^\dagger}(x) > \hat{A}_{l^*}^{\pi^*}(x)$ together with $w_{l^*} \ge 0$ gives $H \ge 0$, while the hypothesis $\hat{A}_l^{\pi^\dagger}(x) < \hat{A}_l^{\pi^*}(x)$ for every $l \neq l^*$ together with $w_l \ge 0$ makes each summand in D non-negative, so $D \ge 0$. Since the proposition's condition 7 is precisely $D > H$, combining the expression $\hat{A}^{\pi^\dagger}(x) - \hat{A}^{\pi^*}(x) = H - D$ with $D > H$ yields

$$\hat{A}^{\pi^\dagger}(x) - \hat{A}^{\pi^*}(x) = H - D < 0,$$

so $\hat{A}^{\pi^\dagger}(x) < \hat{A}^{\pi^*}(x)$, as claimed. Since $m_l(t) \ge 0$, the same argument applies verbatim to the controller's effective weights $\tilde{w}_l(t) = m_l(t)\, w_l$. □

Contrast with scalar rewards. The main text contrasts this conclusion with the scalar-reward case, and we sketch the argument here for completeness. Under any scalar reward $R(x, y)$ and the GRPO normalization of Eq. 1, the advantage $\hat{A}_i = (R(x, y_i) - \mathrm{mean}_j\, R(x, y_j))/(\mathrm{std}_j\, R(x, y_j) + \epsilon)$ is an affine, strictly increasing function of $R(x, y_i)$ for the fixed group $\{y_j\}$ whenever the group has non-zero spread. Hence, if the response drawn from $\pi^\dagger$ receives a higher reward than the response drawn from π^* within the same group, then $\hat{A}^{\pi^\dagger} > \hat{A}^{\pi^*}$ regardless of how the two responses behave on any latent quality axes. Proposition 2 exhibits the opposite verdict for GPRL whenever $k \ge 2$ and Eq. 7 holds, which is the formal basis for the drift monitor of Section 3.2.

D ADDITIONAL ABLATIONS

This appendix expands on the ablations summarized in Section 4.1. Unless stated otherwise, all runs use `Llama-3-8B-Instruct` as the base policy, the 8B GPM, $G = 8$, $k = 3$, the drift controller with our default $\tau = 0.2$, and the three-epoch training budget that the main results use. We also report 2B GPM numbers for most ablations, since the drift behavior we want to characterize is more visible at the smaller reward-model scale where the variance profile is noisier and single-axis exploitation is correspondingly easier to start.

D.1 NUMBER OF GPM SUBSPACES k

Table 4 sweeps k from 1 to 6 at both reward-model scales, and the $k = 1$ row deserves a clarification before we discuss the trend. With a single skew-symmetric block the GPM score $s_1(y_i, y_j \mid x) = v_i^{(2)} v_j^{(1)} - v_i^{(1)} v_j^{(2)}$ defines a pairwise comparison along one axis, and the per-dimension normalization of Eq. 4 reduces to GRPO's group-relative normalization under the induced scalar reward $R_{\mathrm{GPM}}(x, y_i) = \hat{s}_1^{(i)}(x)$, which is the special case formalized in Appendix C.2 with $w_1 = 1$. This row is GRPO under a BT-style scalar reward derived from the GPM block rather than from a separately trained scalar BT model. Although the two are not numerically identical they land in the same neighborhood (44.21 versus 41.92 for GRPO+BT at the 8B scale, and 39.92 versus 39.87 at 2B), which is a consistency check we want, and the rest of the ablation isolates the effect of k rather than of the reward-model architecture. Adding a single subspace ($k = 1 \to 2$) buys 7 to 8 LC. WR points at either scale and is the largest individual jump in the table, while adding a second ($k = 2 \to 3$) buys roughly another 4 to 5. After this the gain disappears and the run mildly regresses by $k = 6$, which we read as a property of the supervision rather than of GPRL itself. Since `Skywork-Reward` aggregates preferences along a small set of facets, this puts a soft ceiling on how many independently informative subspaces a GPM trained on it can carve out, and subspaces beyond that ceiling pick up noise that the per-dimension normalization then amplifies, which Appendix D.9 confirms by showing that the plateau shifts to $k = 2$ on a second corpus.

Table 4: **Ablation on the number of GPM subspaces k.** AlpacaEval 2.0 LC. WR, raw WR, and average response length, with all other hyperparameters fixed at their main-results values and the drift controller enabled. At $k = 1$ the GPM collapses to a single skew-symmetric block, and GPRL with $w_1 = 1$ recovers GRPO under a BT-style scalar reward R_{GPM}.

Subspaces k	2B GPM			8B GPM		
	LC. WR	WR	Avg. Len	LC. WR	WR	Avg. Len
$k = 1$ (GRPO under R_{GPM})	39.92	38.74	1812	44.21	41.07	1738
$k = 2$	47.30	42.96	1742	51.86	45.92	1671
$k = 3$	**51.08**	**45.21**	**1699**	**56.51**	**48.33**	**1600**
$k = 4$	50.62	44.83	1714	56.18	47.94	1612
$k = 6$	49.81	44.27	1758	55.74	47.41	1648

D.2 PER-DIMENSION VS. GLOBAL NORMALIZATION

Table 5 replaces the per-dimension normalization of Eq. 4 with a single global normalization that pools (μ, σ) across all k subspaces. LC. WR drops by roughly 4 points at both reward-model scales while average response length grows by roughly 470 to 500 tokens, moving toward the verbose pattern of the iterative GPM-based methods. The mechanism is exactly the one Section 3.1 predicts, since whichever subspace happens to carry the largest raw magnitudes drives both μ and σ under global normalization, so the rescaled advantages of the other subspaces collapse toward zero and their contribution to the aggregate vanishes. If that dominant subspace correlates with length, as it tends to under judges that favor longer responses, the policy gradient inherits the correlation and response length climbs.

D.3 DRIFT CONTROLLER

The controller's effect is least visible in any single epoch and most visible across the training trajectory, and Table 6 reports two complementary checkpoints that probe the regime where the

Table 5: **Ablation on the normalization scheme.** Replacing the per-dimension normalization of Eq. 4 with a single global normalization that shares (μ, σ) across all k subspaces. AlpacaEval 2.0 with $k = 3$ and the drift controller enabled, at both reward-model scales.

Normalization	2B GPM			8B GPM		
	LC. WR	WR	Avg. Len	LC. WR	WR	Avg. Len
Global (shared μ, σ)	47.13	44.86	2168	52.34	47.86	2104
Per-dimension (ours)	**51.08**	**45.21**	**1699**	**56.51**	**48.33**	**1600**

controller actually does work. At the first epoch on the 8B GPM the controller is essentially decorative, since LC. WR moves by less than a point and terminal drift $\mathcal{D}(t_{\text{end}})$ stays near 0.02 in either case, well under the default threshold $\tau = 0.2$, which is the right outcome for healthy training and a useful sanity check that the controller does not interfere when it has nothing to push back on. By the third epoch the picture has reversed on both reward models, since the controller is worth 3.67 LC. WR points and 0.92 WB-Score on the 8B GPM and 3.95 and 1.61 on the 2B GPM, with the controller-off run developing measurable drift ($\mathcal{D}(t_{\text{end}}) = 0.18$ on 2B, 0.11 on 8B) while the on-runs end at 0.05 and 0.03, respectively. The 2B GPM accumulates more drift by epoch 3 than the 8B, consistent with its variance estimates being noisier and single-axis exploitation correspondingly easier to start, and Appendix D.4 extends the comparison to epoch 5 and shows that the gap continues to widen.

Table 6: **Drift controller ablation.** Disabling the controller corresponds to $\tau = \infty$ and $m_l(t) \equiv 1$. The first-epoch row on the 8B GPM serves as the no-drift sanity check, while the third-epoch rows capture the regime our main results use, with the 2B GPM included to show that drift is larger under noisier supervision. Bold marks the better value within each on/off pair.

Setting	Controller	LC. WR	WB-Score	Avg. Len	$\mathcal{D}(t_{\text{end}})$
8B GPM, epoch 1	off	49.74	36.71	**1718**	0.02
8B GPM, epoch 1	on	**49.83**	**36.84**	1721	0.02
2B GPM, epoch 3	off	47.13	35.45	1864	0.18
2B GPM, epoch 3	on	**51.08**	**37.06**	**1699**	0.05
8B GPM, epoch 3	off	52.84	37.06	1742	0.11
8B GPM, epoch 3	on	**56.51**	**37.98**	**1600**	0.03

D.4 Extended training

The drift controller is a feedback loop rather than a one-shot adjustment, so the cleanest way to see what it buys us is to run the same configuration with and without it across an extended horizon. Table 7 reports five epochs of training on both reward-model scales while Figure 4a (in Section 4.1) plots the LC. WR trajectories.

Three regimes emerge across the table. Through the first epoch the controller is essentially invisible, since the variance profile sits close to $\alpha^{(0)}$ on either reward model, $\mathcal{D}(t)$ never crosses τ, and the on and off curves are within a point of each other, which matches the design intent of Section 3.2 that the controller should remain dormant when the training signal is healthy. Both controller-off runs then peak at epoch 2 and turn over by epoch 3, while the controller-on runs continue to improve, because whenever the variance profile starts to lean on a single subspace, the controller redistributes weight back toward the rest before the policy gradient can lock in. At epoch 2 the 8B controller-off run is in fact 1.96 points ahead, a gap that reverses by epoch 3 as its drift accumulates. Past epoch 3 the controller-off curves enter sustained degradation, with the 2B run dropping 9.35 LC. WR points between epochs 2 and 5 and the 8B run dropping 8.10 over the same span as $\mathcal{D}(t_{\text{end}})$ on the 8B model climbs from 0.02 at epoch 1 through 0.11 at epoch 3 to 0.27 at epoch 5. The controller-on curves plateau within one point of their peak, with the 8B GPM peaking at 56.84 at epoch 4 and settling at 56.47 by epoch 5, and the 2B GPM peaking at 51.08 at epoch 3 and easing to 50.18 by epoch 5. This residual movement is expected rather than being a sign of instability, since the closed-loop dynamics of $m_l(t)$ produce small fluctuations (an empirical observation) rather than monotone convergence.

Table 7: **LC. WR on AlpacaEval 2.0 across five epochs.** GPRL with the drift controller enabled and disabled, both reward-model scales, all other hyperparameters fixed at their main-results values.

RM	Controller	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
2B GPM	off	45.39	48.62	47.13	43.85	39.27
2B GPM	on	45.62	48.93	**51.08**	50.81	50.18
8B GPM	off	49.74	55.92	52.84	51.46	47.82
8B GPM	on	49.83	53.96	56.51	**56.84**	56.47

D.5 Drift threshold τ

Figure 5 sweeps the drift threshold τ at the 8B GPM with three epochs of training, the regime where the controller actually does work (Appendix D.3), and the shape of the curve answers the natural question of why we do not just set τ as small as possible. At $\tau = 0.05$ the controller engages almost from the first step and continuously redistributes weight away from any subspace whose variance share exceeds the initial profile by even a small margin. Since legitimate optimization concentrates variance on whichever subspace the prompt distribution actually rewards (helpfulness on a summarization prompt, faithfulness on a factual one, and so on), this aggressive correction drags the variance profile back toward the initialization $\alpha^{(0)}$ before any genuine signal can accumulate, which amounts to pinning the dimension weights near their initial values and produces a result 4.07 LC. WR points below our default, with a shorter average length that reflects suppressed learning rather than better length control. As τ grows, the controller gives the policy room to concentrate variance for the right reasons before stepping in, and performance peaks at our default $\tau = 0.2$, beyond which the controller engages too late and the run drifts back toward the no-controller numbers. The default $\tau = 0.2$ therefore balances two distinct failure modes, namely premature collapse of the variance profile at small τ and unchecked single-axis exploitation at large τ, rather than trading off two equally good behaviors.

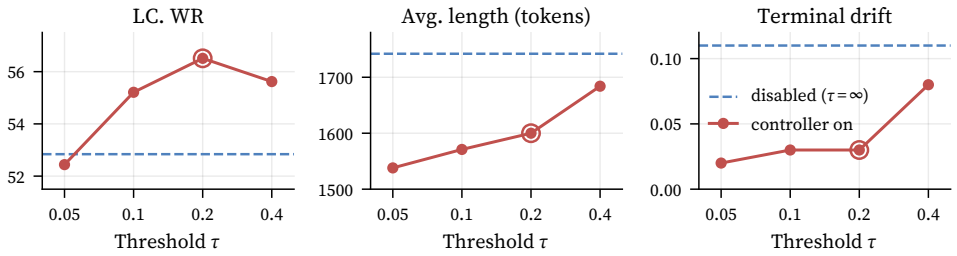


Figure 5: **Ablation on the drift threshold τ.** AlpacaEval 2.0 LC. WR, average response length, and terminal drift $\mathcal{D}(t_{\text{end}})$ with the 8B GPM and three-epoch training, with the default $\tau = 0.2$ circled and the disabled controller ($\tau = \infty$) shown as a dashed line.

D.6 Group size G

Figure 6 sweeps the rollout group size G. Since the drift monitor depends on the per-dimension variance estimates $\mathrm{Var}_{y_i \sim \pi_t}[\hat{s}_l^{(i)}(x)]$ computed from the G responses in the group, the noise of the variance estimate scales as $1/(G-1)$, so at $G = 2$ those estimates are too noisy for the controller to act on reliably and $\mathcal{D}(t)$ runs about three times its $G = 8$ value at both reward-model scales. The LC. WR cost is steep at roughly 9 points below our default, while doubling to $G = 4$ recovers most of the drop but still leaves about 3 points on the table, and gains saturate at $G = 8$ with $G = 16$ producing only a marginal further improvement at twice the rollout cost. We therefore use $G = 8$ throughout the main results.

D.7 Additional base policies

The main text compares all methods on `Llama-3-8B-Instruct` and `Gemma-2-9B-it`. Here we ask whether GPRL's gains persist on two further bases from the open 8B class,

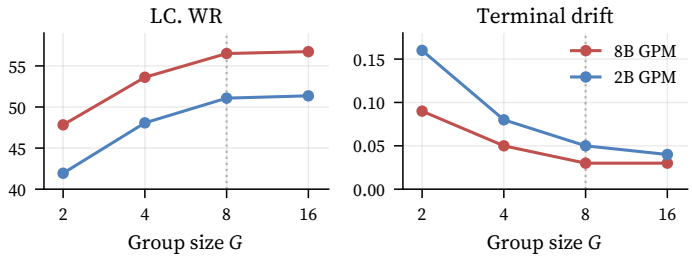


Figure 6: **Ablation on group size G.** AlpacaEval 2.0 LC. WR and terminal drift $\mathcal{D}(t_{\text{end}})$ at both reward-model scales with the controller enabled, with our default $G = 8$ marked by the dotted line.

`Llama-3.1-8B-Instruct` (Grattafiori et al., 2024) and `Qwen3-8B` (Yang et al., 2025), with everything else held at the main-results setting (8B GPM, $G = 8$, $k = 3$, drift controller enabled, three epochs). To put the numbers in context, the AlpacaEval 2.0 LC. WR scores of the bases themselves are roughly 22.6 for `Llama-3-8B-Instruct`, 24.7 for `Llama-3.1-8B-Instruct`, 51.1 for `Gemma-2-9B-it`, and 52.4 for `Qwen3-8B`, so stronger bases leave less headroom for any post-training method.

Table 8 reports GPRL on all four bases, repeating the two main-text rows for reference. We do not compare against published preference-optimization checkpoints on these bases, since they are trained on different preference data and reward annotations (e.g., SimPO's `Gemma-2-9B-it` checkpoint uses on-policy ArmoRM-annotated `UltraFeedback` (Meng et al., 2024)) and their scores are therefore not comparable to ours; Table 1 gives the matched comparison on `Gemma-2-9B-it`. All runs use the same `Skywork-Reward`-trained 8B GPM, so the $k = 3$ plateau is specific to that corpus, as Appendix D.9 shows by moving it to $k = 2$ on `HelpSteer2-Preference`.

Table 8: **GPRL across base policies.** AlpacaEval 2.0 LC. WR, WR, and average response length, with the 8B GPM, $G = 8$, $k = 3$, drift controller enabled, and three epochs of training. The Δ column reports the LC. WR gain over the base policy. The first and third rows repeat Table 1.

Base policy	LC. WR	WR	Avg. Len	Δ
`Llama-3-8B-Instruct` (Grattafiori et al., 2024)	56.51	48.33	1600	+33.91
`Llama-3.1-8B-Instruct` (Grattafiori et al., 2024)	58.27	50.04	1612	+33.55
`Gemma-2-9B-it` (Team et al., 2024)	74.31	67.42	1791	+23.21
`Qwen3-8B` (Yang et al., 2025)	63.79	60.18	2186	+11.39

GPRL improves every base, with gains shrinking as base capability rises. `Qwen3-8B` starts from a verbose base of about 3100 tokens, and GPRL shortens this to 2186 while still raising LC. WR, the same pattern Section 4 attributes to per-dimension normalization preventing a length-correlated subspace from dominating the aggregate. On `Gemma-2-9B-it`, GPRL again produces the shortest responses among matched methods (1791 tokens, Table 1), and the controller engages on roughly the same schedule as on `Llama-3-8B-Instruct` despite the much stronger base, suggesting that the variance profile $\alpha^{(t)}$ is driven by the GPM and the rollouts rather than by the policy's initial capability.

D.8 Transitive multi-dimensional control

The ablations above isolate the number of subspaces, the normalization scheme, and the controller, but they leave open whether GPRL's gains require GPM's relational representation or would follow from any multi-dimensional reward passed through the same pipeline. To separate the two, we train three scalar BT heads $r_l(x, y)$ on the same `Skywork-Reward` comparisons and at the same 8B scale as the GPM, define the transitive per-dimension score $s_l(y_i, y_j \mid x) = r_l(x, y_i) - r_l(x, y_j)$, and pass it through exactly the same normalization, aggregation weights, drift controller, rollouts, and GRPO-style update as GPRL. Since heads trained on a single binary label are unidentified and could collapse onto copies of one utility, we add an orthogonality penalty between them during training, which keeps their held-out pairwise correlations at 0.23, 0.28, and 0.26 and makes the control verifiably multi-dimensional.

Table 9 shows that moving from one scalar head to three transitive heads gains 8.92 LC. WR points, confirming that multi-dimensional supervision helps on its own, while replacing the transitive heads with GPM subspaces gains a further 5.67 points and shortens responses by 216 tokens. Because data, model scale, dimensionality, normalization, controller, and policy optimization are all held fixed, this second gap isolates the preference representation, and GPM retains an edge over the control even under the pooled normalization of Appendix D.2 (52.34 against 50.84), although it then loses the length control that per-dimension normalization provides. We attribute the gap to the relational form of the GPM score rather than specifically to intransitive cycles.

Table 9: **Transitive multi-dimensional control.** AlpacaEval 2.0 LC. WR and average response length at the 8B scale after three epochs of training. The transitive control uses three orthogonality-regularized scalar BT heads trained on the same `Skywork-Reward` comparisons as the GPM.

Preference representation	Normalization	Controller	LC. WR	Avg. Len
Scalar BT (GRPO+BT)	–	off	41.92	1893
Three transitive BT heads	per-dimension	on	50.84	1816
GPM, $k = 3$	global	on	52.34	2104
GPM, $k = 3$ (GPRL)	per-dimension	on	**56.51**	**1600**

Opponent dependence. The same control tests whether GPM's relational structure survives the scalarization in Eq. 5. For a transitive score, any two responses in the same group satisfy $\hat{s}_l^{(a)}(x) - \hat{s}_l^{(b)}(x) = \frac{G}{G-1}\big(r_l(x, y_a) - r_l(x, y_b)\big)$, so once the normalization statistics and weights are frozen, their ordering cannot depend on which opponents were sampled alongside them, whereas under GPM each opponent y_j contributes $s_l(y_a, y_j \mid x) - s_l(y_b, y_j \mid x)$, which is bilinear in $\mathbf{v}_{y_j|x}$ and can change sign with the opponent. Freezing the normalization statistics and weights on a reference response pool and resampling only the opponents of each fixed candidate pair, we find that the transitive control never reverses an ordering (0.0%), as the calculation requires, while GPM reverses 18.6% of them. Although the policy update uses a single aggregate advantage, that advantage is computed from payoffs against the current response population, so the relational structure remains active during online training. We do not claim that every reversal is beneficial, since Table 9 establishes utility while the reversal rate establishes only that the structure is preserved.

D.9 Second reward-model corpus

Appendix D.1 attributes the $k = 3$ plateau to `Skywork-Reward`'s facet structure, which we test by training a GPM on `HelpSteer2-Preference` (Wang et al., 2025d), whose pairwise preferences were collected alongside the Likert-5 ratings of `HelpSteer2` (Wang et al., 2024) for helpfulness, correctness, coherence, complexity, and verbosity on the same prompts and responses. The preferences train the GPM while the attributes provide an independent facet structure to compare against, and since the attributes are themselves far from independent (helpfulness correlates with correctness at 0.94 but with verbosity at only 0.06), the two corpora differ in both the number and the separability of their facets. We match the number of comparisons to the `Skywork-Reward` GPM and keep every other setting unchanged, so that a shifted plateau cannot be explained by the amount of preference data.

Table 10 shows that the plateau moves from $k = 3$ on `Skywork-Reward` to $k = 2$ on `HelpSteer2-Preference`, with LC. WR regressing mildly beyond it as on the original corpus. The main qualitative result therefore transfers, since a multi-dimensional GPM substantially outperforms $k = 1$ on both corpora, while the preferred dimensionality turns out to be a property of the supervision rather than of GPRL.

Because `HelpSteer2` annotates attributes explicitly, it also lets us ask what the learned subspaces capture. Figure 7 reports the Spearman correlation between each subspace's per-response score and the attribute ratings for the $k = 2$ GPM, where the first subspace tracks overall quality through helpfulness, correctness, and coherence, the second tracks style through verbosity and complexity, and every correlation with a non-matching attribute stays below 0.18 in magnitude. This amounts to a coarse separation between a quality axis and a style axis rather than a full disentanglement of the five attributes, and it also shows up downstream, since moving from $k = 1$ to $k = 2$ raises the average MT-Bench math and coding score by 0.68 against 0.19 on writing and role-playing. We read this as

Table 10: **Ablation on the reward-model corpus.** AlpacaEval 2.0 LC. WR for GPMs trained on `Skywork-Reward` (from Table 4) and on a size-matched `HelpSteer2-Preference` corpus, at the 8B scale with all other hyperparameters fixed.

Subspaces k	`Skywork-Reward`	`HelpSteer2-Preference`
$k = 1$	44.21	45.76
$k = 2$	51.86	**53.12**
$k = 3$	**56.51**	52.39
$k = 4$	56.18	51.84

support for the per-category gains of Figure 4 reflecting facets present in the supervision corpus, with the caveat that `HelpSteer2-Preference` may simply supervise these facets more strongly than `Skywork-Reward` does.



Figure 7: **Alignment of learned subspaces with `HelpSteer2` attributes.** Spearman correlation between each subspace's per-response score and the attribute ratings for the $k = 2$ GPM trained on `HelpSteer2-Preference`, with shading proportional to the correlation and gray cells marking correlations below 0.18 in magnitude.

E Additional analyses

This appendix collects two diagnostic analyses that leave the training configuration unchanged but look at two assumptions the method relies on, namely that the frozen GPM remains informative on the evolving rollout distribution and that the sufficient condition of Proposition 2 holds often enough in practice to matter.

E.1 On-policy calibration of the frozen GPM

GPRL keeps the GPM frozen so that $\alpha^{(0)}$ does not move underneath the controller, and although the unit-norm constraint keeps every subspace score on a fixed $[-1, 1]$ scale (Appendix A), boundedness buys numerical stability rather than ranking accuracy on the evolving rollout distribution. We therefore measure calibration directly following a similar protocol to Singhal et al. (2024), i.e., sampling $1{,}000$ on-policy response pairs at epochs 0, 1, 3, and 5 of the 8B controller-on run, labeling them with a `Llama-3.1-70B-Instruct` judge held out from our benchmark setup so that the check is not circular, and reporting the pairwise accuracy, AUROC, and median absolute margin of the aggregate GPM score. We leave individual subspaces unscored against the overall label, since an axis that tracks, say, verbosity should sometimes disagree with overall preference.

Figure 8 shows calibration declining gradually as the rollout distribution moves, with pairwise accuracy falling by 2.9 points over five epochs while staying well above the 50% chance level, even as LC. WR rises from 22.6 for the base policy to 56.47 and drift $\mathcal{D}(t)$ stays between 0.00 and 0.04, so we read this as mild degradation rather than a failure of the frozen GPM. Without the controller, however, drift climbs to 0.27 by epoch 5, pairwise accuracy falls to 63.9, and LC. WR falls to 47.82. The drift monitor does not measure calibration directly, but by limiting how far variance concentrates on a single axis, it also appears to keep the policy away from regions where the frozen GPM becomes substantially less reliable, which complements the tightening of β in Eq. 9.

E.2 Empirical frequency of Proposition 2

Proposition 2 gives a sufficient condition on per-dimension advantages that change throughout training, so its practical relevance depends on how often it holds. Since every step logs the k pairwise

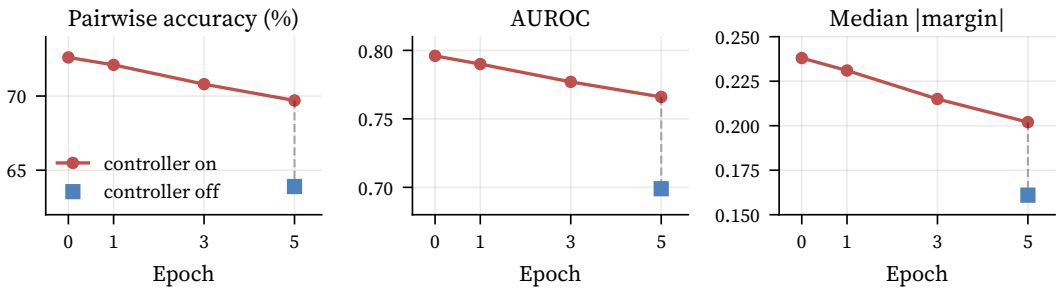


Figure 8: **On-policy calibration of the frozen GPM.** Pairwise accuracy, AUROC, and median absolute margin of the aggregate GPM score on 1,000 on-policy response pairs labeled by a held-out judge, for the 8B controller-on run across epochs (line) and the controller-off run at epoch 5 (square).

score matrices per group and the advantages in Eq. 7 are exactly the per-response quantities of Eq. 4, we evaluate the condition on the 8B controller-on run over within-group response pairs (i, j) satisfying the antecedent, where y_i improves on one dimension l^* and regresses on the other $k-1$ relative to y_j, using the effective weights $\tilde{w}_l(t) = m_l(t)\, w_l$ that the controller applies at each step. We report the fraction of within-group pairs that satisfy the antecedent (coverage), the fraction of those for which Eq. 7 holds, and the fraction for which the aggregate advantage ranks the balanced response y_j above y_i.

Figure 9 shows that the antecedent covers about one fifth of response pairs, as expected from requiring improvement on exactly one axis together with regression on all others. Among applicable pairs, the sufficient inequality holds in about 80% of cases at every checkpoint, and the aggregate advantage ranks the balanced response higher slightly more often because the condition is sufficient but not necessary for that ordering, so Proposition 2 is far from vacuous and remains stable across the five-epoch horizon.

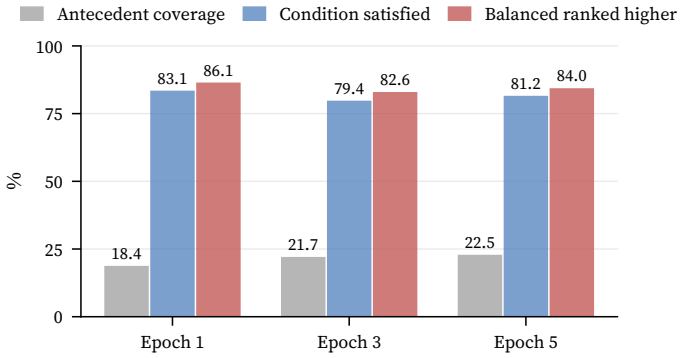


Figure 9: **Empirical frequency of Proposition 2.** Fraction of within-group response pairs that satisfy the antecedent, and among those, the fraction for which Eq. 7 holds and for which the aggregate advantage ranks the balanced response higher, on the 8B controller-on run.

F Implementation Details

This appendix collects the practical settings used to produce the results presented in Section 4 and the corresponding supplementary tables, so that the reader can reproduce our findings or extend our methodology without needing to infer experimental specifics from the broader narrative. Where a hyperparameter is shared with GRPO (Shao et al., 2024) or the SPPO and GPO baselines (Wu et al., 2024b; Zhang et al., 2025b), we retain the previously published values to keep the comparison equitable.

Hardware. All training experiments were executed on a dedicated cluster designated as `REDACTED_FOR_SUBMISSION`, which utilizes NVIDIA DGX H100 nodes equipped with eight

H100 80GB GPUs each. A single computation node provides sufficient processing capacity to derive the primary results at the 8B parameter scale. Unlike the training experiments, all evaluation procedures are computationally lightweight and operate efficiently on a single H100 80GB GPU, since only forward passes through the policy are required for response generation.

Codebase. Our implementation extends the `GRPOTrainer` of the `trl` library (von Werra et al., 2020). We replace its scalar advantage computation with the per-dimension estimation and aggregation of Eqs. 4–5, and add the drift monitor and controller of Section 3.2 as a hook that updates the dimension multipliers $m_l(t)$ and the KL coefficient β between training steps. Other than these modifications, the `trl` codebase remains largely unchanged.

Reproducibility. To provide a clear overview of the experimental framework, Table 11 contains key hyperparameters.

Table 11: **Implementation details for GPRL.** Hyperparameter values used to produce the main results in Section 4 unless an ablation overrides them.

Group	Setting	Value
Policy and dataset	Base policies	`Llama-3-8B-Instruct`, `Gemma-2-9B-it`
	Rollout prompts	`UltraFeedback`
	GPM reward source	8B GPM trained on `Skywork-Reward`
	GPM embedding dimension $2k$	6 ($k = 3$ subspaces)
	Dimension weights w_l	uniform ($w_l = 1$), GPM gate unused
Optimization	Optimizer	AdamW
	Peak learning rate	1×10^{-6}
	LR scheduler	cosine
	Warmup ratio	0.03
	Weight decay	0.1
	Gradient clipping	1.0
	Initial KL coefficient β	0.01
	Per-device batch size	1
	Gradient accumulation steps	8
	Effective batch size	64
	Training epochs	3
	Inner iterations per group	1
	Mixed precision	bfloat16
Rollouts	Group size G	8
	Max completion length	512
	Sampling temperature	1.0
	Backend	vLLM
Drift controller	Threshold τ	0.20
	Redistribution exponent γ	0.5
	KL multiplier κ	1.5
	KL ceiling $\beta_{\max}$	0.20
	Relaxation rate δ	0.99
Hardware	Training	1 DGX H100 node (8× H100 80GB)
	Evaluation	1× H100 80GB