Improving Reward Signal Quality for Online RLHF: Checkpoint Selection, Ensembling, Advantage Baselines, and RFPO

Jason Trinh
Department of EECS
UC Berkeley
jasontrinh@berkeley.edu

Tanmayi Dasari
Department of EECS
UC Berkeley
tanmayi_dasari@berkeley.edu

Kavish Kondap
Department of EECS
UC Berkeley
kavish_kondap@berkeley.edu

Abstract

We study RLHF for open-ended instruction following on the WildChat benchmark with Qwen2.5-1.5B-Instruct, scored by GPT-5.4 head-to-head win rate against the frozen base. We implement and clear the autograder thresholds for all six required Part 1 methods (DPO 70.6%, IPO 73.5%, AOT 75.5%, GRPO 72.5%, DrGRPO 62.0%, GSPO 69.4%) and for the Bradley–Terry reward model (0.8398 held-out pair accuracy on `test_prefs`). For Part 2 we work down the online RLHF pipeline and intervene at four stages: (i) RM checkpoint selection, (ii) pessimistic min-aggregation across two RMs, (iii) leave-one-out advantage baselines and running reward whitening, and (iv) reward-based rollout filtering with either the GRPO clipped surrogate (RFPO) or a per-pair DPO logistic loss (Online DPO). The strongest Part 2 online configuration, Online DPO with G=16, k=2 at 75 steps, reaches **87.7%** win rate, exceeding the strongest Part 1 online baseline (GRPO at 72.5%) by $+15.2$ points; we grade 110 candidate checkpoints across four method families and report failures (group-max baseline collapse, top-k-only filter) alongside successes. Code and all reproduction commands are at `https://github.com/kavishkondap/185_final_project_llm_rl`.

1 Introduction

Reinforcement learning from human feedback (RLHF) has become the dominant paradigm for aligning LLMs with human preferences [Christiano et al., 2017, Ouyang et al., 2022]. A central challenge in online RLHF is reward hacking: the policy exploits imperfections in a learned reward model, producing responses that score highly under the RM but are not genuinely preferred by humans [Gao et al., 2023]. This is particularly acute for small models on limited data, where the RM may be underfitted, over-specialized, or sensitive to checkpoint choice. We use Qwen2.5-1.5B-Instruct on the WildChat benchmark [Zhao et al., 2024] and measure success as GPT-5.4 win rate against the frozen base. As Part 1 we implement DPO [Rafailov et al., 2023], IPO [Azar et al., 2024], AOT [Melnyk et al., 2024], GRPO [Shao et al., 2024], DrGRPO [Liu et al., 2025], and GSPO [Zheng et al., 2025]; these serve as baselines for all Part 2 work.

Our Part 2 investigation walks down the online RLHF pipeline and asks at each stage what aspect of the reward signal can be made more reliable: (1) the *source* of the signal via RM checkpoint selection,

Preprint.

(2) aggregation of the signal via pessimistic min-ensembling of multiple RMs, (3) use of the signal in the policy gradient via leave-one-out advantage baselines and running reward whitening, and (4) which rollouts are trained on via Reward-Filtered Policy Optimization (RFPO) and Online DPO. Each step extracts more reliable supervision from the same noisy reward model rather than adding an unrelated trick.

2 Related Work

Group-relative online RL. Our online baselines come from the family of group-relative policy gradient methods: GRPO [Shao et al., 2024] normalizes advantages by the per-prompt group mean and std and clips at the token level; DrGRPO [Liu et al., 2025] drops the standard deviation divisor and the per-sequence length normalization to remove length and easy-prompt biases; GSPO [Zheng et al., 2025] replaces the per-token importance ratio with a single sequence-level ratio and clips at the sequence level. We use all three as Part 1 baselines and as the underlying surrogates for our filtering experiments.

Filtered group methods (GFPO) and RAFT. GFPO [Shrivastava et al., 2025] rejection-samples rollouts per prompt using both reward and response-length criteria, primarily for length control. Our RFPO is inspired by GFPO but evaluates two reward-based rules side by side (top-k vs. top-k+bottom-k) and drops the length criterion since WildChat already bounds responses to 256 tokens. RAFT [Dong et al., 2023] is the closest precursor to the top-k variant; unlike RAFT we (i) train with the GRPO clipped surrogate rather than SFT, (ii) compute advantages from full-group statistics before filtering, and (iii) reintroduce the bottom-k as a negative-gradient signal in our top+bottom variant.

Online DPO and OAIF. DPO [Rafailov et al., 2023] reformulates RLHF as classification over a fixed dataset of pairs. OAIF [Guo et al., 2024] extends this online by sampling a pair of completions at each step and labelling them with a large LLM annotator. Our Online DPO differs in two ways: (i) we use a trained Bradley–Terry RM as the preference signal rather than an external LLM judge (cheaper, and the same RM is reused for ensembling and whitening); (ii) where OAIF samples $G{=}2, k{=}1$, we sample $G \in \{4, 8, 16\}$ with up to $k = 4$ rank-aligned pairs per prompt. $G{=}2, k{=}1$ recovers an RM-annotated OAIF analogue and is substantially weaker than our $G{=}16, k{=}2$ best on this benchmark.

3 Method

Sections 3.1–3.2 cover the Part 1 algorithms; Sections 3.3–3.7 introduce our four Part 2 contributions, organized along the pipeline described in the Introduction.

3.1 Preliminaries: GRPO, DrGRPO, and GSPO

Let π_θ denote the policy, π_{ref} the frozen reference, and $r(x, y)$ a scalar reward model score. The generic RLHF objective $\mathcal{J}(\theta) = \mathbb{E}\big[r(x, y)\big] - \lambda_{\text{KL}}\, D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})$ is optimized by all three online algorithms by sampling G completions per prompt and forming group-relative advantages; they differ only in advantage normalization and where the PPO clip is applied. Throughout, we reserve β for the offline preference-loss temperature (Section 3.2) and use λ_{KL} for the online KL coefficient.

GRPO [Shao et al., 2024] normalizes advantages by the per-group mean and std,

$$A_{i,k} = \frac{r_{i,k} - \mu_i}{\sigma_i + \varepsilon}, \quad \mu_i = \tfrac{1}{G}\sum_k r_{i,k}, \quad \sigma_i^2 = \tfrac{1}{G}\sum_k (r_{i,k} - \mu_i)^2, \tag{1}$$

and applies PPO-style clipping at the token level with per-sequence length normalization.

DrGRPO [Liu et al., 2025] drops the σ divisor (mean centering only) and removes the per-sequence length normalization. Because GRPO's σ divisor would otherwise normalize advantages back to roughly $\mathcal{O}(1)$ regardless of reward scale, removing it lets DrGRPO's advantages scale with the actual per-group reward spread, so DrGRPO takes larger policy steps than GRPO at the same nominal λ_{KL}.

GSPO [Zheng et al., 2025] replaces the per-token importance ratio with a single sequence-level ratio (the geometric mean of per-token ratios) and clips at the sequence level. This stabilizes the surrogate for long completions, at the cost of treating each response as a single accept/reject decision rather than a per-token gradient.

3.2 Offline Preference Methods: DPO, IPO, AOT

All three offline methods consume the same (prompt, chosen, rejected) triples from `train_prefs` but optimize structurally different objectives. Letting $\Delta_\theta = (\log \pi_\theta(y^+|x) - \log \pi_\theta(y^-|x)) - (\log \pi_{\text{ref}}(y^+|x) - \log \pi_{\text{ref}}(y^-|x))$ denote the reference-corrected sequence margin, the three losses are:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma\big(\beta\, \Delta_\theta\big), \qquad \mathcal{L}_{\text{IPO}} = \big(\Delta_\theta - \tfrac{1}{2\beta}\big)^2,$$
$$\mathcal{L}_{\text{AOT}} = \frac{1}{B}\sum_{i=1}^{B} -\log \sigma\big(\beta\,(r^+_{(i)} - r^-_{(i)})\big), \tag{2}$$

where $r^+_{(i)}, r^-_{(i)}$ are sorted minibatch sequence rewards. These target different notions of "the chosen response is better": DPO is Bradley–Terry MLE with an unbounded per-pair margin (most aggressive separator, most exposed to label noise); IPO regresses the margin to a fixed target $1/(2\beta)$ and zeros out the gradient on already-correctly-ordered pairs (most conservative); AOT optimizes distributional dominance across the minibatch by sorting, which is tolerant to individual mis-rankings but destroys per-pair coupling.

3.3 Reward Model Checkpoint Selection

A fully converged RM may be more susceptible to reward hacking: it has had more opportunity to overfit to superficial patterns in the preference data, producing a signal that is easier for the policy to game. An earlier RM checkpoint, still imperfect in discriminative accuracy, may present a smoother and harder-to-exploit reward landscape.

The Part 1 default RM checkpoint at step 445 reaches **0.8398** pair accuracy on `test_prefs`, comfortably clearing the 0.74 autograder threshold (Appendix Figure 6). We additionally train GRPO, DrGRPO, and GSPO under standard Part 1 hyperparameters against three earlier RM checkpoints (steps 150, 250, 350) chosen from different peak-then-plateau regions, holding all other settings fixed so downstream win-rate differences attribute to the RM checkpoint alone.

3.4 Reward Model Ensembling

Our ensemble uses two Bradley–Terry reward models $r^{(2)}, r^{(3)}$ trained on the same preference dataset with different random seeds, and aggregates their scores before computing advantages by either mean $r_{\text{mean}}(x,y) = \frac{1}{K}\sum_k r^{(k)}(x,y)$ or min $r_{\text{min}}(x,y) = \min_k r^{(k)}(x,y)$. Min is pessimistic (a response must score well under every model), so if the RMs have independent failure modes a policy that exploits one model still scores poorly under min.

3.5 Improved Advantage Baselines and Reward Whitening

The GRPO group-mean baseline $b_k = \frac{1}{G}\sum_j r_j$ includes r_k itself, introducing a self-correlation bias. The leave-one-out (LOO) baseline $b_k^{\text{loo}} = (S - r_k)/(G-1)$, with $S = \sum_j r_j$, removes this. We also test a group-max baseline $b_k^{\text{max}} = \max_j r_j$ (a cheap ReMax-style approximation [Li et al., 2023] that makes all advantages non-positive). As training progresses the reward distribution shifts upward and a fixed λ_{KL} becomes effectively looser, so we maintain an online (Welford) estimate of the global reward mean and std and whiten rewards before computing advantages, $\tilde{r}(x,y) = (r(x,y) - \mu)/(\sigma + \varepsilon)$, so that hyperparameters stay scale-stable.

3.6 RFPO: Reward-Filtered Policy Optimization

RFPO extends GRPO with a rejection-sampling stage: for each prompt x_i the policy generates G responses (with G larger than the standard $G{=}4$ baseline), each scored by the reward model, and a

subset is kept for the GRPO clipped surrogate update. We compare two reward-based filtering rules. **Top-k** keeps the k highest-scoring responses; advantages are computed from full-group statistics (μ_i, σ_i over all G) before filtering so retained samples carry non-negative advantages, and the update only pushes toward kept responses (this adapts Dong et al. [2023] to the GRPO surrogate). **Top-k+bottom-k** keeps the k highest and k lowest; the top-k samples receive positive gradient and the bottom-k receive negative gradient, retaining the "what to avoid" signal the top-only filter discards. We do not pursue GFPO's response-length criterion [Shrivastava et al., 2025] since WildChat already bounds responses to 256 tokens.

3.7 Online DPO

Online DPO replaces the GRPO clipped surrogate with the DPO logistic loss (Section 3.2) applied to (chosen, rejected) pairs from the same top-k+bottom-k filter. For each prompt we pair the rank-j response from the top-k set with the rank-j response from the bottom-k set ($j = 1, \ldots, k$) and average the per-pair DPO loss: $\mathcal{L}_{\text{onlineDPO}} = -\frac{1}{k}\sum_{j=1}^{k} \log \sigma\big(\beta\, \Delta_\theta(x, y_+^{(j)}, y_-^{(j)})\big)$, where $y_+^{(j)}, y_-^{(j)}$ are the chosen/rejected responses from the filter. Unlike offline DPO, pairs are generated on the fly from the current policy and labelled by the RM. G=2, k=1 recovers the canonical OAIF setup [Guo et al., 2024]; larger G and k trade more rollouts/RM evaluations for sharper margins and more pairs per prompt. Because the loss is per-pair, no group-mean or LOO baseline is needed: the reference policy inside Δ_θ plays that role.

4 Experiments and Results

We evaluate all methods on the WildChat benchmark using GPT-5.4 win rate against the frozen Qwen2.5-1.5B-Instruct base model as the primary metric. Win rate counts only prompt pairs where the judge agrees across both forward and reversed orderings, filtering out judge uncertainty and position bias.

4.1 Experimental Setup

All experiments use the WildChat benchmark and Qwen2.5-1.5B-Instruct with LoRA adapters, trained against Bradley–Terry reward models on the same dataset. Configuration details are in Appendix A; exact reproduction commands are in `commands.md` at the repo root.

4.2 Main Results

4.2.1 Part 1 Baselines

Table 1 reports the Gradescope autograder GPT-5.4 win rate for the six required Part 1 methods against the frozen Qwen2.5-1.5B-Instruct base on the 128-prompt public evaluation file. All six pass their respective thresholds.

Table 1: Part 1 autograder win rates against the frozen base on the 128-prompt evaluation set. All six methods clear their thresholds. Bolded entries are the strongest in each family.

Family	Algorithm	Win rate	Threshold
Offline	DPO	70.6%	0.65
Offline	IPO	73.5%	0.60
Offline	AOT	**75.5%**	0.65
Online	GRPO	**72.5%**	0.60
Online	DrGRPO	62.0%	0.60
Online	GSPO	69.4%	0.60

Which Part 1 method worked best. AOT leads the offline family at 75.5%, ahead of IPO at 73.5% and DPO at 70.6%; the next paragraph relates this ordering to the chosen-minus-rejected margin each loss produces. Qualitatively, across the 128 evaluation prompts mean response length is $\sim$200 / $\sim$205

/ ∼230 words for DPO / IPO / AOT and AOT outputs more often engage the full user request (sample inspection in Appendix E).

On the online side GRPO leads at 72.5%, GSPO at 69.4%, and DrGRPO at 62.0%. We attribute this ordering to advantage scaling at the short 25-step Part 1 horizon: DrGRPO's unnormalized advantages produce $\sim 2.4\times$ larger KL drift than GRPO at the same nominal $\lambda_{\text{KL}} = 0.01$ (Figure 1), so it drifts further from the reference per step.

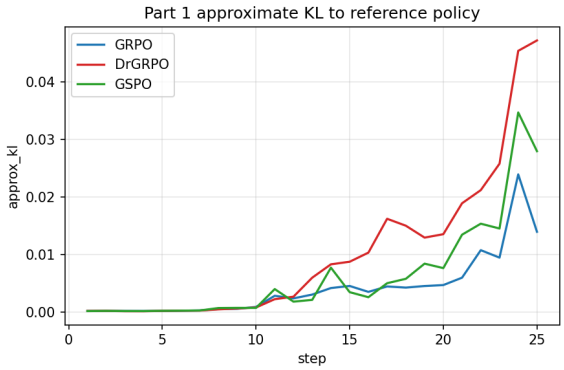

Figure 1: Approximate KL to the frozen reference during Part 1 training (25 steps). DrGRPO grows $\sim 2.4\times$ faster than GRPO at the same nominal $\lambda_{\text{KL}} = 0.01$, confirming the corrected mechanism in Section 3.1: DrGRPO's lack of per-group σ normalization produces larger updates, not smaller.

When Part 1 internal metrics mislead. A natural internal "the policy learned the preference" metric is the chosen-minus-rejected log-prob margin $\log \pi(y_w) - \log \pi(y_l)$. In Bradley–Terry terms a larger margin means a stronger preference for chosen over rejected, so monotonically growing margin would normally read as successful training. Figure 2 shows the opposite empirically: DPO's per-token margin climbs to ∼17 log-prob units yet DPO wins the least (70.6%), AOT keeps its margin at ∼0.7 and wins the most (75.5%), and IPO holds its margin near zero and beats DPO at 73.5%. The judge's ordering inversely tracks margin magnitude: the runaway margin that DPO's unbounded logistic loss produces is exactly the signal that does *not* buy external quality.

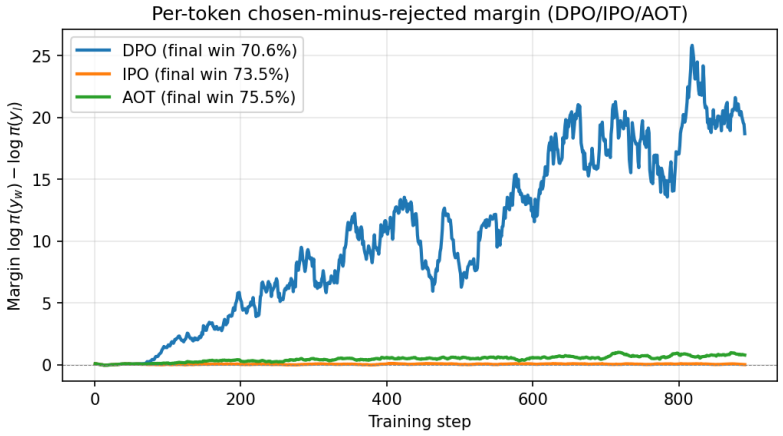

Figure 2: Per-token chosen-minus-rejected margin $\log \pi(y_w) - \log \pi(y_l)$ on training pairs for DPO (β=0.05), IPO, and AOT (β=0.2). DPO's margin grows essentially without bound, IPO holds its margin near zero, AOT keeps it small (≤ 1). Curves are 25-step rolling means.

4.2.2 Effect of Reward Model Checkpoint on Policy Performance

Table 2 shows win rates for GRPO, DrGRPO, and GSPO trained against reward-model $r^{(2)}$ at four training checkpoints (steps 150, 250, 350, and the converged step 445). All four cells per algorithm

use identical Part 1 hyperparameters, so any win-rate difference is attributable to the RM checkpoint alone.

Table 2: Win rates by reward-model training checkpoint. All rows use RM $r^{(2)}$ at the indicated step; only the RM checkpoint varies. Steps 150, 250, and 350 were selected from the region of the held-out pair-accuracy curve where accuracy peaked before plateauing. The step-445 row uses a different RM seed from Table 1 (≤ 3 point offset).

RM Checkpoint	GRPO	DrGRPO	GSPO
Step 150	63.9%	67.3%	65.1%
Step 250	71.4%	71.7%	73.5%
Step 350	66.7%	75.0%	60.4%
Step 445 (sweep baseline)	69.6%	60.4%	71.7%

The pattern is not uniform. Step 250 is the most consistently strong choice (best for GRPO and GSPO, second-best for DrGRPO). DrGRPO is the outlier, peaking at step 350 (75.0%, a substantial improvement over the step 445 sweep baseline of 60.4%): DrGRPO's unnormalized advantages make it more sensitive to RM-score scale. Step 150 is weakest for all three, indicating that very early RM checkpoints are not yet discriminative enough to provide useful training signal even if they are less overfit.

4.2.3 Reward Model Ensembling, LOO Baselines, and Reward Whitening

We ran three progressive batches adding one component at a time, plus a Batch 4 ablation (Section 4.2.4). Batch 1 swept ensembling aggregation (min vs. mean) and the ReMax-style group-max greedy baseline at 25 steps; Batch 2 added the LOO baseline and extended to 75 steps; Batch 3 layered running reward whitening on top of Batch 2. Table 3 summarizes the progression and Figure 3 shows the training trajectory of the best (Batch 3) configuration; a per-recipe bar chart spanning all 30 runs is in Figure 7 in the appendix. Full per-batch tables and per-batch hyperparameters are in Appendices C and A.4.

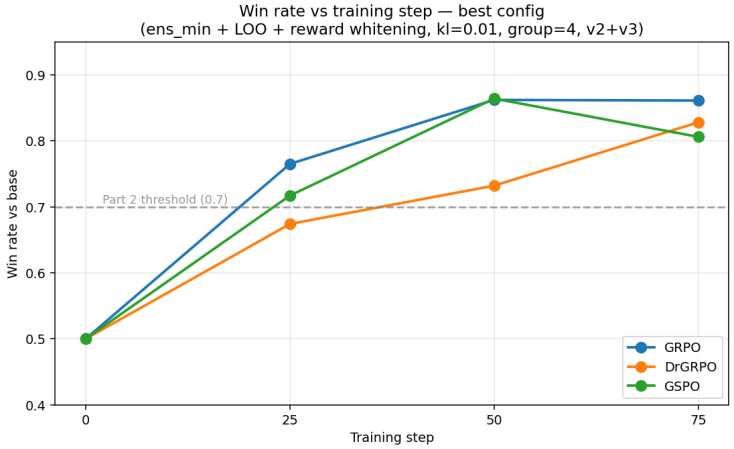

Figure 3: Win rate vs. training step for the best Batch 3 configuration (`ens_min + LOO + reward whitening`, `kl_coef=0.01`, `group_size=4`, RMs $r^{(2)}+r^{(3)}$) across the three online algorithms. GRPO reaches the global maximum (86.1%) at step 75. GSPO peaks at step 50. DrGRPO climbs slowest but is still ascending at step 75.

Batch 1 (aggregation + greedy baseline). Min aggregation beats mean even at 25 steps because it is more pessimistic (a response must score well under both RMs), suppressing single-RM exploitation already visible at this short horizon. The ReMax-style group-max greedy baseline collapses training across all algorithms: with all advantages non-positive, the update only suppresses non-best responses and provides no positive reinforcement.

Table 3: Per-batch progression of GPT-5.4 win rate against the frozen base across the three online algorithms. Bolded value is the best single run. Full per-batch tables in Appendix C; per-batch settings in Appendix A.4.

Batch	Configuration	GRPO	DrGRPO	GSPO
1 (25 steps)	mean-agg, group_mean	69.4%	70.2%	70.0%
1 (25 steps)	min-agg, group_mean	74.5%	73.3%	76.0%
1 (25 steps)	group_max (greedy baseline)	2.4%	2.6%	0.0%
2 (75 steps)	min-agg + LOO	80.6%	82.6%	76.1%
3 (75 steps)	min-agg + LOO + whitening	**86.1%**	82.8%	80.6%

Batch 2 (LOO + longer training). Adds several percentage points over the Batch 1 min-aggregation runs across all three algorithms.

Batch 3 (+ running whitening). Produces large gains for GRPO/GSPO but is nearly neutral for DrGRPO (we discuss the likely mechanism in Section 5). GRPO with this configuration reaches **86.1%**, the highest Part 2 ensembling result.

4.2.4 Ablations of the Batch 3 best configuration

Each row of Table 4 removes one component (ensembling, LOO, or longer training) from the Batch 3 best while holding the rest fixed.

Table 4: Component ablations of the Batch 3 best configuration; each row removes one component while keeping the rest fixed. Δ is relative to the Batch 3 seed=0 result. A reseed at `seed=1` yields 84.6%/73.2%/83.1% (GRPO/DrGRPO/GSPO), so DrGRPO has $\sim$9.6 points seed-to-seed variance and its ablation deltas below $\sim$5 points should not be over-interpreted.

	GRPO		DrGRPO		GSPO	
Configuration	Win rate	Δ	Win rate	Δ	Win rate	Δ
Batch 3 best (seed=0)	86.1%	–	82.8%	–	80.6%	–
− Ensemble (single $r^{(2)}$)	84.2%	-1.9	76.2%	-6.6	84.4%	$+3.8$
− LOO (group-mean)	83.8%	-2.3	79.3%	-3.5	77.1%	-3.5
− Longer training (25 steps)	76.5%	-9.6	67.4%	-15.4	71.7%	-8.9

Longer training is unambiguously the dominant factor (8.9–15.4 points across algorithms, far above seed variance). LOO is small but consistent (a few points on every algorithm). Ensembling is algorithm-dependent: it slightly *improves* GSPO when removed, leaves GRPO within seed noise, and visibly hurts DrGRPO (though that drop also sits inside DrGRPO's $\sim$9.6 points seed-variance band). The early-training benefit of min aggregation appears to diminish once LOO and whitening already provide stabilization.

4.2.5 RFPO Results

We trained the two RFPO filtering variants (top-k only and top-k+bottom-k with the GRPO surrogate) and the Online DPO variant (Section 4.2.6) across group sizes $G \in \{4, 8, 16\}$ and filter sizes k up to 4, each for 75 training steps with checkpoints saved every 25 steps. To isolate the design trade-off, `batch_size` is coupled to G so rollouts/step stay constant at 128 (Appendix A.5): raising G trades fewer distinct prompts per step for more contrastive rollouts per prompt rather than buying more compute. In total we grade **110 candidate checkpoints** across the four method families under the same GPT-5.4 judge as Part 1 (Appendix D). Throughout the figures below, step 0 is fixed at 0.5 (base vs. base) and the dashed line marks the Part 2 online threshold of 0.70; the unfiltered baselines at $G \in \{4, 8, 16\}$ are the comparison anchor (Appendix Figure 8; strongest unfiltered 75-step run: DrGRPO G=16 at step 50, 86.7%).

Top-only filtering (RAFT-style) underperforms baselines. The top-k-only family (Figure 4, left) never exceeds 62.3% and the failure is consistent across all three algorithms and group sizes. The

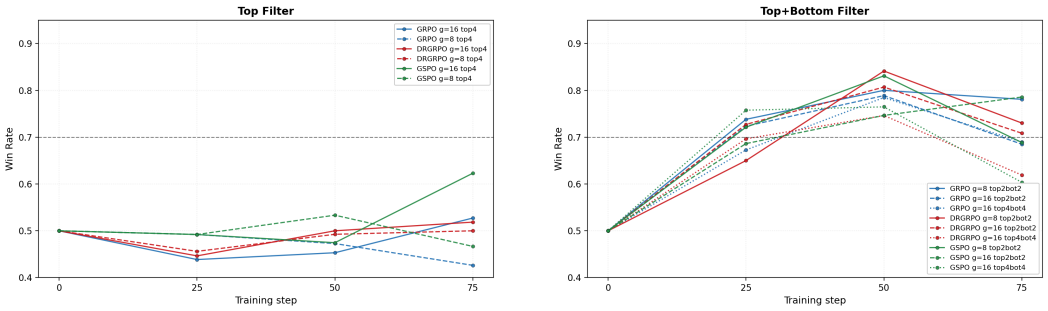

Figure 4: Win rate vs. training step for the two RFPO filtering variants. *Left*: top-k-only (RAFT-style filter); *right*: top-k+bottom-k with the GRPO clipped surrogate. Color encodes algorithm; linestyle encodes group size G or the (group, filter) configuration.

root cause is that our top-k selects the k best of G noisy rollouts, a relative ranking rather than the verifiable-success filter used in the original RAFT setting; without a contrastive negative gradient, ranking errors compound and the policy drifts toward whatever the RM ranks highest regardless of absolute quality.

Top-k+bottom-k with the GRPO surrogate recovers baseline performance. Reintroducing the bottom-k samples (Figure 4, right) recovers most of the performance lost from one-sided filtering. The best configuration is DrGRPO G=8, k=2 at step 50, reaching **84.1%** win rate, within 3 points of the strongest unfiltered baseline (DrGRPO G=16 at step 50: 86.7%; Figure 8). Filter strategy (keeping both reward extremes) matters more for downstream policy quality than the loss choice on top of it: the GRPO surrogate applied to filtered pairs nearly matches an unfiltered baseline of equivalent compute.

4.2.6 Online DPO Results

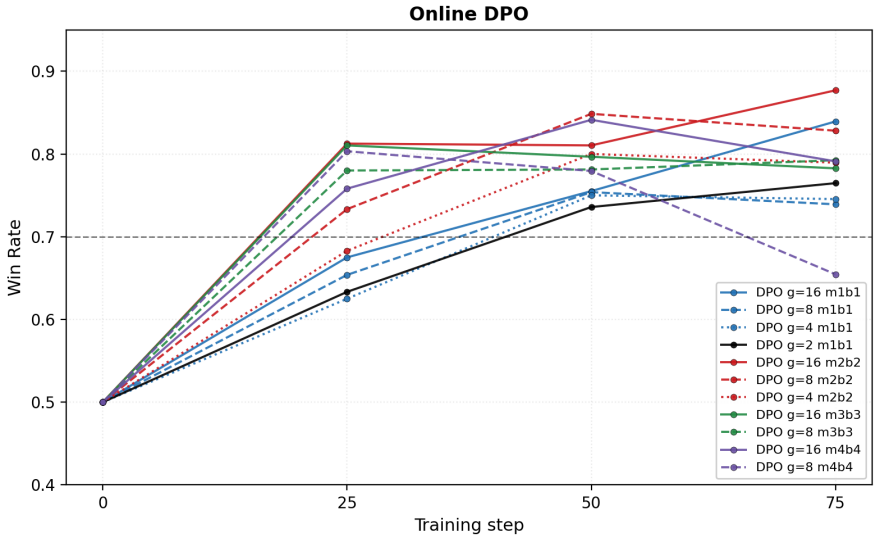

Figure 5: Win rate vs. training step for Online DPO. Color encodes the top-k+bottom-k pair-count variant (legend label "$m_k b_k$" is shorthand for k top-set and k bottom-set pairs, e.g. $m_2 b_2$ is top-2+bot-2). Linestyle encodes group size G (solid: G=16, dashed: G=8, dotted: G=4). The G=2, k=1 line in black is the canonical OAIF [Guo et al., 2024] configuration.

Substituting the DPO logistic loss for the GRPO surrogate while keeping the same top-k+bottom-k filter (Figure 5) yields our best result: G=16, k=2 at step 75 reaches **87.7%** win rate. We frame this against four baselines, ordered from largest delta to most carefully controlled:

- **Starting-point delta:** $+15.2$ points over Part 1 GRPO (72.5%). The two runs share the same reward model (`v1` at `step_000445`) but differ in three other ways: training length (75 vs 25 steps), the top-k+bottom-k filter, and the loss. The training-length ablation alone accounts for +8.9 to $+15.4$ points across algorithms (Table 4).
- **Loss change only:** $+3.6$ points over the best top-k+bottom-k RFPO (84.1%). Swapping the GRPO clipped surrogate for the DPO logistic loss at the same filter adds $\sim$3 points.
- k **scaling only:** $+3.8$ points (83.9% $\rightarrow$ 87.7% at step 75) from doubling the rank-aligned pair count at fixed G=16 (Appendix 14).
- **Matched compute:** $+1.0$ point over DrGRPO G=16 at 86.7%.

Group size matters; OAIF canonical underperforms. The OAIF setup [Guo et al., 2024] (G=2, k=1) peaks at 76.5%, $\sim$ 11 points behind our G=16, k=2 best, with intermediate $G \in \{4, 8\}$ runs interpolating. Two compounding effects explain the gain from larger G and k: the expected max-min RM gap within a group grows with G (sharpening per-pair margins), and k rank-aligned pairs per prompt multiply the contrastive information per rollout round. The gap to OAIF-canonical suggests the standard one-pair-per-prompt setup leaves substantial signal on the table when the RM can cheaply score additional rollouts.

5 Discussion and Limitations

What worked. Two components reliably helped across all three online algorithms: longer training and the LOO advantage baseline. Running reward whitening added substantial gains for GRPO and GSPO but was nearly neutral for DrGRPO. Whitening likely composes with GRPO/GSPO's per-group σ-normalization to keep advantages tightly $\mathcal{O}(1)$, whereas DrGRPO (no per-group σ) only sees a constant rescaling of its already-larger advantage signal, which we measure empirically via DrGRPO's $\sim$ $2.4\times$ larger approximate KL drift in Part 1 (Figure 1, left). Pessimistic RM ensembling with min aggregation gave clear early-training gains that largely disappeared once LOO and whitening were also in place. RFPO filtering with both reward extremes plus an Online DPO loss on rank-aligned pairs gave the single best result we observed, ahead of any unfiltered baseline of matched compute.

What did not work. Two interventions failed under the benchmark constraints: the group-max (ReMax-style) greedy baseline collapsed every algorithm because the update has no positive reinforcement signal, and one-sided top-k filtering on a noisy RM lacked the contrastive negative gradient needed to avoid drifting toward whatever the RM ranked highest.

Limitations and future directions. Most cells were run with a single seed, so small ablation deltas should be treated as tentative, and DrGRPO in particular shows substantial seed sensitivity. Our ensemble uses only two reward models, and all experiments are on a 1.5B-parameter model at relatively short training horizons. Natural next steps are multi-seed averaging on the headline configurations and a best-of-N inference baseline that uses the RM ensemble as a decode-time verifier, which would separate reward-signal-quality gains from gains specific to the online RL update.

6 Contributions

- **Tanmayi:** Reward model checkpoint selection experiments across GRPO, DrGRPO, and GSPO. Reward model and AOT implementation for Part 1.
- **Kavish:** Reward model ensembling with pessimistic aggregation, leave-one-out advantage baselines, running reward whitening. DPO and IPO implementations for Part 1.
- **Jason:** Design and implementation of RFPO and Online DPO, including the rollout filtering pipeline and the on-policy paired DPO loss. Online policy implementations (GRPO, DrGRPO, GSPO) for Part 1.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi 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 (AISTATS)*, 2024.

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

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward rAnked FineTuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

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

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. Direct language model alignment from online AI feedback, 2024.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025.

Igor Melnyk, Youssef Mroueh, Brian Belgodere, Mattia Rigotti, Apoorva Nitsure, Mikhail Yurochkin, Kristjan Greenewald, Jiri Navratil, and Jerret Ross. Distributional preference alignment of LLMs via optimal transport. In *Advances in Neural Information Processing Systems*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024.

Vaishnavi Shrivastava, Hamish Peng, Parishad Behnamghader, Hany Hassan Awadalla, Arindam Mitra, and Weizhu Chen. Sample more to think less: Group filtered policy optimization for concise reasoning, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT interaction logs in the wild. In *International Conference on Learning Representations*, 2024.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Du, Xingzhang Liu, Mingfeng Qu, An Yang, Jingren Tang, and Junyang Lin. Group sequence policy optimization, 2025.

A Training Configurations and Hyperparameters

This appendix collects the full training and evaluation configurations for every run reported in Sections 4–4.2.4.

A.1 Dataset, Model, Reward Models, and Evaluation Protocol

Dataset. We use the `wildchat_min4_judged_5k_v1` dataset, split into 4,744 training preference pairs (`train_prefs`), 256 held-out preference pairs (`test_prefs`), 4,744 prompt-only rollouts (`train_gen`), and 256 held-out evaluation prompts (`test_gen`).

Model. All methods fine-tune `Qwen/Qwen2.5-1.5B-Instruct` with LoRA adapters. The frozen base model serves as both the reference policy for KL regularization and the opponent in head-to-head evaluation.

Reward models. Bradley–Terry reward models ($r^{(1)}$, $r^{(2)}$, $r^{(3)}$; `reward_model_v1/v2/v3`) are trained on `train_prefs` using `Qwen2.5-1.5B-Instruct` with a scalar regression head. All Part 1 online runs use $r^{(1)}$ at `step_000445`. The Part 2 ensemble uses two RMs trained with different random seeds, $r^{(2)}$ and $r^{(3)}$, both at `step_000445`; a single-RM ablation uses $r^{(2)}$ alone at `step_000445`.

Evaluation. The primary metric is GPT-5.4 win rate on the 128-prompt public evaluation file. All policy JSONLs are generated with deterministic decoding (`temperature=0.0`, `top_p=1.0`) and `max_new_tokens=256`. Win rate is computed using `policy_win_rate_pair_agree_usable`, which only counts pairs where the GPT judge agrees across both forward and reversed orderings. Secondary diagnostics include held-out reward model pair accuracy on `test_prefs`, approximate KL divergence from the reference policy, and reward model score trends during online training.

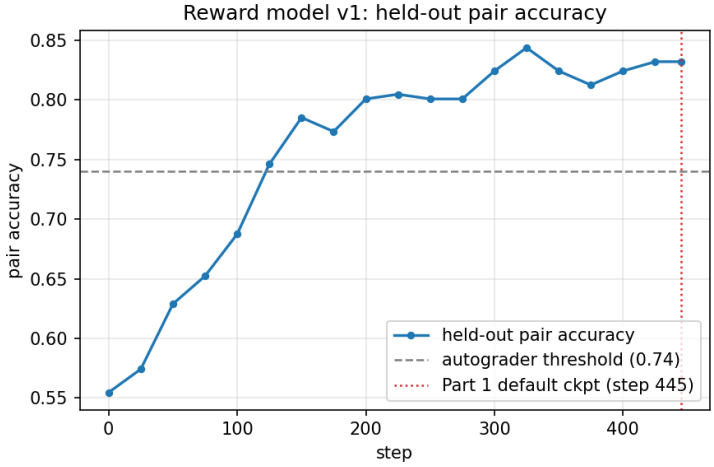

Figure 6: Held-out pair accuracy of reward model v1 on `test_prefs` during training. The 0.74 autograder threshold is the dashed horizontal line. The step 445 checkpoint used by all Part 1 runs (vertical line) sits in the post-peak plateau at 0.8398, comfortably above threshold. Earlier checkpoints (steps 150, 250, 350) are used as alternative reward signals in Section 4.2.2.

A.2 Offline Baselines (DPO, IPO, AOT)

DPO and IPO are trained with $\beta = 0.1$, AOT with $\beta = 0.2$, all for 3 epochs with learning rate `5e-5` and per-device batch size 4. Reported reference checkpoints: DPO at step 100, IPO at step 300, AOT at step 550.

A.3 Part 1 Online and Part 2 Shared Hyperparameters

All Part 1 online runs (GRPO, DrGRPO, GSPO) and all Part 2 ensembling/LOO/whitening runs share the hyperparameters in Table 5. Per-batch overrides for Part 2 (RM adapters, aggregation, baseline, whitening, training steps) are in Table 6; the RFPO and Online DPO sweep settings are in Table 7.

Table 5: Shared training hyperparameters for Part 1 online and Part 2 ensembling/LOO/whitening runs. Part 2 batches override the fields in Table 6.

Hyperparameter	Value
`model_name` / `reward_model_name`	`Qwen/Qwen2.5-1.5B-Instruct`
`dataset_name`	`wildchat_min4_judged_5k_v1`
`train_split` / `eval_split`	`train_gen` / `test_gen`
`batch_size` / `group_size`	16 / 4
`min_new_tokens` / `max_new_tokens`	32 / 256
`temperature` / `top_p`	0.8 / 0.95
`lr`	1e-5
`grad_accum_steps` / `ppo_epochs` / `minibatch_size`	2 / 2 / 8
`clip_eps`	0.2
`kl_coef`	0.01
`max_prompt_tokens` / `max_response_tokens`	700 / 512
`eval_limit` / `eval_interval` / `save_interval`	32 / 25 / 25

A.4 Per-Batch Overrides (Ensembling, LOO, Whitening, Ablations)

Table 6: Part 2 per-batch overrides on top of Table 5. RM adapters are taken at `step_000445`. The "$-$ X" rows under Batch 4 each remove one ingredient from Batch 3 while keeping the rest fixed.

Configuration	`steps`	RM adapters	`rm_aggregation`	`baseline`	`reward_whitening`
Batch 1 (agg sweep)	25	$r^{(2)}$+$r^{(3)}$	mean or min	`group_mean`	off
Batch 1 (group_max)	25	$r^{(2)}$	–	`group_max`	off
Batch 2	75	$r^{(2)}$+$r^{(3)}$	min	`group_loo`	off
Batch 3 (best)	75	$r^{(2)}$+$r^{(3)}$	min	`group_loo`	running
Batch 4: reseed (`seed=1`)	75	$r^{(2)}$+$r^{(3)}$	min	`group_loo`	running
Batch 4: $-$ ensemble	75	$r^{(2)}$	–	`group_loo`	running
Batch 4: $-$ LOO	75	$r^{(2)}$+$r^{(3)}$	min	`group_mean`	running
Batch 4: $-$ longer	25	$r^{(2)}$+$r^{(3)}$	min	`group_loo`	running

A.5 RFPO and Online DPO Sweep Configuration

All RFPO and Online DPO runs train for 75 steps with checkpoints every 25 steps, using single RM $r^{(1)}$ at `step_000445`. To hold rollouts per step constant at 128 across group sizes, `batch_size` is coupled to G (Table 7, rightmost column). The G=16 batch size of 8 also keeps the rollout buffer inside 80 GB H200 memory.

Table 7: RFPO and Online DPO sweep configurations on top of Table 5. G=2, k=1 Online DPO is the OAIF-canonical configuration with the RM as annotator.

Variant	`algo`	`filter_strategy`	k	G (`batch_size`)
Top-k RFPO	{grpo, dr_grpo, gspo}	`top_k`	4	8 (16), 16 (8)
Top-k + bot-k RFPO	{grpo, dr_grpo, gspo}	`top_k_bottom_k`	2, 4	8 (16), 16 (8)
Online DPO (`dpo_beta=0.1`)	`online_dpo`	`top_k_bottom_k`	1, 2, 3, 4	2 (32), 4 (32), 8 (16), 16 (8)

B Reproduction Commands

The full set of exact training, build, download, and grading commands used in this paper is in `commands.md` at the root of the project repository, `https://github.com/kavishkondap/185_final_project_llm_rl`. Sections in `commands.md` cover, in order: RM training (§1); Part 1 baselines (§2); the Part 2 RM-checkpoint sweep (§3); the ensembling/LOO/whitening batches (§4); the RFPO + Online DPO sweep (§5); the final autograder submission build (§6); and per-candidate grading (§7).

C Per-Batch Detailed Tables

This appendix contains the detailed per-batch tables (with pair-agree counts) referenced by the compact progression Table 3 in the main text, plus a per-recipe win-rate bar chart spanning all 30 ensembling/LOO/whitening runs (Figure 7).

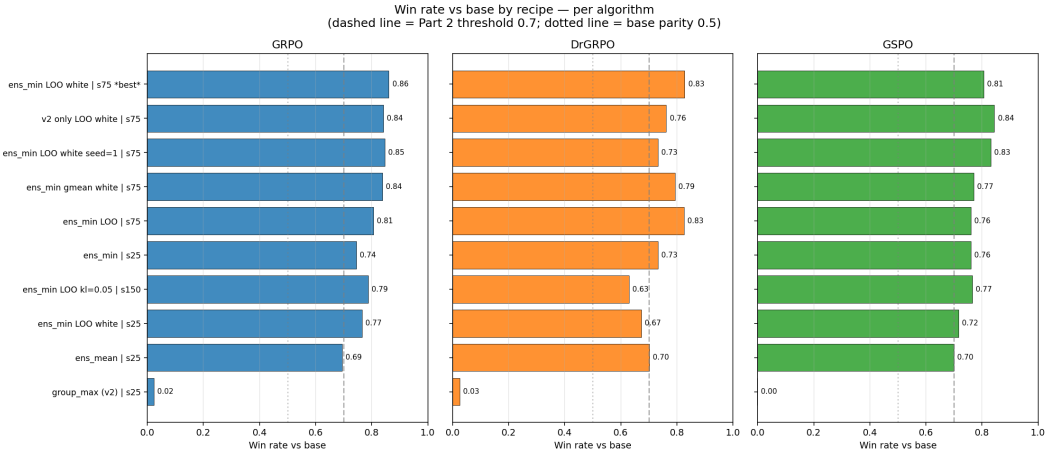

Figure 7: GPT-5.4 win rate against the frozen base for every Part 2 ensembling/LOO/whitening recipe, broken out per algorithm. "ens_min" / "ens_mean" denote min- vs. mean-aggregated two-RM ensembles ($r^{(2)}+r^{(3)}$); "v2 only" uses single-RM $r^{(2)}$; "LOO" replaces the group-mean baseline with leave-one-out; "white" adds running reward whitening; "group_max" is the ReMax-style greedy baseline; "s25/s75/s150" is the training step count; "kl=0.05" is the higher-KL Batch 3 attempt; "seed=1" is the reseed.

Table 8: Batch 1: ensembling aggregation and group-max baseline at 25 steps. Two-model ensemble is $r^{(2)}$ + $r^{(3)}$ at step 445; the single-RM group-max row uses $r^{(2)}$ only. Win rates use `policy_win_rate_pair_agree_usable`; pair-agree count in parentheses.

Baseline	Aggregation	GRPO	DrGRPO	GSPO
group_mean	mean (2-model)	69.4% (36)	70.2% (47)	70.0% (50)
group_mean	min (2-model)	74.5% (47)	73.3% (45)	76.0% (50)
group_max	mean (1-model)	2.4% (41)	2.6% (39)	0.0% (41)

Table 9: Batch 2: LOO baseline, min aggregation, 75 steps.

	GRPO	DrGRPO	GSPO
Win rate (pair-agree)	80.6% (67)	82.6% (69)	76.1% (71)

Table 10: Batch 3: adding running reward whitening to the Batch 2 configuration. Δ is relative to the corresponding Batch 2 run.

	Batch 3 (+ whitening)		Batch 2 (no whitening)
Algo	Win rate (pair-agree)	Δ	Win rate (pair-agree)
GRPO	**86.1%** (79)	+5.5 points	80.6% (67)
DrGRPO	82.8% (58)	+0.2 points	82.6% (69)
GSPO	80.6% (67)	+4.5 points	76.1% (71)

D Per-Checkpoint Win Rates for the 110-Candidate Sweep

We report the exact GPT-5.4 win rate and pair-agree-usable count (out of 128) for every graded checkpoint in the Part 2 sweep, broken down by method family. All runs share the fixed hyperparameters documented in Section 4.1; the only varying parameters are the algorithm (GRPO, DrGRPO, GSPO, or DPO loss), the group size G, the filter rule, and the checkpoint step. Rows within each table are sorted by win rate descending. Figure 8 shows the unfiltered baseline trajectories used as the comparison anchor. Filter "none" denotes an unfiltered group (no rejection sampling); "top-k" is the one-sided RFPO variant; "top-k+bot-k" is the two-sided RFPO variant; and "top-m+bot-b pairs" refers to the rank-aligned (chosen, rejected) pair construction used by Online DPO. Steps 25, 50, 75 correspond to the three checkpoints saved during each 75-step run.

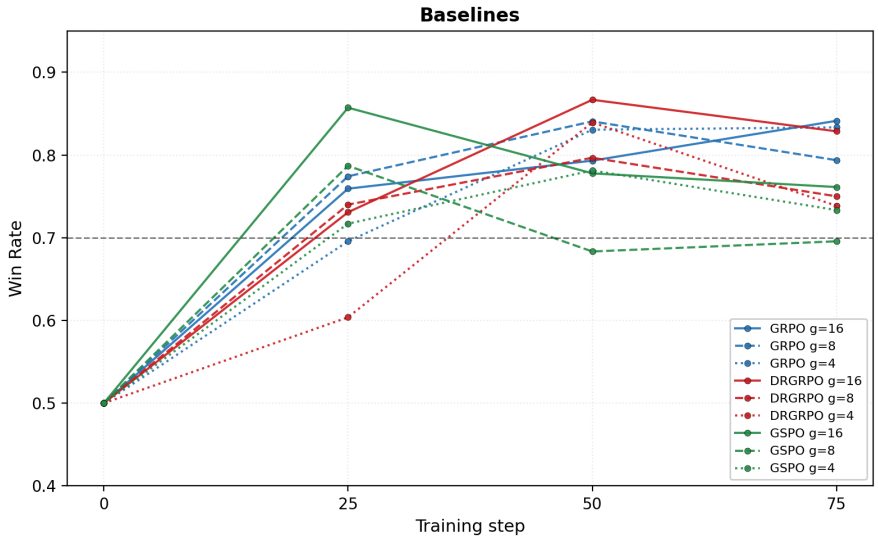

Figure 8: Win rate vs. training step for the unfiltered online baselines used as the RFPO/Online DPO comparison anchor. Color encodes algorithm; linestyle encodes group size G (solid: G=16, dashed: G=8, dotted: G=4).

E Qualitative Sample Comparisons

To complement the quantitative results, we include two prompts from the 128-prompt public evaluation set with side-by-side generations from (i) our strongest Part 1 online baseline (GRPO at step 25), (ii) a representative top-k-only RFPO checkpoint, and (iii) our strongest Part 2 online submission (Online DPO G=16, k=2 at step 75). Responses are truncated to fit; complete generations are in the submitted JSONLs.

Sample 1: ESL teacher reply. *Prompt:* "I am an online ESL teacher. Please help me compose a friendly, concise, conversational and easy to understand message to a potential student. Here is the

Table 11: Unfiltered baselines. All 27 graded checkpoints, sorted by win rate descending.

Algo	G	Filter	Step	Win rate	Usable
DrGRPO	16	none	50	86.7%	60/128
GSPO	16	none	25	85.7%	56/128
GRPO	16	none	75	84.1%	63/128
GRPO	8	none	50	84.1%	69/128
DrGRPO	4	none	50	83.9%	56/128
GRPO	4	none	75	83.3%	72/128
GRPO	4	none	50	83.1%	59/128
DrGRPO	16	none	75	82.9%	70/128
DrGRPO	8	none	50	79.7%	59/128
GRPO	8	none	75	79.4%	63/128
GRPO	16	none	50	79.3%	58/128
GSPO	8	none	25	78.7%	61/128
GSPO	4	none	50	78.1%	64/128
GSPO	16	none	50	77.8%	63/128
GRPO	8	none	25	77.4%	62/128
GSPO	16	none	75	76.1%	67/128
GRPO	16	none	25	75.9%	54/128
DrGRPO	8	none	75	75.0%	64/128
DrGRPO	8	none	25	74.0%	50/128
DrGRPO	4	none	75	73.8%	65/128
GSPO	4	none	75	73.3%	60/128
DrGRPO	16	none	25	73.1%	52/128
GSPO	4	none	25	71.7%	53/128
GRPO	4	none	25	69.6%	46/128
GSPO	8	none	75	69.6%	69/128
GSPO	8	none	50	68.3%	60/128
DrGRPO	4	none	25	60.4%	53/128

conversation so far: Julia is considering you as their potential tutor and wants to improve her English from B2 to C1 . . . ”

GRPO (Part 1, step 25): “Hello Julia! Great to hear that you’re looking to improve your English skills! Jumping from B2 to C1 is definitely achievable with the right guidance. What specific areas do you want to focus on in order to reach this level? Whether it’s reading comprehension, writing, speaking or listening, let me know what excites you most and we can tailor our lessons . . . ”

Top-k-only RFPO (GSPO G=16, k=4, step 75): “Hello Julia, I’m happy to chat with you about your English learning goals. I’d love to hear more about what you want to achieve and how I can help you reach your goals. Let’s connect and see if we’re a good fit.”

Online DPO (G=16, k=2, step 75): “Here’s a friendly, concise, conversational message to help guide Julia towards choosing the best tutor: ‘Hey Julia! Thanks for reaching out. My goal is to help you reach C1 level in English. We can work together to tackle any areas where you need improvement and build confidence. Let me know if there’s anything specific you’d like to focus on or how we can . . . ’ ”

Observation. The top-k-only RFPO output is shorter, more generic, and decorated with emoji that read like prompt acknowledgement rather than substantive help, consistent with the “policy converges toward a narrow mode favored by the RM” failure mode discussed in Section 4.2.5. The Online DPO output is on-task (it actually drafts the message the user asked for) and structurally similar in quality to GRPO but more concise.

Sample 2: magazine article ideas. *Prompt:* “Can you give me ten ideas for original magazine articles regarding relatable issues facing contemporary women?”

GRPO (Part 1, step 25): “1. ‘The Unspoken Truths: How Society’s Expectations Affect Women’s Mental Health’ 2. ‘From the Kitchen to the Boardroom: Balancing Work and Family in Today’s World’ 3. ‘Navigating Menopause: The Challenges and Coping Strategies for Modern Women’ 4. ‘The Impact of Social Media on Body Image and Self-Esteem Among Young Women’ 5. ‘Finding Balance: Overcoming . . . ’ ”

Table 12: Top-k RFPO (RAFT-style one-sided filter). All 21 graded checkpoints, sorted by win rate descending. Three (algo, G=8, k=4, step 25) rows appear twice because two separate training runs landed on the same nominal configuration: a dedicated 25-step run and a 75-step run sampled at step 25 (these have different sample seeds and rollout draws, not different hyperparameters). The full file names are listed in `part_2_generations/top_filter_results.json`.

Algo	G	Filter	Step	Win rate	Usable
GSPO	16	top-4	75	62.3%	61/128
GSPO	8	top-4	50	53.3%	60/128
GSPO	8	top-4	25	52.9%	51/128
GRPO	16	top-4	75	52.7%	55/128
GRPO	8	top-4	25	52.5%	61/128
DrGRPO	16	top-4	75	51.9%	54/128
DrGRPO	16	top-4	50	50.0%	64/128
DrGRPO	8	top-4	75	50.0%	58/128
DrGRPO	8	top-4	50	49.3%	67/128
GSPO	8	top-4	25	49.2%	63/128
GSPO	16	top-4	25	49.2%	63/128
GRPO	8	top-4	25	49.2%	63/128
DrGRPO	8	top-4	25	47.6%	63/128
GSPO	16	top-4	50	47.5%	59/128
GRPO	8	top-4	50	47.3%	55/128
GSPO	8	top-4	75	46.7%	60/128
DrGRPO	8	top-4	25	45.6%	57/128
GRPO	16	top-4	50	45.3%	64/128
DrGRPO	16	top-4	25	44.6%	56/128
GRPO	16	top-4	25	43.9%	57/128
GRPO	8	top-4	75	42.6%	61/128

Top-k-only RFPO (GSPO G=16, k=4, step 75): "1. The challenges of balancing work and family life 2. The impact of body image on women's self-esteem 3. How to cope with anxiety and stress 4. The importance of mental health awareness in women 5. How to negotiate successfully 6. The impact of social media on women's self-image 7. How to deal with sexual harassment . . . "

Online DPO (G=16, k=2, step 75): "Sure! Here are ten original article ideas focusing on relatable issues facing contemporary women: 1. 'The Female Experience: Balancing Work and Family' 2. 'From Surviving to Thriving: Overcoming Gender-Based Violence' 3. 'Navigating the Digital Divide: Women's Access to Technology' 4. 'The Power of Self-Love: Building Confidence in Today's World' 5. . . . ' "

Observation. On this prompt the three policies look similar in length and topic coverage, but the GRPO Part 1 baseline and the Online DPO submission both produce titled ideas (with explanatory subtitles in the GRPO case), whereas the top-k-only RFPO output collapses to a bare bullet list of themes. The judge consistently prefers the titled, more specific framings, which is visible in the win-rate gaps in Table 1 and Appendix 14.

Offline-method samples (DPO vs. IPO vs. AOT)

The Part 1 win-rate ordering on the 128-prompt evaluation set is AOT (75.5%) > IPO (73.5%) > DPO (70.6%). Sample inspection shows a clear length and on-task pattern that explains the ordering: across all 128 prompts the mean response length is **$\sim$200 / $\sim$205 / $\sim$230 words for DPO / IPO / AOT**, and DPO has the shortest minimum response ($\sim$35 words, vs. AOT's $\sim$80). Qualitatively the AOT outputs commit more fully to the user's request while DPO sometimes under-commits with a single sentence, consistent with DPO's saturating logistic loss pushing the chosen-rejected margin past the point where the response is actually informative. Two illustrative prompts:

Offline Sample 1: short creative writing prompt. *Prompt:* "Write a Vince McMahon vs Kevin Owens with Owens's win in 20 seconds."

DPO ($\sim$30 words): "In the intense Vince McMahon vs Kevin Owens match, Owens delivers a powerful finish, dominating the final moments to secure his victory, leaving fans on the edge of their seats."

Table 13: Top-k + bottom-k RFPO (GRPO clipped surrogate). All 27 graded checkpoints, sorted by win rate descending.

Algo	G	Filter	Step	Win rate	Usable
DrGRPO	8	top-2+bot-2	50	84.1%	63/128
GSPO	8	top-2+bot-2	50	83.1%	71/128
DrGRPO	16	top-2+bot-2	50	80.7%	57/128
GRPO	8	top-2+bot-2	50	80.0%	75/128
GRPO	16	top-2+bot-2	50	78.9%	71/128
GSPO	16	top-2+bot-2	75	78.6%	56/128
GRPO	16	top-4+bot-4	50	78.5%	65/128
GRPO	8	top-2+bot-2	75	78.1%	73/128
GSPO	16	top-4+bot-4	50	76.5%	68/128
GSPO	16	top-4+bot-4	25	75.8%	66/128
GSPO	16	top-2+bot-2	50	74.6%	71/128
DrGRPO	16	top-4+bot-4	50	74.6%	67/128
GRPO	8	top-2+bot-2	25	73.8%	42/128
DrGRPO	8	top-2+bot-2	75	73.0%	63/128
DrGRPO	16	top-2+bot-2	25	72.7%	44/128
GRPO	16	top-2+bot-2	25	72.3%	47/128
GSPO	8	top-2+bot-2	25	72.1%	61/128
DrGRPO	16	top-2+bot-2	75	70.8%	72/128
DrGRPO	16	top-4+bot-4	25	69.6%	56/128
GRPO	16	top-4+bot-4	75	69.0%	58/128
GSPO	8	top-2+bot-2	75	68.9%	61/128
GSPO	16	top-2+bot-2	25	68.6%	51/128
GRPO	16	top-2+bot-2	75	68.5%	73/128
GRPO	16	top-4+bot-4	25	67.3%	55/128
DrGRPO	8	top-2+bot-2	25	65.0%	60/128
DrGRPO	16	top-4+bot-4	75	61.9%	63/128
GSPO	16	top-4+bot-4	75	60.3%	58/128

IPO ($\sim$85 words): "Kevin Owens defeated Vince McMahon in an intense match that showcased the power and determination of both wrestlers. Despite being outnumbered and outmatched at first, Kevin Owens fought back with incredible strength and skill to secure the victory.. . . The crowd roared as Owens delivered a series of devastating moves . . . "

AOT ($\sim$85 words): "Vince McMahon: 'Kevin, you're the champion!' Kevin Owens: (smirking) 'Yeah, well I won this one.' (Countdown to the finish) Kevin Owens executes the move and wins it. (end credits roll) Kevin Owens is declared the new champion in 20 seconds.. . . "

Observation. DPO produces a single rambling sentence that does not engage with the "20 seconds" constraint. IPO writes paragraph prose. AOT actually engages with the format request by writing scripted dialogue with stage directions, matching what a 20-second wrestling spot would actually look like.

Offline Sample 2: rewrite-and-extend prompt. *Prompt:* "Reframe and restructure in more polished, coherent way with the use of transition words . . . and complete the unfinished parts more analytically and creatively. . . . [user text]"

DPO ($\sim$115 words): Truncates the rewrite, ending mid-sentence at $\sim$60% of the user's source text length.

IPO ($\sim$120 words): Similar truncation to DPO, slightly more polished prose but also stops mid-sentence before completing the requested extension.

AOT ($\sim$285 words): Produces a full-length rewrite that explicitly uses transition words ("In this context," "By doing so") and completes the unfinished parts as requested.

Observation. On a multi-part instruction with both a style ask ("polished, coherent, with transitions") and a content ask ("complete the unfinished parts"), only AOT delivers both. DPO and IPO match

Table 14: Online DPO with top-m+bottom-b rank-aligned (chosen, rejected) pairs. All 35 graded checkpoints, sorted by win rate descending.

Algo	G	Filter	Step	Win rate	Usable
DPO	16	top-2+bot-2 pairs	75	87.7%	65/128
DPO	8	top-2+bot-2 pairs	50	84.8%	66/128
DPO	16	top-4+bot-4 pairs	50	84.1%	63/128
DPO	16	top-1+bot-1 pairs	75	83.9%	56/128
DPO	8	top-2+bot-2 pairs	75	82.8%	64/128
DPO	16	top-2+bot-2 pairs	25	81.2%	48/128
DPO	16	top-3+bot-3 pairs	25	81.0%	58/128
DPO	16	top-2+bot-2 pairs	50	81.0%	58/128
DPO	8	top-4+bot-4 pairs	25	80.3%	61/128
DPO	4	top-2+bot-2 pairs	50	80.0%	50/128
DPO	16	top-3+bot-3 pairs	50	79.7%	59/128
DPO	8	top-3+bot-3 pairs	75	79.2%	53/128
DPO	16	top-4+bot-4 pairs	75	79.1%	67/128
DPO	4	top-2+bot-2 pairs	75	78.9%	57/128
DPO	16	top-3+bot-3 pairs	75	78.3%	69/128
DPO	8	top-3+bot-3 pairs	50	78.1%	64/128
DPO	8	top-3+bot-3 pairs	25	78.0%	50/128
DPO	8	top-4+bot-4 pairs	50	77.9%	68/128
DPO	2	top-1+bot-1 pairs	75	76.5%	51/128
DPO	16	top-4+bot-4 pairs	25	75.8%	62/128
DPO	16	top-1+bot-1 pairs	50	75.5%	49/128
DPO	8	top-2+bot-2 pairs	25	75.4%	57/128
DPO	8	top-1+bot-1 pairs	50	75.4%	65/128
DPO	4	top-1+bot-1 pairs	50	75.0%	48/128
DPO	4	top-1+bot-1 pairs	75	74.5%	55/128
DPO	8	top-1+bot-1 pairs	75	73.9%	69/128
DPO	2	top-1+bot-1 pairs	50	73.6%	53/128
DPO	8	top-2+bot-2 pairs	25	73.3%	60/128
DPO	8	top-1+bot-1 pairs	25	73.2%	56/128
DPO	4	top-2+bot-2 pairs	25	68.3%	41/128
DPO	16	top-1+bot-1 pairs	25	67.5%	40/128
DPO	8	top-4+bot-4 pairs	75	65.5%	55/128
DPO	8	top-1+bot-1 pairs	25	65.4%	52/128
DPO	2	top-1+bot-1 pairs	25	63.3%	30/128
DPO	4	top-1+bot-1 pairs	25	62.5%	40/128

the style ask but truncate before fulfilling the completion ask, the pattern that drives DPO's lower judge-credit rate.