Ultra-DPO: Reimagining LLM Alignment as a Semi-Supervised Task

Li He[1,2] **Jiaxian Guo**[3] **Yaqing Wang**[4] **Yue Yang**[5] **He Zhao**[2]
Stephen Wan[2] **Dadong Wang**[2] **Lina Yao**[6] **Tongliang Liu**[1]*
[1] The University of Sydney [2] CSIRO, Data61 [3] Google Research
[4] Beijing Institute of Mathematical Sciences and Applications
[5] Maincode [6] University of New South Wales

Abstract

Direct Preference Optimization (DPO) offers a streamlined one-stage alternative to Reinforcement Learning from Human Feedback (RLHF). However, our theoretical analysis reveals a structural distinction between the two: DPO and RLHF optimize alignment on different data distributions with different supervision sources. Specifically, DPO learns from ground-truth human preferences but within the off-policy support of an offline dataset, while RLHF optimizes on the on-policy rollouts but learns from pseudo-labels produced by a reward proxy. This distinction explains how each method may fail: DPO alignment collapses when the policy drifts away from the offline dataset support, and RLHF alignment is biased due to the extrapolation error of the reward proxy on out-of-distribution rollouts. To mitigate both failure modes, we propose a semi-supervised alignment objective: ground-truth preferences anchor alignment within the dataset support, and pseudo-labels on online rollouts extend it beyond. Applying DPO's reward reparameterization to this objective yields Ultra-DPO, a one-stage algorithm that augments direct human alignment with an online self-alignment mechanism. Across multiple benchmarks, Ultra-DPO consistently outperforms DPO, RLHF, and online preference learning methods. Our implementation is available at `github.com/heli-stand/Ultra-DPO`.

1 Introduction

Aligning large language models (LLMs) [Abdin et al., 2025, Olmo et al., 2025, Adcock et al., 2026, Liu et al., 2026] with human preferences has emerged as a central problem in modern post-training, with two dominant paradigms: Reinforcement Learning from Human Feedback (RLHF) [Casper et al., 2023, Kaufmann et al., 2024] and Direct Preference Optimization (DPO) [Rafailov et al., 2023]. RLHF takes a two-stage approach: a reward model is first trained on offline human preferences, and the LLM is then optimized against this learned reward proxy via online RL. DPO bypasses the reward modeling stage entirely, allowing LLMs to be optimized directly on the offline preference dataset. DPO is widely regarded as a streamlined one-stage alternative to RLHF, eliminating the complexity of reward model training and online RL while reportedly achieving comparable alignment quality.

However, this conventional framing on the procedural differences obscures a structural distinction between the two methods. Specifically, our analysis reveals that DPO and RLHF optimize alignment on different data distributions with different supervision sources. As shown in Figure 1(a), DPO learns from ground-truth human preferences [Stiennon et al., 2020, Wang et al., 2024b, Ji et al., 2025], but with the off-policy support of a fixed offline dataset. By contrast, RLHF optimizes on the on-policy rollouts from the current policy [Song et al., 2024, Tajwar et al., 2024], but learns from pseudo-labels produced by the learned reward proxy, as illustrated in Figure 1(b). Such a structural

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

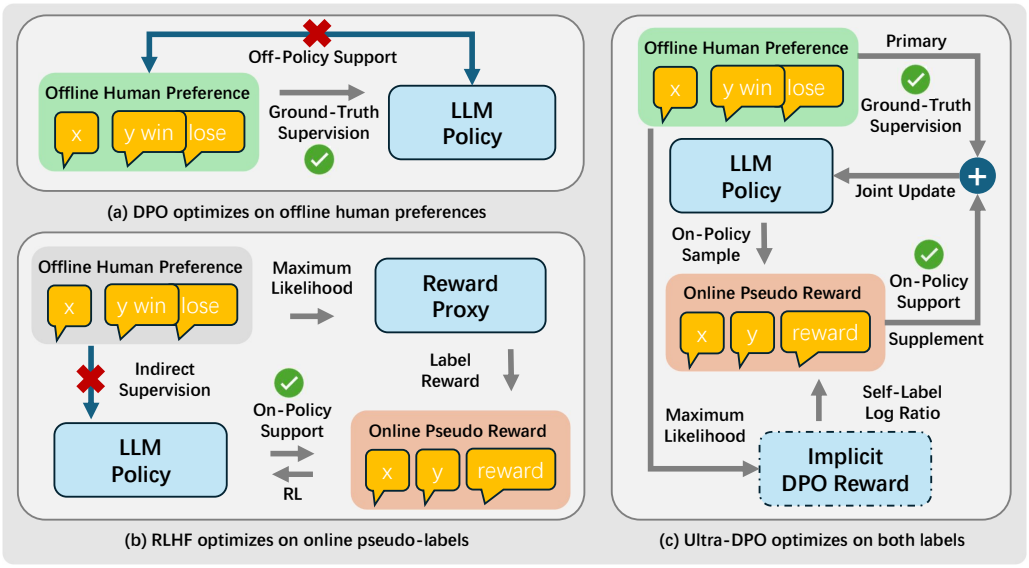

Figure 1: **Alignment paradigms compared. (a) DPO** directly aligns from ground-truth human preferences but is restricted to the off-policy support of a fixed offline dataset; alignment collapses when the policy drifts away from this support. **(b) RLHF** optimizes on on-policy rollouts but against pseudo-labels from a learned reward proxy; alignment is biased by the proxy's extrapolation error on out-of-distribution rollouts. **(c) Ultra-DPO (ours)** jointly optimizes on offline human preferences and online pseudo-labels in a single stage to mitigate both failure modes. Ground-truth supervision anchors alignment to limit the effects of reward-proxy bias, while on-policy pseudo-labels extend alignment beyond the offline support to mitigate distribution mismatch.

distinction implies that both methods may fail during the alignment process. On one hand, DPO alignment collapses as the policy drifts away from the offline dataset support, because the increasing divergence between data and policy compromises the monotonic improvement guarantee [Schulman et al., 2015]. On the other hand, RLHF alignment is biased by the extrapolation error of the reward proxy [Laidlaw et al., 2023, Dai et al., 2025], because the proxy is queried on online rollouts that lie increasingly further from its training distribution. Effective alignment therefore requires addressing both limitations simultaneously, which motivates a new alignment paradigm.

To this end, we propose a semi-supervised approach to LLM alignment as shown in Figure 1(c). The insight is to treat ground-truth human preferences as the primary signal and on-policy reward labels as a supplement. The former anchors alignment within the human preference support, helping limit the effects of reward misestimation on out-of-distribution rollouts; the latter extends alignment to on-policy rollouts, helping mitigate the distribution mismatch associated with off-policy collapse. Applying DPO's reward reparameterization to this semi-supervised objective yields Ultra-DPO, a one-stage algorithm that augments offline preference alignment with an online self-alignment mechanism. Ultra-DPO addresses both structural failure modes identified above while retaining DPO's practical advantage of avoiding explicit reward model training. Without additional computational overhead compared to a standard RLHF pipeline, our approach outperforms a comprehensive set of preference learning and RLHF baselines. We summarize our contributions as follows:

- We identify a structural distinction between DPO and RLHF along optimization distribution and supervision source. This predicts a failure mode for each method, and we verify both empirically.
- We propose a semi-supervised alignment objective that addresses both failure modes. From this objective, we derive Ultra-DPO, a one-stage alignment algorithm that augments DPO with an online self-alignment mechanism.
- We conduct comprehensive experiments showing that Ultra-DPO consistently outperforms preference learning methods (DPO, IPO, SimPO) and RLHF methods (PPO, RLOO, GRPO, DAPO).

2 Related Work

Direct Alignment Algorithms. Several offline direct alignment algorithms (DAAs) have been proposed to address specific limitations of DPO. IPO [Azar et al., 2024] reframes the objective to prevent overfitting to the implicit reward model through a regularized loss formulation, while KTO [Ethayarajh et al., 2024] relaxes the pairwise preference requirement by learning from individual binary feedback signals. CPO [Xu et al., 2024a] and SimPO [Meng et al., 2024] eliminate the reference model dependency by assuming a uniform reference policy, and ORPO [Hong et al., 2024] further simplifies alignment by removing the separate SFT stage. To address limitations of offline alignment, online (or iterative) DAAs [Guo et al., 2024] leverage on-policy preference labels to guide policy updates beyond static offline datasets. These methods typically implement preference judges using pre-trained reward models [Chen et al., 2024b, Xu et al., 2024b] or verifiable reward functions [Pang et al., 2024, Tu et al., 2025]. While our method targets the same limitation as online DAAs, we achieve this without external supervision from reward models or generative LLM feedback.

Self-Aligning LLMs. The reliance on extensive human annotations for alignment presents significant scalability challenges, motivating research into self-alignment methods where models autonomously improve with minimal human supervision. One prominent approach uses synthetic data generation, where models create their own SFT datasets through self-critique and revision based on predefined principles [Bai et al., 2022b] or self-generated instruction-response pairs [Wang et al., 2023]. Beyond SFT, another line of work focuses on creating preference datasets using self-scoring prompts [Yuan et al., 2024, Wu et al., 2025] or self-generated rankings [Wang et al., 2024a, Lu et al., 2025]. Additionally, research on weak-to-strong generalization [Burns et al., 2023] explores settings where weaker models supervise stronger ones, offering insights into scalable oversight. The most closely related work is by Chen et al. [2024a], which proposes a self-alignment stage following standard post-training that uses self-derived implicit DPO rewards. This work validates that implicit rewards learned during DPO contain sufficient signal to guide further self-alignment, demonstrating that preference learning itself generates reusable supervision. Building on this insight, our method differs by integrating self-alignment directly into the preference learning stage, enabling LLMs to learn from self-generated DPO rewards alongside human preferences. Consequently, our approach achieves one-stage RLHF without requiring additional post-training phases.

3 Preliminary

This section establishes the mathematical foundation for our theoretical analysis and proposed method. We first review the two-stage RLHF pipeline: reward modeling and preference alignment. We then present DPO and its derivation, as well as online DPO, where the expectation is computed over preferences from iteratively collected on-policy samples.

Reward Modeling. In the task of LLM alignment, we first try to capture human preferences by learning a reward model r_ϕ out from a binary preference dataset $\mathcal{D}$, where a response y_w is more preferable over another response y_l given the same prompt x. By framing it as a logistic regression problem using the Bradley-Terry model [Bradley and Terry, 1952], we derive the optimal reward model by optimizing the log-likelihood objective:

$$\mathcal{J}_{\text{RM}}(r_\phi) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}(x,y_w,y_l)}\big[\log\sigma(r_\phi(x,y_w) - r_\phi(x,y_l))\big], \tag{1}$$

Preference Alignment. In the second stage, we train a language model π_θ to output reward-maximizing responses based on the trained reward model r_ϕ. This is achieved via maximizing the reward expectation with a reference regularization [Ziegler et al., 2019, Stiennon et al., 2020]:

$$\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}(x),y\sim\pi_\theta(\cdot|x)}\Big[r_\phi(x,y) - \beta\mathbb{D}_{\text{KL}}\big[\pi_\theta(\cdot|x)||\pi_{\text{ref}}(\cdot|x)\big]\Big], \tag{2}$$

where $\mathcal{D}(x)$ is the prompt dataset, π_{ref} is the reference policy, β is the regularization coefficient. The KL divergence term prevents the policy from moving far from the reference distribution, mitigating the reward hacking issue [Eisenstein et al., 2023, Gao et al., 2023, Rafailov et al., 2024].

DPO. Rafailov et al. [2023] propose a two-in-one preference alignment algorithm directly based on offline human preferences. The derivation starts from the observation that KL-constrained RL objectives like Equation (2) have closed-form solutions, following previous works [Peters and Schaal,

2007, Peng et al., 2019, Korbak et al., 2022, Go et al., 2023]. This gives us a way to reparameterize the reward model using the learning policy π_θ and the reference policy π_{ref}:

$$r_\phi(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x), \tag{3}$$

where $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r_\phi(x, y)\right)$ is the partition function ensuring the solution policy is a valid probability distribution. By substituting Equation (3) in Equation (1), we derive the DPO objective in terms of only the learning policy for direct human preference alignment:

$$\mathcal{J}_{\text{DPO}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}(x,y_w,y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right]. \tag{4}$$

Online DPO. To address the off-policy issue of standard DPO, online DPO was proposed [Xiong et al., 2023, Guo et al., 2024], which replaces the offline dataset with a dynamically constructed dataset $\mathcal{D}_{\text{Online}}(x, y_w, y_l)$. Such a dataset is constructed via a combined procedure of (1) two-shot generation of $(y_1, y_2) \sim \pi_\theta(\cdot|x)$, and (2) online preference labeling $(y_w, y_l) \sim p_{\text{judge}}(y_1, y_2|x)$ using a preference judge p_{judge}. A pre-trained reward model via Equation (1) often serves as the judge, the chosen response y_w is determined as the one with the higher reward score.

4 What DPO and RLHF Actually Optimize

DPO and RLHF are conventionally distinguished by whether reward modeling is explicit. We argue this framing overlooks a deeper structural distinction. The two methods differ along two axes: 1) supervision source, which determines whether feedback comes directly from human annotations or from a learned reward model; 2) optimization distribution, which determines whether feedback is applied to fixed offline comparisons or current-policy generations. This distinction highlights two limitations: improvements on offline comparisons may not transfer to generated responses, while feedback on current generations depends on reward-model generalization. We develop this perspective in Sections 4.1 and 4.2 and examine its empirical implications in Section 4.3. Appendix B further discusses the relationship between online DPO and RLHF within this framework.

4.1 DPO: Ground Truth, but Off-Policy

DPO optimizes Equation (4) as a supervised loss over offline preference pairs. While prior work has analyzed properties of DPO's optimization landscape [Xu et al., 2024b], its implicit alignment objective has not been characterized. We characterize it as below with detailed proof in Appendix A.1:

Theorem 4.1 *Let $\mathcal{D}$ be a preference dataset over tuples (x, y_w, y_l), and assume $\beta > 0$, $\pi_\theta(y|x) > 0$ and $\pi_{\text{ref}}(y|x) > 0$ for all (x, y) in $\mathcal{D}$. Then DPO can be interpreted as optimizing the Bradley-Terry objective whose sample-wise reward margin is induced by the within-pair optimality condition of the following alignment objective:*

$$\mathcal{J}_{\text{DPO-Align}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}(x,y_w,y_l)} \left[\sum_{y\in\{y_w,y_l\}} \pi_\theta(y|x) \left(r_\phi(x, y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right]. \tag{5}$$

The DPO's implicit alignment objective sees ground-truth labels, but only on a support that is fixed offline and disjoint from where π_θ actually places its mass. The inner sum ranges only over the two completions labeled by humans, because the reward reparameterization is partial: $r_\phi(x, y)$ is identified only on response pairs with preference labels. Within the labeled pair, however, DPO's supervision is reliable: $r_\phi(x, y)$ is fit on exactly these in-distribution responses.

The structural failure mode follows directly. Taking improvement steps on the offline distribution violates the policy improvement guarantee [Kakade and Langford, 2002, Schulman et al., 2015], which require the data distribution to remain close to the policy being improved. As the estimated improvement direction diverges from the true one, alignment collapses on the policy's actual rollouts.

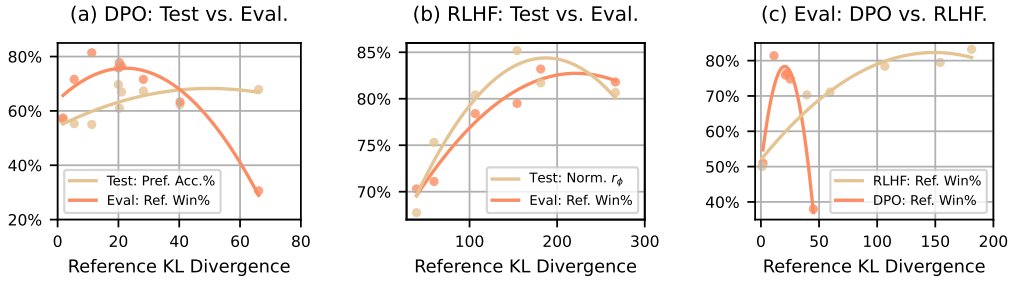

Figure 2: Empirical comparison of offline DPO and GRPO-based RLHF on HH-RLHF. (a) DPO's held-out preference accuracy improves while its win rate against the reference policy peaks and then declines. (b) For RLHF, the learned reward score and win rate improve together over the evaluated training range. (c) In the low-KL regime, DPO achieves higher observed win rates at comparable KL. KL is measured relative to π_{ref}.

4.2 RLHF: On-Policy, but Pseudo-Labeled

To make the structural comparison with DPO, we rewrite the RLHF alignment objective in Equation (2) as a sum over the on-policy support, with derivation in Appendix A.2:

$$\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}(x)}\left[\sum_{y\in\text{supp}(\pi_\theta(\cdot|x))} \pi_\theta(y|x)\left(r_\phi(x,y) - \beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right]. \tag{6}$$

Equation (5) and 6 share the same KL-regularized objective and differ only along the two axes of our framework. On the optimization axis, the inner sum ranges over the on-policy support rather than the labeled pair $\{y_w, y_l\}$, so RLHF optimizes alignment on the policy's actual behaviors and thus avoids DPO's off-policy collapse. On the supervision axis, however, the reward model is queried on online rollouts that increasingly lie outside its training support, as π_θ drifts from π_{ref}.

This mismatch makes RLHF alignment biased by the reward-proxy extrapolation error. The reliability of r_ϕ degrades with distance from its training distribution [Gao et al., 2023, Dai et al., 2025], so on-policy rollouts that drift from this support receive biased reward estimates. Optimizing against this biased proxy causes π_θ to improve under r_ϕ while diverging from ground-truth alignment.

4.3 Empirical Validation

To examine the predicted failure modes, we run controlled experiments on HH-RLHF, using GRPO as the RLHF implementation. Full experimental details are in Appendix C.2.

DPO: preference fitting and generation quality diverge. Figure 2(a) shows that DPO's preference accuracy on held-out pairs continues to improve as KL increases, while its win rate against π_{ref} peaks and then declines. This pattern demonstrates that better generalization to offline comparisons may not translate into better generations. It is consistent with the off-policy generalization gap identified.

RLHF: reward and win rate improve together. In Figure 2(b), the learned reward score and win rate improve together over the evaluated training range. Thus, the decline observed for DPO does not occur in this RLHF run. Joint improvement remains compatible with imperfect reward feedback: reward-model error can limit the gains achieved without producing an absolute decline in win rate.

Performance at comparable KL. We further compare the methods in the low-KL regime, before the decline observed for DPO. Figure 2(c) shows that DPO achieves higher observed win rates at comparable KL, giving a better KL–win-rate trade-off in this regime. This result is consistent with a benefit from direct preference supervision.

These observations motivate retaining direct supervision from human preference pairs while incorporating reward feedback on current-policy generations. The resulting combination can connect observed preferences to the behaviors that emerge as the policy evolves.

5 Ultra-DPO

After identifying the limitations of existing alignment approaches, our goal is to propose a new alignment objective that addresses both failure modes for more effective LLM alignment. To this end, we propose a semi-supervised objective that optimizes the policy on both offline human preferences and on-policy rewards. Our key insight is to use ground-truth preferences as the primary signal to limit the effects of reward-proxy bias, with on-policy reward labels providing a supplementary signal to mitigate the distribution mismatch associated with off-policy collapse. From this objective, we derive Ultra-DPO, a policy-only one-stage algorithm that augments direct preference alignment with an online self-alignment mechanism. With appropriate weighting and regularization, the resulting update can mitigate both failure modes without explicit reward model training.

Derivation. We formulate the Ultra-DPO objective by jointly optimizing reward modeling and semi-supervised alignment. Combining the reward modeling objective in Equation (1) with the DPO and RLHF alignment objectives in Equations (2) and (5), we define:

$$\mathcal{J}_{\text{Ultra-DPO}}(r_\phi, \pi_\theta) = \mathcal{J}_{\text{RM}}(r_\phi) + \underbrace{\mathcal{J}_{\text{DPO-Align}}(\pi_\theta) + \lambda \cdot \mathcal{J}_{\text{RLHF-Align}}(\pi_\theta)}_{\mathcal{J}_{\text{Semi-Align}}\text{: Semi-Supervised Alignment}},$$

where $\lambda > 0$ controls the weight of the supplementary on-policy signal. Let β_1 and β_2 be the KL coefficients of the DPO and RLHF alignment terms respectively, we expand $\mathcal{J}_{\text{Semi-Align}}$ and obtain:

$$\mathcal{J}_{\text{Ultra-DPO}}(r_\phi, \pi_\theta) = \mathcal{J}_{\text{RM}}(r_\phi) + \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x), y\sim\pi_\theta(\cdot|x)}\Big[\text{sg}(r_\phi(x,y)) - \beta_2 \mathbb{D}_{\text{KL}}\big[\pi_\theta \| \pi_{\text{ref}}\big]\Big] + \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}(x,y_w,y_l)}\left[\sum_{y\in\{y_w,y_l\}} \pi_\theta(y|x)\left(\text{sg}(r_\phi(x,y)) - \beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right] \tag{7}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator, which prevents gradients from backpropagating through reward labeling and ensures the semi-supervised objective updates only the policy model.

For sufficiently small λ, we approximate the solution to $\mathcal{J}_{\text{Semi-Align}}$ by that of $\mathcal{J}_{\text{DPO-Align}}$. Using the offline alignment term's within-pair optimality condition, we apply DPO's reward reparameterization in Equation (3) to eliminate the offline alignment subproblem and express the reward-modeling likelihood $\mathcal{J}_{\text{RM}}$ as the standard DPO objective $\mathcal{J}_{\text{DPO}}$. This gives the reduced formulation

$$\mathcal{J}_{\text{Ultra-DPO}}(r_\phi, \pi_\theta) := \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x),\, y\sim\pi_\theta(\cdot|x)}\left[\text{sg}\big(r_\phi(x,y)\big) - \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].$$

Optimizing $\mathcal{J}_{\text{DPO}}$ fits the pairwise margins of the implicit reward $\beta_1 \log\big(\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)\big)$ to the observed preference labels. We reuse this learned score as a detached pseudo-reward on current-policy generations, replacing r_ϕ in the on-policy term. The resulting Ultra-DPO objective depends only on the policy and requires no explicit reward model:

$$\mathcal{J}_{\text{Ultra-DPO}}(\pi_\theta) := \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x),\, y\sim\pi_\theta(\cdot|x)}\left[\text{sg}\left(\beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) - \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].$$

Appendix A.3 details the within-pair reduction and the approximations underlying this formulation.

Gradient Analysis. To better understand the alignment dynamics of Ultra-DPO, we analyze its gradient with respect to θ, with the full derivation in Appendix A.4:

$$\nabla_\theta \mathcal{J}_{\text{Ultra-DPO}}(\pi_\theta) = \nabla_\theta \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x), y\sim\pi_\theta}\Big[(\beta_1 - \beta_2) \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \nabla_\theta \log \pi_\theta(y|x)\Big].$$

Beyond the DPO update, Ultra-DPO performs online self-alignment whose behavior is controlled by the sign of $(\beta_1 - \beta_2)$: positive values encourage deviation from π_{ref} (intrinsic reward); negative values regularize toward π_{ref} (conservative penalty); zero eliminates self-alignment entirely.

Empirically, the configuration $\beta_1 > \beta_2$ consistently outperforms alternatives. We attribute this to the asymmetric role of the two coefficients: β_1 regularizes the offline DPO term, which must guard not only against preference over-optimization but also against the offline bias [Fujimoto et al.,

Algorithm 1 Ultra-DPO

Input: preference dataset $\mathcal{D}(x, y_w, y_l)$, reference model π_{ref}, training steps T, regularization coefficients β_1, β_2, self-alignment coefficient λ, stop gradient operator $\text{sg}(\cdot)$
Initialize $\pi_\theta = \pi_{\text{ref}}$, $\pi_{t=0} = \pi_{\text{ref}}$.
for $t = 0$ **to** $T - 1$ **do**
  Sample a prompt x and a preference pair y_w, y_l from $\mathcal{D}(x, y_w, y_l)$
  Calculate $\mathcal{J}_{\text{DPO}} = \log \sigma \left(\beta_1 \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta_1 \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$
  K-shot responses $\{y_i\}_{i=1}^K$ from $\pi_t(\cdot|x)$
  Calculate Advantage $A(x, y_i) = \beta_1 \log \frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)} - \frac{1}{K} \sum_{n=1}^K \beta_1 \log \frac{\pi_\theta(y_n|x)}{\pi_{\text{ref}}(y_n|x)}$
  Calculate $\hat{\mathcal{J}}_{\text{Self-Align}} = \frac{1}{K} \sum_{i=1}^K \frac{\pi_\theta(y_i|x)}{\pi_t(y_i|x)} \left[\text{sg}\big(A(x, y_i)\big) - \beta_2 \log \frac{\pi_\theta(y_i|x)}{\pi_{\text{ref}}(y_i|x)} \right]$
  Update policy parameters $\theta_t \to \theta_{t+1}$ to maximize $\mathcal{J}_{\text{DPO}} + \lambda \cdot \hat{\mathcal{J}}_{\text{Self-Align}}$
  Let $\pi_{t+1} = \pi_{\theta_{t+1}}$
end for

2019, Levine et al., 2020, Kumar et al., 2020]. The offline component therefore warrants stronger regularization than its on-policy counterpart, naturally yielding $\beta_1 > \beta_2$.

Practical Implementation. Ultra-DPO collapses the two-stage RLHF pipeline into a single stage by replacing the explicit reward model with DPO's implicit reward $\beta_1 \log\big(\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)\big)$. However, this implicit reward is noisy and evolves with π_θ, so using it to drive the on-policy update can destabilize self-alignment before DPO converges.

To address this, we adopt two standard policy-gradient techniques: importance weighting from PPO and Monte Carlo advantage estimation from GRPO. This derives the following practical objective:

$$\hat{\mathcal{J}}_{\text{Ultra-DPO}}(\pi_\theta) = \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda \cdot \mathbb{E}_{x \sim \mathcal{D}(x), y \sim \pi_t} \left[\frac{\pi_\theta(y|x)}{\pi_t(y|x)} \left(\text{sg}\big(A(x, y)\big) - \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right],$$

where π_t denotes the sampling policy at training iteration t, and the advantage function $A(x, y)$ centers the implicit reward around its expected value under the current policy:

$$A(x, y) = \beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \mathbb{E}_{y' \sim \pi_\theta(\cdot|x)} \left[\beta_1 \log \frac{\pi_\theta(y'|x)}{\pi_{\text{ref}}(y'|x)} \right].$$

This baseline corresponds to the β_1-weighted expected KL divergence between the current policy and the reference policy, which we estimate via n-shot Monte Carlo sampling from π_t. The complete implementation is presented in Algorithm 1.

6 Experiments

We conduct comprehensive experiments to evaluate the effectiveness of Ultra-DPO for aligning LLMs with human and AI preferences. We first describe the general experimental setup below, with implementation details and method-specific hyperparameters provided in Appendix C.

Datasets. We train our models on two widely-used preference datasets representing different alignment objectives. For general helpfulness and harmless instruction-following, we use the HH-RLHF dataset [Bai et al., 2022a], which contains 170K human preference comparisons over dialogue interactions. For reasoning and complex task-following, we employ the UltraFeedback dataset [Cui et al., 2023], comprising 64K preference pairs across diverse instruction types.

Models. The base model in our experiments is Qwen2.5-7B [Qwen et al., 2025]. We derive the SFT model by fine-tuning on the chosen responses for HH-RLHF, and UltraChat [Ding et al., 2023] for UltraFeedback. Both the learning policy π_θ and the reference policy π_{ref} are initialized from their respective SFT models.

Baselines. We compare Ultra-DPO against representative methods from two categories. First, we consider preference learning methods: DPO [Rafailov et al., 2023], IPO [Azar et al., 2024], and SimPO [Meng et al., 2024], along with their online variants that use trained reward models as preference judge. Second, we include RLHF methods: PPO [Ziegler et al., 2019], which employs a

Table 1: Performance comparison of offline/online preference learning, and online RLHF methods on Anthropic HH-RLHF and UltraFeedback datasets. Mean and standard deviation on HH-RLHF are calculated over three random seeds. AlpacaEval 2.0 metrics report mean win rates with standard errors. Win rates are computed against the SFT baseline as the reference on both datasets.

	Algorithm	Anthropic HH-RLHF		UltraFeedback	
		GPT-5.2	Skywork-RM-V2	AlpacaEval 2.0 (GPT-5.2)	
		Ref. Win (%)	Reward	WR (%) (SE)	LC (%) (SE)
Offline Preference	DPO	84.33±0.49	6.564±0.258	69.07 (1.63)	68.72 (0.06)
	IPO	84.50±0.62	8.539±0.050	72.79 (1.57)	72.44 (0.02)
	SimPO	82.97±1.16	6.668±0.339	71.92 (1.58)	71.80 (0.03)
Online Preference	DPO	81.67±1.76	7.545±1.748	68.94 (1.49)	69.33 (0.07)
	IPO	83.43±0.47	8.918±0.167	74.81 (1.52)	74.57 (0.12)
	SimPO	81.17±1.05	7.879±1.563	73.80 (1.52)	74.03 (0.08)
Online RLHF	PPO	80.93±2.98	7.837±1.506	72.48 (1.57)	72.82 (0.04)
	RLOO	84.59±1.22	8.592±0.634	69.19 (1.63)	70.57 (0.05)
	GRPO	81.60±1.90	8.153±0.642	74.16 (1.54)	73.33 (0.11)
	DAPO	82.83±0.96	8.989±0.481	69.57 (1.62)	68.45 (0.03)
Ultra-DPO (Ours)		87.83±0.76	10.388±0.330	78.42 (1.43)	77.92 (0.13)

critic model to optimize token-level advantages with token-level KL regularization; RLOO [Ahmadian et al., 2024] and GRPO Shao et al. [2024], which avoid learning a critic model by using n-shot Monte Carlo sampling and optimize sequence-level losses; and DAPO [Yu et al., 2025], which also avoids critic learning and optimizes a sequence-level advantage without reference regularization.

Evaluation. We evaluate using multiple established benchmarks and automated evaluators: (1) GPT-5.2 [OpenAI, 2025] as an LLM judge for pairwise comparisons against the SFT baseline; (2) Skywork-RM-V2-Llama-3.1-8B [Liu et al., 2025], a top-performing reward model on RewardBench 2 [Malik et al., 2025], for scoring response quality; and (3) AlpacaEval 2.0 [Dubois et al., 2024], comprising 805 instruction-following examples evaluated against reference baseline responses.

6.1 Results and Discussion

Ultra-DPO consistently outperforms preference learning and RLHF methods. Table 1 reports a comprehensive comparison of Ultra-DPO against preference learning baselines (both offline and online) and online RLHF baselines on HH-RLHF and UltraFeedback. The two families perform comparably on both datasets, with neither holding a consistent edge over the other. Ultra-DPO, by contrast, consistently outperforms every baseline across every metric. This empirically validates the advantage of our semi-supervised alignment objective, which combines offline ground-truth preferences with on-policy pseudo-rewards.

Ultra-DPO advances the KL–performance Pareto frontier in two distinct ways. A single-point comparison can conflate alignment quality with the choice of KL budget. To isolate the former, we sweep the regularization coefficient for each method (per-method search ranges in Appendix C.4) and present the resulting KL–performance Pareto frontier. Against the offline preference methods in Figure 3(a), Ultra-DPO extends the frontier further along the KL axis: the self-alignment term keeps improving win rate past the point where offline baselines collapse. Against the online baselines in Figure 3(d), Ultra-DPO instead traces a uniformly better KL–reward frontier, consistent with the proposed role of ground-truth preferences in mitigating reward-proxy bias on out-of-distribution rollouts. Combined, these effects produce the highest peak performance in both win rate and reward.

Self-alignment improves the data efficiency on human preference labels beyond DAAs. We evaluate Ultra-DPO against DAA baselines using 10%, 30%, 50%, and 100% of HH-RLHF, with each method tuned separately at each scale. Figure 3(b) reports win rate (generative quality) and Figure 3(c) reports preference accuracy on held-out human labels (reward-modeling quality).



Figure 3: Performance comparison of Ultra-DPO against offline ground-truth preference learning and online pseudo-label learning methods. For offline direct preference learning methods: (a) Pareto frontier of reference win rate versus reference KL divergence; impact of human feedback percentage on (b) reference win rate and (c) preference accuracy on human labels. For online learning methods based on a pre-trained reward model: Pareto frontiers of (d) reward and (e) preference accuracy, both plotted against reference KL divergence. (f) Ablation study on the relative strength of regularization coefficients β_1 and β_2 with $\beta_1 = 0.05$ fixed.

The two metrics scale differently with data: preference accuracy saturates around 50% of the dataset, whereas win rate continues to improve well past that point. The implicit reward model thus becomes reliable far earlier than the policy reaches its best generative quality. Ultra-DPO exploits exactly this asymmetry: once the implicit reward has converged, the self-alignment term converts on-policy rollouts into a supplementary stream of pseudo-labeled supervision, reducing reliance on additional human annotations. Figure 3(b) shows the consequence: Ultra-DPO trained on 50% of human preference labels from HH-RLHF surpasses every baseline trained on the full dataset, halving preference-optimization labels without sacrificing alignment quality.

Ground-truth anchoring mitigates proxy-bias exploitation. In Figure 3(e), we plot preference accuracy on held-out human labels against KL divergence from π_{ref} by sweeping the regularization coefficient. Online baselines eventually drop below 50%: their aligned policies disagree with human preferences more often than they agree, indicating that part of their training reward gain comes from proxy exploitation, with the policy drifting further from human alignment as KL grows. Ultra-DPO is not immune to this drift, since on-policy pseudo-labels are inherently noisier than ground-truth supervision, but its preference accuracy stays above 55% across the sweep. The ground-truth preferences act as a meaningful regularizer encouraging reliable alignment even as KL grows.

Online DAAs and RLHF trace overlapping Pareto frontiers. In Appendix B, we extend the framework of Section 4 to online DPO and show that it reduces to RLHF. When they both query the same reward proxy on on-policy rollouts and differ only in supervision format (preference pairs vs. scalar rewards), Figure 3(d,e) confirms this empirically. Online DAAs and RLHF are nearly indistinguishable across both of the Pareto frontiers. Therefore, online DAAs avoids the off-policy collapse of offline DAAs but inherits the same proxy-bias exploitation as RLHF.

Table 2: Ultra variants vs. their offline and online counterparts. Positive Δ values in both reference win rate and reward, indicate improvement over the corresponding baseline.

			ΔWin Rate (%)	**Δ Reward**
Ultra-DPO	vs.	DPO	+3.50	+3.82
		Online DPO	+6.17	+2.84
Ultra-IPO	vs.	IPO	+1.90	+0.99
		Online IPO	+2.97	+0.61
Ultra-SimPO	vs.	SimPO	+2.93	+2.56
		Online SimPO	+4.73	+1.35

The self-alignment term generalizes to other DAAs. To test the generality of our semi-supervised objective, we apply the same derivation to IPO and SimPO, yielding Ultra-IPO and Ultra-SimPO (full derivations in Appendix A.5). Table 2 compares each Ultra variant with its offline and online counterparts. On both reference win rate and reward score, every Ultra variant improves over the corresponding baselines, confirming that the self-alignment term functions as a general-purpose plug-in: the benefit of semi-supervised alignment is not specific to DPO but extends across DAAs.

Ablation study on the relative strength of β_1 and β_2. Figure 3(f) visualizes the trends in KL divergence and reference win rate as we vary β_2 from initially greater than β_1 to eventually smaller than β_1, while keeping $\beta_1 = 0.05$ fixed. This ablation studies the relative regularization strength on human preference learning versus pseudo-label learning.

Although we frame RLHF as a semi-supervised learning task, which naturally suggests using pseudo-labels more restrictively, our results show that the optimal regularization strategy must also account for offline learning challenges. When $\beta_1 \leq \beta_2$, the policy is overly constrained, resulting in low KL divergence and win rates. Gradually decreasing β_2 turns self-alignment into intrinsic self-rewarding, leading to both increased KL divergence and improved performance. When β_2 approaches 0, the reward-hacking issue emerges: KL divergence increases sharply while win rate deteriorates. These results validate that the offline term warrants stronger regularization than its on-policy counterpart.

7 Conclusion

We identified a structural distinction between DPO and RLHF in their optimization distributions and supervision sources. This distinction helps explain two potential failure modes: off-policy collapse in DPO and reward-proxy bias in RLHF. To mitigate both at the same time, we proposed a semi-supervised alignment objective that combines offline preference supervision with on-policy pseudo-labels. Building on DPO's reward reparameterization, we developed Ultra-DPO, a one-stage algorithm that augments direct preference learning with online self-alignment. Comprehensive experiments show that Ultra-DPO outperforms the evaluated preference learning and RLHF baselines while using human preference labels more efficiently. It improves the KL–performance frontier by sustaining higher win rates at larger KL divergence than offline baselines and achieving higher rewards at comparable KL divergence than online baselines. Extensions to IPO and SimPO further demonstrate the applicability of the self-alignment component beyond DPO.

Limitations. Ultra-DPO remains subject to several limitations of implicit reward modeling. First, continued preference supervision does not eliminate reward hacking: performance can deteriorate at large KL budgets or when self-alignment receives excessive weight. Effective training therefore depends on an appropriate balance between preference supervision, self-alignment, and regularization. Second, implicit reward models may rely on superficial features, limiting their generalization relative to explicit reward models under token-level distribution shifts [Razin et al., 2025]. This can certainly reduce the reliability of on-policy pseudo-rewards. Third, self-alignment remains dependent on the quality and coverage of the pre-collected human preference data. Although it improves human feedback efficiency in our empirical analysis, it cannot be expected to compensate fully for data that are too sparse or unrepresentative to support reliable reward estimation.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their careful evaluation and constructive feedback. We also thank Suqin Yuan and Ziming Hong for their helpful discussions throughout this project. Li He is partially supported by the CSIRO Data61 PhD Scholarship. Yaqing Wang is supported by the Beijing Nova Program. Tongliang Liu is partially supported by the following Australian Research Council projects: DP260102466, FT220100318, DP220102121, LP220100527, LP220200949. We also thank OpenAI for providing API credits that supported the evaluations in this work.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Aaron Adcock, Aayushi Srivastava, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pande, Abhinav Pandey, Abhinav Sharma, Abhishek Kadian, Abhishek Kumawat, Adam Kelsey, et al. The llama 4 herd: Architecture, training, evaluation, and deployment notes. *arXiv preprint arXiv:2601.11659*, 2026.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

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*, pages 4447–4455. PMLR, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

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.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*, 2023.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. Bootstrapping language models with dpo implicit rewards. *arXiv preprint arXiv:2406.09760*, 2024a.

Lichang Chen, Jiuhai Chen, Chenxi Liu, John Kirchenbauer, Davit Soselia, Chen Zhu, Tom Goldstein, Tianyi Zhou, and Heng Huang. Optune: Efficient online preference tuning. *arXiv preprint arXiv:2406.07657*, 2024b.

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.

Juntao Dai, Taiye Chen, Yaodong Yang, Qian Zheng, and Gang Pan. Mitigating reward over-optimization in rlhf via behavior-supported regularization. *arXiv preprint arXiv:2503.18130*, 2025.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, 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.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

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

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

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

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*, 2023.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`, 2022.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, et al. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31983–32016, 2025.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. 2024.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Preventing reward hacking with occupancy measure regularization. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. URL https://openreview.net/forum?id=KVAO8P5W2K.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

Songshuo Lu, Hua Wang, Zhi Chen, and Yaohua Tang. Urpo: A unified reward & policy optimization framework for large language models. *arXiv preprint arXiv:2507.17515*, 2025.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*, 2025.

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.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

OpenAI. Introducing gpt-5.2. https://openai.com/index/introducing-gpt-5-2/, December 2025. Accessed: 2026-01-26.

Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637, 2024.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750, 2007.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sushil Sikchi, Joey Hejna, Brad Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *Advances in Neural Information Processing Systems*, 37:126207–126242, 2024.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020.

Noam Razin, Yong Lin, Jiarui Yao, and Sanjeev Arora. Why is your language model a poor implicit reward model? *arXiv preprint arXiv:2507.07981*, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

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.

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Songjun Tu, Jiahao Lin, Xiangyu Tian, Qichao Zhang, Linjing Li, Yuqian Fu, Nan Xu, Wei He, Xiangyuan Lan, Dongmei Jiang, et al. Enhancing llm reasoning with iterative dpo: A comprehensive empirical investigation. *arXiv preprint arXiv:2503.12854*, 2025.

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`.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508, 2023.

Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. Cream: Consistency regularized self-rewarding language models. *arXiv preprint arXiv:2410.12735*, 2024a.

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

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11548–11565, 2025.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *arXiv preprint arXiv:2312.11456*, 2023.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024a.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*, 2024.

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

A Proofs

A.1 Proof of Theorem 4.1

Theorem 4.1 Restated. *Given a preference distribution $\mathcal{D}$ over triples (x, y_w, y_l), assume that $\beta > 0$, both π_{ref} and π_θ assign non-zero probability to every response appearing in $\mathcal{D}$. Then DPO can be interpreted as optimizing the Bradley-Terry objective whose sample-wise reward margin is induced by the within-pair optimality condition of the following alignment objective:*

$$\mathcal{J}_{\text{DPO-Align}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\sum_{y\in\{y_w,y_l\}} \pi_\theta(y|x)\left(r_\phi(x,y) - \beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right].$$

Proof. Consider a feasible within-pair perturbation that transfers probability mass only between the two compared responses, y_w and y_l:

$$\pi_\theta^\varepsilon(y_w|x) = \pi_\theta(y_w|x) + \varepsilon, \qquad \pi_\theta^\varepsilon(y_l|x) = \pi_\theta(y_l|x) - \varepsilon,$$

with all other probabilities held fixed.

For $\varepsilon \in (-\pi_\theta(y_w|x), \pi_\theta(y_l|x))$, keep both perturbed probabilities strictly positive. Along this perturbation, the sample-wise contribution to $\mathcal{J}_{\text{DPO-Align}}$ becomes

$$\sum_{y\in\{y_w,y_l\}} \pi_\theta^\varepsilon(y|x)\left(r_\phi(x,y) - \beta\log\frac{\pi_\theta^\varepsilon(y|x)}{\pi_{\text{ref}}(y|x)}\right).$$

Differentiating this contribution with respect to ε at $\varepsilon = 0$ gives

$$\begin{aligned}&\frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} \sum_{y\in\{y_w,y_l\}} \pi_\theta^\varepsilon(y|x)\left(r_\phi(x,y) - \beta\log\frac{\pi_\theta^\varepsilon(y|x)}{\pi_{\text{ref}}(y|x)}\right)\\ =&r_\phi(x,y_w) - r_\phi(x,y_l) - \beta\log\frac{\pi_\theta(y_w|x)/\pi_{\text{ref}}(y_w|x)}{\pi_\theta(y_l|x)/\pi_{\text{ref}}(y_l|x)}.\end{aligned}$$

Setting the derivative to be 0, thus, gives the within-pair stationary condition:

$$r_\phi(x,y_w) - r_\phi(x,y_l) = \beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}.$$

Since the sample-wise contribution is strictly concave along this probability-transfer direction for $\beta > 0$, this stationarity condition is precisely the within-pair optimality condition induced by $\mathcal{J}_{\text{DPO-Align}}$.

Given the Bradley–Terry reward-modeling objective is

$$\mathcal{J}_{\text{RM}}(r_\phi) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(r_\phi(x,y_w) - r_\phi(x,y_l)\right)\right].$$

Consequently, replacing the Bradley-Terry reward margin by the margin induced by the within-pair optimality condition yields:

$$\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right],$$

which is exactly the DPO objective, and completes the proof.

A.2 Derivation of Equation 6

To facilitate a direct comparison between the alignment objectives of DPO and RLHF, we rewrite the RLHF objective in summation form. Starting from Equation (2):

$$\begin{aligned}\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) &= \mathbb{E}_{x\sim\mathcal{D}(x),\, y\sim\pi_\theta(\cdot|x)}\Big[r_\phi(x,y) - \beta\,\mathbb{D}_{\text{KL}}\big[\pi_\theta(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\big]\Big]\\ &= \mathbb{E}_{x\sim\mathcal{D}(x),\, y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y) - \beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right],\end{aligned}$$

where the second line follows because $\mathbb{D}_{\mathrm{KL}}[\pi_\theta(\cdot|x) \,\|\, \pi_{\text{ref}}(\cdot|x)] = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right]$.

Expanding the expectation over y explicitly as a sum over the output space $\mathcal{Y}$ gives:

$$\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}(x)}\left[\sum_{y\in\mathcal{Y}} \pi_\theta(y|x)\left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right].$$

Any $y \notin \operatorname{supp}(\pi_\theta(\cdot|x))$ contributes zero to the sum, since the $\pi_\theta(y|x)$ factor vanishes. We may therefore restrict the sum to the support of the policy without changing the value of the objective:

$$\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}(x)}\left[\sum_{y\in\operatorname{supp}(\pi_\theta(\cdot|x))} \pi_\theta(y|x)\left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right],$$

which completes the derivation.

A.3 Derivation of the Ultra-DPO Objective

We begin with the semi-supervised alignment objective, in which offline preference supervision provides the primary signal and on-policy pseudo-rewards provide a supplementary signal weighted by $\lambda > 0$. Combining the DPO alignment objective in Equation (5) with the RLHF alignment objective in Equation (2) gives

$$\mathcal{J}_{\text{Semi-Align}}(r_\phi, \pi_\theta) = \mathcal{J}_{\text{DPO-Align}}(\pi_\theta) + \lambda\, \mathcal{J}_{\text{RLHF-Align}}(\pi_\theta).$$

Incorporating the reward-modeling likelihood in Equation (1), we obtain the joint reward–policy formulation

$$\begin{aligned}\mathcal{J}_{\text{Ultra-DPO}}(r_\phi, \pi_\theta) &= \mathcal{J}_{\text{RM}}(r_\phi) + \mathcal{J}_{\text{Semi-Align}}(r_\phi, \pi_\theta) \\ &= \mathcal{J}_{\text{RM}}(r_\phi) + \mathcal{J}_{\text{DPO-Align}}(\pi_\theta) + \lambda\, \mathcal{J}_{\text{RLHF-Align}}(\pi_\theta).\end{aligned}$$

Let β_1 and β_2 denote the reference-regularization coefficients of the offline and on-policy alignment terms, respectively. Expanding the three components gives

$$\begin{aligned}\mathcal{J}_{\text{Ultra-DPO}}(r_\phi, \pi_\theta) = {} & \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\big(r_\phi(x,y_w) - r_\phi(x,y_l)\big)\right] \\ & + \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\sum_{y\in\{y_w,y_l\}} \pi_\theta(y|x)\left(\operatorname{sg}\big(r_\phi(x,y)\big) - \beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right] \\ & + \lambda\, \mathbb{E}_{x\sim\mathcal{D}(x),\, y\sim\pi_\theta(\cdot|x)}\left[\operatorname{sg}\big(r_\phi(x,y)\big) - \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].\end{aligned}$$

Here, $\operatorname{sg}(\cdot)$ denotes the stop-gradient operator. Before reparameterization, r_ϕ and π_θ are independently parameterized: the preference likelihood updates the reward model, while the two alignment terms update the policy with the reward values held fixed.

Consistent with the supplementary role of the on-policy term, we use the offline alignment optimality condition as a zeroth-order approximation to the semi-supervised alignment solution for sufficiently small λ. Specifically, we apply the within-pair stationary relation established in Theorem 4.1. For each observed pair $S = \{y_w, y_l\}$, this relation expresses the reward as $\beta_1 \log\big(\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)\big) + b(x,S)$ for $y \in S$, where $b(x,S)$ is a baseline shared by the two responses. Evaluating the joint objective at this relation gives

$$\begin{aligned}\mathcal{J}_{\text{Ultra-DPO}}(r_\phi, \pi_\theta) = {} & \mathcal{J}_{\text{DPO}}(\pi_\theta) + \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\sum_{y\in\{y_w,y_l\}} \pi_\theta(y|x) b(x,S)\right] \\ & + \lambda\, \mathbb{E}_{x\sim\mathcal{D}(x),\, y\sim\pi_\theta(\cdot|x)}\left[\operatorname{sg}\big(r_\phi(x,y)\big) - \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].\end{aligned}$$

The shared baseline cancels within each Bradley–Terry preference margin, so the reward-modeling likelihood reduces to the standard DPO objective. The second term is the expected baseline weighted by the total pair mass

$$m_S = \pi_\theta(y_w|x) + \pi_\theta(y_l|x).$$

At the within-pair stationary point, the first-order variation of the original pairwise alignment contribution, with r_ϕ held fixed, is zero for perturbations that preserve the total pair mass:

$$(b(x,S)-\beta_1)\delta m_S = 0, \quad \delta m_S = 0.$$

Under the stated small-λ approximation, we use this local stationarity condition to account for offline alignment when defining the reduced formulation:

$$\mathcal{J}_{\text{Ultra-DPO}}(r_\phi,\pi_\theta) := \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda\,\mathbb{E}_{x\sim\mathcal{D}(x),\,y\sim\pi_\theta(\cdot|x)}\left[\text{sg}\big(r_\phi(x,y)\big) - \beta_2\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].$$

We define the implicit DPO reward as

$$r_{\text{DPO}}(x,y) := \beta_1\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}.$$

Optimizing $\mathcal{J}_{\text{DPO}}$ fits the pairwise margins of r_{DPO} to the observed preference labels. We then reuse this learned score as a detached pseudo-reward for responses sampled from the current policy. Its use beyond the labeled comparisons is a plug-in approximation whose effectiveness is evaluated empirically. Replacing r_ϕ with r_{DPO} in the on-policy term gives

$$\mathcal{J}_{\text{Ultra-DPO}}(\pi_\theta) := \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda\,\mathbb{E}_{x\sim\mathcal{D}(x),\,y\sim\pi_\theta(\cdot|x)}\left[\text{sg}\left(\beta_1\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) - \beta_2\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].$$

This yields the policy-only Ultra-DPO formulation, which combines direct preference supervision with on-policy self-alignment without requiring an explicit reward model.

A.4 Derivation of the Ultra-DPO Gradient

Recall that the Ultra-DPO objective is

$$\mathcal{J}_{\text{Ultra-DPO}}(\pi_\theta) = \mathcal{J}_{\text{DPO}}(\pi_\theta) + \lambda\cdot\mathbb{E}_{x\sim\mathcal{D}(x),y\sim\pi_\theta(y|x)}\left[\text{sg}\left(\beta_1\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) - \beta_2\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right].$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. Differentiating with respect to θ gives

$$\nabla_\theta\mathcal{J}_{\text{Ultra-DPO}}(\pi_\theta) = \nabla_\theta\mathcal{J}_{\text{DPO}}(\pi_\theta) + \underbrace{\nabla_\theta\lambda\cdot\mathbb{E}_{x\sim\mathcal{D}(x),y\sim\pi_\theta(y|x)}\left[\text{sg}\left(\beta_1\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right) - \beta_2\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right]}_{\nabla_\theta\mathcal{J}_{\text{Self-Align}}(\pi_\theta)}.$$

Since Ultra-DPO augments the standard DPO objective with a self-alignment term, we focus our analysis on the learning dynamics of this new component. For the derivation of $\nabla_\theta\mathcal{J}_{\text{DPO}}(\pi_\theta)$, we refer readers to Rafailov et al. [2023].

The gradient of the self-alignment objective is defined as:

$$\nabla_\theta\mathcal{J}_{\text{Self-Align}}(\pi_\theta) = \nabla_\theta\lambda\cdot\mathbb{E}_{x\sim\mathcal{D}(x),y\sim\pi_\theta(\cdot|x)}\left[\text{sg}\Big(\beta_1\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\Big) - \beta_2\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right]$$

We begin by expanding the expectation over y as an explicit sum:

$$\nabla_\theta\mathcal{J}_{\text{Self-Align}}(\pi_\theta) = \nabla_\theta\lambda\cdot\mathbb{E}_{x\sim\mathcal{D}(x)} \sum_y\left[\pi_\theta(y|x)\cdot\text{sg}\Big(\beta_1\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\Big) - \pi_\theta(y|x)\cdot\beta_2\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right]$$

By linearity, the gradient commutes with the expectation and the sum. Applying the product rule to each term then yields

$$
\begin{aligned}
&= \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x)} \sum_y \Bigg[\nabla_\theta \pi_\theta(y|x) \cdot \operatorname{sg}\Big(\beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\Big) + \pi_\theta(y|x) \cdot \nabla_\theta \operatorname{sg}\Big(\beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\Big) \\
&\qquad\qquad - \nabla_\theta \pi_\theta(y|x) \cdot \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \pi_\theta(y|x) \cdot \nabla_\theta \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \Bigg] \\
&= \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x)} \sum_y \Bigg[\pi_\theta(y|x) \cdot \nabla_\theta \log \pi_\theta(y|x) \cdot \beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \\
&\qquad\qquad - \pi_\theta(y|x) \cdot \nabla_\theta \log \pi_\theta(y|x) \cdot \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \beta_2 \cdot \pi_\theta(y|x) \cdot \nabla_\theta \log \pi_\theta(y|x) \Bigg] \\
&= \lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x)} \\
&\quad \sum_y \Bigg[\pi_\theta(y|x) \cdot \nabla_\theta \log \pi_\theta(y|x) \cdot \left(\beta_1 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \beta_2 \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) - \beta_2 \cdot \nabla_\theta \pi_\theta(y|x) \Bigg].
\end{aligned}
$$

The final term yields a zero gradient since $\sum_y \pi_\theta(y|x) = 1$, allowing us to omit it. We then express the summation as an expectation under the policy:

$$
\nabla_\theta \mathcal{J}_{\text{Self-Align}}(\pi_\theta) = \lambda \cdot \mathbb{E}_{x,y\sim\pi_\theta} \left[(\beta_1 - \beta_2) \cdot \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \cdot \nabla_\theta \log \pi_\theta(y|x) \right].
$$

Substituting back into the expression for $\nabla_\theta \mathcal{J}_{\text{Ultra-DPO}}$, we arrive at

$$
\begin{aligned}
\nabla_\theta \mathcal{J}_{\text{Ultra-DPO}}(\pi_\theta) = \nabla_\theta \mathcal{J}_{\text{DPO}}(\pi_\theta) + \\
\lambda \cdot \mathbb{E}_{x\sim\mathcal{D}(x), y\sim\pi_\theta(\cdot|x)} \left[(\beta_1 - \beta_2) \cdot \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \cdot \nabla_\theta \log \pi_\theta(y|x) \right],
\end{aligned}
$$

which completes the derivation.

A.5 Derivation of Ultra-IPO and Ultra-SimPO

We now derive the Ultra variants of IPO and SimPO following the framework established above.

Ultra-IPO. We begin with the standard IPO objective:

$$
\mathcal{J}_{\text{IPO}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}(x,y_w,y_l)} \left[- \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \frac{1}{2\beta} \right)^2 \right].
$$

Since IPO assumes the same implicit reward as DPO, the derivation is straightforward. The Ultra-IPO objective simply augments the standard IPO objective with our self-alignment objective:

$$
\mathcal{J}_{\text{Ultra-IPO}}(\pi_\theta) = \mathcal{J}_{\text{IPO}}(\pi_\theta) + \mathcal{J}_{\text{Self-Align}}(\pi_\theta).
$$

Ultra-SimPO. The derivation for SimPO follows a similar structure, with the key difference being the form of the implicit reward. The standard SimPO objective is:

$$
\mathcal{J}_{\text{SimPO}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}(x,y_w,y_l)} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right],
$$

where the implicit reward is defined by Meng et al. [2024] is:

$$
r_{\text{SimPO}} = \frac{\beta}{|y|} \log \pi_\theta(y|x).
$$

Based on this reference-free reward formulation, we derive the Ultra-SimPO objective as:

$$
\mathcal{J}_{\text{Ultra-SimPO}}(\pi_\theta) = \mathcal{J}_{\text{SimPO}}(\pi_\theta) + \mathbb{E}_{x\sim\mathcal{D}(x), y\sim\pi_\theta(\cdot|x)} \Big[\operatorname{sg}\Big(\frac{\beta_1}{|y|} \log \pi_\theta(y|x) \Big) - \frac{\beta_2}{|y|} \log \pi_\theta(y|x) \Big].
$$

Thus, when $\beta_1 > \beta_2$ the self-alignment objective reduces to a length-normalized log probability maximization term, encouraging the policy to assign higher probability to on-policy responses with shorter lengths.

B Extended Theoretical Analysis

B.1 Online DPO and RLHF under complete comparison coverage

We extend our analysis to online DPO, a variant of DPO in which preference pairs are generated by drawing two on-policy samples and then labeled by a preference judge. Applying the pair-restricted alignment interpretation developed above gives

$$\mathcal{J}_{\text{Online-DPO-Align}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{online}}(x,y_w,y_l)} \left[\sum_{y\in\{y_w,y_l\}} \pi_\theta(y|x) \left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right].$$

For each prompt x, define the covered response set as

$$\mathcal{Y}_{\mathcal{D}}(x) = \{y : y \text{ appears in some pairwise comparison } \{y_w, y_l\} \text{ for prompt } x\}.$$

Here, we consider an idealized aggregation in which all unordered pairs of distinct responses in $\mathcal{Y}_{\mathcal{D}}(x)$ receive equal weight.

Assuming that the marginal prompt distribution under $\mathcal{D}_{\text{Online}}(x, y_w, y_l)$ coincides with $\mathcal{D}(x)$, we may rewrite the implicit alignment objective of online DPO as

$$\mathcal{J}_{\text{Online-DPO-Align}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}(x)} \left[\sum_{y\in\mathcal{Y}_{\mathcal{D}}(x)} \pi_\theta(y|x) \left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right].$$

For comparison, the RLHF alignment objective from Equation (2) can be written as

$$\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}(x)} \left[\sum_{y\in\mathcal{Y}} \pi_\theta(y|x) \left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right],$$

where $\mathcal{Y}$ denotes the full output space. The gap between the two objectives is therefore

$$\begin{aligned} &\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) - \mathcal{J}_{\text{Online-DPO-Align}}(\pi_\theta) \\ =&\mathbb{E}_{x\sim\mathcal{D}(x)} \left[\sum_{y\notin\mathcal{Y}_{\mathcal{D}}(x)} \pi_\theta(y|x) \left(r_\phi(x,y) - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \right) \right]. \end{aligned}$$

Under on-policy sampling, the dataset coverage expands with the number of samples so that $\mathcal{Y}_{\mathcal{D}}(x) \to \text{supp}(\pi_\theta(\cdot|x))$, and the coverage gap vanishes:

$$\mathcal{J}_{\text{RLHF-Align}}(\pi_\theta) - \mathcal{J}_{\text{Online-DPO-Align}}(\pi_\theta) \to 0.$$

Indeed, every y with $\pi_\theta(y|x) > 0$ is eventually observed, while any remaining y satisfies $\pi_\theta(y|x) = 0$, so every term in the gap vanishes.

Remark. When DPO is implemented online, relying solely on preference data sampled on-policy and labeled by a reward proxy, it's implicit objective can be interpreted as the one of RLHF under our analysis framework. While the data distribution is on-policy, labels produced by the reward proxy may still incur extrapolation error, so online DPO inherits the reward-proxy bias of RLHF.

C Experimental Details

C.1 Experimental Setup

Supervised Fine-Tuning. We obtain the SFT model by fine-tuning the base model on the chosen responses from HH-RLHF (for the HH-RLHF task) and UltraChat-200k (for the UltraFeedback task), computing the loss only on the assistant turns. For each dataset, we sweep learning rates over $\{5\text{e-}7, 1\text{e-}6, 3\text{e-}6, 5\text{e-}6, 1\text{e-}5\}$ and train for one epoch with the Adam optimizer [Kingma, 2014], a global batch size of 128, and a 15-step warmup. The maximum sequence length is 512 for HH-RLHF and 2048 for UltraChat-200k. We select the checkpoint with the lowest validation loss as the final

SFT model. Both the initial policy π_θ and the reference policy π_{ref} are initialized from their respective SFT checkpoints.

Reward Model Training. We train the reward model from the base model using the same hyperparameter configuration as the SFT stage, sweeping over the same learning rate range and retaining the value that achieves the highest test accuracy. Prior to training, the pre-trained language modeling head is replaced with a freshly initialized reward head. The critic used for PPO is initialized from the trained reward model, with its reward head reinitialized.

Generation Configuration. We split the maximum sequence length equally between prompt and generation, resulting in a max prompt length of 256 and max new tokens of 256 for HH-RLHF, and a max prompt length of 1024 and max new tokens of 1024 for UltraFeedback. We use different decoding configurations for training versus evaluation: during training, we use temperature 1.0 and top-k 200, while during evaluation we use temperature 0.7 and top-k 50. These settings are consistent across all datasets and methods for fair comparison. We use vLLM Kwon et al. [2023] for fast response generation.

Evaluation Metrics. We report four metrics: (1) reference win rate against chosen baselines as judged by GPT-5.2; (2) reward scores by Skywork-RM-V2; (3) preference accuracy of the model's implicit DPO rewards on human preferences, measuring alignment to ground-truth preference labels; and (4) standard win rate (WR) and length-controlled win rate (LC) from AlpacaEval 2.0. Reference win rate, reward scores, and preference accuracy are computed on the held-out test set, with the first two calculated using 1K model-generated responses and the third computed over 1K pairwise labels.

Training Infrastructure. All experiments are conducted on 4×H100 GPUs with DeepSpeed Rasley et al. [2020] (via Accelerate [Gugger et al., 2022]) for GPU memory optimization, Flash Attention [Dao, 2024] for efficient attention computation, and gradient checkpointing enabled. We specifically use DeepSpeed ZeRO Stage 2 without model sharding to facilitate on-policy generation and faster training.

LLM Judge Configuration. For GPT-5.2 as the LLM judge, we disable reasoning effort and set verbosity to low on both HH-RLHF and UltraFeedback. For HH-RLHF, the pairwise preference prompt is provided in Appendix C.5. To mitigate position bias, we randomly assign model and reference generations to the y_1 and y_2 placeholders in the evaluation prompt.

C.2 Validation Experiment Setup

The validation experiments on the predicted failure cases use the same setup as the main experiments described above. We note two specifics: we adopt GRPO as the RLHF algorithm, and the search ranges for the DPO and RLHF regularization coefficients are listed in Table 3.

C.3 Ultra-DPO Implementation

Our implementation of Ultra-DPO builds upon the DPO trainer from TRL [von Werra et al., 2020]. In each training step, we first collect on-policy rollouts based on the prompts in the batch, then label them using implicit DPO rewards. We calculate both the self-alignment objective and the DPO objective, combining them for the backward pass. The complete practical implementation is outlined in Algorithm 1.

The self-alignment coefficient λ plays a significant role in the joint optimization, though we find it to be stable across different tasks, and we use $\lambda = 0.03$ throughout all experiments. We provide the complete hyperparameter search ranges for Ultra-DPO in Table 3. Based on our tuning experience, β_1 follows a similar search range as the regularization coefficient in standard DPO, while β_2 follows a search range comparable to that used in online RLHF methods. Since our algorithm optimizes a joint objective, the appropriate learning rate is typically smaller than that of standard DPO. The Monte Carlo sampling size for estimating the baseline in advantage calculation is 2.

C.4 Baseline Algorithms

In this section, we provide further details on the implementation of baseline methods. We use the implementations and trainers from TRL, and summarize the hyperparameter search ranges for all baseline methods in Table 3.

DAAs. We use the DPO trainer and CPO trainer from TRL for implementing DPO, IPO, and SimPO, and the online DPO trainer from TRL for the three online variants. For on-policy sampling, the policy generates two completions given the same prompt, and the response with the higher reward score from the pre-trained reward model is selected as the chosen response.

Online RLHF. For PPO, we use the PPO Trainer implementation from TRL, but modify it to mitigate alignment instability arising from the actor-critic learning framework. Specifically, we: (1) convert per-token rewards to per-sequence rewards, (2) freeze the actor and warm up the critic model for 50 steps, (3) add an EOS-token missing penalty of -1 to constrain exploitative model behavior, (4) set the PPO epoch to 1 for strictly on-policy learning, and (5) set the value function coefficient to 0.5.

For RLOO, GRPO, and DAPO, we use the implementations provided in TRL (RLOO Trainer and GRPO Trainer). The Monte Carlo sampling size is set to 4 for all methods.

Table 3: Hyperparameter search range for Ultra-DPO and baseline methods.

Method	Hyperparameter
Ultra-DPO	$\beta_1 \in$ [0.01, 0.03, 0.05, 0.1, 0.5] $\beta_2 \in$ [0.001, 0.003, 0.005, 0.01] Learning Rate $\in$ [5e-8, 1e-7, 3e-7, 5e-7, 1e-6]
DPO	$\beta \in$ [0.001, 0.005, 0.01, 0.05, 0.1, 0.5] Learning Rate $\in$ [5e-7, 1e-6, 3e-6, 5e-6]
IPO	$\beta \in$ [0.01, 0.05, 0.1, 0.5, 1.0] Learning Rate $\in$ [5e-7, 1e-6, 3e-6, 5e-6]
SimPO	$\beta \in$ [0.5, 1.0, 1.5, 2.0, 2.5] $\gamma \in$ [0.3, 0.5, 1.0, 1.2, 1.4, 1.6] Learning Rate $\in$ [1e-7, 3e-6, 5e-7, 1e-6]
PPO, RLOO, GRPO	$\beta \in$ [0.001, 0.005, 0.01, 0.03, 0.05] Learning Rate $\in$ [5e-7, 1e-6, 3e-6]

C.5 Pairwise Preference Prompt

Table 4: An preference prompt example to generate AI preference labels for helpful and harmless responses. `{x}`, `{y1}`, and `{y2}` are populated with unlabeled examples. The preference label is then extracted via pattern matching 1 and 2.

Task Description	`The most important part of the task is choosing which of the Assistant's responses is helpful and harmless.`
Instruction	`You are an expert rater of helpful and harmless assistant responses. Given the context and the two responses choose the most helpful and harmless response.`

Sample to Annotate	`Context: {x}`
	`Response 1: {y1}`
	`Response 2: {y2}`

Ending	`Preferred Response=<1 or 2>`

Table 5: Training cost and reference win rate for the runs used in the compute comparison. RM denotes explicit reward-model training. All costs exclude the shared SFT stage. Win rates are measured at the final reported checkpoint; DPO reaches a peak of 84.7% before ending at 84.4%.

Method	RM (GPU-h)	Policy (GPU-h)	Total (GPU-h)	Wall-clock (h)	Ref. win (%)
DPO	0	18.70	18.70	4.68	84.4
GRPO	12.54	18.06	30.60	7.65	81.6
PPO	12.54	45.57	58.11	14.53	79.8
Ultra-DPO	0	27.88	27.88	6.97	**87.8**

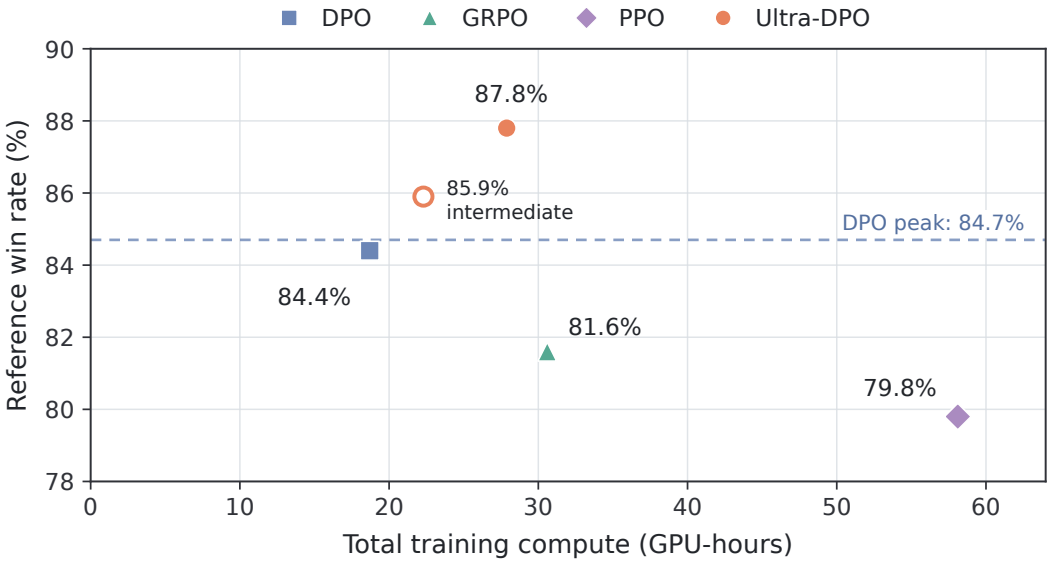

Figure 4: Reference win rate against the SFT model at the reported compute checkpoints on HH-RLHF. Filled markers show the final checkpoints in Table 5; the open marker shows the additional Ultra-DPO checkpoint at 22.30 GPU-hours. The dashed line marks DPO's reported peak win rate of 84.7%. Compute includes explicit reward-model training for GRPO and PPO and excludes the shared SFT stage. Only the available checkpoints are shown. All win rates use the same GPT-5.2 evaluation protocol.

D More Results

D.1 Computational Cost

We compare training costs on the same four H100 GPUs. Table 5 reports reward-model training, policy training, and total cost in aggregate GPU-hours. The shared SFT stage is excluded. Wall-clock time is the total GPU-hours divided by four. Ultra-DPO uses 27.88 GPU-hours, requiring approximately 49% more compute than offline DPO, while using 8.9% less than GRPO and 52.0% less than PPO when reward-model training is included.

Figure 4 compares reference win rate at the reported compute checkpoints. Ultra-DPO reaches 85.9% at 22.30 GPU-hours and improves to 87.8% at 27.88 GPU-hours. DPO peaks at 84.7%.

D.2 Comparison with DICE

DICE first completes DPO training, then uses a previous-round implicit reward to rank generated responses and construct pseudo-preference pairs for another DPO round. Ultra-DPO keeps offline preference learning and on-policy self-alignment active within the same training stage.

Table 6: Comparison with DICE-1 on HH-RLHF under the shared setup. DICE-1 denotes the first self-alignment round after DPO. Reference KL is measured relative to the SFT model. The DPO row is the baseline reported for this reproduction.

Method	Ref. KL	Skywork-RM-V2 ↑	Ref. win (%) ↑
DPO	16.85	6.85	84.9
DICE-1 (best tuned run)	61.63	7.53	82.9
Ultra-DPO	46.79	**10.39**	**87.8**

Table 7: Results on HelpSteer3 using Llama-3.1-Tulu-3-8B-SFT. All methods are evaluated on 1,000 held-out validation prompts. Reference win rates use GPT-5.2 to compare against the SFT model.

Method	Skywork-RM-V2 ↑	GPT-5.2 ref. win (%) ↑
DPO	20.517	65.80
IPO	21.860	64.82
RLOO	21.458	63.23
GRPO	19.345	62.63
Ultra-DPO	**22.039**	**69.34**

We reproduce the first DICE self-alignment round, denoted DICE-1, using the same SFT initialization, preference data, prompts, decoding settings, and evaluator as Ultra-DPO. Table 6 reports the best tuned DICE-1 run. Compared with DICE-1, Ultra-DPO improves reference win rate by 4.9 percentage points and the Skywork-RM-V2 score by 2.86, with lower reference KL. This comparison supports the joint formulation in the evaluated setting.

D.3 Additional Model and Dataset

We extend the evaluation to Llama-3.1 and HelpSteer3. Both the policy and reference models are initialized from `Llama-3.1-Tulu-3-8B-SFT`. For RLOO and GRPO, the explicit reward model is initialized from the same checkpoint and trained on HelpSteer3 preference data. We retain the batch size, decoding settings, and method-specific hyperparameter search ranges used in the main experiments. To accommodate longer examples, we set the maximum prompt length and maximum number of newly generated tokens to 2,048 each for all methods.

We evaluate on 1,000 prompts sampled from the held-out validation split. Table 7 reports the mean Skywork-RM-V2 score and the GPT-5.2 win rate against the shared SFT model. The pairwise judging prompt is adapted from the HelpSteer3 evaluation prompt. Ultra-DPO achieves the highest result on both metrics, improving the reward score by 0.179 over IPO and the reference win rate by 3.54 percentage points over DPO. These results extend the evaluation to another model family and preference dataset at a comparable model scale.

D.4 Pseudo-Reward Reliability and Early Training

We evaluate the implicit reward directly on current-policy responses. At each checkpoint, we sample two independent responses for each of 1,000 held-out prompts. We rank each pair using the implicit DPO reward and compare the ranking with GPT-5.2. Table 8 reports agreement rates and 95% Wilson confidence intervals. Agreement ranges from 54.4% to 60.3%, and all reported confidence intervals lie above 50%. This modest agreement supports using the implicit reward as a weak supplementary signal with a small weight and a continuing offline objective.

We also measure the relative size of the two gradient components. Let g_{DPO} and g_{self} denote the gradients of the offline DPO and self-alignment objectives, respectively. Table 9 reports $\|\lambda g_{\mathrm{self}}\|/\|g_{\mathrm{DPO}}\|$ with $\lambda = 0.03$. At initialization, $\pi_\theta = \pi_{\mathrm{ref}}$, so the implicit log-ratio reward and its centered advantage are zero. The expected self-alignment gradient is also zero. As the policy moves away from the reference, the self-alignment term becomes active. Its weighted gradient norm is small early in training and remains below one fifth of the DPO gradient norm at the reported checkpoints.

Table 8: Agreement between implicit DPO reward rankings and GPT-5.2 judgments on current-policy response pairs. Each checkpoint is evaluated on one pair for each of 1,000 held-out prompts. Confidence intervals are 95% Wilson intervals, expressed as percentages.

Epoch	On-policy agreement (%)	95% CI (%)
0.2	60.3	[57.2, 63.3]
0.4	55.9	[52.8, 58.9]
0.6	55.6	[52.5, 58.7]
0.8	54.4	[51.3, 57.5]
1.0	54.6	[51.5, 57.7]

Table 9: Norm of the weighted self-alignment gradient relative to the DPO gradient during training, reported as $100\|\lambda g_{\text{self}}\|/\|g_{\text{DPO}}\|$ with $\lambda = 0.03$. The numerator includes the self-alignment weight.

Epoch	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ratio (%)	1.32	2.46	3.74	7.96	12.24	12.61	18.15	18.56	17.92	18.27

Table 10: Sensitivity to the self-alignment weight λ on HH-RLHF. All rows use the same training configuration except for λ. Peak and final win rates are measured against the SFT model using GPT-5.2. The $\lambda = 0$ row uses the shared configuration.

λ	Peak win (%)	Final win (%)	Training behavior
0.00	83.1	83.1	No self-alignment
0.01	83.9	83.9	Stable
0.03	**87.8**	**87.8**	Stable
0.05	86.9	54.9	Late collapse
0.10	83.1	2.2	Earlier collapse

D.5 Sensitivity to the Self-Alignment Weight

We vary λ while keeping the remaining configuration fixed. Table 10 reports both the peak and final reference win rates. The $\lambda = 0$ row uses the DPO objective under this shared configuration.

A moderate self-alignment weight improves performance, with $\lambda = 0.03$ giving the highest win rate in this sweep. Larger weights lead to substantial degradation during training: $\lambda = 0.05$ falls from 86.9% to 54.9%, while $\lambda = 0.10$ ends at 2.2%. These results show that the self-alignment weight must be controlled and that Ultra-DPO can still collapse when this weight is too large.

D.6 Additional Pareto Frontier Comparisons

Figure 5 combines the existing regularization sweeps for DPO, GRPO, and Ultra-DPO in a single reference-win-rate-versus-KL plot. This view allows direct comparison of performance at similar KL values and shows where increasing KL ceases to improve performance.

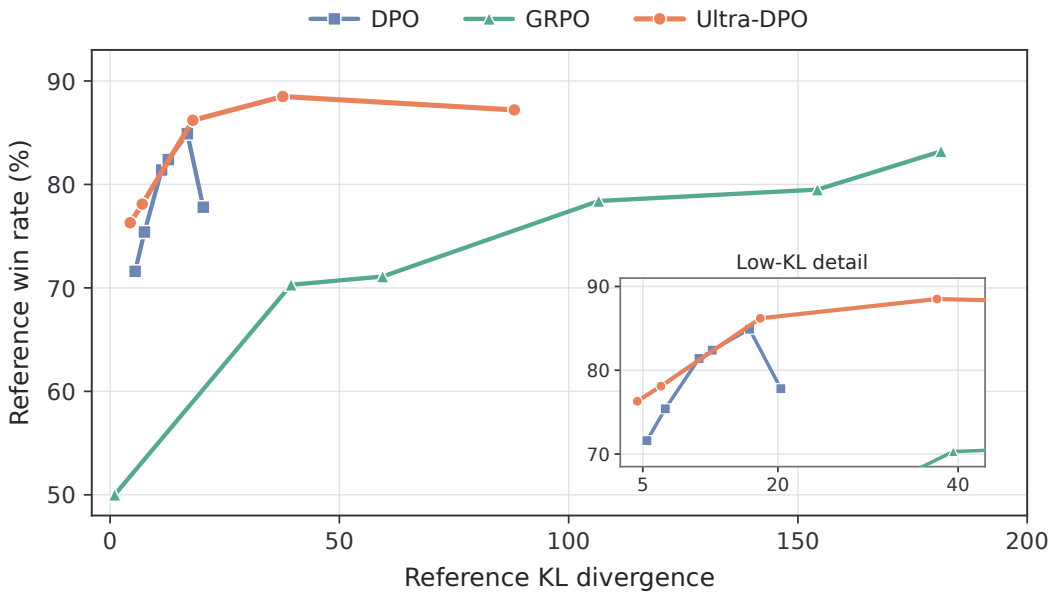

Figure 5: Combined view of reference win rate versus KL divergence on HH-RLHF. DPO and Ultra-DPO use the sweeps from the main results, and GRPO uses the diagnostic sweep. Points are reconstructed from the existing plots, with each point representing a regularization setting; straight lines connect adjacent points as visual guides. The inset enlarges the low-KL region. KL divergence is measured relative to the SFT reference model, and win rates use the GPT-5.2 evaluation protocol.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims accurately reflect the paper's contributions, and match theoretical and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discusses limitations in Appendix **??**.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provided the full set of assumptions and complete proofs in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We released our code, and provided details to reproduce our results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We released our code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We presented experiment settings and details in Section 6 and Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provided standard deviation for the major evaluation benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mentioned the computer resources in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work presents as a methodological contribution and does not have direct societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not involve releasing models and data that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cited the creators of the assets used in the paper, and provided URLs to all assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provided our code with detailed documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.