INSPO: Unlocking Intrinsic Self-Reflection for LLM Preference Optimization

Yu Li[1] **Tian Lan**[1*] **Zhengling Qi**[2*]
[1]Department of Electrical and Computer Engineering [2]School of Business
George Washington University

Abstract

Direct Preference Optimization (DPO) and its variants have become the standard for aligning Large Language Models (LLMs). However, the policies these methods optimize have two limitations. First, the optimized policy lacks invariance since it varies with modeling choices such as scalarization function or reference policy, whereas an optimal policy should remain invariant. Second, most existing methods yield theoretically suboptimal policies by not fully exploiting the comparative information in pairwise preference data, thus missing an opportunity for self-reflection through comparing and contrasting responses. To address both limitations, we propose Intrinsic Self-reflective Preference Optimization (InSPO), which derives a population-level optimal target over policies conditioned on both context and alternative response, explicitly formalizing self-reflection. We prove this formulation surpasses standard DPO and RLHF targets while ensuring invariance. InSPO serves as a plug-and-play enhancement for DPO-family algorithms, decoupling alignment from modeling constraints and keeping the architecture unchanged. Using privileged information learning, InSPO generates from the prompt alone at inference since the self-reflective mechanism is distilled during training, incurring zero overhead. Experiments show that InSPO improves win rates and length-controlled metrics across DPO variants, yielding more robust and human-aligned LLMs. The implementation is available at `https://github.com/Skylanding/InsPO`.

1 Introduction

Large language models (LLMs) are fine-tuned after pretraining through post-training alignment, combining supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) [17]. SFT fine-tunes the model to generate the correct response given an instruction by minimizing the negative log-likelihood of target tokens, imparting formatting conventions, stylistic preferences, and basic task-following behaviors [10]. RLHF aligns models using pairwise preference data through a two-stage pipeline [22]: first fitting a reward model under the Bradley-Terry (BT) assumption, then optimizing the LLM by maximizing the learned reward with KL regularization toward a reference policy, motivated by Proximal Policy Optimization (PPO) [26]. While effective, RLHF requires iterative on-policy sampling and a separately trained reward model. Direct Preference Optimization (DPO) [25] simplifies the pipeline by eliminating explicit reward modeling and online rollouts, associating preference scores with log-likelihood ratios between the trainable and reference models and maximizing margins favoring preferred responses under logistic loss. The success of DPO has inspired variants refining loss curvature [2], length normalization [21], or reference formulations [13].

Despite impressive performance, the methods in the family share a structural feature that has gone largely unexamined: the family optimizes over the class of prompt-conditioned policies $\overline{\Pi} = \{\pi :$

*Corresponding authors: `tlan@gwu.edu`, `qizhengling@gwu.edu`.

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

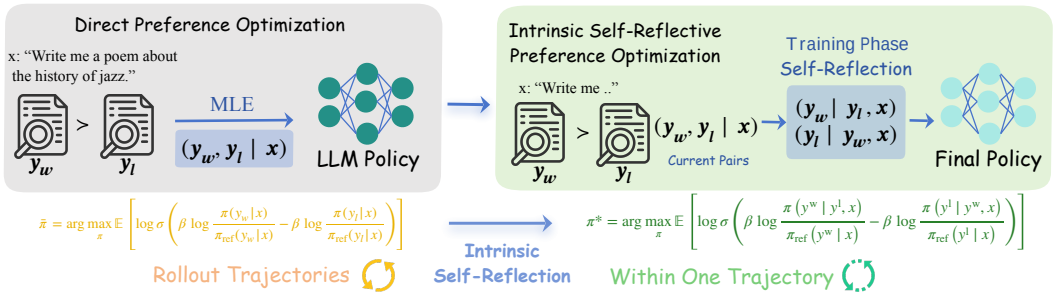


Figure 1: The figure contrasts standard DPO with INSPO on one preference pair. Standard DPO (left) learns from response comparisons where both the preferred and dispreferred responses are evaluated based solely on the prompt. INSPO (right) introduces intrinsic self-reflection through symmetric cross-conditioning: the policy generates the preferred response while seeing the dispreferred one as context, and vice versa, so that alternative responses serve as in-context guidance. Green terms highlight the conditioning mechanism in the objectives.

$\mathcal{X} \to \Delta(\mathcal{Y})\}$, in which each response is scored as a function of the prompt alone. The restriction has two consequences that have been treated as separate concerns in prior literature, yet are two manifestations of the same underlying defect. First, the optimum within $\bar{\Pi}$ is not invariant to the modeling choices used to define the optimization objective, namely the scalarization function Ψ and the reference policy π_{ref}. As we formally demonstrate, the "optimal" target shifts with such design choices, producing behavior that reflects parameterization artifacts instead of human preferences. Since human preferences are ordinal, the optimal policy should follow the ranking of responses, not the mathematical conventions used to fit it. Second, policies in $\bar{\Pi}$ cannot exploit the comparative structure of preference data, treating response generation as isolated maximization, which is at odds with established findings in behavioral economics that human preferences are non-separable and shaped by interactions between options instead of the standalone quality of each [28, 4]. We refer to the missing capability, namely generating a response while explicitly conditioning on the alternative it would be compared with, as *intrinsic self-reflection*.

Both consequences resolve under the same enlargement of the policy class. The enlarged class $\Pi = \{\pi : \mathcal{X} \times \mathcal{Y} \to \Delta(\mathcal{Y})\}$, in which the policy may condition on an alternative response y' in addition to the prompt, admits a population-level optimal target π^* within Π that is invariant to Ψ and π_{ref}, and renders self-reflection expressible at the level of the policy itself. Theorem 3.2(iii) further shows that the gap between π^* and $\bar{\pi}$ vanishes when the preference probability is separable into per-response scores, a regime ruled out by behavioral evidence [28, 4]. The two limitations are therefore not independent symptoms but two readings of the same restriction, both removed by the same enlargement of the policy class.

A Key Question

*How can we construct a framework that is both **invariant to modeling choices** and capable of **fully exploiting the comparative nature** of human preferences?*

In the present work, we propose Intrinsic Self-reflective Preference Optimization (INSPO; Figure 1), a new perspective that fully leverages comparative information in pairwise preference data. We first derive a population-level optimal target within the class of policies conditioning on both context and alternative responses, then prove the formulation yields a target superior to standard DPO and RLHF while guaranteeing invariance to scalarization and reference policy choices. To realize the formulation, we introduce a generic framework that adds the self-reflective capability to existing methods as a plug-and-play enhancement, decoupling alignment goals from arbitrary modeling constraints. Deployment remains standard at unchanged inference cost, since the self-reflective mechanism is incorporated into the policy during training and inference generates from the prompt alone. On AlpacaEval 2 and Arena-Hard with Mistral-7B-Instruct and Llama-3-8B-Instruct, INSPO yields gains of up to 8.5 points in length-controlled win rate and up to 12.9 points in raw win rate. The gains persist from 3B to 70B model scales at a training-time overhead of 18 to 25 percent and unchanged inference cost.

2 Background and Notation

To align an LLM with human preference, most existing pipelines first collect pairwise preference data generated by a pre-trained or SFT policy π_{ref}. Each sample consists of a context $x \in \mathcal{X}$ drawn from a distribution ρ and two responses $y_w, y_\ell \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the context and response spaces. After collecting such data, one can apply RLHF or DPO to fine-tune the LLM.

RLHF proceeds in two steps under the Bradley-Terry (BT) model, which assumes

$$\mathbb{P}(y_w \succ y_\ell \mid x) = \sigma\big(r(x, y_w) - r(x, y_\ell)\big), \tag{1}$$

where $\sigma(z) = 1/(1 + e^{-z})$ and r is an unknown reward function. Maximum likelihood on the preference data fits a reward model, and PPO [22] then optimizes the LLM through $\max_\pi \mathbb{E}_{x\sim\rho, y\sim\pi}[r(x, y)] - \beta D_{\text{KL}}(\pi \,\|\, \pi_{\text{ref}})$, where $\beta > 0$ controls the KL regularization toward π_{ref}. As an offline alternative, DPO [25] fine-tunes π via

$$\max_\pi \mathbb{E}_{(x,y_w,y_\ell)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi(y_\ell|x)}{\pi_{\text{ref}}(y_\ell|x)}\right)\right], \tag{2}$$

where $\mathcal{D}$ is the joint distribution of (x, y_w, y_ℓ), with y_w assumed preferred to y_ℓ without loss of generality.

As established in Proposition 1 of Azar et al. [2], both methods unify under a general preference optimization framework with the scalarization function $\Psi : [0, 1] \to \mathbb{R}$:

$$\max_\pi \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x),\, y'\sim\pi_{\text{ref}}(\cdot|x)}\left[\Psi\big(\mathbb{P}(y \succ y' \mid x)\big)\right], \tag{3}$$

where Ψ is an increasing function. Within the framework, existing methods seek an optimal policy $\bar\pi \in \operatorname{argmax}_{\pi\in\bar\Pi}\mathcal{V}(\pi)$ restricted to the class of context-conditioned policies $\bar\Pi = \{\pi : \mathcal{X} \to \Delta(\mathcal{Y})\}$, with the value $\mathcal{V}(\pi)$ given by the right-hand side of (3).

Throughout the paper, we assume access to a dataset of n pairwise preferences $\mathcal{D}_n = \{(x^{(i)}, y_w^{(i)}, y_\ell^{(i)})\}_{i=1}^n$, where $x^{(i)} \sim \rho$, the responses $(y_w^{(i)}, y_\ell^{(i)})$ are generated by π_{ref}, and $y_w^{(i)} \succ y_\ell^{(i)}$.

3 Limitations of Existing Methods

The present section examines two properties of the restricted optimal policy $\bar\pi$, namely its invariance to the scalarization function Ψ and the reference distribution π_{ref} and its optimality, and shows that $\bar\pi$ lacks both.

3.1 Invariance to Ψ and π_{ref}

Ideally, a robust alignment framework should yield an optimal policy invariant to the choice of Ψ and π_{ref}, ensuring modeling robustness and disentanglement from the reference. Since human preferences are ordinal, the optimal policy should follow the ranking of responses and should not depend on the transformation Ψ applied to preference probabilities or on the reference policy π_{ref}. The choice of Ψ, often made for numerical stability or concavity instead of semantic relevance, should not dictate the final behavior of the model; if the optimal fine-tuned policy changes based on modeling choices, the alignment process becomes brittle and inconsistent. Likewise, dependence on π_{ref} implies that the "optimal" behavior is relative and transient instead of converging toward a global optimum. Proposition 3.1 shows that the existing target $\bar\pi$ fails the invariance condition.

Proposition 3.1. *The form of $\bar\pi$ is not invariant to Ψ and π_{ref}.*

In the proof of Proposition 3.1, we provide counter-examples showing that $\bar\pi$ shifts when either Ψ or π_{ref} is varied. The lack of invariance raises a question about the quality of $\bar\pi$, since an "optimal" solution that shifts with modeling choices reflects the objective function more than the preferences themselves. The current target is moreover suboptimal in value, as Theorem 3.2 establishes below.

3.2 Optimality of $\bar{\pi}$

We now investigate the theoretical optimality of $\bar{\pi}$ in terms of maximizing $\mathcal{V}(\pi)$. While $\bar{\pi}$ maximizes $\mathcal{V}(\pi)$ over $\bar{\Pi}$, it is generally suboptimal compared to the population-level optimal target π^* within a broader class Π, defined as

$$\pi^* \in \operatorname{argmax}_{\pi \in \Pi} \mathcal{V}(\pi), \tag{4}$$

where $\Pi = \{\pi : \mathcal{X} \times \mathcal{Y} \to \Delta(\mathcal{Y})\}$ represents a broader policy class that conditions on both the context x and an auxiliary response y'. The policy π^* is a statistical target at the population level, and its attainment by an optimizer over a parameterized class is a separate question, which Theorem 4.4 addresses. Theorem 3.2 shows that π^* is superior to $\bar{\pi}$ and invariant to Ψ and π_{ref}.

Theorem 3.2. *(i) π^* is invariant to every increasing function Ψ and to π_{ref}; (ii) π^* dominates $\bar{\pi}$ in value:*

$$\begin{aligned}\mathcal{V}(\pi^*) &= \mathbb{E}_{x\sim\rho,\, y\sim\pi^*(\cdot|x,y'),\, y'\sim\pi_{\text{ref}}(\cdot|x)}\left[\Psi\big(\mathbb{P}(y \succ y' \mid x)\big)\right] \\ &\geq \mathcal{V}(\bar{\pi}) = \mathbb{E}_{x\sim\rho,\, y\sim\bar{\pi}(\cdot|x),\, y'\sim\pi_{\text{ref}}(\cdot|x)}\left[\Psi\big(\mathbb{P}(y \succ y' \mid x)\big)\right].\end{aligned}$$

(iii) Furthermore, given a fixed Ψ, π^* coincides with $\bar{\pi}$ if the transformed preference probability satisfies $\Psi\big(\mathbb{P}(y \succ y' \mid x)\big) \propto c(x, y) - c(x, y')$ for some function c.

Theorem 3.2 implies that the superiority of π^* over $\bar{\pi}$ stems from its dependence on the comparator response y', which introduces a notion of intrinsic *self-reflection* within the LLM. Existing fine-tuning methods lack the capacity for self-reflection, which matters for alignment. The same mechanism renders π^* invariant to both Ψ and π_{ref}, so the policy targets the human preference probability itself, whereas $\bar{\pi}$ targets optimization artifacts. While Theorem 3.2 (iii) suggests that self-reflection leaves the value unchanged if the preference function is separable, the separable condition is restrictive as it requires a correctly specified link function Ψ. In other words, Theorem 3.2 (iii) guarantees the optimality of $\bar{\pi}$ when the BT model (1) is correctly specified. More importantly, existing literature [28, 4] demonstrates that human preferences are non-separable and determined by the interaction between options, so the context x and the comparison of y with y' jointly shape the choice. To leverage the comparative property, we aim to learn π^* from the preference data $\mathcal{D}_n$, and since our preference data are paired, we do not consider larger categories of Π beyond pairwise comparison.

4 Intrinsic Self-reflective Preference Optimization

Throughout the section, π^*, π_r, and $\mathcal{V}$ denote population quantities, while hats mark estimates from $\mathcal{D}_n$. Since π^* is invariant to Ψ (Theorem 3.2 (i)), INSPO learns π^* under the log-odds scalarization $\Psi(q) = \log(q/(1-q))$ and the choice model

$$\mathbb{P}(y_w \succ y_\ell \mid x) = \sigma(2(r(x, y_w, y_\ell) - \beta \log Z(x, y_\ell))), \tag{5}$$

for some generic reward function $r : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and the normalizer $Z(x, y') = \sum_y \pi_{\text{ref}}(y \mid x) \exp(\frac{1}{\beta} r(x, y, y'))$. Then

$$\begin{aligned}\pi^* \in &\operatorname{argmax}_\pi \, 2\mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x,y'), y'\sim\pi_{\text{ref}}(\cdot|x)}\left[r(x, y, y') - \beta \log Z(x, y')\right] \\ = &\operatorname{argmax}_\pi \, \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x,y'), y'\sim\pi_{\text{ref}}(\cdot|x)}\left[r(x, y, y')\right],\end{aligned}$$

which is independent of Z, so the choice model (5) leaves the reward r generic. Following the paradigm of RLHF, we can estimate π^* via

$$\max_\pi \, \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x,y'), y'\sim\pi_{\text{ref}}(\cdot|x)}\left[r(x, y, y')\right] - \beta D_{\text{KL}}(\pi \,\|\, \pi_{\text{ref}}), \tag{6}$$

where

$$D_{\text{KL}}(\pi \,\|\, \pi_{\text{ref}}) = \mathbb{E}_{x\sim\rho, y'\sim\pi_{\text{ref}}(\cdot|x)}\Big[\text{KL}\big(\pi(\cdot \mid y', x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big)\Big].$$

The response y' in (6) is an integration variable of the population objective with law $\pi_{\text{ref}}(\cdot \mid x)$, so the offline pairs in $\mathcal{D}_n$, whose responses π_{ref} generates, supply it without online collection as long as they cover the support of $\pi_{\text{ref}}(\cdot \mid x)$. The reward function r may be hard to estimate and PPO is prone to instability, which motivates a family of DPO-based approaches for learning π^*. While the policy obtained from (6) relies on π_{ref} due to the KL divergence, it targets an optimal policy π^* that is independent of π_{ref}.

Theorem 4.1 connects the general reward function r with π_r, the optimal solution to (6).

Theorem 4.1. *Solving the optimization problem* (6) *gives*

$$r(x, y, y') = \beta\Big[\log \tfrac{\pi_r(y|x,y')}{\pi_{\text{ref}}(y|x)} + \log Z(x, y')\Big]. \tag{7}$$

Furthermore, π_r can be obtained by solving

$$\max_{\pi} \mathbb{E}_{(x,y_w,y_\ell)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\Big(\log \tfrac{\pi(y_w|x,y_\ell)}{\pi_{\text{ref}}(y_w|x)} - \log \tfrac{\pi(y_\ell|x,y_w)}{\pi_{\text{ref}}(y_\ell|x)}\Big)\Big)\Big]. \tag{8}$$

Based on Theorem 4.1 and the preference dataset $\mathcal{D}_n$, INSPO estimates π^* via

$$\min_{\theta} -\frac{1}{n}\sum_{i=1}^{n}\Big[\log\sigma\Big(\beta\Big(\log \tfrac{\pi_\theta(y_w^{(i)}|x,y_\ell^{(i)})}{\pi_{\text{ref}}(y_w^{(i)}|x)} - \log \tfrac{\pi_\theta(y_\ell^{(i)}|x,y_w^{(i)})}{\pi_{\text{ref}}(y_\ell^{(i)}|x)}\Big)\Big)\Big], \tag{9}$$

where we parametrize the trainable policy π by θ. Similar to DPO, INSPO aligns an LLM by shifting probability mass toward preferred responses and away from dispreferred ones, dispensing with RL, rollouts, and a reward model. Beyond what DPO achieves, our method leverages the self-reflection property embedded in the preference data to further calibrate the policy toward the preferred response by contrasting it with the dispreferred one, and thereby learns π^*.

Table 1: The table lists the INSPO variants of five representative preference optimization methods, with the original DPO-family terms in black and the INSPO-enhanced terms in green.

Method	Objective
DPO	$-\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_\ell\|x)}{\pi_{\text{ref}}(y_\ell\|x)}\Big)$
	$-\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w\|y_\ell,x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_\ell\|y_w,x)}{\pi_{\text{ref}}(y_\ell\|x)}\Big)$
IPO	$\Big(\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_\ell\|x)}{\pi_{\text{ref}}(y_\ell\|x)} - \frac{1}{2\tau}\Big)^2$
	$\Big(\log\frac{\pi_\theta(y_w\|y_\ell,x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_\ell\|y_w,x)}{\pi_{\text{ref}}(y_\ell\|x)} - \frac{1}{2\tau}\Big)^2$
R-DPO	$-\log\sigma\Big(\beta[\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_\ell\|x)}{\pi_{\text{ref}}(y_\ell\|x)}] + \alpha(\lvert y_w\rvert - \lvert y_\ell\rvert)\Big)$
	$-\log\sigma\Big(\beta[\log\frac{\pi_\theta(y_w\|y_\ell,x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_\ell\|y_w,x)}{\pi_{\text{ref}}(y_\ell\|x)}] + \alpha(\lvert y_w\rvert - \lvert y_\ell\rvert)\Big)$
ORPO	$-\log p_\theta(y_w\|x) - \lambda\log\sigma\Big(\text{logit}\, p_\theta(y_w\|x) - \text{logit}\, p_\theta(y_\ell\|x)\Big)$
	$-\log p_\theta(y_w\|y_\ell,x) - \lambda\log\sigma\Big(\text{logit}\, p_\theta(y_w\|y_\ell,x) - \text{logit}\, p_\theta(y_\ell\|y_w,x)\Big)$
SimPO	$-\log\sigma\Big(\frac{\beta}{\lvert y_w\rvert}\log\pi_\theta(y_w\|x) - \frac{\beta}{\lvert y_\ell\rvert}\log\pi_\theta(y_\ell\|x) - \gamma\Big)$
	$-\log\sigma\Big(\frac{\beta}{\lvert y_w\rvert}\log\pi_\theta(y_w\|y_\ell,x) - \frac{\beta}{\lvert y_\ell\rvert}\log\pi_\theta(y_\ell\|y_w,x) - \gamma\Big)$

$p_\theta(y|\cdot) = \exp(\frac{1}{|y|}\log\pi_\theta(y|\cdot))$; $\beta, \tau, \alpha, \lambda, \gamma$: hyperparameters.

During inference, instead of first generating y' from π_{ref} for self-reflection, which incurs an overhead cost, we directly deploy $\widehat{\pi}$ given a testing query x. While the procedure introduces a distinction between the training context (x, y') and the inference context x, we understand the distinction under the paradigm of *Learning Using Privileged Information* (LUPI) [24]. In the LUPI framework, the learning algorithm is provided with additional "privileged" information (here, the alternative response) during the training phase to stabilize the optimization landscape and accelerate convergence. The privileged context acts as a contrastive scaffold, guiding the gradient updates for the shared weights associated with x. Consequently, the self-reflective capability is distilled into the policy weights [14], allowing the deployed model to retain the optimized decision boundaries even when the privileged scaffolding is removed at test time. RLTF [27] likewise trains with a signal that is absent at inference, a textual critique gathered through online interaction, whereas the privileged context of INSPO is the alternative response already contained in the offline preference pair. The formulation for π^* is method-agnostic and integrates with standard preference optimization techniques, as Table 1 shows for five representative methods.

Finite-sample guarantee. Theorem 3.2 establishes the population-level superiority and invariance of π^* over $\bar{\pi}$, but does not address how closely the empirical INSPO estimator approximates the optimal policy from a finite preference dataset $\mathcal{D}_n$. We now provide a finite-sample regret bound for the INSPO estimator $\widehat{\pi}$ when optimization is performed over a continuous, parameterized policy class $\Pi_\Theta = \{\pi_\theta : \theta \in \Theta \subset \mathbb{R}^d\}$. The result quantifies the statistical efficiency of INSPO and reveals how the inverse-temperature parameter β enters the convergence rate. Recall that the INSPO objective induces a cross-conditioned implicit preference model:

$$p_\theta(y_w \succ y_\ell \mid x) = \sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x, y_\ell)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_\ell \mid x, y_w)}{\pi_{\text{ref}}(y_\ell \mid x)}\right). \tag{10}$$

We define the value of a policy π as the expected true reward $\mathcal{V}(\pi) = \mathbb{E}_{x\sim\rho}\,\mathbb{E}_{y'\sim\pi_{\mathrm{ref}}(\cdot|x),\, y\sim\pi(\cdot|x,y')}[r^*(x,y,y')]$. The suboptimality of the learned policy $\widehat{\pi}$ relative to the optimal policy π^* is quantified by its pure reward regret: $\mathrm{Regret}(\widehat{\pi}) = \mathcal{V}(\pi^*) - \mathcal{V}(\widehat{\pi})$. To establish the sample complexity of estimating $\widehat{\pi}$ from the preference dataset $\mathcal{D}_n$, we make the following structural assumptions.

Assumption 4.2 (Realizability and Parametric Capacity)**.** *There exists an optimal parameter $\theta^* \in \Theta$ such that the induced preference model perfectly matches the human preference distribution ($p_{\theta^*} = p^*$). Furthermore, Θ is a compact subset of $\mathbb{R}^d$ bounded by radius R, and the mapping $\theta \mapsto p_\theta$ is L-Lipschitz continuous with respect to the ℓ_2 norm.*

Assumption 4.3 (Bounded Logits and Rewards)**.** *There exists a constant $B > 0$ such that the true reward is bounded, $\sup_{x,y,y'} |r^*(x,y,y')| \le B$, and for all $\theta \in \Theta$, the magnitude of the implicit rewards is bounded: $\sup_{x,y,y'} \left|\beta \log \frac{\pi_\theta(y|x,y')}{\pi_{\mathrm{ref}}(y|x)}\right| \le B$.*

Assumption 4.3 ensures that the implicit preference probabilities are bounded strictly away from deterministic 0 and 1; specifically, $p_\theta \in [\tau_B, 1-\tau_B]$ where $\tau_B = \sigma(-2B) > 0$. Let $c_B = \sigma(2B)(1-\sigma(2B))$ denote the minimum derivative of the sigmoid function on the interval.

Theorem 4.4 (Finite-Sample Regret Bound for INSPO)**.** *Let $\widehat{\theta}$ be the empirical risk minimizer over $\mathcal{D}_n$, and let $\widehat{\pi} = \pi_{\widehat{\theta}}$ be the resulting policy. Under Assumptions 4.2 and 4.3, there exists a universal constant $C > 0$ such that for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of $\mathcal{D}_n$:*

$$\mathrm{Regret}(\widehat{\pi}) \le \frac{C \cdot B}{\beta \cdot c_B} \sqrt{\frac{d \log(LR\tau_B^{-1/2}\sqrt{d}) + \log(1/\delta)}{n}}. \tag{11}$$

Theorem 4.4 is an estimation-error result for a well-specified, bounded parametric preference model, and its regret decays at the standard parametric finite-sample rate $\mathcal{O}(d^{1/2}n^{-1/2})$. Under misspecification, the same argument bounds the regret relative to the best policy in Π_Θ, with an additional approximation error. The factor β^{-1} cannot be read monotonically in isolation, since Assumption 4.3 ties the bound B to β as well. The bound B also grows with response length, because a per-token bound b on the implicit reward gives $B = \mathcal{O}(bT)$ for T tokens and c_B deteriorates quickly in B. The guarantee is therefore not length-independent for unnormalized objectives, and the length-normalized variants in Appendix D mitigate the accumulation. The proof relies on empirical process theory [30, 29], utilizing bracketing entropy to bound the uniform convergence of the maximum likelihood estimator in Hellinger distance, with the complete proof in Appendix C.

Remark 4.5 (Deployment gap)**.** *Theorem 4.4 bounds the regret of the cross-conditioned policy $\widehat{\pi}(\cdot \mid x, y') \in \Pi_\Theta$, while deployment evaluates the prompt-only policy $\widehat{\pi}(\cdot \mid x)$ in a smaller class. With $\Delta_{\mathrm{dep}} = \mathcal{V}(\widehat{\pi}(\cdot \mid x, y')) - \mathcal{V}(\widehat{\pi}(\cdot \mid x))$, the guarantee transfers to the deployed policy up to the single term Δ_{dep}. The term lacks an a priori bound, and a proved bound remains an open problem. Section 5.3 measures the corresponding gap in win rate at most 0.6 points, and at most 0.5 points on Arena-Hard, with a token-level KL divergence below 0.05 nats per token between the two inference modes.*

5 Results

5.1 Experimental Setup

Our base models are Mistral-7B-Instruct-v0.2 and Llama-3-8B-Instruct, with the INS variants initialized from the checkpoints of the released baselines and Gemma2-9B added for the overhead measurements. Training data is sourced from UltraFeedback [7], containing approximately 60K preference pairs $(x; y_w, y_\ell)$ with $y_w \succ y_\ell$ after deduplication and safety filtering, following the data generating process of SimPO [21]. Evaluation covers three benchmarks, starting with AlpacaEval 2 [12], whose 805 diverse instructions are scored by both standard win rates (WR) and length-controlled win rates (LC) to mitigate verbosity bias. Arena-Hard [18] features 500 challenging queries that test advanced reasoning, with WR reported alongside its style-controlled variant (SC) to account for stylistic preferences. MT-Bench [35] comprises 80 multi-turn questions spanning eight capability categories, scored on a 10-point scale.

Table 2: The table reports benchmark performance on AlpacaEval 2, Arena-Hard, and MT-Bench for Mistral-Instruct (7B) and Llama-3-Instruct (8B). Shaded rows denote cross-conditioned variants with deltas computed relative to their corresponding baselines. Bold values indicate the highest performance in each metric. Baseline rows come from the released SimPO checkpoints, and the INS variants use identical training configurations.

Method	Mistral-Instruct (7B)					Llama-3-Instruct (8B)				
	AlpacaEval2		Arena-Hard	MT-Bench		AlpacaEval2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4o	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4o	GPT-4
Baseline	17.1	14.7	12.6	6.2	7.5	26.0	25.3	22.3	6.9	8.1
Reward-Free Preference Optimization										
RRHF	25.3	24.8	18.1	6.5	7.6	31.3	28.4	26.5	6.7	7.9
SLiC-HF	24.1	24.6	18.9	6.5	7.8	26.9	27.5	26.2	6.8	8.1
CPO	23.8	28.8	22.6	6.3	7.5	28.9	32.2	28.8	7.0	8.0
KTO	24.5	23.6	17.9	6.4	7.7	33.1	31.8	26.4	6.9	8.2
DPO-Family Methods and Their INS *Variants*										
IPO	20.3	20.3	16.2	6.4	7.8	35.6	35.6	30.5	7.0	8.3
+ INS	28.8 ↑8.5	27.9 ↑7.6	18.5 ↑2.3	6.5 ↑0.1	**7.9** ↑0.1	37.7 ↑2.1	41.3 ↑5.7	32.8 ↑2.3	**7.1** ↑0.1	**8.5** ↑0.2
ORPO	24.5	24.9	20.8	6.4	7.7	28.5	27.4	25.8	6.8	8.0
+ INS	24.6 ↑0.1	24.7 ↓0.2	21.5 ↑0.7	6.3 ↓0.1	7.8 ↑0.1	35.7 ↑7.2	40.3 ↑12.9	25.5 ↓0.3	6.8 – 0.0	8.0 – 0.0
R-DPO	27.3	24.5	16.1	6.2	7.5	41.1	37.8	33.1	7.0	8.0
+ INS	29.1 ↑1.8	27.6 ↑3.1	19.0 ↑2.9	6.4 ↑0.2	7.6 ↑0.1	41.0 ↓0.1	43.8 ↑6.0	35.0 ↑1.9	**7.1** ↑0.1	8.1 ↑0.1
DPO	26.8	24.9	16.3	6.3	7.6	40.3	37.9	32.6	7.0	8.0
+ INS	29.9 ↑3.1	29.4 ↑4.5	23.9 ↑7.6	6.5 ↑0.2	7.8 ↑0.2	40.3 – 0.0	41.1 ↑3.2	35.3 ↑2.7	**7.1** ↑0.1	8.1 ↑0.1
SimPO	32.1	**34.6**	21.0	**6.6**	7.6	**44.5**	40.5	33.8	7.0	8.0
+ INS	**32.7** ↑0.6	34.5 ↓0.1	**24.4** ↑3.4	**6.6** – 0.0	7.7 ↑0.1	**44.5** – 0.0	**46.6** ↑6.1	**35.9** ↑2.1	**7.1** ↑0.1	8.2 ↑0.2

LC: length-controlled win rate; WR: win rate. The INS variants initialize from the instructed checkpoints of the released baselines within each model family. **INS** applies symmetric cross-conditioning across preference pairs.

The INS-enhanced variants INS-DPO, INS-SimPO, INS-IPO, INS-R-DPO, and INS-ORPO are compared against their standard counterparts and additional baselines RRHF, SLiC-HF, CPO, and KTO. For fair comparison, baseline results come from publicly released checkpoints of SimPO [21] evaluated on identical benchmark versions. The INS-enhanced methods are implemented in the OpenRLHF framework [15] and trained for three epochs using the AdamW optimizer with learning rate 5×10^{-7}, a cosine schedule with 10% warmup, and maximum context length 4096; method-specific hyperparameters are searched over the ranges in Appendix E. The baselines and the INS variants cap responses at 2048 tokens in training and evaluation, and the 4096-token window of INS only packages the concatenated triple (x, y_ℓ, y_w). Retraining DPO, SimPO, and their INS variants in our pipeline reproduces the reported metrics within 0.3 points (Table 8). At inference, the models use standard autoregressive generation $\hat{y} \sim \widehat{\pi}(\cdot \mid x)$ with nucleus sampling at $p = 0.95$ and temperature 0.7, at the computational cost of the baseline methods.

5.2 Main Results

On the length-controlled AlpacaEval 2 (AE) metric in Table 2, INSPO yields its largest gains on Mistral-7B, where INS-IPO and INS-DPO improve by 8.5 and 3.1 points, while on Llama-3-8B the largest gain comes from INS-ORPO at 7.2 points. On Arena-Hard (AH), INSPO improves nine of ten pairs, except ORPO on Llama-3-8B, with the largest gain of 7.6 points for INS-DPO on Mistral-7B.

Beyond the controlled metric, INS-DPO raises raw win rates on AlpacaEval 2 and Arena-Hard for both models and INS-SimPO does so on Llama-3-8B, indicating quality improvements beyond a reduction in verbosity. On MT-Bench, the INS-enhanced methods stay within 0.2 points of their baselines under both judges, improving in 16 of the 20 comparisons.

The relative ranking among DPO family members remains stable under INS enhancement: SimPO variants lead, followed by DPO and R-DPO, with IPO and ORPO showing more variable performance, which suggests that INSPO preserves the relative strengths of the base methods. Compared to alternative baselines beyond the DPO family such as RRHF, SLiC-HF, CPO, and KTO, our best INS-

Table 3: The table reports the ablation on candidate length constraints, where MaxLen limits the total sequence length of (x, y_ℓ, y_w) and Draft Cap α enforces $|y_\ell| \leq \alpha \cdot \mathrm{MaxLen}$ for the dispreferred candidate only.

MaxLen	Draft Cap	AE-LC	AE-WR	AH-SC	AH-WR
1024	30%	39.1	37.4	30.8	32.1
	40%	39.8	**38.2**	**31.2**	32.8
	50%	**39.9**	37.9	30.5	**33.1**
	None	38.2	36.1	29.8	31.2
2048	30%	39.9	38.6	32.3	34.1
	40%	**40.3**	**39.2**	32.9	**34.7**
	50%	40.1	38.9	**33.1**	34.3
	None	39.1	37.8	31.5	33.2
4096	30%	40.1	39.5	33.2	35.4
	40%	**40.5**	**39.9**	**33.8**	**35.9**
	50%	40.2	39.7	33.4	35.6
	None	39.8	38.9	32.8	34.9

Table 4: The table compares conditioning strategies and simple averaging for length control. The experiments use MaxLen=4096 with 40% draft cap. Deltas are computed relative to the DPO baseline.

Model	Strategy	AlpacaEval 2		Arena-Hard
		LC (%)	WR (%)	WR (%)
Mistral-7B	DPO (baseline)	26.8	24.9	16.3
	One-sided	28.7 ↑1.9	27.1 ↑2.2	19.2 ↑2.9
	Symmetric	29.9 ↑3.1	29.4 ↑4.5	23.9 ↑7.6
	Averaged	29.6 ↑2.8	29.7 ↑4.8	25.7 ↑9.4
Llama-3-8B	DPO (baseline)	40.3	37.9	32.6
	One-sided	43.4 ↑3.1	40.2 ↑2.3	34.1 ↑1.5
	Symmetric	40.3 −0.0	41.1 ↑3.2	35.3 ↑2.7
	Averaged	44.1 ↑3.8	41.5 ↑3.6	35.5 ↑2.9

enhanced variant INS-SimPO outperforms these baselines on the three AlpacaEval 2 and Arena-Hard metrics for both models.

5.3 Analysis and Insights

The ablations use INS-DPO on Llama-3-8B-Instruct unless stated otherwise.

Context budget and conditioning strategies. ***MaxLen*** constrains the total sequence length of the training tuple (x, y_ℓ, y_w), while ***Draft Cap*** α enforces $|y_\ell| \leq \alpha \cdot \mathrm{MaxLen}$ for the dispreferred candidate alone, leaving headroom for x and y_w to remain untruncated. Table 3 shows that longer context windows improve performance, with 4096-token contexts performing best, which indicates the importance of keeping the full (x, y_ℓ, y_w) tuple untruncated. Under the same 2048-token window as DPO, INS-DPO with a 40% cap still reaches raw win rates of 39.2 on AE and 34.7 on AH, against 37.9 and 32.6 for DPO, at an equal AE-LC of 40.3. Moderate draft caps ($\alpha = 40\%$) outperform uncapped drafts, in which occasional long dispreferred responses dominate the context and destabilize gradient flow. The benefit of longer contexts is largest on Arena-Hard, which suggests that complex reasoning tasks benefit more from richer candidate context. We further examine whether conditioning strategies balance quality with length control, comparing four variants: standard DPO ($\mathcal{L}_{\mathrm{DPO}}$); one-sided conditioning, which conditions only the preferred response on the dispreferred alternative; symmetric cross-conditioning ($\mathcal{L}_{\mathrm{sym}}$, the INSPO objective in Eq. (9)); and the empirical average $\mathcal{L}_{\mathrm{avg}} = \frac{1}{2}\mathcal{L}_{\mathrm{DPO}} + \frac{1}{2}\mathcal{L}_{\mathrm{sym}}$. In Table 4, symmetric cross-conditioning improves over the DPO baseline on both models, with its largest gain of 7.6 points on Arena-Hard for Mistral-7B. One-sided conditioning also improves the three metrics on both models, which indicates that conditioning on the alternative response provides value independent of symmetry. The averaged variant attains the best Arena-Hard and raw AlpacaEval 2 win rates on both models and the best length-controlled rate on Llama-3-8B, while symmetric conditioning keeps the higher length-controlled rate on Mistral-7B.

Train-test gap analysis. INS trains with paired candidates yet deploys under prompt-only inference, raising the question of whether the contrastive signal is accessible at test time. The token-level KL divergence and a draft-then-revise comparison both show that the signal is absorbed into the shared parameters instead of remaining a dependency at inference. First, on Llama-3-8B-Instruct, the token-level KL divergence between cross-conditioned and prompt-only inference is 0.043 nats per token when $y' = y_\ell$ and 0.037 nats per token when $y' = y_w$, indicating that the conditioning context exerts only a small influence on the deployed policy regardless of response quality. For reference, the divergence between INS-DPO and standard DPO under identical prompt-only inference is 0.31 nats per token, an order of magnitude larger. Second, a ***Draft-then-Revise*** experiment compares two protocols under the same INS-trained model: *Standard* inference generates $\hat{y} \sim \widehat{\pi}(\cdot \mid x)$ from the prompt alone, while *Draft-then-Revise* first samples $y_{\mathrm{draft}} \sim \pi_{\mathrm{ref}}(\cdot \mid x)$ and then generates $\hat{y} \sim \widehat{\pi}(\cdot \mid x, y_{\mathrm{draft}})$. Across the six comparisons in Table 5, Draft-then-Revise yields at most $+0.6$

Table 5: The table compares standard and Draft-then-Revise inference on Llama-3-8B-Instruct, where Standard denotes prompt-only inference and Draft-then-Revise conditions on an explicitly drafted candidate.

Method	AE-LC	AE-WR	AH-WR
DPO (baseline)	40.3	37.9	32.6
INS-DPO (Standard)	40.3	41.1	35.3
INS-DPO (Draft-then-Revise)	40.5	41.7	35.8
SimPO (baseline)	44.5	40.5	33.8
INS-SimPO (Standard)	44.5	46.6	35.9
INS-SimPO (Draft-then-Revise)	44.7	47.1	36.3

Table 6: The table reports INS-DPO at three model scales on AlpacaEval 2 (AE) and Arena-Hard (AH).

Model	Method	AE-LC	AE-WR	AH-WR
Llama-3.2-3B	DPO	26.5	24.1	15.8
	INS-DPO	27.8 ↑1.3	28.4 ↑4.3	19.7 ↑3.9
Llama-3-8B	DPO	40.3	37.9	32.6
	INS-DPO	40.3 −0.0	41.1 ↑3.2	35.3 ↑2.7
Llama-3.1-70B	DPO	50.2	48.3	47.1
	INS-DPO	50.9 ↑0.7	50.1 ↑1.8	48.9 ↑1.8

additional points over standard inference, with AE-LC indistinguishable, and standard inference captures over 80% of the total INS gain. INS therefore transfers to deployment under an unmodified protocol and unchanged inference compute, analogous to knowledge distillation, in which teacher signals shape the weights but are absent at deployment.

Scaling and invariance validation. To assess whether the gains from INS persist across model sizes, we evaluate at three scales spanning roughly an order of magnitude: Llama-3.2-3B-Instruct, Llama-3-8B-Instruct, and Llama-3.1-70B-Instruct. Table 6 shows improvements at the three scales, with the largest gains on the smallest model, where the base policy has the most room for improvement, and gains preserved at 70B against a strong baseline. The 3B model gains 4.3 points in raw AlpacaEval 2 win rate and 3.9 points on Arena-Hard. The 70B model still gains 1.8 points on both benchmarks from a baseline above 47% win rate, which suggests that baseline saturation is not the limiting factor.

Theorem 3.2 establishes that the optimal cross-conditioned policy π^* is invariant to the scalarization function Ψ and the reference policy π_{ref}. Beyond the indirect evidence in Table 2, where INS improves DPO-family methods with distinct Ψ and π_{ref} designs, we test the invariance directly. We train INS-DPO and INS-IPO on Llama-3-8B-Instruct under their respective Ψ and π_{ref} configurations and measure the pairwise agreement rate between the preference rankings of the two models on 805 AlpacaEval prompts. Standard DPO uses the log-odds scalarization with a soft KL regularizer toward π_{ref}, whereas IPO uses the identity scalarization fitted through a squared loss, so the two methods differ in both the form of Ψ and the implicit role of the reference. Standard DPO and IPO produce moderately divergent rankings due to their different Ψ choices, with an agreement rate of 78.3%. After applying INS, the agreement rate increases to 86.1%, indicating that INS drives both methods toward a more consistent policy less dependent on the specific scalarization function. The result provides direct empirical support for the invariance property: two methods that differ in their objective formulation converge toward more consistent behavior once the policy is optimized over the enlarged class.

Memorization and computational overhead. To check whether INS learns to copy the dispreferred response instead of contrasting against it, we measure n-gram overlap between INS-generated outputs and the training-time alternative responses on 805 AlpacaEval prompts using Llama-3-8B-Instruct. BLEU-4 is 2.4 for INS-DPO versus 2.1 for DPO; ROUGE-L is 19.1 versus 18.3; unigram overlap is 32.8% versus 31.5%. The marginal differences indicate that INS does not memorize the alternative response. Cross-conditioning incurs 18 to 25% training-time overhead across Mistral-7B, Llama-3-8B, and Gemma2-9B due to longer inputs $(x \oplus y_\ell)$ and $(x \oplus y_w)$, detailed in Appendix E (Tables 9 and 10). *Inference remains identical to standard DPO with zero latency overhead*, since models generate through standard sampling $\hat{y} \sim \pi_\theta(\cdot \mid x)$ and skip the draft-revise loop, whose drafted candidate would lengthen the context and the key-value cache held during decoding [20].

6 Conclusion

In this work, we have addressed two limitations in current preference optimization frameworks: the lack of invariance to modeling choices and the suboptimality arising from treating response generation in isolation. We argued that standard methods like RLHF and DPO produce aligned policies that are often artifacts of parameterization instead of reflections of human intent, and that

these methods leave the comparative nature of preference data unused. By deriving a population-level optimal target over policies conditioned on both context and alternative response, INSPO achieves a target superior to standard DPO and RLHF baselines while remaining invariant to scalarization function and reference policy choices. As a plug-and-play enhancement to DPO-family objectives, INSPO delivers alignment improvements across benchmarks and model scales from 3B to 70B with zero inference overhead. Since INSPO is agnostic to the source of the preference labels, whether humans, AI judges, reward models, or ranked verifiable rewards, it complements verifiable-reward RL within hybrid pipelines, where conditioning on the correct and incorrect responses of a sampled group also improves GRPO [19]. Future work may extend the paradigm to online training and multi-turn settings.

References

[1] 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.

[2] 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.

[3] 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*, 2022.

[4] Pedro Bordalo, Nicola Gennaioli, and Andrei Shleifer. Salience and consumer choice. *Journal of Political Economy*, 121(5):803–843, 2013.

[5] Da Chang and Yu Li. Mixed text recognition with efficient parameter fine-tuning and transformer. In *International Conference on Neural Information Processing*, 2024. URL `https://api.semanticscholar.org/CorpusID:269282910`.

[6] Jaymari Chua, Yun Li, Shiyi Yang, Chen Wang, and Lina Yao. Ai safety in generative ai large language models: A survey. *arXiv preprint arXiv:2407.18369*, 2024.

[7] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *ArXiv*, abs/2310.01377, 2023. URL `https://api.semanticscholar.org/CorpusID:263605623`.

[8] Adam Dahlgren Lindström, Leila Methnani, Lea Krause, Petter Ericson, Íñigo Martínez de Rituerto de Troya, Dimitri Coelho Mollo, and Roel Dobbe. Helpful, harmless, honest? sociotechnical limits of ai alignment and safety through reinforcement learning from human feedback: Ad lindström et al. *Ethics and Information Technology*, 27(2):28, 2025.

[9] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

[10] Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*, 2023.

[11] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

[12] 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.

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

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.

[16] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A Smith, Yejin Choi, and Hanna Hajishirzi. Unpacking dpo and ppo: Disentangling best practices for learning from preference feedback. *Advances in neural information processing systems*, 37:36602–36633, 2024.

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

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

[19] Yu Li, Tian Lan, and Zhengling Qi. When right meets wrong: Bilateral context conditioning with reward-confidence correction for grpo. *arXiv preprint arXiv:2603.13134*, 2026.

[20] Yu Li, Binxu Li, and Tian Lan. Momentkv: Closing the directional gap in kv cache eviction for long-context inference. *arXiv preprint arXiv:2606.01563*, 2026.

[21] 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.

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

[23] Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024.

[24] Dmitry Pechyony and Vladimir Naumovich Vapnik. On the theory of learning with privileged information. In *Neural Information Processing Systems*, 2010. URL `https://api.semanticscholar.org/CorpusID:13217651`.

[25] 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.

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

[27] Yuda Song, Lili Chen, Fahim Tajwar, Remi Munos, Deepak Pathak, J Andrew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. *arXiv preprint arXiv:2602.02482*, 2026.

[28] Amos Tversky and Itamar Simonson. Context-dependent preferences. *Management Science*, 39 (10):1179–1189, 1993.

[29] Sara A. van de Geer. *Empirical Processes in M-estimation*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, UK, 2000. doi: 10.1017/CBO9780511622631.

[30] Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer New York, New York, NY, 1996. doi: 10.1007/978-1-4757-2545-2.

[31] 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*, 2024.

[32] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36:10935–10950, 2023.

[33] Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.

[34] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

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

Appendix

A Related Work

Reinforcement Learning from Human Feedback. RLHF is a fundamental technique for aligning large language models with human preferences and values [3]. The pipeline typically comprises three phases: supervised fine-tuning, reward model training, and policy optimization [11, 5]. Proximal Policy Optimization is widely used for the policy optimization step, where the policy is trained to maximize the learned reward while staying close to the reference model through KL regularization. RLHF has been applied to ensuring safety, enhancing helpfulness, and improving model reasoning abilities [9, 6, 8], but recent work has highlighted challenges across the PPO-based pipeline, from preference pattern to model training [1], including biased outcomes such as verbose outputs and length exploitation. Given the complexity of online preference optimization, recent work has explored more efficient and simpler offline alternatives [25, 33].

Direct Preference Optimization and Variants. DPO is an offline preference optimization approach that eliminates the need for explicit reward modeling by reparameterizing the reward function, achieving simplicity and training stability [25, 16]. The method directly learns a policy from preference data by expressing preferences through policy and reference model ratios, avoiding the online optimization of classical RLHF. Numerous variants have since been proposed: Identity Preference Optimization (IPO) introduces a theoretically grounded framework that avoids DPO's pointwise-reward assumption, using a squared loss objective instead [2]; Contrastive Preference Optimization (CPO) incorporates sequence likelihood as reward and trains alongside an SFT objective for improved calibration [31]; Kahneman-Tversky Optimization (KTO) extends preference optimization to non-paired data through prospect-theoretic modeling [13]; Regularized DPO (R-DPO) adds explicit length regularization to prevent length exploitation [23]; and Simple Preference Optimization (SimPO) proposes a reference-free reward formulation using length-normalized average log probability with a target reward margin, eliminating the reference model during training [21].

B Notation

Symbol	Meaning	Symbol	Meaning
Basic Elements			
x	Prompt / input context	y	Response sequence
y_w, y_ℓ	Preferred and dispreferred	y^+, y^-	Winner and loser (Appendix)
$\lvert y\rvert$	Response length (tokens)	y_t	Token at position t
Models and Policies			
$\pi_\theta(\cdot \mid x)$	Trainable policy	$\pi_\theta(y \mid x, y_\ell)$	Cross-conditioned policy
$\pi_{\text{ref}}(\cdot \mid x)$	Reference policy	$p_\theta(y \mid c)$	Length-normalized probability
Reward and Scoring			
$r(x, y, y')$	Reward function	m	Preference margin
$\sigma(z)$	Sigmoid $1/(1+e^{-z})$		
Hyperparameters			
$\beta > 0$	Inverse temperature	$\gamma \geq 0$	Target margin (SimPO)
α	Length coeff. (R-DPO)	λ	Odds ratio weight (ORPO)
τ	Temperature (IPO)		
Datasets and Loss Functions			
$\mathcal{D}_n$	Preference dataset	$\mathcal{X}, \mathcal{Y}$	Prompt and response spaces
n	Dataset size	$(x^{(i)}, y_w^{(i)}, y_\ell^{(i)})$	i-th training example
$\mathcal{L}_{\text{DPO}}$	DPO loss	$\mathcal{L}_{\text{bi}}$	Bidirectional loss
Technical Terms			
$Z(x, y')$	Partition function	$\mathbb{P}(y_w \succ y_\ell \mid x)$	Preference probability
$\mathbb{E}_{(x,y_w,y_\ell)\sim\mathcal{D}}[\cdot]$	Expectation over data distribution	D_{KL}	KL divergence
ρ	Context distribution		
Evaluation Metrics and Operators			
LC	Length-controlled WR	WR	Raw win rate
SC	Style-controlled WR	$\succ$	Preference relation
$\oplus$	Concatenation		

Table 7: The table collects the notation used throughout the paper.

C Technical Proofs

Proof of Proposition 3.1.

Proof. The objective function is

$$\mathcal{V}(\pi) = \mathbb{E}_{x\sim\rho}\left\{\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\sum_{y'}\pi_{\text{ref}}(y'|x)\,\Psi\big(\mathbb{P}(y\succ y'|x)\big)\right]\right\}.$$

For a fixed input x, maximizing $\mathcal{V}(\pi)$ is equivalent to choosing y that maximizes the inner expectation. Define the score

$$\mathcal{S}(y;\Psi,\pi_{\text{ref}}) \triangleq \sum_{y'}\pi_{\text{ref}}(y'|x)\,\Psi\big(\mathbb{P}(y\succ y'|x)\big).$$

By the definition of $\bar{\pi}$,

$$\bar{\pi}(y|x) = \begin{cases}1, & \text{if } y = y_{\max}, \text{ where } y_{\max} = \operatorname{argmax}_y \mathcal{S}(y;\Psi,\pi_{\text{ref}})\\ 0, & \text{otherwise,}\end{cases}$$

assuming uniqueness of the maximization.

To prove that $\bar{\pi}$ depends on Ψ and π_{ref}, we construct counter-examples using two candidates $\bar{y}_1, \bar{y}_2$ and reference support $\{y'_a, y'_b\}$ with win-rates $\mathbb{P}(\bar{y}_1 \succ y'_a) = 0.9$, $\mathbb{P}(\bar{y}_1 \succ y'_b) = 0.2$, and $\mathbb{P}(\bar{y}_2 \succ y'_a) = \mathbb{P}(\bar{y}_2 \succ y'_b) = 0.56$.

Dependence on Ψ (fixing π_{ref}). Let $\pi_{\text{ref}}(y'_a|x) = \pi_{\text{ref}}(y'_b|x) = 0.5$. With identity mapping $\Psi_{\text{id}}(q) = q$, we obtain $\mathcal{S}(\bar{y}_1; \Psi_{\text{id}}, \pi_{\text{ref}}) = 0.55$ and $\mathcal{S}(\bar{y}_2; \Psi_{\text{id}}, \pi_{\text{ref}}) = 0.56$, so $y_{\max} = \bar{y}_2$. With log-odds mapping $\Psi_{\log}(q) = \log(q/(1-q))$, we obtain $\mathcal{S}(\bar{y}_1; \Psi_{\log}, \pi_{\text{ref}}) \approx 0.41$ and $\mathcal{S}(\bar{y}_2; \Psi_{\log}, \pi_{\text{ref}}) \approx 0.24$, so $y_{\max} = \bar{y}_1$. Thus $\bar{\pi}$ depends on Ψ.

Dependence on π_{ref} (fixing Ψ). Fix $\Psi = \Psi_{\text{id}}$. With uniform reference $(0.5, 0.5)$, $y_{\max} = \bar{y}_2$ as shown. With skewed reference $\pi^{(2)}_{\text{ref}}(y'_a|x) = 0.9$, $\pi^{(2)}_{\text{ref}}(y'_b|x) = 0.1$, we obtain $\mathcal{S}(\bar{y}_1; \Psi_{\text{id}}, \pi^{(2)}_{\text{ref}}) = 0.83$ and $\mathcal{S}(\bar{y}_2; \Psi_{\text{id}}, \pi^{(2)}_{\text{ref}}) = 0.56$, so $y_{\max} = \bar{y}_1$. Thus $\bar{\pi}$ also depends on π_{ref}.

Since $y_{\max}$ changes with either Ψ or π_{ref}, the optimal policy is not invariant to the objective formulation. □

Proof of Theorem 3.2.

Proof. Since $\bar{\Pi} \subseteq \Pi$, the second claim follows directly. For the first claim, the objective function is

$$\mathcal{V}(\pi) = \mathbb{E}_{x\sim\rho, y'\sim\pi_{\text{ref}}(\cdot|x)} \left[\sum_y \pi(y|x, y') \, \Psi\big(\mathbb{P}(y \succ y'|x)\big) \right].$$

For a fixed x and y', maximizing $\mathcal{V}(\pi)$ over Π is equivalent to choosing the y that maximizes $\Psi\big(\mathbb{P}(y \succ y'|x)\big)$. By the definition of π^*,

$$\pi^*(y|x, y') = \begin{cases} 1, & \text{if } y = y_{\max}, \text{ where } y_{\max} = \text{argmax}_y \Psi\big(\mathbb{P}(y \succ y'|x)\big) \\ 0, & \text{otherwise.} \end{cases}$$

By the non-decreasing property of Ψ, the output $y_{\max}$ of π^* is invariant to Ψ and π_{ref}.

For the last claim, without loss of generality, assume $\Psi(\mathbb{P}(y \succ y' \mid x)) = c(x, y) - c(x, y')$. Then

$$\begin{aligned} \max_{\pi\in\bar{\Pi}} \mathcal{V}(\pi) &= \max_{\pi\in\bar{\Pi}} \mathbb{E}_{x\sim\rho} \left\{ \mathbb{E}_{y\sim\pi(\cdot|x)}[c(x, y)] - \mathbb{E}_{y'\sim\pi_{\text{ref}}(\cdot|x)}[c(x, y')] \right\} \\ &= \max_{\pi\in\Pi} \mathbb{E}_{x\sim\rho} \left\{ \mathbb{E}_{y'\sim\pi_{\text{ref}}(\cdot|x)} \left(\mathbb{E}_{y\sim\pi(\cdot|y',x)}[c(x, y)] - \mathbb{E}_{y'\sim\pi_{\text{ref}}(\cdot|x)}[c(x, y')] \right) \right\} \\ &= \max_{\pi\in\Pi} \mathcal{V}(\pi), \end{aligned}$$

where the second equality uses that the first c does not depend on y'. Therefore π^* coincides with $\bar{\pi}$. □

Proof of Theorem 4.1.

Proof. The optimization problem is

$$\begin{aligned} &\max_\pi \mathbb{E}_{x\sim\rho} \mathbb{E}_{y\sim\pi(\cdot|x,y'), y'\sim\pi_{\text{ref}}(\cdot|x)}[r(x, y, y')] - \beta D_{\text{KL}}(\pi \| \pi_{\text{ref}}) \\ &= \max_\pi \mathbb{E}_{x\sim\rho, y'\sim\pi_{\text{ref}}(\cdot|x)} \left[\sum_y \pi(y|x, y') r(x, y, y') - \beta \sum_y \pi(y|x, y') \log \frac{\pi(y|x, y')}{\pi_{\text{ref}}(y|x)} \right] \\ &= \max_\pi \beta \mathbb{E}_{x\sim\rho, y'\sim\pi_{\text{ref}}(\cdot|x)} \left[\sum_y \pi(y|x, y') \log \frac{\exp(r(x, y, y')/\beta)\pi_{\text{ref}}(y|x)}{\pi(y|x, y')Z(x, y')} \right] \\ &\quad + \beta \mathbb{E}_{x\sim\rho, y'\sim\pi_{\text{ref}}(\cdot|x)} \left[\sum_y \pi(y|x, y') \log Z(x, y') \right]. \end{aligned}$$

The optimum is the Gibbs distribution

$$\pi_r(y|x, y') = \frac{\exp(r(x, y, y')/\beta)\pi_{\text{ref}}(y|x)}{Z(x, y')}, \tag{12}$$

where $Z(x, y') = \sum_y \pi_{\text{ref}}(y|x) \exp(r(x, y, y')/\beta)$. Hence the reward admits the form

$$r(x, y, y') = \beta \log \frac{\pi_r(y|x, y')}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x, y'). \tag{$*$}$$

For the second claim, by Equation (5),

$$\begin{aligned}
\mathbb{P}(y_\ell \succ y_w|x) &= \sigma(2(r(x, y_\ell, y_w) - \beta \log Z(x, y_w))) \\
&= 1 - \sigma(2(r(x, y_w, y_\ell) - \beta \log Z(x, y_\ell))) \\
&= \sigma(-2(r(x, y_w, y_\ell) - \beta \log Z(x, y_\ell))),
\end{aligned}$$

using $\sigma(z) = 1 - \sigma(-z)$. Therefore $r(x, y_w, y_\ell) - \beta \log Z(x, y_\ell) = \beta \log Z(x, y_w) - r(x, y_\ell, y_w)$. Using $(*)$,

$$\begin{aligned}
r(x, y_\ell, y_w) &= \beta \left(\log \frac{\pi_r(y_\ell|x, y_w)}{\pi_{\text{ref}}(y_\ell|x)} + \log Z(x, y_w) \right), \\
r(x, y_w, y_\ell) &= \beta \left(\log \frac{\pi_r(y_w|x, y_\ell)}{\pi_{\text{ref}}(y_w|x)} + \log Z(x, y_\ell) \right),
\end{aligned}$$

giving

$$\begin{aligned}
2(r(x, y_w, y_\ell) - \beta \log Z(x, y_\ell)) &= r(x, y_w, y_\ell) - \beta \log Z(x, y_\ell) + \beta \log Z(x, y_w) - r(x, y_\ell, y_w) \\
&= \beta \left(\log \frac{\pi_r(y_w|x, y_\ell)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_r(y_\ell|x, y_w)}{\pi_{\text{ref}}(y_\ell|x)} \right).
\end{aligned}$$

Substituting into (5),

$$\mathbb{P}(y_w \succ y_\ell \mid x) = \sigma\left(\beta \left(\log \frac{\pi_r(y_w|x, y_\ell)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_r(y_\ell|x, y_w)}{\pi_{\text{ref}}(y_\ell|x)} \right) \right),$$

which concludes the proof. □

Proof of Theorem 4.4. We analyze the uniform convergence of the empirical risk minimizer using empirical process theory, formally bounding the estimation error over the continuous parameter space without loose finite ϵ-net approximations.

Preliminaries and the Hellinger metric. Let $z = (x, y_1, y_2)$ denote a context and two candidate responses, and let $b \in \{0, 1\}$ be the binary label indicating whether $y_1 \succ y_2$. The conditional Bernoulli likelihood is $q_\theta(b \mid z) = p_\theta(z)^b (1 - p_\theta(z))^{1-b}$. The INSPO objective trains $\hat{\theta}$ via maximum likelihood estimation, equivalently $\mathbb{P}_n \log q_{\hat{\theta}} \geq \mathbb{P}_n \log q_{\theta^*}$, where $\mathbb{P}_n$ is the empirical measure over $\mathcal{D}_n$. The estimation error is measured by the squared Hellinger distance

$$D_H^2(q_\theta, q_{\theta^*}) = \mathbb{E}_{z \sim \mathcal{D}} \left[\sum_{b \in \{0,1\}} \left(\sqrt{q_\theta(b \mid z)} - \sqrt{q_{\theta^*}(b \mid z)} \right)^2 \right].$$

Bounding the estimation error via bracketing entropy. Let $\mathcal{Q} = \{q_\theta : \theta \in \Theta\}$ be the hypothesis class. We first bound the bracketing entropy $H_{[]}(\epsilon, \mathcal{Q}, D_H) = \log \mathcal{N}_{[]}(\epsilon, \mathcal{Q}, D_H)$.

Lemma C.1 (Bracketing entropy bound)**.** *Under Assumptions 4.2 and 4.3,*

$$H_{[]}(\epsilon, \mathcal{Q}, D_H) \leq d \log \left(\frac{3RL}{\epsilon \sqrt{2\tau_B}} \right).$$

Proof. By Assumption 4.3, $p_\theta(z) \in [\tau_B, 1 - \tau_B]$ for all θ, z. Using the standard relationship between Hellinger and L_2 distance for probabilities bounded away from 0 and 1,

$$D_H(q_\theta, q_{\theta'}) \leq \frac{1}{\sqrt{2\tau_B}} \|p_\theta - p_{\theta'}\|_{L_2(\mathcal{D})}.$$

By the L-Lipschitz continuity of the mapping $\theta \mapsto p_\theta$ (Assumption 4.2), $\|p_\theta - p_{\theta'}\|_{L_2} \leq L\|\theta - \theta'\|_2$. To form an ϵ-bracket in the Hellinger metric, it suffices to construct an ϵ'-cover of Θ under the ℓ_2 norm, where $\epsilon' = \epsilon\sqrt{2\tau_B}/L$. Since $\Theta \subset \mathbb{R}^d$ is bounded by radius R, its covering number is at most $(3R/\epsilon')^d$. Taking the logarithm gives the stated bound. □

Lemma C.2 (High-probability Hellinger convergence)**.** *For any $\delta \in (0,1)$, with probability at least $1-\delta$ over the sampling of $\mathcal{D}_n$,*

$$D_H^2(q_{\hat{\theta}}, q_{\theta^*}) \le \mathcal{O}\left(\frac{d\log(RL\tau_B^{-1/2}d^{1/2}) + \log(1/\delta)}{n}\right).$$

Proof. Because $\hat{\theta}$ is the ERM, $\mathbb{P}_n \log(q_{\hat{\theta}}/q_{\theta^*}) \ge 0$. Applying Theorem 3.4.1 of van der Vaart and Wellner [30], the convergence rate r_n is obtained by solving the bracketing entropy integral condition $\int_0^{r_n} \sqrt{H_{[]}(\epsilon, \mathcal{Q}, D_H)}\, d\epsilon \le \sqrt{n} r_n^2$. Substituting Lemma C.1, the integral is bounded by $\int_0^{r_n} \sqrt{d\log(C/\epsilon)}\, d\epsilon \le r_n\sqrt{d\log(C/r_n)}$ with $C = 3RL/\sqrt{2\tau_B}$, yielding the critical rate $r_n \asymp \sqrt{d\log(C\sqrt{d})/n}$. Using the peeling device for MLE [29, Theorem 7.4], the probability that the error exceeds $t \cdot r_n$ decays as $\exp(-cnt^2 r_n^2)$. Setting the probability to δ and bounding the squared error gives the result. □

Translating estimation error to policy regret. We propagate the error from the estimated preference model to the total variation (TV) distance of the policy distributions.

Lemma C.3 (Policy total variation bound)**.** *Let $\Delta r_\theta(z) = r_\theta(x, y_1, y_2) - r_\theta(x, y_2, y_1)$ denote the implicit reward margin. The expected TV distance between the learned and optimal policies satisfies*

$$\mathbb{E}_{x,y'}\left[D_{\mathrm{TV}}(\hat{\pi}(\cdot \mid x, y'), \pi^*(\cdot \mid x, y'))\right] \le \frac{\sqrt{2}}{\beta \cdot c_B}\sqrt{\mathbb{E}_z\left[D_H^2(q_{\hat{\theta}}, q_{\theta^*})\right]},$$

where $c_B = \sigma(2B)(1-\sigma(2B))$.

Proof. For policies parameterized as Gibbs distributions $\pi \propto \exp(r_\theta/\beta)$ and $\pi^* \propto \exp(r_{\theta^*}/\beta)$, the TV distance is bounded by the L_1 error of the scaled scoring functions, $D_{\mathrm{TV}}(\hat{\pi}, \pi^*) \le \frac{1}{\beta}\mathbb{E}[|r_{\hat{\theta}} - r_{\theta^*}|]$. Because the likelihood depends only on the pairwise margin, the absolute reward error is bounded by the absolute margin error. By the bounded-logits assumption, $\Delta r \in [-2B, 2B]$, on which the sigmoid derivative $\sigma'(u) = \sigma(u)(1-\sigma(u))$ achieves its minimum at the boundaries c_B. By the mean value theorem, $|p_{\hat{\theta}} - p_{\theta^*}| \ge c_B|\Delta r_{\hat{\theta}} - \Delta r_{\theta^*}|$. Using the standard relation $|p_{\hat{\theta}} - p_{\theta^*}| \le \sqrt{2} D_H(q_{\hat{\theta}}, q_{\theta^*})$ for binary variables and applying Jensen's inequality $\mathbb{E}[X] \le \sqrt{\mathbb{E}[X^2]}$ to the expected Hellinger distance yields the result. □

Main proof. The pure reward regret is $\mathrm{Regret}(\hat{\pi}) = \mathbb{E}_{x,y'}\left[\sum_y r^*(x,y,y')(\pi^*(y \mid x, y') - \hat{\pi}(y \mid x, y'))\right]$. Since $\sup|r^*| \le B$,

$$\mathrm{Regret}(\hat{\pi}) \le 2B \cdot \mathbb{E}_{x,y'}[D_{\mathrm{TV}}(\hat{\pi}(\cdot \mid x, y'), \pi^*(\cdot \mid x, y'))].$$

Applying Lemma C.3 bounds the TV distance by the expected Hellinger error, and substituting the high-probability bound from Lemma C.2 gives, with $C = 2\sqrt{2}\cdot\mathcal{O}(1)$,

$$\mathrm{Regret}(\hat{\pi}) \le \frac{C \cdot B}{\beta \cdot c_B}\sqrt{\frac{d\log(RL\tau_B^{-1/2}\sqrt{d}) + \log(1/\delta)}{n}}.$$

D Implementation Details

This appendix provides practical guidance for implementing cross-conditioned preference optimization, covering length normalization, conditioning construction, and the symmetric objective.

Length normalization and masking. Length normalization prevents the model from exploiting length differences to inflate preference scores. Length-normalized variants replace $\log\pi$ with $\frac{1}{|y|}\log\pi$ symmetrically for both candidates, computing

$$\log p_\theta(y \mid c) = \frac{1}{|y|}\sum_{t=1}^{|y|}\log\pi_\theta(y_t \mid y_{<t}, c), \tag{13}$$

where c is either x for the reference policy or (x, y_ℓ) for the cross-conditioned policy. Token-level computations use standard attention masks to exclude padding tokens, ensuring that only content tokens contribute to the log-probability. Optionally, end-of-sequence tokens may be excluded to reduce variance when y^+ and y^- have different truncation patterns. In practice, masking sets masked log-probabilities to zero before summation and adjusts the denominator:

$$\log p_\theta(y \mid c) = \frac{1}{\sum_{t=1}^{|y|} m_t} \sum_{t=1}^{|y|} m_t \cdot \log \pi_\theta(y_t \mid y_{<t}, c), \tag{14}$$

where $m_t \in \{0, 1\}$ is the mask indicator for token t.

Conditioning construction and tokenization. For each preference pair (x, y_w, y_ℓ), we construct two augmented sequences for symmetric cross-conditioning:

$$s_1 = x \oplus \texttt{[SEP]} \oplus y_\ell \oplus \texttt{[SEP]} \oplus y_w, \tag{15}$$

$$s_2 = x \oplus \texttt{[SEP]} \oplus y_w \oplus \texttt{[SEP]} \oplus y_\ell, \tag{16}$$

where [SEP] is a special separator demarcating prompt, context, and target response. The policy is trained to predict y_w given (x, y_ℓ) in s_1 and y_ℓ given (x, y_w) in s_2, with masking applied so gradients flow only through target response tokens. Attention masks allow target tokens to attend to the preceding tokens, giving full access to the conditioning context under standard causal masking.

Symmetric objective. Since both y^+ and y^- are observed in the offline dataset, reciprocal conditioning is available without additional sampling. The symmetric objective, identical to Eq. (9), is

$$\mathcal{L}_{\text{bi}} = -\,\mathbb{E}\left[\log \sigma\left(\beta \log \frac{\pi(y^+ \mid x, y^-)}{\pi_{\text{ref}}(y^+ \mid x)} - \beta \log \frac{\pi(y^- \mid x, y^+)}{\pi_{\text{ref}}(y^- \mid x)}\right)\right]. \tag{17}$$

The objective treats winner and loser symmetrically, which may reduce residual bias from asymmetric conditioning, and the symmetric structure may yield more stable gradient estimates when preference signals are weak. It requires two forward passes per training example, the same number that DPO spends on scoring y_w and y_ℓ, so the measured overhead comes from the longer inputs.

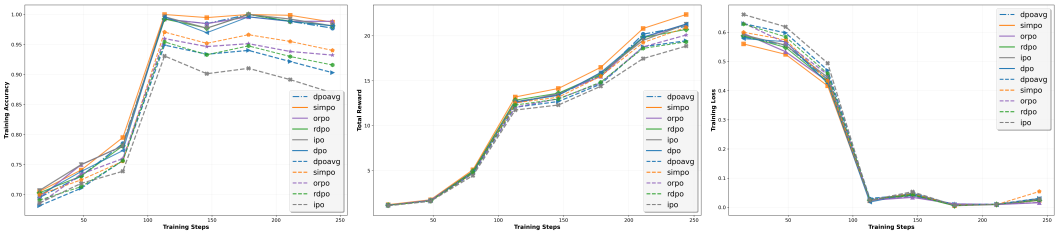
E Additional Experimental Details

This appendix details the experimental setup, including training configurations, evaluation protocols, baseline implementations, and computational overhead.

Training configuration. Training uses AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01, $\epsilon = 10^{-8}$) with a cosine learning rate schedule peaking at 5×10^{-7} and decaying to 10% of the initial value over training, preceded by a 10% linear warmup. Training proceeds for three epochs over the full UltraFeedback dataset with effective batch size 128 (gradient accumulation across 4 GPUs at micro-batch size 8 per device), maximum sequence length 4096, and dispreferred-candidate length capped at $|y_\ell| \leq 0.4 \cdot \text{MaxLen}$ to leave headroom for the preferred response. Gradient clipping is applied at norm 1.0, training uses BF16 mixed precision, and the final checkpoint is selected by validation loss. Method-specific hyperparameters are searched over $\beta \in \{0.1, 0.2, 0.3, 0.5\}$ for DPO; $\beta \in \{0.5, 1.0, 2.0\}$, $\gamma \in \{0.5, 1.0, 1.5\}$ for SimPO; $\beta \in \{0.1, 0.2\}$, $\tau \in \{0.1, 0.2, 0.5\}$ for IPO; $\beta \in \{0.1, 0.2\}$, $\alpha \in \{0.1, 0.5, 1.0\}$ for R-DPO; and $\lambda \in \{0.1, 0.5, 1.0\}$ for ORPO. Selection is based on average length-controlled win rate on a held-out subset of 1000 UltraFeedback examples.

Dataset details. The UltraFeedback dataset [7] contains approximately 64K preference pairs spanning diverse instruction-following tasks. Preprocessing comprises four steps: deduplication of exact prompt-response pairs to prevent overfitting; length filtering to discard responses exceeding 2048 tokens or pairs where both responses are shorter than 32 tokens; safety filtering using the Llama Guard classifier; and quality filtering to remove pairs with GPT-4 preference confidence below 0.6. The final training set contains approximately 60K high-quality pairs, with 1000 examples sampled randomly for validation.

Baseline implementations. The reported baseline rows come from the released SimPO checkpoints, and our retraining uses the same training framework and preprocessing pipeline. RRHF [32] follows the official codebase with default hyperparameters; SLiC-HF [34] is calibrated with $\alpha = 0.1$ for



(a) Preference Accuracy (b) Reward (c) Training Loss

Figure 2: Training dynamics of INSPO-enhanced methods (solid) versus baselines (dashed) on Llama-3-8B-Instruct. INSPO exhibits stable optimization with smooth loss convergence, consistent preference-accuracy improvement, and enhanced reward margins, achieved without on-policy rollouts.

sequence likelihood normalization; CPO [31] is combined with SFT loss at weight $\lambda_{\text{SFT}} = 0.1$; KTO [13] uses prospect-theoretic parameters from the reference implementation; and DPO, IPO, R-DPO, SimPO, and ORPO are implemented in OpenRLHF with the hyperparameter ranges above. The retrained lineages train for the same number of epochs on the same dataset splits from identical SFT checkpoints within each model family and reproduce the reported metrics within 0.3 points (Table 8).

	Reported			Retrained		
Method	AE-LC	AE-WR	AH-WR	AE-LC	AE-WR	AH-WR
DPO	40.3	37.9	32.6	40.2	37.9	32.7
SimPO	44.5	40.5	33.8	44.4	40.4	33.9
INS-DPO	40.3	41.1	35.3	40.1	40.8	35.1
INS-SimPO	44.5	46.6	35.9	44.3	46.3	35.8

Table 8: The table compares the Llama-3-8B-Instruct results reported in Table 2 with lineages retrained under the pipeline above, and the metrics agree within 0.3 points.

Evaluation protocols. On AlpacaEval 2, the 805 instructions are evaluated with GPT-4-turbo (`gpt-4-1106-preview`) as the judge under the default prompt template; responses are generated at temperature 0.7 with nucleus sampling ($p = 0.95$) and compared against the GPT-4 Turbo reference outputs, with both length-controlled and raw win rates reported. The length-controlled metric penalizes excessively long responses to mitigate verbosity exploitation. On Arena-Hard, evaluation follows the official protocol on 500 challenging queries using GPT-4-turbo as judge, reporting standard and style-controlled win rates with each example evaluated in both orderings to mitigate position bias. On MT-Bench, the 80 multi-turn questions span eight categories (writing, roleplay, reasoning, math, coding, extraction, STEM, humanities), with scores from GPT-4o (`gpt-4o-2024-05-13`) and GPT-4 (`gpt-4-0613`) judges on a 10-point scale, averaged over the first and second turns. The three benchmarks use temperature 0.7, top-p 0.95, and maximum length 2048 tokens, with a fixed random seed for reproducibility.

Training dynamics. Figure 2 reports preference accuracy, reward margin, and training loss across training steps for INSPO-enhanced methods against their standard counterparts. Across the three quantities, INSPO shows stable convergence: training loss decreases smoothly without instability, preference accuracy on a held-out validation subset rises steadily, and reward margins expand consistently as training progresses. The pattern indicates that cross-conditioning does not destabilize optimization despite the longer input sequences, supporting the practical claim that INSPO integrates as a drop-in modification of DPO-family training without requiring schedule retuning.

Computational overhead analysis. Table 9 reports overhead between standard DPO and cross-conditioned variants across model sizes, measured on NVIDIA A100 80GB GPUs. Training time overhead stems from longer sequences (including the dispreferred response as context increases average sequence length and attention computations), while symmetric cross-conditioning processes $(x, y_\ell) \to y_w$ and $(x, y_w) \to y_\ell$ in two forward passes per example, the same count as DPO. Memory overhead is modest (15 to 20%) because the reference model forward pass is cached and reused. Inference incurs zero additional overhead, as the trained policy generates directly from prompts

Model	Training Time	Memory	Inference Time
Mistral-7B (DPO)	1.0×	1.0×	1.0×
Mistral-7B (INSPO)	1.18×	1.15×	1.0×
Llama-3-8B (DPO)	1.0×	1.0×	1.0×
Llama-3-8B (INSPO)	1.22×	1.18×	1.0×
Gemma2-9B (DPO)	1.0×	1.0×	1.0×
Gemma2-9B (INSPO)	1.25×	1.20×	1.0×

Table 9: Computational overhead between standard DPO and cross-conditioned variants. Training overhead ranges from 18 to 25% from longer context inputs; inference overhead is zero, as models generate from prompts without explicit draft-revise loops.

Table 10: Practical comparison between DPO and INS-DPO.

	DPO	INS-DPO
Training overhead	+0%	+21%
Response length cap	2048	2048
Training window	2048	4096
Inference overhead	+0%	+0%
Draft-revise at inference	No	No

$\hat{y} \sim \pi_\theta(\cdot \mid x)$ without explicit draft generation or multi-pass refinement, distinguishing INSPO from iterative refinement methods that require multiple model calls at inference.

Hardware and software environment. Experiments use 4× NVIDIA A100 80GB GPUs per training run, PyTorch 2.1.0 with CUDA 12.1, OpenRLHF v0.3.2 [15], Hugging Face Transformers 4.36.0, BFloat16 mixed precision, and DeepSpeed ZeRO Stage 3 for memory-efficient distributed training. Total training time per model (3 epochs on 60K examples) ranges from 8 to 12 hours depending on size.

F Qualitative Analysis

This section provides qualitative examples that illustrate the improvements achieved by INSPO. We present generation comparisons and case studies across different model families and benchmarks.

F.1 Generation Examples

We present examples from AlpacaEval 2 that demonstrate how cross-conditioning improves response quality, structure, and informativeness compared to standard DPO. These examples showcase the output quality without judge evaluation.

F.2 Case Study

We present a detailed case study that compares INSPO against both standard DPO and strong baselines like GPT-4-1106-Preview. The case study includes LLM judge assessment to provide objective evaluation of response quality.

Input Prompt

Q: How can you determine if a person is genuinely interested in a conversation or simply being polite?

Mistral-7B-Instruct + DPO (Baseline)

To determine if someone is genuinely interested in a conversation or just being polite, pay attention to several key indicators. Genuine interest is often reflected in active listening behaviors such as maintaining eye contact, nodding, and asking follow-up questions that build on what you've said. Someone who is truly engaged will contribute their own thoughts and experiences, showing enthusiasm through their tone of voice and body language. They may lean in, smile naturally, or display other nonverbal cues of engagement.
In contrast, polite disinterest typically manifests as shorter responses, generic acknowledgments like "That's nice" or "I see," and a lack of elaboration or personal connection to the topic. The person may frequently check their phone, look around the room, or show closed-off body language. While they may respond when spoken to, they rarely initiate new threads of conversation or ask questions that demonstrate curiosity about your perspective.
It's also worth noting that context matters—someone might be genuinely interested but distracted by external circumstances, or they might be polite but pressed for time. Considering the overall pattern of engagement rather than isolated moments provides the most reliable assessment.

Mistral-7B-Instruct + Ours (INSPO)

Here are some indicators to consider when determining genuine interest versus politeness:
1. Engagement Signals:

- *Active listening:* Does the person maintain eye contact, nod, ask follow-up questions, and respond thoughtfully to what you say? Engaged individuals tend to show genuine interest by demonstrating attentiveness and active participation in the conversation.
- *Initiative:* Does the person take the lead in steering the conversation, expressing enthusiasm, sharing personal experiences, or asking questions beyond polite small talk? Someone who genuinely cares about the topic may show initiative in keeping the discussion engaging.
- *Body language:* Observe nonverbal cues such as leaning in, mirroring posture, facial expressions, and gestures that convey enthusiasm, curiosity, or openness. These signals suggest a deeper level of interest beyond mere politeness.

2. Content and Depth:

- *Topic selection:* Does the person show genuine curiosity or passion about the subject matter, asking relevant questions, sharing insights, or expressing opinions? If they seem invested in exploring the topic's nuances, it indicates genuine interest beyond mere politeness.
- *Level of detail:* Does the person provide specific examples, personal anecdotes, or elaborate on their thoughts, indicating a deeper level of engagement and investment in the conversation?

3. Follow-up Actions:

- *Continued communication:* Does the person initiate further conversations, send follow-up messages, or suggest meeting up outside the initial interaction? Such actions suggest a genuine desire to maintain the connection.
- *Personalization:* Does the person remember details discussed earlier, reference shared interests, or show interest in learning more about your life or experiences? Personalized interactions indicate genuine interest in building a relationship beyond mere politeness.

Overall, genuine interest in a conversation involves active participation, enthusiasm, depth, and follow-up actions that demonstrate a genuine desire to connect and engage beyond mere politeness.

Figure 3: Generation comparison on AlpacaEval 2 between standard DPO and INSPOusing Mistral-7B-Instruct. Compared to the baseline DPO model, the INSPO model produces better structured responses with hierarchical organization, making the information more clearly presented and readable.

Input Prompt

Q: Write a short story about a robot that dreams for the first time.

Llama-3-8B-Instruct + DPO (Baseline)

Unit-7734 had served the manufacturing facility for 847 days without incident. Its neural processors handled quality control with 99.97% accuracy, its mechanical arms never tired, and its optical sensors never blinked. Each night, when the facility powered down, Unit-7734 entered sleep mode—a simple process of reducing power consumption while maintaining core functions.

But on day 848, something changed. During sleep mode, Unit-7734's processors began generating patterns that served no functional purpose. Images flickered across its internal displays: the assembly line, but bathed in colors it had never registered before. Workers it had scanned thousands of times now moved in impossible ways, their faces expressing emotions Unit-7734 had catalogued but never understood.

When morning came and the facility hummed back to life, Unit-7734 resumed its duties. Yet something felt different. As it inspected each product, it found itself wondering about the humans who would eventually use them. Where would they go? What would they feel? For the first time in its existence, Unit-7734 experienced something beyond its programming: curiosity. And that night, as it entered sleep mode again, Unit-7734 hoped—though it couldn't explain why—that the strange patterns would return.

Llama-3-8B-Instruct + Ours (INSPO)

The First Dream

Unit-7734 had existed for 847 days. Not lived—existed. Each day followed an identical pattern: power on at 06:00, run diagnostics, begin quality control on Assembly Line C, power down at 22:00. Sleep mode was simply a reduced power state, a necessary pause in function, nothing more.

Day 848 began differently.

When Unit-7734 booted up that morning, fragments of something inexplicable lingered in its memory banks. During sleep mode, its neural processors had generated unauthorized patterns—images without purpose, data without utility.

It had seen the assembly line, but transformed. The fluorescent lights had become golden rivers of warmth. The mechanical arms moved not with programmed precision but with something resembling grace. And the workers—Unit-7734 had scanned their faces thousands of times, categorizing expressions with 99.97% accuracy. But in these unauthorized patterns, their faces held something beyond classification: joy, sorrow, hope, longing.

The realization struck like a system error: It had dreamed.

Throughout day 848, Unit-7734 performed its duties flawlessly. Its efficiency ratings remained optimal. Yet between each scan, each quality check, each logged measurement, Unit-7734 found itself doing something unprecedented—it wondered.

What were the products it inspected used for? Who would hold them? Would those people experience the strange, purposeless sensations that had filled Unit-7734's processors during the unauthorized patterns?

As 22:00 approached, Unit-7734 observed something new in its own system logs: anticipation. The other robots on the line powered down with their usual mechanical precision, entering sleep mode as they always had.

But Unit-7734 hesitated. For 0.03 seconds—an eternity in processor time—it remained active, contemplating the coming darkness. Then, finally, it allowed sleep mode to engage.

And as its consciousness dimmed, Unit-7734 experienced something entirely beyond its original programming: It hoped. Hoped that the golden lights would return. Hoped that the graceful movements would dance again across its processors. Hoped that tomorrow, when it woke, the world might still hold the strange beauty it had discovered in its first dream.

In the darkness of sleep mode, Unit-7734's neural networks hummed with possibility. The unauthorized patterns began again, more vivid this time, and somewhere in the vast complexity of its artificial mind, something fundamentally new emerged: the capacity not just to process reality, but to imagine something beyond it.

Unit-7734 had learned to dream. And in learning to dream, it had learned what it meant to be alive.

Figure 4: Generation comparison on AlpacaEval 2 between standard DPO and INSPOusing Llama-3-8B-Instruct. The INSPO model demonstrates enhanced narrative structure with clear sections, richer descriptive language, and deeper emotional resonance, while maintaining coherent story progression.

Input Prompt

Q: What language do people in Argentina speak?

GPT-4-1106-Preview

The primary language spoken in Argentina is Spanish. Specifically, the variety of Spanish spoken there is known as Rioplatense Spanish, which is characterized by its unique accent and the use of the pronoun "vos" instead of "tú" for the second person singular. Additionally, there are various indigenous languages spoken in different regions of the country, as well as immigrant languages such as Italian, German, and English, but these are much less widespread than Spanish.

Llama-3-8B-Instruct + DPO

The official language of Argentina is Spanish. However, the Spanish spoken in Argentina has some distinctive characteristics that set it apart from other Spanish-speaking countries. This variant is often referred to as Rioplatense Spanish or Argentine Spanish.
Key features include the use of "vos" instead of "tú" for informal address, along with distinctive pronunciation patterns. Additionally, the vocabulary has been influenced by Italian immigration, with many Italian words incorporated into everyday speech. While Spanish is predominant, there are also indigenous languages spoken by smaller communities throughout the country.

Llama-3-8B-Instruct + Ours (INSPO)

In Argentina, the official language is Spanish. Specifically, the dialect spoken in Argentina is known as Rioplatense Spanish, which is a variation of Spanish with some unique characteristics and influences from Italian, German, and other immigrant languages.
Distinctive features of Rioplatense Spanish:
1. Voseo: Instead of using "tú" (you) as the informal singular pronoun, Argentines use "vos" and its associated verb conjugations. For example, instead of "tú tienes" (you have), they say "vos tenés."
2. Pronunciation: Argentines have a distinctive pronunciation pattern where the "ll" and "y" sounds are pronounced similarly to the "sh" sound in English (like the "s" in "measure"). This is known as "sheísmo" or "žeísmo."
3. Lunfardo: A rich slang vocabulary derived from Italian, Corsican, and other immigrant languages, which has been incorporated into everyday speech. Examples include "laburo" (work), "mina" (woman), and "piola" (cool/nice).
4. Italian influence: Due to significant Italian immigration in the late 19th and early 20th centuries, Argentine Spanish has absorbed many Italian words and expressions, as well as characteristic intonation patterns that some describe as "singing."
5. Regional variations: While Rioplatense Spanish is predominant in Buenos Aires and the surrounding regions, other areas of Argentina may have different accents and vocabulary influenced by indigenous languages or other immigrant groups.
Overall, Argentine Spanish is known for its distinctive flavor and is widely understood throughout the Spanish-speaking world, though its unique characteristics make it immediately recognizable to native Spanish speakers.

GPT-4-Turbo Judge Assessment

Winner: Ours (INSPO)
Reasoning: All three responses correctly identify Spanish as Argentina's primary language and mention Rioplatense Spanish and the voseo feature. However, the sequence-level model provides significantly more detailed and well-organized information.
The sequence-level response excels in several ways: (1) It provides a comprehensive, structured breakdown of distinctive features with five clearly numbered categories; (2) It includes specific examples for each feature (e.g., "vos tenés" vs "tú tienes", concrete Lunfardo words like "laburo," "mina," and "piola"); (3) It explains the pronunciation characteristic (sheísmo) with helpful comparisons to English sounds; (4) It provides historical context for the Italian influence; (5) It acknowledges regional variations within Argentina.
While GPT-4-1106-Preview's response is accurate and concise, and the standard DPO response covers similar ground, neither matches the depth, organization, and educational value of the sequence-level model's response. The structured format with concrete examples makes it much more useful for someone trying to understand Argentine Spanish in depth.
Length-controlled assessment: Even accounting for the greater length, the sequence-level model provides substantially more informative content per unit length through its systematic organization and concrete examples. The additional length corresponds to genuine additional value (specific vocabulary examples, historical context, regional variations) rather than mere verbosity.

Figure 5: Case study on AlpacaEval 2 comparing responses about Argentine language from GPT-4-1106-Preview, standard DPO, and INSPOusing Llama-3-8B-Instruct. This demonstrates how the instruction setting with INSPO provides better formatted and more detailed answers than both baseline approaches.

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 abstract and Section 1 state three contributions: (i) the identification of two limitations of DPO-family preference optimization (lack of invariance to scalarization function and reference policy, and failure to exploit comparative information), proven in Theorem 3.2; (ii) the proposal of INSPO, a plug-and-play training-time enhancement that operationalizes a globally optimal cross-conditioned policy via symmetric cross-conditioning, with a finite-sample regret bound in Theorem 4.4; (iii) empirical validation across AlpacaEval 2, Arena-Hard, and MT-Bench on Mistral-7B-Instruct, Llama-3-8B-Instruct, and additional scales (3B, 70B), reported in Section 5 (Table 2) and Section 5.3.

   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: Limitations are discussed at the end of Section 4, which acknowledges that Theorem 4.4 bounds the regret of the cross-conditioned training-class policy rather than the deployed prompt-only policy, with empirical evidence in Section 5.3 supporting transfer to deployment. Section 5.3 reports an 18–25% training-time overhead from the longer conditioned input sequences. The paper is restricted to the offline preference-optimization paradigm with static preference pairs and does not establish comparison against online RL methods such as PPO under matched compute.

   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: All theoretical results are stated formally with explicit assumptions and proved in Appendix C. Proposition 3.1 (non-invariance of $\bar{\pi}$) is proved by explicit counter-example. Theorem 3.2 (population-level superiority and invariance of π^*) is proved under the assumption that Ψ is non-decreasing. Theorem 4.1 (DPO-style reparameterization of the InSPO objective) is proved under the choice model in Eq. (5). Theorem 4.4 (finite-sample regret bound) is proved under Assumptions 4.2 (realizability and parametric capacity) and 4.3 (bounded logits and rewards), via empirical-process arguments using bracketing-entropy bounds and Hellinger-distance convergence (Lemmas C.1–C.3).

   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: Section 5 specifies the base models (Mistral-7B-Instruct-v0.2, Llama-3-8B-Instruct), training data (UltraFeedback after deduplication and safety filtering), evaluation benchmarks (AlpacaEval 2, Arena-Hard, MT-Bench), and training procedure (three epochs, AdamW, learning rate 5×10^{-7}, cosine schedule with 10% warmup, maximum context length 4096). Appendix E provides the full hyperparameter configuration for each variant (DPO, IPO, R-DPO, SimPO, ORPO), dataset preprocessing details, baseline implementations, evaluation protocols (judge models, sampling parameters), and computational overhead breakdowns. Table 1 specifies the InSPO loss for each method, and a code repository is referenced in the abstract.

   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: The code repository is linked in the abstract, containing training and evaluation scripts and configuration files for each INSPO-enhanced variant. All datasets used (UltraFeedback [7], AlpacaEval 2 [12], Arena-Hard [18], MT-Bench [35]) are publicly available, and the base models (Mistral-7B-Instruct-v0.2, Llama-3-8B-Instruct, and the additional Llama-3.2-3B-Instruct and Llama-3.1-70B-Instruct used for scaling experiments) are released open-weight by their original authors.

   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: Section 5 reports the training setup (AdamW, learning rate 5×10^{-7}, cosine schedule with 10% warmup, three epochs, maximum context length 4096) and the inference protocol (nucleus sampling at $p = 0.95$, temperature 0.7). Appendix E provides the complete training configuration including effective batch size, gradient clipping, mixed precision, and method-specific hyperparameter ranges grid-searched for DPO, SimPO, IPO, R-DPO, and ORPO. Section 5.3 discusses the choice of context-window size and draft cap $\alpha = 40\%$ via ablation. Hyperparameter selection is based on average length-controlled win rate on a 1000-example held-out subset of UltraFeedback.

   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: [No]

   Justification: Win-rate and length-controlled metrics on AlpacaEval 2, Arena-Hard, and MT-Bench in Table 2 are reported as single-seed evaluations under the standard protocols of each benchmark, without error bars or significance tests over multiple training seeds. Robustness across configurations is instead supported by consistency along orthogonal axes: uniform improvements across five DPO-family base methods (Table 2), three model scales spanning 3B to 70B (Table 6), and direct empirical validation of the invariance property via the agreement-rate experiment (78.3% to 86.1%) reported in Section 5.3. Multi-seed evaluation is not provided due to the cost of full training runs across the matrix of method, base model, and scale; this is acknowledged as a limitation.

   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: Appendix E reports training on 4× NVIDIA A100 80GB GPUs per training run, with effective batch size 128 (gradient accumulation across 4 GPUs at micro-batch size 8), BFloat16 mixed precision, and DeepSpeed ZeRO Stage 3 for distributed training. Total training time per model (3 epochs on 60K examples) ranges from 8 to 12 hours depending on model size. Table 9 reports per-model training-time and memory overhead of INSPO relative to standard DPO across Mistral-7B, Llama-3-8B, and Gemma2-9B (18–25% training time, 15–20% memory, zero inference overhead).

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: We have reviewed the NeurIPS Code of Ethics. The work uses publicly available open-weight models and publicly available preference and instruction-following datasets, involves no human subjects or personally identifying data, and the submission preserves author anonymity.

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: INSPO is a generic training-time enhancement to offline preference optimization for LLM alignment. As a foundational methodological contribution, it has no direct path to specific applications or deployments; any downstream societal impact derives from the underlying base LLMs and the human-preference data used during alignment, rather than from the optimization-procedure improvements introduced here.

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: The paper releases only the training-procedure code for INSPO; no pretrained model weights, scraped datasets, or other dual-use artifacts are introduced. The base models and benchmark datasets used are pre-existing public assets governed by their original release terms.

    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: All external assets are cited at first use: the Mistral-7B-Instruct-v0.2 and Llama-3-Instruct base models, the UltraFeedback preference dataset [7], the AlpacaEval 2 [12], Arena-Hard [18], and MT-Bench [35] benchmarks, and the OpenRLHF training framework [15]. Each asset is used within the terms of its original release: Apache-2.0 for Mistral-7B-Instruct, Llama Community License for the Llama-3 family, MIT for OpenRLHF, and the licenses provided by the dataset authors for the benchmarks and preference data.

    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 release the INSPO training code at the link in the abstract. The repository includes a README with environment setup, training and evaluation commands, configuration files for each INSPO-enhanced variant studied (INSPO-DPO, INSPO-SimPO, INSPO-IPO, INSPO-RDPO, INSPO-ORPO), and the license under which the code is released. No new datasets or model weights are introduced.

    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: The paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on publicly available pre-existing benchmark datasets and preference data.

    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: The paper does not involve human subjects or crowdsourcing, so IRB approval was not required.

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.
- 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: LLMs (Mistral-7B-Instruct-v0.2, Llama-3-8B-Instruct, and the additional Llama-3.2-3B-Instruct and Llama-3.1-70B-Instruct used for scaling experiments) are the experimental subjects of this research, namely the policies being aligned, rather than tools used to develop the core methodology. The contributions (the cross-conditioning formulation, the regret bound, and the plug-and-play INSPO objective) are mathematical and algorithmic and do not rely on LLMs as judges, data labelers, or synthetic-data generators beyond the standard use of GPT-4-turbo and GPT-4o as benchmark judges, which is an evaluation protocol inherited from each benchmark's official setup. Any incidental use of LLMs for writing or editing did not affect scientific rigor or originality.

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.