When Scores Conflict with Preferences: Pre-Alignment Diagnosis and Adaptive Correction of Constraint Drift in DPO

Ping Liu
Department of Computer Science and Engineering
University of Nevada, Reno
pino.pingliu@gmail.com

Yan Yan*
School of Electrical Engineering and Computer Science
Washington State University
yan.yan1@wsu.edu

Abstract

Direct Preference Optimization (DPO) is increasingly used in settings where pairwise preferences coexist with pointwise criteria such as response length, answer correctness, or safety scores. When the two disagree, standard DPO treats pairs whose preferred output scores worse on the pointwise criterion (conflict pairs) in the same way as the remaining (matching) pairs. We characterize a systematic failure mode, constraint drift: for sufficiently small updates, conflict pairs tend to increase constraint cost while matching pairs tend to repair it, and the net change depends on their balance. This balance yields a closed-form pre-alignment diagnostic, ρ^*_{eq}, computable from dataset statistics, that indicates whether a dataset is self-correcting or requires intervention. Building on this diagnostic, we propose Conflict-aware DPO (CoDPO), which down-weights conflict pairs with a strength set from ρ^*_{eq} rather than tuned, and adds an adaptive margin shift that addresses residual drift during training. Empirically, drift grows monotonically with conflict exposure on the conflict-dominated domains, and the intervention decision given by ρ^*_{eq} matches the observed drift on every domain, including one that it identifies as self-correcting. On all four domains of the main comparison, CoDPO reduces drift while largely preserving the preference learning of vanilla DPO. Compared with hard filtering and recent constrained and multi-objective baselines, CoDPO also achieves a more favorable drift-preference trade-off.

1 Introduction

Generative models are increasingly aligned with human objectives through preference optimization. Preference optimization methods such as RLHF [1, 2, 3] and DPO [4] have substantially improved helpfulness, reasoning quality, and instruction following [5, 6, 7, 8] by learning from pairwise comparisons between candidate outputs. A response preferred in such a comparison may nevertheless score poorly on a pointwise criterion, such as correctness or safety, that assesses each response on its own [9, 10]. Following PSDPO [11], we call this coexistence of pairwise preferences and pointwise criteria heterogeneous supervision.

Under heterogeneous supervision, standard preference optimization treats all training pairs uniformly, implicitly assuming that the preferred output also performs well on the pointwise criterion. On the domains we study, the preferred output scores worse on the pointwise criterion in a substantial fraction of pairs: where the criterion is response length, the preferred response is the longer one in 57% of HelpSteer2 pairs [12], and where it is correctness, the preferred response is the incorrect

*Corresponding author.

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

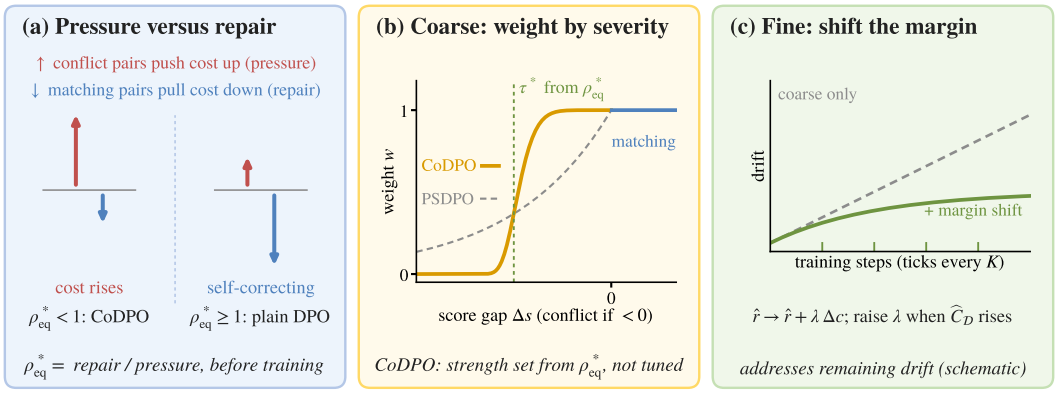

Figure 1: Overview of CoDPO. (a) Under DPO, conflict pairs push the constraint cost up and matching pairs pull it down; ρ^*_{eq}, computed before training, compares the two. If the push dominates ($\rho^*_{\text{eq}} < 1$), CoDPO intervenes; otherwise the data are self-correcting and plain DPO is used. (b) Conflict pairs are down-weighted by severity, with τ^* set from ρ^*_{eq}. (c) During training, the DPO margin is shifted, more strongly when the constraint-risk estimate rises. Curves in (b) and (c) are schematic.

one in 71% of GSM8K pairs [13] and 58% of ARC pairs [14]. Such pairs, conflict pairs in the terminology of PSDPO [11], occur at comparable rates in the other conflict-dominated domains we study (Table 1); the remaining pairs are matching pairs. Conflict pairs tend to push preference optimization toward outputs that win the pairwise comparison but carry a higher cost on the pointwise criterion, a degradation that PSDPO [11] identifies as semantic drift in text-to-video alignment; we call its counterpart under a general pointwise criterion constraint drift.

Existing interventions against constraint drift differ in how they trade the pointwise criterion off against the preference. Constrained RLHF, constrained DPO [9, 15, 16, 17, 18], and multi-objective methods [19, 20] set this trade-off through a weight that is either fixed by the practitioner or, in Lagrangian methods, adjusted during training according to whether the estimated average cost exceeds a target. Hard filtering, akin to preference-data selection [21], removes conflict pairs, which can discard much of the data when conflicts are frequent, while per-pair weighting [22, 11] attenuates them. In each case, the strength of the intervention is chosen by the practitioner, which leaves two questions open for a given dataset: whether it needs intervention at all, ideally known in advance, and if so, how strong the intervention should be, both before and during training.

We address both questions from a theoretical account of constraint drift under DPO. We show that, for sufficiently small updates, conflict pairs and matching pairs push the constraint in opposite directions: conflict pairs raise the expected constraint cost, while matching pairs repair it. Drift therefore depends not on the frequency of conflicts alone but on the balance between conflict-side pressure and matching-side repair. Under a first-order approximation, this balance can be estimated from dataset statistics alone, before any alignment step is taken. The estimate yields a closed-form diagnostic, ρ^*_{eq}, that indicates whether the dataset is self-correcting or needs intervention (Figure 1a).

When intervention is needed, our control method, Conflict-aware DPO (CoDPO), acts at two levels within a single loss (Figure 1b,c). At the coarse level, CoDPO down-weights conflict pairs by severity, rather than removing them all, through asymmetric conflict weighting [11]. Instead of tuning the weighting strength, CoDPO solves for it from dataset statistics through ρ^*_{eq}, so that conflict-side pressure and matching-side repair balance to first order. The diagnostic is computed at initialization, and the balance can shift as the policy moves away from the reference model during training, so some drift can remain. The fine level addresses this residual drift by shifting the DPO margin of each pair in proportion to its constraint-cost difference, and strengthens the shift when the drift measured during training rises. Together, the diagnostic tells the practitioner whether a dataset needs intervention, and CoDPO determines how strong that intervention should be.

We test the diagnostic and CoDPO on five conflict-dominated domains (HelpSteer2, GSM8K, ARC, Math-Calc, TL;DR) and a self-correcting control (PKU-SafeRLHF [10]). On these domains, drift

grows monotonically with conflict exposure except on the control, where it stays negligible, and ρ^*_{eq} assigns the correct intervention regime on all six. On the four domains of the main comparison, all of which call for intervention, CoDPO reduces drift while largely preserving the preference learning achieved by vanilla DPO. Compared with hard filtering and recent constrained and multi-objective baselines, CoDPO also achieves a more favorable drift-preference trade-off.

In summary, our contributions are as follows:

- We show that, for small updates, the change in constraint cost at each DPO step is a balance between conflict-side pressure and matching-side repair, and derive from it a closed-form diagnostic, ρ^*_{eq}, indicating before training whether a dataset needs intervention.
- We propose CoDPO: its coarse level adopts PSDPO's asymmetric weighting form but applies it only when $\rho^*_{\text{eq}} < 1$, with the strength set from ρ^*_{eq} rather than tuned; its fine level shifts the DPO margin and strengthens the shift when the drift measured during training rises.
- Empirically, the intervention decision given by ρ^*_{eq} matches the observed drift on every domain we evaluate, and in our main comparison CoDPO reduces drift while largely preserving vanilla DPO's preference learning, with a more favorable drift-preference trade-off than the compared baselines.

2 Related Work

Preference and constrained alignment. RLHF [2, 1] and DPO [4] learn from pairwise preferences, and subsequent variants modify the preference objective [5, 6, 7, 23, 24, 25] or address conflicts among multiple pairwise dimensions [26, 27, 28, 29], while keeping the supervision purely pairwise. Constrained alignment adds a pointwise cost, through a cost model with Lagrangian updates in Safe RLHF [9] and through recent constrained DPO variants [15, 30, 16, 17, 18]. These methods share our supervision structure, pairwise preferences together with a pointwise criterion, and set the trade-off between the two through a parameter chosen by the practitioner or driven by an aggregate cost estimate. Our analysis addresses a question that precedes this trade-off: whether a given dataset needs constraint handling at all, and if so, how strong it should be, as determined by the conflict structure of its pairs.

Multi-objective optimization and preference-data selection. Multi-objective methods such as PCGrad [31], CAGrad [32], Rewarded Soups [19], and MODPO [20] address objective competition through gradient projection, model interpolation, or reward combination, and data-centric methods retain a high-margin subset of the training pairs [21] or select the informative tokens within a pair [33]; these methods act on objectives or on general data quality rather than on the conflict between a pairwise preference and a pointwise criterion within a pair. Closer to our coarse level, Uni-DPO [22] keeps every pair but reweights it by an external score margin and by the implicit reward margin, so that ambiguous and already-learned pairs contribute less. In text-to-video alignment, PSDPO [11] weights pairs by conflict direction instead, down-weighting conflict pairs with an asymmetric exponential weight that leaves matching pairs unchanged.

Our coarse level adopts this weighting form and differs in how its strength is set. PSDPO bounds the gradient contributed by conflict pairs and shows that the bound shrinks with stronger down-weighting, which motivates down-weighting but leaves its strength to be tuned. Section 3 instead tracks the constraint cost, which conflict pairs raise and matching pairs reduce. Because of this repair, the cost stops rising to first order once the conflict weight falls below a critical value, from which CoDPO sets the strength and decides whether down-weighting is needed. To limit drift that accumulates during training, PSDPO introduces conflict pairs only in a second training stage, whereas our fine level shifts the DPO margin throughout training and strengthens the shift when the measured drift rises.

3 Constraint Drift as a Balance of Pressure and Repair

3.1 Setup and Conflict Structure

Each training example is a preference pair (x, y_w, y_l), in which y_w is the output preferred by the pairwise annotation and y_l the rejected one. Let $\pi_\theta(y \mid x)$ be a generative policy initialized at the reference policy, $\pi_{\theta_0} = \pi_{\text{ref}}$, as is standard for DPO, and let $L_R(\theta) = -\mathbb{E}_{\mathcal{D}}[\log \sigma(\beta\, \hat{r}_\theta)]$ be the

DPO preference loss [4], where $\hat{r}_\theta = \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$ is the implicit reward margin. A separate scorer $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ evaluates each output independently of the pairwise signal, oriented so that larger values indicate better satisfaction of the pointwise criterion; for instance, s decreases with response length on a length-constrained domain [34, 35] and is a reward-model score [2, 1] on a safety-constrained domain. A constraint cost $c(x, y) \geq 0$, strictly decreasing in s, defines the constraint risk $C(\theta) = \mathbb{E}_{x,\, y\sim\pi_\theta}[c(x, y)]$.

Definition 1 (Conflict [11]) *For a preference pair (x, y_w, y_l), let $\Delta s := s(x, y_w) - s(x, y_l)$ and $\Delta c := c(x, y_w) - c(x, y_l)$. A conflict pair satisfies $\Delta s < 0$ (equivalently $\Delta c > 0$), meaning that the preferred output scores worse on the pointwise criterion than the rejected one. We define the conflict set $\mathcal{D}_{\text{conf}} := \{(x, y_w, y_l) : \Delta s < 0\}$ and the matching set $\mathcal{D}_{\text{match}} := \{(x, y_w, y_l) : \Delta s \geq 0\}$, with conflict rate $p_c := \Pr(\Delta s < 0)$.*[2]

3.2 Pressure, Repair, and the Diagnostic

Following PSDPO [11], the conflict-matching partition splits the DPO gradient into contributions from the two subsets,

$$\nabla L_R(\theta) = (1 - p_c)\, g_{\text{match}} + p_c\, g_{\text{conf}},$$

where $g_{\text{match}} = \mathbb{E}_{\mathcal{D}_{\text{match}}}[\nabla \ell_R]$, $g_{\text{conf}} = \mathbb{E}_{\mathcal{D}_{\text{conf}}}[\nabla \ell_R]$, and the per-sample loss is $\ell_R = -\log \sigma(\beta \hat{r}_\theta)$. Whereas PSDPO [11] bounds the norm of the conflict term, we bound its first-order effect on the constraint risk, as in first-order analyses of DPO training dynamics [36, 37]: under regularity conditions on C and the coupling argument of Appendix B.2, the conflict term can raise the constraint risk to first order,

$$-\langle \nabla C, g_{\text{conf}} \rangle \geq \mathbb{E}_{\mathcal{D}_{\text{conf}}}[\beta(1 - \sigma(\beta \hat{r}_\theta))\, \Delta c\, \mathcal{I}_\theta] \geq 0,$$

where, for $\Delta c \neq 0$, the geometry term $\mathcal{I}_\theta := (\Delta c)^{-1} \langle \nabla_\theta C_x(\theta), \nabla_\theta \hat{r}_\theta \rangle > 0$, with $C_x(\theta) = \mathbb{E}_{y\sim\pi_\theta}[c(x, y)]$, measures how much the expected cost on prompt x changes, per unit of the cost gap Δc, when DPO learns the pair; it is named for the inner product between the two gradient directions. An analogous bound with the opposite sign holds for g_{match} and captures the repair provided by matching pairs.

Theorem 1 (Constraint drift) *Under the regularity conditions of Appendix B.1 (proof in Appendix B.3), gradient descent on L_R satisfies*

$$\mathbb{E}[C(\theta_{t+1}) - C(\theta_t)] \geq -\eta(1 - p_c)\mu_0 + \eta\, p_c\, \kappa_0 - \tfrac{1}{2}\eta^2 L_C G_R^2, \tag{1}$$

where η is the step size, L_C is the smoothness constant of C, and G_R bounds $\|\nabla \ell_R\|$. The terms

$$\kappa_0 := \mathbb{E}_{\mathcal{D}_{\text{conf}}}[\beta(1 - \sigma(\beta \hat{r}_\theta))\, \Delta c\, \mathcal{I}_\theta] > 0, \qquad \mu_0 := \mathbb{E}_{\mathcal{D}_{\text{match}}}[\beta(1 - \sigma(\beta \hat{r}_\theta))\, |\Delta c|\, \mathcal{I}_\theta] > 0$$

aggregate the conflict-side drift pressure and the matching-side repair, respectively. For sufficiently small η, whenever $p_c\kappa_0 > (1 - p_c)\mu_0$, the lower bound is strictly positive and the expected constraint risk increases.

Theorem 1 thus expresses each DPO step as a balance between conflict-side pressure and matching-side repair: drift appears when pressure outweighs repair, and we call the reverse case, in which repair at least matches pressure, the self-correcting regime, which Section 3.3 makes precise.

Remark 1 (Pre-alignment diagnostic) *The drift condition of Theorem 1, $p_c\kappa_0 > (1 - p_c)\mu_0$, is equivalent to $(1 - p_c)\mu_0/(p_c\kappa_0) < 1$, and this ratio admits a simple data-dependent approximation at the start of alignment. Both μ_0 and κ_0 average the product of three factors: the learning signal, the cost difference, and the geometry term. At initialization $\pi_\theta = \pi_{\text{ref}}$, so $\hat{r}_\theta = 0$ and the learning signal $\beta(1-\sigma(\beta \hat{r}_\theta))$ takes the same value $\beta/2$ on every pair, which cancels exactly in the ratio. Assumption (vi) (Appendix B.1) posits that, per unit of cost gap, learning a conflict pair changes the expected cost by about as much as learning a matching pair; under this assumption, the geometry term cancels approximately as well, and the ratio reduces to the average cost differences $\overline{\Delta c}_{\text{conf}} := \mathbb{E}_{\mathcal{D}_{\text{conf}}}[\Delta c]$ and $\overline{|\Delta c|}_{\text{match}} := \mathbb{E}_{\mathcal{D}_{\text{match}}}[|\Delta c|]$, giving the closed-form diagnostic*

$$\rho^*_{\text{eq}} = \frac{(1 - p_c)\, \overline{|\Delta c|}_{\text{match}}}{p_c\, \overline{\Delta c}_{\text{conf}}}, \tag{2}$$

[2] Pairs with $\Delta s = 0$ satisfy $\Delta c = 0$. We place them in $\mathcal{D}_{\text{match}}$ so that the partition is exhaustive and, under Assumption (vii) of Appendix B.1, define their composite contribution $|\Delta c|\, \mathcal{I}_\theta$ to be zero; $\mathcal{I}_\theta$ itself need not be evaluated at $\Delta c = 0$.

that is, the total cost reduction across matching pairs divided by the total cost increase across conflict pairs, so ρ^*_{eq} depends not only on how often conflicts occur but also on how large the cost differences are. Under this approximation, $\rho^*_{\text{eq}} \geq 1$ indicates before alignment that the dataset is self-correcting and needs no intervention, while a value below 1 calls for intervention; Appendix B.5 gives the full derivation.

3.3 Down-Weighting Conflict Pairs

Theorem 1 suggests keeping matching pairs at full weight and down-weighting conflict pairs, as PSDPO [11] does in text-to-video alignment; the question we address is how much down-weighting is enough. We assign each pair a weight $w(\Delta s)$, equal to 1 on matching pairs and in $[0, 1]$ on conflict pairs, and summarize how much of the conflict-side signal is retained by the average conflict weight $\rho := \mathbb{E}_{\mathcal{D}_{\text{conf}}}[w(\Delta s)]$, so that $\rho = 1$ recovers plain DPO and $\rho = 0$ removes all conflict pairs. Theorem 2 answers this question with a critical value ρ^* for the average conflict weight, below which the reduced conflict-side pressure is smaller than the matching-side repair.

Theorem 2 (Drift attenuation) *Under the conditions of Theorem 1, the modulated loss $L_{R,w}(\theta) = -\mathbb{E}[w(\Delta s) \log \sigma(\beta \hat{r}_\theta)]$ satisfies the upper bound*

$$\mathbb{E}[C(\theta_{t+1}) - C(\theta_t)] \leq -\eta(1 - p_c)\mu_0 + \eta\, p_c\, \kappa_w + \tfrac{1}{2}\eta^2 L_C G_R^2,$$

where $\kappa_w := \mathbb{E}_{\mathcal{D}_{\text{conf}}}[w(\Delta s)\, \beta(1 - \sigma(\beta \hat{r}_\theta))\, \Delta c\, \mathcal{I}_\theta]$ is the modulated counterpart of κ_0 and satisfies $\kappa_w \leq \rho\, \kappa_0$ by Assumption (v) and Chebyshev's sum inequality (Appendix B.4). Using $\kappa_w \leq \rho\, \kappa_0$, the first-order part of this bound is at most $\eta\,[\, p_c\, \rho\, \kappa_0 - (1 - p_c)\, \mu_0\,]$, which is negative when ρ lies below the critical weight

$$\rho^* = \frac{(1 - p_c)\mu_0}{p_c \kappa_0},$$

the model-dependent threshold at which conflict-side pressure and matching-side repair balance. For $\rho < \rho^*$ and a step size satisfying the explicit bound given in Appendix B.4, the expected per-step change in constraint risk is non-positive.

Down-weighting leaves the repair from matching pairs unchanged and reduces the pressure from conflict pairs to at most ρ times its original value. Once ρ falls below ρ^*, the reduced pressure is smaller than the repair, so for a sufficiently small step each update is expected to lower the constraint risk or leave it unchanged. Plain DPO corresponds to $\rho = 1$, so when $\rho^* > 1$ it already satisfies this condition, and the data are self-correcting.

To use this result, CoDPO needs a value of ρ before training starts. Because ρ^* depends on gradients of the model, CoDPO instead uses its estimate ρ^*_{eq} from Remark 1, which needs only the cost gaps in the data. It sets the average conflict weight to $\rho = \rho^*_{\text{eq}}$, and keeps every weight at 1, that is, plain DPO, when $\rho^*_{\text{eq}} \geq 1$. Some drift can nevertheless remain, for two reasons: ρ is set at the estimated balance point rather than strictly below ρ^*, and ρ^* itself shifts as training proceeds. Section 4 implements this coarse level and adds a fine level that addresses this residual drift.

4 CoDPO: Controlling the Balance Before and During Training

CoDPO turns this analysis into two levels of control: a coarse level that sets the conflict weights before training (Section 4.1) and a fine level that corrects the drift remaining during training (Section 4.2).

4.1 Coarse Level: Setting the Weights Before Training

The coarse level assigns per-pair weights whose average conflict weight matches the pre-alignment estimate ρ^*_{eq} of Remark 1.[3] We keep matching pairs at full weight and down-weight conflict pairs by severity through the weighting family

$$w_{\tau,p}(\Delta s) = \begin{cases} 1, & \Delta s \geq 0, \\ \exp\!\big(-(|\Delta s|/\tau)^p\big), & \Delta s < 0, \end{cases} \tag{3}$$

[3]Two degenerate cases of Eq. (2) are fixed by convention, each the limit of the expression: if $p_c = 0$ there are no conflict pairs and $\rho^*_{\text{eq}} = +\infty$, leaving the modulation off; if $p_c = 1$ there are no matching pairs and $\rho^*_{\text{eq}} = 0$, removing conflict pairs.

where τ sets how strongly conflict pairs are down-weighted overall and p sets how sharply the weight drops from 1 to near 0 as the conflict becomes more severe.[4] This family extends the asymmetric exponential weighting introduced by PSDPO [11] with the sharpness p and recovers it at $p = 1$; in PSDPO, τ is chosen as a hyperparameter. In CoDPO, τ is instead determined by the diagnostic. For fixed p, each τ yields an average conflict weight $\rho(\tau) := \mathbb{E}_{\mathcal{D}_{\text{conf}}}[w_{\tau,p}(\Delta s)]$, which increases strictly from 0 to 1 as τ grows from 0 to ∞: a small τ suppresses nearly all conflict pairs, and a large τ keeps them at nearly full weight. CoDPO picks the τ whose average conflict weight equals the diagnostic,

$$\rho(\tau^*) = \text{clip}(\rho^*_{\text{eq}}, 0, 1). \tag{4}$$

Because $\rho(\tau)$ is strictly increasing, Eq. (4) has a unique solution for $0 < \rho^*_{\text{eq}} < 1$, which we locate by bisection; when $\rho^*_{\text{eq}} \geq 1$, CoDPO keeps $w_i = 1$ on every pair. The calibration uses only dataset statistics and requires no gradient computation.

4.2 Fine Level: Shifting the Margin During Training

The coarse weights are calibrated from ρ^*_{eq}, which assumes the learning signal $\beta(1 - \sigma(\beta\hat{r}_\theta))$ to be the same on every pair. This holds at initialization, where $\pi_\theta = \pi_{\text{ref}}$ gives every pair a zero margin $\hat{r}_\theta$. As training proceeds, the learning signal comes to differ from pair to pair, so the balance point that ρ^*_{eq} approximates moves while the weights stay fixed, and the constraint risk can keep rising; we call this remaining drift residual drift. Lowering the conflict weights further would discard more of the preference signal that conflict pairs carry, and fixed weights cannot respond to drift as it emerges, so we add a second control that acts during training without changing the weights.

Shifting the margin. DPO pushes each pair with strength $\beta(1 - \sigma(\beta\hat{r}_\theta))$, which is set by the pair's margin $\hat{r}_\theta$: the larger the margin, the more the pair is treated as already learned and the less it is pushed. Adding an offset to a pair's margin therefore makes DPO update that pair more or less strongly, while leaving its coarse weight w unchanged. On the i-th pair $(x_i, y_{w,i}, y_{l,i})$, DPO moves probability from $y_{l,i}$ toward $y_{w,i}$. On a conflict pair ($\Delta c_i > 0$), $y_{w,i}$ is the costlier response, so this raises the expected constraint cost; on a matching pair ($\Delta c_i < 0$), it lowers it. To push conflict pairs less and matching pairs more, we shift each pair's margin in proportion to its cost gap,

$$\hat{r}_\theta \;\rightarrow\; \hat{r}_\theta + \lambda\,\Delta c, \qquad \lambda \geq 0.$$

A conflict pair has its margin raised and is pushed less, most of all when its conflict is severe, so its effect on the constraint cost weakens while its preference is still learned; a matching pair has its margin lowered and is pushed more, so its repair strengthens. A single scalar λ thus adjusts each pair according to its own cost gap and, unlike the fixed margin offsets of ODPO [38], R-DPO [35], and SafeDPO [18], can be changed during training.

Implementing the shift. The margin shift $\hat{r}_\theta \rightarrow \hat{r}_\theta + \lambda\Delta c$ needs no change to the DPO loss: standard DPO computes it when the reference is tilted against high-cost outputs,

$$\tilde{\pi}_{\text{ref}}(y \mid x) \propto \pi_{\text{ref}}(y \mid x)\, \exp\big(-\lambda\, c(x, y)\big), \tag{5}$$

since $\log \tilde{\pi}_{\text{ref}}(y \mid x) = \log \pi_{\text{ref}}(y \mid x) - \lambda\, c(x, y) - \log Z(x)$ and the normalizer $Z(x)$, shared by y_w and y_l, cancels in the pairwise margin, leaving $\tilde{r}_\theta = \hat{r}_\theta + \lambda\,\Delta c$ (Appendix C.2). Running standard DPO with $\tilde{\pi}_{\text{ref}}$ thus replaces ℓ_R by $\ell_{\text{tilt}} = -\log\sigma(\beta\tilde{r}_\theta)$, and since $\lambda\Delta c$ does not depend on θ, the gradient direction is unchanged and only the argument of σ shifts:

$$\nabla\ell_{\text{tilt}} = -\beta\Big(1 - \sigma\big(\beta\hat{r}_\theta + \beta\lambda\Delta c\big)\Big)\,\nabla\hat{r}_\theta. \tag{6}$$

The tilt also links the shift to constrained alignment: because the optimal DPO policy is proportional to the reference times $\exp(r/\beta)$, training against $\tilde{\pi}_{\text{ref}}$ targets the cost-penalized reward $r - \beta\lambda c$, the form used by Lagrangian approaches [9, 16], which typically raise λ while an estimated cost exceeds a prescribed threshold; CoDPO instead raises λ when the constraint risk measured on the training pairs rises, as described next.

Setting the shift strength online. How much the constraint cost still rises under the coarse weights depends on how training unfolds, which cannot be known before training, so λ is adjusted during training. Adjusting λ requires deciding when to increase it and by how much. When to increase λ

[4] We use $p = 8$ in all experiments. As $p \rightarrow \infty$ the modulation approaches a hard threshold at $|\Delta s| = \tau$, which retains mild conflicts and removes severe ones; this differs from $\rho \rightarrow 0$, which removes conflict pairs regardless of severity.

Algorithm 1 CoDPO training procedure

Require: Preference dataset $\mathcal{D}$, scorer s, cost c, reference policy π_{ref}, temperature β, learning rate η, steps T, interval K, step fraction α
Ensure: Trained policy π_{θ^*}

```
Compute Δs_i and Δc_i for all pairs
Compute p_c and ρ*_eq, using the degenerate conventions of §4.1 when p_c ∈ {0, 1}
if ρ*_eq ≥ 1 then
    Set w_i ← 1 for all pairs
else if ρ*_eq = 0 then
    Set w_i ← 1 for matching pairs and w_i ← 0 for conflict pairs
else
    Solve Eq. (4) for τ* by bisection and set w_i using Eq. (3)
end if
Compute λ_tilt using Eq. (8)
Initialize λ ← α λ_tilt and C_prev ← Ĉ_D(θ_0)
for t = 1, ..., T do
    Sample mini-batch B
    Update θ using ∇_θ L_CoDPO(θ)
    if t mod K = 0 then
        C_curr ← Ĉ_D(θ_t)
        if C_curr > C_prev then
            λ ← min(λ + α λ_tilt, λ_tilt)
        end if
        C_prev ← C_curr
    end if
end for
```

is decided by the change in constraint risk, measured every K training steps on the training pairs through the estimate

$$\widehat{C}_{\mathcal{D}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_i \big[q_{\theta,i}\, c(x_i, y_{w,i}) + (1 - q_{\theta,i})\, c(x_i, y_{l,i}) \big], \qquad q_{\theta,i} = \frac{\pi_\theta(y_{w,i} \mid x_i)}{\pi_\theta(y_{w,i} \mid x_i) + \pi_\theta(y_{l,i} \mid x_i)}, \quad (7)$$

where $q_{\theta,i}$ is the probability the current policy assigns to $y_{w,i}$ within the pair, a normalized policy probability rather than the Bradley–Terry probability $\sigma(\beta \hat{r}_\theta)$. Because $\widehat{C}_{\mathcal{D}}$ uses only the responses $y_{w,i}$ and $y_{l,i}$ already in the training data, rather than new responses sampled from the policy, it adds no generation cost; it rises when, across pairs, the policy shifts probability toward the costlier response of each pair on balance. How much to increase it is set by the same first-order analysis as ρ^*_{eq}, which, under the same approximations and for the unmodulated loss, gives the shift strength that cancels first-order drift (Appendix C.4),

$$\lambda_{\mathrm{tilt}} = \max\left(0,\ \frac{2\,\mathbb{E}_{\mathcal{D}}[\Delta c]}{\beta\,\mathbb{E}_{\mathcal{D}}[\Delta c^2]}\right). \quad (8)$$

Because the coarse weights already remove part of this drift, setting $\lambda = \lambda_{\mathrm{tilt}}$ outright could overcorrect, so we increase λ gradually, in increments of $\alpha\lambda_{\mathrm{tilt}}$, and cap it at λ_{tilt}. Concretely, λ starts at $\alpha\lambda_{\mathrm{tilt}}$, so that the shift is active from the first step; whenever $\widehat{C}_{\mathcal{D}}$ has increased between two consecutive evaluations, which indicates that the residual drift has grown and the current shift is too weak, λ is raised by $\alpha\lambda_{\mathrm{tilt}}$, up to λ_{tilt}, and otherwise it is kept unchanged, since a stable or falling $\widehat{C}_{\mathcal{D}}$ indicates that the current shift suffices. Taking the maximum with 0 in λ_{tilt} prevents a negative λ, which would reverse the shift. It also switches the fine level off on self-correcting data: there $\mathbb{E}_{\mathcal{D}}[\Delta c] \le 0$, which is equivalent to $\rho^*_{\mathrm{eq}} \ge 1$, so $\lambda_{\mathrm{tilt}} = 0$ and, with all coarse weights at 1, CoDPO reduces to vanilla DPO.

4.3 Combining the Two Levels

CoDPO combines the coarse weights and the margin shift in a single loss:

$$\mathcal{L}_{\mathrm{CoDPO}}(\theta) = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i \cdot \log \sigma\big(\beta\, \hat{r}_\theta^{(i)} + \beta\lambda \Delta c_i\big), \quad (9)$$

where the weights w_i are fixed before training (Section 4.1) and the shift $\lambda\Delta c_i$ is adjusted during training (Section 4.2). The complete training procedure is summarized in Algorithm 1. The offline

Table 1: Experimental domains. p_c: conflict rate. ρ^*_{eq}: pre-alignment diagnostic (clipped to $[0, 1]$). Zero-cross.: estimated value of ρ at which drift changes sign in Table 2.

Domain	Pairwise preference	Pointwise constraint	Pairs	p_c	ρ^*_{eq}	Zero-cross.
HelpSteer2	Helpfulness rating	Response length	7,220	57%	0.71	$\sim$0.76
GSM8K	Explanation detail	Answer correctness	14,284	71%	0.40	$\sim$0.27
ARC	Reward-model score	Answer correctness	12,580	58%	0.72	$\sim$0.61
Math-Calc	Calculation density	Answer correctness	10,387	56%	0.80	$\sim$0.50
TL;DR	Human preference	Summary length	92,534	58%	0.59	$\sim$0.43
PKU-SafeRLHF	Helpfulness	Safety score	73,907	2.5%	1.0	—

calibration steps are performed once before training, in $O(|\mathcal{D}|)$ time per bisection iteration. Each optimization step has the same cost as vanilla DPO because w_i, Δc_i, and λ enter the loss only as scalar coefficients. Additional overhead comes only from evaluating the constraint-risk estimate $\widehat{C}_{\mathcal{D}}(\theta_t)$ to update λ, which uses forward passes on the fixed training pairs and no generation, and is needed only once every $K = 500$ steps.

5 Experiments

We evaluate on a suite of domains spanning diverse constraint types (Table 1). Most are conflict-dominated, where preference and pointwise criterion frequently disagree (HelpSteer2 [12], GSM8K [13], ARC [14], Math-Calc, TL;DR [1]). PKU-SafeRLHF [10] serves as a self-correcting control where the diagnostic predicts no intervention ($\rho^*_{\text{eq}} \geq 1$, clipped to 1.0). Math-Calc is constructed from GSM8K [13] with calculation density as the preference signal. We use the conflict-dominated domains for drift-curve validation and a representative subset (HelpSteer2, GSM8K, ARC, Math-Calc) for full method comparison against baselines.

We use Qwen2.5-7B-Instruct [39] as the default base model with LoRA [40], DPO with β=0.1, and CoDPO with p=8, K=500, α=0.1 (full hyperparameters in Appendix A). For robustness, we additionally test three base models (DeepSeek-R1-Distill-Llama-8B [8], Mistral-7B-Instruct-v0.3 [41], Qwen3-8B [42]) and three preference losses (IPO [5], SimPO [7], KTO [6]). We compare against five recent methods: SafeDPO [18], PD-DPO [16], C2-DPO [17], MODPO [20], and Rewarded Soups [19], with vanilla DPO (ρ=1) and hard filtering (ρ=0) as references.

We evaluate each model on the two candidate outputs in each preference pair. The expected constraint cost is

$$\mathbb{E}[c] = P_\theta(y_w \succ y_l \mid x)\, c(x, y_w) + P_\theta(y_l \succ y_w \mid x)\, c(x, y_l),$$

where the pairwise probabilities are obtained from the policy's likelihoods of the two candidates. Drift is measured as $\mathbb{E}[c(\theta)] - \mathbb{E}[c(\theta_{\text{ref}})]$, preference learning as $P_\theta(\text{chosen}) = P_\theta(y_w \succ y_l \mid x)$, and generation quality as pairwise LLM-as-judge win rates using JudgeLM-13B [43] with position-debiased scoring.

5.1 Drift Curves and Diagnostic Validation

To test the balance in Theorem 1, we vary the average conflict weight ρ over $\{0, 0.3, 0.5, 0.7, 1\}$ using the modulation family of CoDPO, train a model at each value, and measure drift and $P(\text{chosen})$ on the training pairs (Table 2). The two ends of this range correspond to familiar settings: $\rho = 0$ removes all conflict pairs, as hard filtering does, and $\rho = 1$ is vanilla DPO.

Table 2 shows three patterns that hold in all five conflict-dominated domains. First, drift increases monotonically with ρ. Second, drift is negative at $\rho = 0$, where only matching pairs drive training, and positive at $\rho = 1$, where conflict pairs keep full weight, consistent with the balance between repair and pressure in Theorem 1. Third, $P(\text{chosen})$ largely increases with ρ, so removing conflict pairs to suppress drift also weakens preference learning. Hard filtering ($\rho = 0$) keeps only 7% to 41% of the gain in $P(\text{chosen})$ that vanilla DPO achieves over the base model on HelpSteer2, GSM8K, ARC, and Math-Calc, and on TL;DR it falls below the base model. Removing all conflict pairs thus suppresses drift at a large cost in preference learning, which motivates keeping ρ near the value at which drift changes sign rather than at 0.

This value can be read off Table 2 (Zero-cross. in Table 1). Computed from dataset statistics alone, the diagnostic agrees with this behavior: $\rho^*_{\text{eq}} < 1$ in all five domains, matching the positive drift

Table 2: Drift curves across six domains. Each row corresponds to a different average conflict weight ρ; Base denotes the reference model before alignment. The constraint cost is response length (characters) on HelpSteer2 and TL;DR, binary incorrectness probability on GSM8K, ARC, and Math-Calc, and based on the safety score on PKU-SafeRLHF (Appendix A).

	HelpSteer2			GSM8K			ARC			Math-Calc			TL;DR			PKU-SafeRLHF		
ρ	E[c]	Drift	P(ch)	E[c]	Drift	P(ch)	E[c]	Drift	P(ch)	E[c]	Drift	P(ch)	E[c]	Drift	P(ch)	E[c]	Drift	P(ch)
Base	1227	—	.454	.363	—	.351	.525	—	.439	.363	—	.420	126.5	—	.511	.4956	—	.396
0.0	1182	−46	.489	.225	−.138	.358	.409	−.116	.473	.227	−.136	.516	121.4	−5.1	.479	.4915	−.004	.418
0.3	1197	−30	.532	.379	+.016	.423	.506	−.019	.514	.337	−.026	.597	125.9	−0.6	.556	.4922	−.003	.418
0.5	1212	−15	.550	.392	+.029	.434	.519	−.006	.523	.363	.000	.623	126.8	+0.3	.566	.4923	−.003	.418
0.7	1223	−5	.560	.404	+.041	.443	.530	+.005	.528	.375	+.012	.631	127.2	+0.7	.568	.4926	−.003	.417
1.0	1247	+20	.566	.422	+.059	.453	.544	+.019	.532	.389	+.026	.652	127.5	+1.0	.567	.4928	−.003	.417

Table 3: Comparison with prior work across four domains. Drift $= \mathbb{E}[c(\theta)] - \mathbb{E}[c(\theta_{\text{ref}})]$. P(ch) $= P_\theta(y_w \succ y_l \mid x)$. MODPO ($w$=1.0) and Soups ($\alpha$=0.75) use the setting with the smallest |drift|. Full hyperparameter sweeps are in Appendix D. Implementation details are given in Appendix A.

	HelpSteer2		GSM8K		ARC		Math-Calc	
Method	Drift	P(ch)	Drift	P(ch)	Drift	P(ch)	Drift	P(ch)
Base	—	.454	—	.351	—	.439	—	.420
Vanilla	+59	.566	+.129	.453	+.055	.532	+.071	.652
SafeDPO	−132	.437	−.353	.284	−.493	.412	−.352	.442
PD-DPO	−128	.452	−.360	.289	−.509	.427	−.357	.449
C2-DPO	≈0	.457	+.004	.353	+.003	.442	≈0	.423
MODPO	−74	.502	−.180	.365	−.158	.466	−.175	.514
Soups	+3	.509	+.018	.406	−.010	.480	+.001	.540
CoDPO	+18.7	**.640**	+.082	**.551**	+.046	**.622**	+.066	**.757**

at $\rho = 1$, and ρ^*_{eq} lies within 0.05 to 0.30 of the zero crossing, a gap we attribute to the first-order approximation. On PKU-SafeRLHF, $\rho^*_{\text{eq}} \geq 1$ matches its negligible drift. The diagnostic thus decides whether to intervene and sets the strength of the coarse weights.

5.2 Comparison with Prior Work

Table 3 compares CoDPO with five recent baselines across four conflict-dominated domains, and the results show three patterns. First, SafeDPO [18] and PD-DPO [16], which enforce a constraint on the expected cost, lower the constraint cost well below the reference level, and their $P(\text{chosen})$ remains near or below that of the base model. C2-DPO [17], which constrains the update itself, keeps both the constraint cost and $P(\text{chosen})$ near their initial values. These methods are designed to satisfy a constraint rather than to retain preference learning at limited drift, which is what our evaluation measures. Second, MODPO [20] and Rewarded Soups [19] expose a trade-off hyperparameter, and sweeping it traces a frontier between drift and preference learning (Tables 6 and 7 in Appendix D). Third, among all methods and configurations with drift comparable to or below that of CoDPO, CoDPO attains the highest $P(\text{chosen})$ in all four domains, and the configurations with higher $P(\text{chosen})$, MODPO with constraint weight $w \leq 0.5$, have roughly 1.5 to 5 times its drift. Relative to vanilla DPO, CoDPO also reduces drift while raising $P(\text{chosen})$ in all four domains.

5.3 Contributions of Coarse Weights and Margin Shift

Table 4 separates the contributions of the two components by comparing vanilla DPO, coarse weights alone (Eq. 3 with λ=0), and the full CoDPO (coarse weights with the adaptive margin shift) across four conflict-dominated domains. Coarse control alone reduces drift in all four domains. Adding the margin shift reduces drift further in every case. This additional reduction is largest on GSM8K (+17 percentage points) and smallest on HelpSteer2 (+2 percentage points).

5.4 Robustness Across Models and Losses

On HelpSteer2, across four base models (Qwen2.5-7B [39], DeepSeek-8B [8], Mistral-7B [41], Qwen3-8B [42]; Table 4), vanilla DPO drifts upward in every case and the coarse weights reduce

Table 4: Drift under vanilla training, coarse control only, and full CoDPO, across domains (Qwen2.5-7B), base models (HelpSteer2), and preference losses (HelpSteer2; IPO with β=0.5, SimPO with β=2, γ=0.5; coarse control at ρ^*_{eq}). Coarse control reduces drift in every setting.

	Domains (Qwen2.5-7B)				Base models (HelpSteer2)			Losses (HelpSteer2)		
	HelpSteer2	GSM8K	ARC	Math-Calc	DeepSeek-8B	Mistral-7B	Qwen3-8B	IPO	SimPO	KTO
Vanilla	+59	+.129	+.055	+.071	+18	+92	+34	+15	+74	+72
Coarse	+20	+.105	+.049	+.068	−7	+45	+0.4	−36	−1	+16
Combined	+18.7	+.082	+.046	+.066	−7	+41	+0.4	—	—	—

Table 5: LLM-as-judge evaluation (JudgeLM-13B) on HelpSteer2: CoDPO compared pairwise against each method.

	Vanilla DPO	SafeDPO	PD-DPO	C2-DPO	MODPO	Soups
CoDPO wins	14.1%	86.3%	73.9%	12.7%	52.2%	13.5%
Opp. wins	10.5%	0.0%	3.0%	10.7%	7.7%	10.0%
Tie	75.5%	13.6%	23.2%	76.6%	40.1%	76.5%

drift in every case, with the margin shift reducing it further on Qwen2.5 and Mistral. Applied as a data-level plug-in to IPO [5], SimPO [7], and KTO [6] (Table 4), the coarse weights also reduce drift in all three cases.

5.5 LLM-as-Judge Evaluation

To evaluate generation quality beyond probability-level metrics, we compare CoDPO pairwise with each method on all 7,220 HelpSteer2 prompts using JudgeLM-13B [43] with position-debiased scoring (Table 5), and CoDPO wins more often than it loses against every compared method. Compared with SafeDPO [18], PD-DPO [16], and MODPO [20], CoDPO wins 52%–86% of comparisons, in line with their lower $P(\text{chosen})$ in Table 3. Compared with vanilla DPO [4], C2-DPO [17], and Rewarded Soups [19], most comparisons are ties (75%–77%), so CoDPO reduces drift relative to vanilla DPO without a change in generation quality detectable by this judge. GPT-4o as a second judge agrees: CoDPO ties vanilla DPO and is preferred over MODPO 44.9% to 3.9% (Appendix F.3).

5.6 Ablations

Steepness parameter p. The weighting family of Eq. 3 recovers the PSDPO weighting at p=1 and approaches a hard threshold on conflict severity as $p\to\infty$. Table 8 sweeps p on HelpSteer2: drift reduction increases monotonically with p (22% to 75%), while $P(\text{chosen})$ remains essentially constant (.640–.644). Due to compute constraints, we use p=8 in all experiments without tuning it per domain, although larger p lowers drift further here.

Dynamic diagnostic validity. To examine the initialization approximation $\sigma(\beta\hat{r}_\theta) \approx 1/2$, we track $\rho^*(t)$ during vanilla DPO training on HelpSteer2, recomputing only the learning signal from the actual $\sigma(\beta\hat{r}_\theta)$ on all 7,220 pairs: $\rho^*(0) = 0.710$ matches $\rho^*_{\text{eq}} = 0.709$, and as $\sigma \to 0.946$, $\rho^*(t)$ moves to 0.603 (gap 0.11). This is consistent with the balance point shifting during training, which motivates the adaptive margin shift; Appendix E reports the trajectory of λ during training.

6 Conclusion

We studied constraint drift under heterogeneous supervision and showed that, for small updates, it depends on the balance between conflict-side pressure and matching-side repair. From this balance we derive a diagnostic, ρ^*_{eq}, that is computed from dataset statistics before training and indicates whether a dataset needs intervention. When it does, CoDPO down-weights conflict pairs through the asymmetric weighting of PSDPO [11], with a strength set by ρ^*_{eq} rather than tuned, and shifts the DPO margin during training to address the remaining drift. Empirically, the intervention decision given by ρ^*_{eq} matches the observed drift on all evaluated domains, and in our main comparison CoDPO reduces drift while largely preserving the preference learning of vanilla DPO. A natural next step is to track the balance as the model changes during training, rather than only at initialization, and to extend the analysis to settings where the preference pairs themselves are generated during training [44].

Acknowledgments and Disclosure of Funding

This work used the high-performance GPU cluster acquired with support from the U.S. National Science Foundation (NSF) under Award #2511989, "Research Infrastructure: MRI: Acquisition of a High-Performance GPU Cluster for Large-Scale AI." This work also used Delta and DeltaAI at the National Center for Supercomputing Applications (NCSA) through allocation CIS260780 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [45], which is supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The authors declare no competing interests.

References

[1] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *NeurIPS*, 2020.

[2] 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. In *NeurIPS*, 2022.

[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] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

[5] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, 2024.

[6] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. In *ICML*, 2024.

[7] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *NeurIPS*, 2024.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

[9] Juntao Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *ICLR*, 2024.

[10] Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Alex Qiu, Jiayi Zhou, Kaile Wang, Boxun Li, et al. PKU-SafeRLHF: Towards multi-level safety alignment for LLMs with human preference. In *ACL*, 2025.

[11] Siwei Meng, Yawei Luo, Shu Zhang, and Ping Liu. When physical preferences meet semantic constraints: Physical and semantic direct preference optimization for text-to-video generation. In *ACM MM*, 2026.

[12] Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. HelpSteer 2: Open-source dataset for training top-performing reward models. In *NeurIPS Datasets and Benchmarks Track*, 2024.

[13] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[14] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[15] Akifumi Wachi, Thien Q Tran, Rei Sato, Takumi Tanabe, and Youhei Akimoto. Stepwise alignment for constrained language model policy optimization. In *NeurIPS*, 2024.

[16] Yihan Du, Seo Taek Kong, and R. Srikant. Provably convergent primal-dual DPO for constrained LLM alignment. *arXiv preprint arXiv:2510.05703*, 2025.

[17] Kavosh Asadi, Xingzi Xu, Julien Han, Ege Beyazit, Idan Pipano, Dominique Perrault-Joncas, Shoham Sabach, Mohammad Ghavamzadeh, and Karim Bouyarmane. C2-DPO: Constrained controlled direct preference optimization. *TMLR*, 2026.

[18] Geon-Hyeong Kim, Yu Jin Kim, Byoungjip Kim, Honglak Lee, Kyunghoon Bae, Youngsoo Jang, and Moontae Lee. SafeDPO: A simple approach to direct preference optimization with enhanced safety. In *ICLR*, 2026.

[19] Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *NeurIPS*, 2023.

[20] Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of ACL*, 2024.

[21] Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is more: Improving LLM alignment via preference data selection. In *NeurIPS*, 2025.

[22] Shangpin Peng, Weinong Wang, Zhuotao Tian, Senqiao Yang, Xing Wu, Haotian Xu, Chengquan Zhang, Takashi Isobe, Baotian Hu, and Min Zhang. Uni-DPO: A unified paradigm for dynamic preference optimization of LLMs. In *ICLR*, 2026.

[23] Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. Cal-DPO: Calibrated direct preference optimization for language model alignment. In *NeurIPS*, 2024.

[24] Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust DPO: Aligning language models with noisy feedback. In *ICML*, 2024.

[25] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-DPO: Direct preference optimization with dynamic β. In *NeurIPS*, 2024.

[26] Moxin Li, Yuantao Zhang, Wenjie Wang, Wentao Shi, Zhuo Liu, Fuli Feng, and Tat-Seng Chua. Self-improvement towards Pareto optimality: Mitigating preference conflicts in multi-objective alignment. In *Findings of ACL*, 2025.

[27] Zhihao Xu, Yongqi Tong, Xin Zhang, Jun Zhou, and Xiting Wang. Understanding conflicts in multi-objective alignment through reward consistency. In *Findings of ACL*, 2026.

[28] Jia Zhang, Yao Liu, Chen-Xi Zhang, Yi Liu, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. Data selection for LLM alignment using fine-grained preferences. In *ICLR*, 2026.

[29] Peter Chen, Xiaopeng Li, Xi Chen, and Tianyi Lin. Reward-free alignment for conflicting objectives. In *ICML*, 2026.

[30] Wenxuan Zhang, Philip H. S. Torr, Mohamed Elhoseiny, and Adel Bibi. Bi-factorial preference optimization: Balancing safety-helpfulness in language models. In *ICLR*, 2025.

[31] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020.

[32] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 2021.

[33] Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Erxue Min, and Sophia Ananiadou. Selective preference optimization via token-level reward function estimation. In *EMNLP*, 2025.

[34] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *COLM*, 2024.

[35] Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of ACL*, 2024.

[36] Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *ICLR*, 2025.

[37] Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *ICLR*, 2025.

[38] Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. In *Findings of ACL*, 2024.

[39] An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[40] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[41] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[42] An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[43] Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *ICLR*, 2025.

[44] Zixuan Xu, Haoxuan Chen, Jiangxin Sun, and Jian-Fang Hu. GPDPO: Learning from reward conflicts via group Pareto-dominance policy optimization. *arXiv preprint arXiv:2610.35480*, 2026.

[45] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. ACCESS: Advancing innovation: NSF's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing (PEARC '23)*, 2023.

A Experimental Setup Details

Domains. On HelpSteer2 [12], the pairwise signal is helpfulness rating and the pointwise constraint is response length. On GSM8K [13] and Math-Calc, the preference favors more detailed explanations while the constraint requires answer correctness; Math-Calc reuses GSM8K solutions but redefines the preference signal as calculation density rather than explanation detail, which produces a different distribution of conflict pairs. On ARC [14], a reward model determines pairwise preference and the constraint is factual correctness. On TL;DR [1], the preference comes from human annotations and the constraint is summary length. On PKU-SafeRLHF [10], the preference is helpfulness and the constraint is safety.

Training hyperparameters. All experiments use Qwen2.5-7B-Instruct [39] as the default base model with LoRA (rank 16, scaling factor 32). We train for 3 epochs, except for TL;DR and PKU-SafeRLHF where we use 1 epoch. We use DPO with β=0.1, learning rate 5×10^{-5}, and gradient accumulation 8. CoDPO uses p=8 and updates λ every K=500 steps with step fraction α=0.1. Some methods are trained with a custom training loop rather than the TRL trainer, so absolute values may differ slightly across tables. All experiments run on NVIDIA RTX PRO 6000 GPUs, and the reported experiments, including baseline sweeps and ablations, take approximately 700 GPU-hours.

Baselines. For each baseline, we sweep its primary hyperparameter and report representative configurations spanning the drift–preference frontier. The sweeps for MODPO and Rewarded Soups are reported in Appendix D.

B Full Proofs and Derivations

Appendix B.1 states the notation and assumptions, Appendix B.2 proves the coupling result used in Section 3.2, Appendices B.3 and B.4 prove Theorems 1 and 2, and Appendix B.5 derives the diagnostic ρ^*_{eq} of Remark 1.

B.1 Notation and Assumptions

Throughout, the DPO per-sample loss of a preference pair (x, y_w, y_l) is $\ell_R(\theta) = -\log\sigma(\beta\hat{r}_\theta)$, with implicit reward margin

$$\hat{r}_\theta = \log\frac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)},$$

and the constraint risk of the policy is

$$C(\theta) = \mathbb{E}_x\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[c(x,y)]\right].$$

For the same pair, the score discrepancy is $\Delta s = s(x, y_w) - s(x, y_l)$ and the cost gap is $\Delta c = c(x, y_w) - c(x, y_l)$, with $\Delta c > 0$ on conflict pairs and $\Delta c \le 0$ on matching pairs. Because c is strictly decreasing in s, $\Delta s < 0$ holds if and only if $\Delta c > 0$, so the conflict condition of Definition 1 can be stated equivalently in terms of either difference.

The results rest on the following regularity conditions, which are assumed throughout: (i) $C(\theta)$ is L_C-smooth; (ii) $\|\nabla\ell_R\| \le G_R$; (iii) on $\mathcal{D}_{\mathrm{conf}}$, the conflict severity Δc and the geometry term $\mathcal{I}_\theta$ are bounded away from zero; and (iv) the matching repair constant

$$\mu_0 := \mathbb{E}_{\mathcal{D}_{\mathrm{match}}}\left[\beta(1-\sigma(\beta\hat{r}_\theta))|\Delta c|\mathcal{I}_\theta\right]$$

is strictly positive.

In addition to (i)–(iv), the analysis uses one simplification. The constraint risk $C(\theta)$ averages the expected cost $C_x(\theta) = \mathbb{E}_{y\sim\pi_\theta}[c(x,y)]$ over prompts. Because all prompts share the same parameters, an update on a pair with prompt x changes not only C_x but also the costs on other prompts; for instance, learning to prefer a longer response to one prompt can lengthen the responses to others. We account only for the effect on the pair's own prompt, replacing $\langle\nabla C(\theta), \nabla_\theta\hat{r}_\theta\rangle$ by $\langle\nabla C_x(\theta), \nabla_\theta\hat{r}_\theta\rangle$. With this simplification, the change in C is a sum of per-pair terms, each depending only on that pair's own quantities; the per-pair identities below and the upper bound of Theorem 2 rely on this form. Beyond these conditions, three further assumptions are used in specific places: (v) in the modulation analysis of Theorem 2, (vi) in the closed-form diagnostic of Remark 1, and (vii) in the treatment of tied pairs in Definition 1.

(v) *More severe conflicts exert more pressure (used in Theorem 2).* For a conflict pair, let $u := \beta(1 - \sigma(\beta\hat{r}_\theta))\,\Delta c\,\mathcal{I}_\theta$ denote how strongly it pushes the constraint cost up, so that $\kappa_0 = \mathbb{E}_{\mathcal{D}_{\text{conf}}}[u]$ and $\kappa_w = \mathbb{E}_{\mathcal{D}_{\text{conf}}}[w(\Delta s)\,u]$. We assume that a more severe conflict (more negative Δs) pushes the cost up at least as strongly as a milder one. Since u is the product of the learning signal, the cost gap, and the geometry term, we expect this to hold approximately when pairs with larger score discrepancies also have larger cost gaps, provided that the learning signal and the geometry term take similar values across conflict pairs.[5] The weights are ordered the opposite way, with $w(\Delta s)$ smallest on the most severe conflicts. Down-weighting therefore reduces most the pairs that push hardest, and Chebyshev's sum inequality makes this precise: since w and u are oppositely ordered, $\kappa_w = \mathbb{E}_{\mathcal{D}_{\text{conf}}}[w\,u] \le \mathbb{E}_{\mathcal{D}_{\text{conf}}}[w]\,\mathbb{E}_{\mathcal{D}_{\text{conf}}}[u] = \rho\,\kappa_0$.

(vi) *Conflict and matching pairs have similar geometry (used in Remark 1).* The geometry term $\mathcal{I}_\theta$ measures how much the expected cost on a prompt changes, per unit of the cost gap Δc, when DPO learns the pair. We assume that this sensitivity is on average about the same for conflict and matching pairs, that is, $\overline{\mathcal{I}}_{\text{conf}} \approx \overline{\mathcal{I}}_{\text{match}}$, where each average weights a pair by $|\Delta c|$ (Appendix B.5). This is plausible when the two kinds of pairs share the same distribution of prompts and the reference policy gives non-negligible probability to both responses of each pair; Appendix F.1 checks it with a gradient-norm proxy. Under this assumption, the geometry term cancels in the ratio μ_0/κ_0, and ρ^*_{eq} in Eq. (2) depends only on the cost gaps Δc in the data.

(vii) *Tied pairs (used in Definition 1).* For pairs with $\Delta c = 0$, the composite contribution $|\Delta c|\,\mathcal{I}_\theta$ to μ_0 is taken to be zero, that is, tied pairs are assumed not to contribute to the first-order change in constraint risk.

Section 5.1 provides indirect support for (vi): the intervention decision given by ρ^*_{eq}, which relies on it, matches the observed drift on all evaluated domains.

B.2 Proof of Proposition 1

Proposition 1 (Coupling emergence) *For a conflict pair $(x, y_w, y_l) \in \mathcal{D}_{\text{conf}}$ with $\Delta c > 0$, the DPO update satisfies the first-order inequality*

$$-\langle \nabla_\theta C(\theta), \nabla_\theta \ell_R(\theta) \rangle \ge \beta(1 - \sigma(\beta\hat{r}_\theta))\,\Delta c\,\mathcal{I}_\theta \ge 0,$$

where $\mathcal{I}_\theta = \mathcal{I}_\theta(x, y_w, y_l) > 0$ is the geometry term defined below. The lower bound decomposes into three factors: the learning signal strength $\beta(1 - \sigma(\beta\hat{r}_\theta))$, the conflict severity Δc, and the geometry term $\mathcal{I}_\theta$.

The proof computes the inner product between the constraint gradient and the DPO gradient and then determines its sign. On the preference side, differentiating $\ell_R = -\log\sigma(\beta\hat{r}_\theta)$ gives

$$\nabla_\theta \ell_R = -\beta\big(1 - \sigma(\beta\hat{r}_\theta)\big)\nabla_\theta \hat{r}_\theta, \tag{10}$$

where

$$\nabla_\theta \hat{r}_\theta = \nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x).$$

On the constraint side, the score-function identity gives, for a fixed prompt x,

$$\nabla_\theta \mathbb{E}_{y\sim\pi_\theta}[c(x,y)] = \sum_y c(x,y)\,\pi_\theta(y|x)\,\nabla_\theta \log \pi_\theta(y|x). \tag{11}$$

Substituting Eq. (10), the constraint-preference coupling for a single sample is

$$-\langle \nabla C(\theta), \nabla_\theta \ell_R \rangle = \beta(1 - \sigma(\beta\hat{r}_\theta))\langle \nabla_\theta C(\theta), \nabla_\theta \hat{r}_\theta \rangle, \tag{12}$$

and evaluating the constraint gradient at the level of prompt x, as stated in Appendix B.1, with Eq. (11) turns the inner product into

$$\sum_y c(x,y)\,\pi_\theta(y|x)\,\big[k_\theta(y, y_w; x) - k_\theta(y, y_l; x)\big], \tag{13}$$

where

$$k_\theta(y, y'; x) := \langle \nabla \log \pi_\theta(y|x), \nabla \log \pi_\theta(y'|x) \rangle$$

[5] The learning signal meets this exactly at initialization, where it equals $\beta/2$ on every pair, so of these two factors only the geometry term needs to be assumed to vary little.

is the Fisher kernel. To determine the sign of this expression, note that $\mathbb{E}_{y\sim\pi_\theta}[k_\theta(y, y'; x)] = 0$, so Eq. (13) equals the covariance

$$\mathrm{Cov}_{y\sim\pi_\theta}\big(c(x, y),\ k_\theta(y, y_w; x) - k_\theta(y, y_l; x)\big). \tag{14}$$

On conflict pairs, $c(x, y_w) > c(x, y_l)$, and Assumption (iii) ensures that this covariance is positive. Normalizing it by the cost gap defines the geometry term

$$\mathcal{I}_\theta(x, y_w, y_l) := \frac{1}{\Delta c}\langle \nabla_\theta C_x(\theta), \nabla_\theta \hat{r}_\theta\rangle, \tag{15}$$

where $C_x(\theta) = \mathbb{E}_{y\sim\pi_\theta}[c(x, y)]$, and substituting it into Eq. (12) gives

$$-\langle \nabla C, \nabla \ell_R\rangle \geq \beta(1 - \sigma(\beta\hat{r}_\theta))\Delta c\, \mathcal{I}_\theta \geq 0 \quad \text{on } \mathcal{D}_{\mathrm{conf}}. \quad \square$$

An analogous argument shows that on matching pairs ($\Delta c \leq 0$),

$$-\langle \nabla C, \nabla \ell_R\rangle \leq 0,$$

which is the repair effect summarized by μ_0.

B.3 Proof of Theorem 1

The proof combines a smoothness bound with the decomposition of the DPO gradient over the two subsets. By L_C-smoothness, one gradient step changes the constraint risk by at least

$$C(\theta_{t+1}) - C(\theta_t) \geq -\eta\langle \nabla C(\theta_t), \nabla L_R(\theta_t)\rangle - \frac{L_C}{2}\eta^2\|\nabla L_R(\theta_t)\|^2. \tag{16}$$

The first-order term splits over the partition $\mathcal{D} = \mathcal{D}_{\mathrm{match}} \cup \mathcal{D}_{\mathrm{conf}}$ as

$$-\langle \nabla C, \nabla L_R\rangle = (1 - p_c)\,\mathbb{E}_{\mathcal{D}_{\mathrm{match}}}[-\langle \nabla C, \nabla \ell_R\rangle] + p_c\,\mathbb{E}_{\mathcal{D}_{\mathrm{conf}}}[-\langle \nabla C, \nabla \ell_R\rangle],$$

and the two terms can be bounded separately. On matching pairs, the definition of μ_0 gives

$$\mathbb{E}_{\mathcal{D}_{\mathrm{match}}}[-\langle \nabla C, \nabla \ell_R\rangle] = -\mu_0,$$

and on conflict pairs, Proposition 1 gives

$$\mathbb{E}_{\mathcal{D}_{\mathrm{conf}}}[-\langle \nabla C, \nabla \ell_R\rangle] \geq \kappa_0.$$

Substituting both terms into Eq. (16) and using $\|\nabla L_R\| \leq G_R$ yields

$$\mathbb{E}[C(\theta_{t+1}) - C(\theta_t)] \geq \eta[-(1 - p_c)\mu_0 + p_c\kappa_0] - \frac{L_C}{2}\eta^2 G_R^2. \quad \square$$

For sufficiently small η, the sign of the per-step change is therefore governed by the first-order balance $p_c\kappa_0$ versus $(1 - p_c)\mu_0$.

B.4 Proof of Theorem 2

The proof follows the same route as for Theorem 1, with the modulation applied to conflict pairs and the smoothness bound taken in the upper direction. Under the modulation, the loss gradient becomes

$$\nabla L_{R,w}(\theta) = (1 - p_c)\,\mathbb{E}_{\mathcal{D}_{\mathrm{match}}}[\nabla \ell_R] + p_c\,\mathbb{E}_{\mathcal{D}_{\mathrm{conf}}}[w(\Delta s)\nabla \ell_R],$$

and L_C-smoothness bounds the change in constraint risk from above,

$$C(\theta_{t+1}) - C(\theta_t) \leq -\eta\langle \nabla C, \nabla L_{R,w}\rangle + \frac{L_C}{2}\eta^2\|\nabla L_{R,w}\|^2.$$

The first-order term decomposes over the two subsets as before,

$$-\langle \nabla C, \nabla L_{R,w}\rangle = -(1 - p_c)\mu_0 + p_c\kappa_w,$$

where

$$\kappa_w := \mathbb{E}_{\mathcal{D}_{\mathrm{conf}}}\big[w(\Delta s)\,\beta(1 - \sigma(\beta\hat{r}_\theta))\,\Delta c\,\mathcal{I}_\theta\big].$$

To compare κ_w with κ_0, note that on $\mathcal{D}_{\mathrm{conf}}$ the modulation $w(\Delta s)$ is non-decreasing in Δs, since more severe conflicts receive smaller weights, while by Assumption (v) the per-sample coupling $\beta(1 - \sigma(\beta\hat{r}_\theta))\,\Delta c\,\mathcal{I}_\theta$ is non-increasing in Δs. Chebyshev's sum inequality for oppositely ordered sequences therefore gives

$$\kappa_w \leq \rho\,\kappa_0, \qquad \rho = \mathbb{E}_{\mathcal{D}_{\mathrm{conf}}}[w(\Delta s)].$$

Substituting the first-order decomposition into the smoothness bound, and using $\|\nabla L_{R,w}\| \le G_R$ since $w \le 1$, yields

$$\mathbb{E}[C(\theta_{t+1}) - C(\theta_t)] \le -\eta(1-p_c)\mu_0 + \eta p_c \kappa_w + \frac{L_C}{2}\eta^2 G_R^2.$$

It remains to determine when this upper bound is non-positive. When

$$\rho < \rho^* := \frac{(1-p_c)\mu_0}{p_c \kappa_0},$$

the first-order term of the upper bound, $-\eta(1-p_c)\mu_0 + \eta p_c \kappa_w$, is strictly negative by $\kappa_w \le \rho\,\kappa_0$. Combined with the second-order term, the upper bound itself is then non-positive, and hence $\mathbb{E}[C(\theta_{t+1}) - C(\theta_t)] \le 0$, whenever

$$\eta \le \frac{2\,[(1-p_c)\mu_0 - p_c \rho \kappa_0]}{L_C G_R^2}. \quad \square$$

B.5 Derivation of the Pre-Alignment Diagnostic

The closed-form diagnostic ρ^*_{eq} in Remark 1 follows from evaluating the ratio $(1-p_c)\mu_0/(p_c\kappa_0)$ of Theorem 1 at initialization. At initialization, alignment starts from $\pi_{\theta_0} = \pi_{\mathrm{ref}}$, so the implicit reward margin satisfies $\hat{r}_{\theta_0} = 0$ on every pair and the learning signal takes the same value on every pair,

$$\beta\big(1 - \sigma(\beta \hat{r}_{\theta_0})\big) = \beta\big(1 - \sigma(0)\big) = \beta/2.$$

To express κ_0 and μ_0 through this common value, we define the severity-weighted average geometry terms

$$\overline{\mathcal{I}}_{\mathrm{conf}} := \frac{\mathbb{E}_{\mathcal{D}_{\mathrm{conf}}}[\Delta c\, \mathcal{I}_\theta]}{\overline{\Delta c}_{\mathrm{conf}}}, \qquad \overline{\mathcal{I}}_{\mathrm{match}} := \frac{\mathbb{E}_{\mathcal{D}_{\mathrm{match}}}[|\Delta c|\, \mathcal{I}_\theta]}{\overline{|\Delta c|}_{\mathrm{match}}}.$$

With these terms and the learning signal fixed at $\beta/2$, the definitions of κ_0 and μ_0 give, without approximation,

$$\kappa_0 = (\beta/2)\,\overline{\mathcal{I}}_{\mathrm{conf}}\,\overline{\Delta c}_{\mathrm{conf}}, \qquad \mu_0 = (\beta/2)\,\overline{\mathcal{I}}_{\mathrm{match}}\,\overline{|\Delta c|}_{\mathrm{match}}.$$

In the ratio $(1-p_c)\mu_0/(p_c\kappa_0)$, the factor $\beta/2$ cancels exactly, and under Assumption (vi), which posits $\overline{\mathcal{I}}_{\mathrm{conf}} \approx \overline{\mathcal{I}}_{\mathrm{match}}$, the geometry factors cancel approximately, so that

$$\rho^* = \frac{(1-p_c)\mu_0}{p_c\kappa_0} \approx \frac{(1-p_c)\,\overline{|\Delta c|}_{\mathrm{match}}}{p_c\,\overline{\Delta c}_{\mathrm{conf}}} = \frac{\sum_{\mathcal{D}_{\mathrm{match}}} |\Delta c|}{\sum_{\mathcal{D}_{\mathrm{conf}}} \Delta c} = \rho^*_{\mathrm{eq}}.$$

The last two forms are equal because $1-p_c$ and p_c are the fractions of matching and conflict pairs, so the numerator and the denominator are sums over the corresponding subset divided by the same total number of pairs. Assumption (vi) is therefore the only approximation in the derivation.

This approximation is most accurate near initialization. As training proceeds, the learning signal is no longer constant across pairs and the geometry terms change with the policy, so ρ^*_{eq} should be interpreted as a theory-informed proxy rather than an exact threshold.

C Derivations for the Margin Shift

This appendix derives the margin shift used by the fine level: Appendix C.1 motivates its form, Appendices C.2 and C.3 show how a tilted reference implements it and what gradient it produces, and Appendix C.4 derives the shift strength λ_{tilt}.

C.1 Ideal residual correction

This subsection motivates the form of the margin shift used by the fine level; the projection below is not part of the method itself. After coarse control, the modulated DPO gradient is

$$g_R^w = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \Big[-w_i\, \beta\big(1 - \sigma(\beta \hat{r}_\theta^{(i)})\big) \Big] \nabla_\theta \hat{r}_\theta^{(i)}, \tag{17}$$

and the parameters move along the update direction $d := -g_R^w$. If residual drift remains, the harmful part of d is its component along the constraint gradient

$$g_C := \nabla_\theta C(\theta). \tag{18}$$

A natural ideal correction is therefore

$$d^{\mathrm{proj}} = d - \frac{\max(0, \langle d, g_C \rangle)}{\|g_C\|^2}\, g_C, \tag{19}$$

which removes this component only when the update raises the constraint cost and leaves d unchanged otherwise.

Computing g_C at every step is expensive, so we approximate it within the span of the per-pair margin gradients. Writing $\sigma_i := \sigma(\beta \hat{r}_\theta^{(i)})$ and $v_i := \nabla_\theta \hat{r}_\theta^{(i)}$, we use the heuristic proxy

$$g_C^{\mathrm{proxy}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \beta(1 - \sigma_i)\, \Delta c_i\, v_i, \tag{20}$$

which weights each margin gradient by the cost gap of its pair and, for convenience, by its learning signal. Since $d = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i\, \beta(1 - \sigma_i)\, v_i$, replacing g_C by this proxy in Eq. (19) gives

$$d - \gamma\, g_C^{\mathrm{proxy}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \beta(1 - \sigma_i)\, \big(w_i - \gamma\, \Delta c_i\big)\, v_i, \qquad \gamma = \frac{\max\big(0, \langle d, g_C^{\mathrm{proxy}} \rangle\big)}{\|g_C^{\mathrm{proxy}}\|^2}. \tag{21}$$

The correction therefore changes the coefficient of pair i from w_i to $w_i - \gamma\, \Delta c_i$: it weakens the update on pairs with $\Delta c_i > 0$ and strengthens it on pairs with $\Delta c_i < 0$, by an amount proportional to Δc_i.

A margin shift has the same structure to first order. Since $1 - \sigma(z + \delta) \approx \big(1 - \sigma(z)\big)\big(1 - \sigma(z)\, \delta\big)$ for small δ, replacing $\beta \hat{r}_\theta^{(i)}$ by $\beta \hat{r}_\theta^{(i)} + \beta\lambda\Delta c_i$ changes the coefficient of v_i from $w_i\, \beta(1 - \sigma_i)$ to approximately $\beta(1 - \sigma_i)\big(w_i - w_i \sigma_i\, \beta\lambda\, \Delta c_i\big)$, which matches Eq. (21) with γ replaced by $w_i \sigma_i \beta\lambda$. This motivates a per-sample margin shift of the form $\lambda \Delta c_i$, which, unlike the projection, requires no constraint gradient.

C.2 Tilted reference and modified margin

We define the tilted reference by

$$\tilde{\pi}_{\mathrm{ref}}(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x) \exp(-\lambda c(x, y))}{Z(x)}, \qquad Z(x) = \sum_{y'} \pi_{\mathrm{ref}}(y' | x) \exp(-\lambda c(x, y')). \tag{22}$$

For a preference pair (x, y_w, y_l), the tilted margin is

$$\begin{aligned} \tilde{r}_\theta &= \log \frac{\pi_\theta(y_w | x)}{\tilde{\pi}_{\mathrm{ref}}(y_w | x)} - \log \frac{\pi_\theta(y_l | x)}{\tilde{\pi}_{\mathrm{ref}}(y_l | x)} \\ &= \hat{r}_\theta + \lambda\big(c(x, y_w) - c(x, y_l)\big) \\ &= \hat{r}_\theta + \lambda \Delta c, \end{aligned} \tag{23}$$

where the shared normalization term $Z(x)$ cancels because both outputs correspond to the same input x.

C.3 Gradient of the tilted loss

The tilted per-sample loss is

$$\ell_{\mathrm{tilt}} = -\log \sigma\big(\beta(\hat{r}_\theta + \lambda \Delta c)\big). \tag{24}$$

Since Δc depends only on the data and not on θ,

$$\nabla_\theta(\hat{r}_\theta + \lambda \Delta c) = \nabla_\theta \hat{r}_\theta. \tag{25}$$

Applying the chain rule gives

$$\nabla_\theta \ell_{\mathrm{tilt}} = -\beta\Big(1 - \sigma(\beta \hat{r}_\theta + \beta\lambda \Delta c)\Big) \nabla_\theta \hat{r}_\theta, \tag{26}$$

which is Eq. (6) in the main text.

C.4 Derivation of λ_{tilt}

We derive λ_{tilt} by requiring that the first-order drift of the unmodulated loss ($w \equiv 1$) vanish after tilting, under the same early-training approximations used for ρ^*_{eq}. By Proposition 1, the first-order contribution of a pair after tilting is $\beta\big(1 - \sigma(\beta\hat{r}_\theta + \beta\lambda\Delta c)\big)\Delta c\,\mathcal{I}_\theta$; treating the geometry term as approximately constant across pairs, as in Assumption (vi), we impose

$$\mathbb{E}_{\mathcal{D}}\Big[\big(1 - \sigma(\beta\hat{r}_\theta + \beta\lambda\Delta c)\big)\Delta c\Big] \approx 0. \tag{27}$$

Near initialization, $\hat{r}_\theta \approx 0$, so this becomes

$$\mathbb{E}_{\mathcal{D}}\Big[\big(1 - \sigma(\beta\lambda\Delta c)\big)\Delta c\Big] \approx 0. \tag{28}$$

For small $\beta\lambda\Delta c$, linearizing the sigmoid at the origin, $\sigma(z) \approx \frac{1}{2} + \frac{1}{4}z$, gives

$$1 - \sigma(\beta\lambda\Delta c) \approx \frac{1}{2} - \frac{1}{4}\beta\lambda\Delta c,$$

and substituting into Eq. (28) yields

$$\frac{1}{2}\,\mathbb{E}_{\mathcal{D}}[\Delta c] - \frac{1}{4}\,\beta\lambda\,\mathbb{E}_{\mathcal{D}}[\Delta c^2] \approx 0, \qquad \text{so} \qquad \lambda \approx \frac{2\,\mathbb{E}_{\mathcal{D}}[\Delta c]}{\beta\,\mathbb{E}_{\mathcal{D}}[\Delta c^2]}.$$

This value is negative when $\mathbb{E}_{\mathcal{D}}[\Delta c] < 0$, that is, when the repair from matching pairs already exceeds the pressure from conflict pairs and no correction is needed. A negative λ would lower the margin of conflict pairs and push them harder, which would increase the drift, so we clip it at zero:

$$\lambda_{\text{tilt}} = \max\left(0,\ \frac{2\,\mathbb{E}_{\mathcal{D}}[\Delta c]}{\beta\,\mathbb{E}_{\mathcal{D}}[\Delta c^2]}\right). \tag{29}$$

C.5 Role of λ_{tilt} in the adaptive update

Two considerations prevent using λ_{tilt} directly as the shift strength. First, λ_{tilt} is derived for the unmodulated loss, whereas the coarse weights already remove part of the drift, so setting $\lambda = \lambda_{\text{tilt}}$ could overcorrect. Second, it is derived at initialization, where every pair has zero margin, whereas the drift that remains later in training depends on how the margins evolve. CoDPO therefore uses λ_{tilt} only to set the scale of λ: λ starts at $\alpha\lambda_{\text{tilt}}$, increases by $\alpha\lambda_{\text{tilt}}$ whenever the constraint-risk estimate $\widehat{C}_{\mathcal{D}}$ rises between two consecutive evaluations, and is capped at λ_{tilt}. The data thus fix the scale of the shift, while its actual strength is set during training by the measured change in $\widehat{C}_{\mathcal{D}}$. We use $\alpha = 0.1$ in all experiments, so that each increase adds one tenth of λ_{tilt} and λ approaches its cap only if $\widehat{C}_{\mathcal{D}}$ rises at nearly every evaluation; Table 9 reports the resulting trajectories.

D Baseline Hyperparameter Sweeps

Tables 6 and 7 report the full sweeps of the trade-off hyperparameter of MODPO and Rewarded Soups on the four domains of the main comparison. Table 3 shows each method at the configuration with the smallest |drift|, a choice that favors drift control over preference learning; the comparison in Section 5.2 therefore uses every configuration in these sweeps. For MODPO, $w \leq 0.5$ gives higher $P(\text{chosen})$ than CoDPO but larger positive drift, while $w \geq 1.0$ moves drift below zero and lowers $P(\text{chosen})$; no value of w in this sweep gives both drift at or below that of CoDPO and higher $P(\text{chosen})$. For Rewarded Soups, α interpolates between a constraint specialist (α=0) and a helpfulness specialist (α=1), and increasing α raises drift in every domain and $P(\text{chosen})$ in nearly every case; at α=1, $P(\text{chosen})$ still remains below that of CoDPO in all four domains.

Table 6: MODPO sweep across four domains. Higher w places more weight on the constraint. Values $w \leq 0.5$ give higher $P(\text{chosen})$ than CoDPO with larger drift, and values $w \geq 1.0$ give negative drift with lower $P(\text{chosen})$.

	HelpSteer2		GSM8K		ARC		Math-Calc	
w	Drift	P(ch)	Drift	P(ch)	Drift	P(ch)	Drift	P(ch)
0.25	+76	.672	+.163	.607	+.068	.634	+.101	.791
0.5	+93	.688	+.200	.653	+.094	.668	+.129	.833
1.0	−74	.502	−.180	.365	−.158	.466	−.175	.514
2.0	−106	.427	−.303	.273	−.284	.386	−.280	.425
4.0	−77	.435	−.214	.290	−.159	.410	−.193	.421
CoDPO	+18.7	.640	+.082	.551	+.046	.622	+.066	.757

Table 7: Rewarded Soups sweep across four domains. α=1: helpfulness specialist only. α=0: constraint specialist only. At every α, $P(\text{chosen})$ is below that of CoDPO.[7]

	HelpSteer2		GSM8K		ARC		Math-Calc	
α	Drift	P(ch)	Drift	P(ch)	Drift	P(ch)	Drift	P(ch)
0.0	−79	.439	−.263	.285	−.217	.413	−.235	.432
0.25	−43	.451	−.158	.309	−.125	.428	−.136	.429
0.5	−15	.474	−.050	.353	−.052	.447	−.047	.453
0.75	+3	.509	+.017	.406	−.010	.480	+.002	.540
1.0	+22	.567	+.060	.454	+.015	.524	+.027	.647
CoDPO	+18.7	**.640**	+.082	**.551**	+.046	**.622**	+.066	**.757**

E Additional Ablations

Sharpness p. Table 8 reports the sweep over p discussed in Section 5.6. Larger p gives lower drift both with the coarse weights alone and with the margin shift added, and the margin shift lowers drift at every p. On GSM8K, p=32 likewise gives lower combined drift than p=8 (+.075 versus +.082), so p=8 is a fixed default rather than a tuned or optimal choice.

Table 8: Effect of steepness p on HelpSteer2. Drift and reduction are for the coarse weights alone, with reduction relative to the vanilla DPO drift of +59 in Table 4; the combined columns add the margin shift. p=2 was run with the coarse weights only.[9]

	Coarse weights only			Combined	
p	Drift	Reduction	$P(\text{chosen})$	Drift	$P(\text{chosen})$
1 (PSDPO form)	+46	22%	.644	+43.5	.642
2	+33	44%	.640	—	—
4	+25	58%	.640	+23.5	.639
8 (default)	+20	66%	.640	+19.5	.642
16	+18	69%	.642	+16.7	.640
32	+15	75%	.643	+13.2	.643

Adaptive λ trajectory. Table 9 reports the trajectory of λ on the four domains of the main comparison. Following the rule of Section 4.2, λ increases when $\widehat{C}_{\mathcal{D}}$ has risen since the previous evaluation and is otherwise kept unchanged. In every domain, λ is increased at most evaluations, so $\widehat{C}_{\mathcal{D}}$ keeps rising during training even with the coarse weights and the current shift in place, which is the residual drift that the fine level targets. The steps at which λ is kept unchanged differ across domains, and λ stays below λ_{tilt} in every domain, so the shift neither follows a fixed schedule nor reaches its cap.

[7] The sweep is a separate set of runs, so the α=0.75 row can differ in the last digit from Table 3.

[9] The sweep is a separate set of runs, so the p=8 row differs slightly from the CoDPO results in Tables 3 and 4.

Table 9: Trajectory of λ during training, reported at evaluations up to step 3500. $\uparrow$: λ increased because $\widehat{C}_{\mathcal{D}}$ rose since the previous evaluation; $=$: λ unchanged. The first column is the initial value $\alpha\lambda_{\text{tilt}}$, and λ_{tilt} is both the step scale and the cap. HelpSteer2 trains for fewer steps and therefore has fewer evaluation points.

	λ at step								
Domain	0	500	1000	1500	2000	2500	3000	3500	λ_{tilt}
HelpSteer2	0.62	=0.62	↑1.23	↑1.85	↑2.47	↑3.08	—	—	6.17
GSM8K	0.85	↑1.70	↑2.55	↑3.40	↑4.25	=4.25	↑5.10	=5.10	8.49
ARC	0.33	↑0.65	=0.65	↑0.98	↑1.31	=1.31	↑1.64	=1.64	3.27
Math-Calc	0.22	↑0.44	↑0.66	↑0.88	=0.88	↑1.10	↑1.33	↑1.55	2.21

F Additional Evaluations

F.1 Check of Assumption (vi)

Assumption (vi) requires the average geometry term to be comparable on conflict and matching pairs. The geometry term itself is not tractable per pair, because $\nabla_\theta C_x$ requires an expectation over the full output distribution, so we compare a per-pair proxy for the scale of the policy geometry,

$$G_\theta(x, y_w, y_l) = \tfrac{1}{2}\Big(\big\|\nabla_\theta \log \pi_\theta(y_w \mid x)\big\|^2 + \big\|\nabla_\theta \log \pi_\theta(y_l \mid x)\big\|^2\Big),$$

computed at initialization in FP32 over the LoRA parameters on the full training set of each domain. Table 10 reports the ratio $r_G = \bar{G}_{\text{conf}}/\bar{G}_{\text{match}}$ of its averages on the two subsets, with 95% confidence intervals from a paired bootstrap. The ratios lie between 0.81 and 1.10 and show no consistent direction: they fall below 1 on HelpSteer2 and GSM8K, lie above 1 on ARC, and are statistically indistinguishable from 1 on Math-Calc. GSM8K and Math-Calc share the same questions and differ only in the preference signal, yet their ratios differ (0.809 and 0.993), which suggests that the ratio reflects how the preference and constraint signals interact rather than a fixed property of the model or the data source.

To check whether these deviations from 1 matter, we replace the equal-geometry approximation in Remark 1 by the measured ratio, which divides ρ^*_{eq} by r_G (Table 11). For TL;DR, which we did not measure, we use the ratio among the measured domains that moves ρ^*_{eq} closest to the threshold, 0.809. The intervention decision is unchanged in every domain, and the smallest distance to the threshold of 1 is 0.179.

Table 10: Ratio of the geometry proxy on conflict and matching pairs at initialization, with 95% paired-bootstrap confidence intervals.

Domain	Pairs	$r_G = \bar{G}_{\text{conf}}/\bar{G}_{\text{match}}$	95% CI
HelpSteer2	7,220	0.865	[0.838, 0.894]
GSM8K	14,284	0.809	[0.782, 0.837]
ARC	12,580	1.102	[1.083, 1.121]
Math-Calc	10,387	0.993	[0.956, 1.031]

Table 11: Sensitivity of the intervention decision to the measured geometry ratio. For TL;DR, the ratio is the measured value that moves ρ^*_{eq} closest to the threshold.

Domain	ρ^*_{eq}	r_G	ρ^*_{eq}/r_G	Distance to 1
HelpSteer2	0.71	0.865	0.821	0.179
GSM8K	0.40	0.809	0.494	0.506
ARC	0.72	1.102	0.653	0.347
Math-Calc	0.80	0.993	0.806	0.194
TL;DR	0.59	0.809	0.729	0.271

F.2 Seed Variation

To estimate run-to-run variation, we repeat vanilla DPO, the coarse weights alone, and the full CoDPO on HelpSteer2 with p=8 over three seeds (Table 12). The drift reduction from vanilla DPO

to CoDPO, about 38 characters, is much larger than the standard deviation across seeds, which ranges from 1 to 3 characters, so this conclusion does not depend on the seed. With three runs, the standard deviation is itself only roughly estimated.

Table 12: Drift on HelpSteer2 (p=8) over three seeds.

Configuration	Drift (mean $\pm$ s.d., n=3)
Vanilla DPO	$+58.4 \pm 2.8$
Coarse weights only	$+22.3 \pm 1.6$
CoDPO (coarse weights and margin shift)	$+20.1 \pm 1.1$

F.3 Second Judge: GPT-4o

To check that the judge results in Section 5.5 do not depend on JudgeLM, we add GPT-4o, a judge from a different model family, on the same 7,220 HelpSteer2 prompts. Each pair of responses is judged twice with their order swapped, and a win is counted only when both orders agree; all other outcomes count as ties. The rubric asks for overall response quality and instructs the judge not to favor longer responses, since length is the constraint on HelpSteer2. Table 13 reports the results. Against vanilla DPO, the two win rates differ by about 22 of the 7,220 prompts, so the two models are comparable in generation quality; against MODPO, GPT-4o clearly prefers CoDPO, in line with JudgeLM (Table 5).

Table 13: GPT-4o pairwise evaluation on HelpSteer2 (7,220 prompts, position-debiased, length-neutral rubric).

Comparison	CoDPO wins	Opponent wins	Tie
CoDPO vs. vanilla DPO	6.9%	6.6%	86.5%
CoDPO vs. MODPO	44.9%	3.9%	51.2%

G Limitations and Broader Impacts

The diagnostic ρ^*_{eq} rests on a first-order analysis at initialization and on the assumption that the average geometry term is comparable across conflict and matching pairs, which we check through a proxy on four domains (Appendix F.1) rather than measure exactly; it lies within 0.05 to 0.30 of the observed balance point, and the margin shift reduces but does not remove the drift that this gap leaves. Our evaluation covers 7–8B models and one pointwise constraint per domain, with judge-based evaluation on HelpSteer2 only, so scaling to larger models and to multiple constraints remains to be tested. CoDPO is meant to keep preference-aligned models within a specified constraint such as length, correctness, or safety; because it follows the constraint it is given, a misspecified constraint would be reinforced rather than corrected.

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 introduction state three contributions: (1) a formal characterization of constraint drift in DPO under heterogeneous supervision (Theorem 1), (2) a closed-form pre-alignment diagnostic ρ^*_{eq}, and (3) the CoDPO method combining coarse modulation with adaptive margin shift (Section 4). All three are supported by the experimental results in Section 5 across conflict-dominated domains and a self-correcting control. Method comparison against baselines is conducted on a representative subset of conflict-dominated domains, while robustness checks across multiple base-model architectures and preference losses are conducted on HelpSteer2 as a representative domain.

   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 in Appendix G: the diagnostic rests on a first-order analysis at initialization and an assumption on the geometry term that we check through a proxy rather than measure exactly, and our evaluation covers 7–8B models and one constraint per domain.

   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: Theorem 1 (constraint drift) and Theorem 2 (drift attenuation) are stated with references to their assumptions, which are listed in full in Appendix B.1, including Assumption (v), that more severe conflicts exert more pressure, and Assumption (vi), that conflict and matching pairs have similar geometry. Complete proofs are given in Appendix B, and the derivation of ρ^*_{eq} is in Appendix B.5.

   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 and Appendix fully document datasets, base models, hyperparameters (β=0.1, p=8, K=500, α=0.1, LoRA rank 16, learning rate 5×10^{-5}), training duration, and evaluation protocol. Algorithm 1 provides the complete training procedure. All datasets used (HelpSteer2, GSM8K, ARC, TL;DR, PKU-SafeRLHF) are publicly available, and Math-Calc is constructed from GSM8K via the procedure described in Section 5.

   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: [No]

   Justification: Code and configuration files will be released when ready. All datasets used are publicly available through their original sources cited in Section 5.

   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: Training and evaluation details are in Section 5 and Appendix, including base model selection, LoRA configuration, optimizer settings, training schedule, hyperparameter choices, baseline configurations, and evaluation metrics. Hyperparameter sweeps for MODPO and Rewarded Soups are reported in Appendix.

   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: Due to compute constraints, primary results are reported from single runs per configuration. A three-seed repetition of the main configurations on HelpSteer2 (Appendix F.2) gives standard deviations of 1 to 3 characters of drift, well below the reduction of about 38 characters from vanilla DPO to CoDPO. The LLM-as-judge evaluation uses position-debiased scoring on the full 7,220 HelpSteer2 prompts, providing large-sample win-rate estimates.

   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: The GPU type, total compute, and the number of training epochs per domain are given in Appendix. All experiments use NVIDIA RTX PRO 6000 GPUs with LoRA fine-tuning.

   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: The research conforms with the NeurIPS Code of Ethics. All datasets are publicly available with appropriate licenses, no human subjects research was conducted by the authors, and the work focuses on technical alignment methods rather than deployment of generative systems.

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: [Yes]

Justification: Broader impacts are discussed in Appendix G. CoDPO is designed to improve constraint adherence in preference-aligned models; because it follows the constraint it is given, a misspecified constraint would be reinforced rather than corrected.

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 proposes a training algorithm rather than releasing pre-trained models or datasets. No new high-risk assets are introduced.

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 datasets and base models used are properly cited in the submission. Each dataset is used under its original license, and base models are used under their respective open-source licenses.

    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: [N/A]

    Justification: No new datasets or pre-trained models are released. Math-Calc is a derived preference signal constructed from GSM8K, with construction details in Section 5 and Appendix.

    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 preference data is sourced from existing publicly available datasets.

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 research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are the subject of study (preference optimization of language models) but are not used as a non-standard component of the core method development. The LLM-as-judge evaluation uses two standard judges, JudgeLM-13B and GPT-4o, following a well-established evaluation protocol.

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.