*Compass*DPO : Dynamics-Controlled Direct Preference Optimization for Robust Safety Alignment

Anonymous ACL submission

Abstract

Direct Preference Optimization (DPO) has become a standard framework for safety alignment, but its reliance on pairwise preference updates makes training sensitive to imperfect supervision. Existing robust DPO methods often address this sensitivity through global loss corrections or external data-level interventions, while largely overlooking how unreliable comparisons distort batch-level optimization dynamics. We propose *Compass*DPO , a reward-free DPO framework that stabilizes preference optimization through dynamics control. Using the implicit DPO reward margin as a training-time compass, *Compass*DPO regulates sample influence along two complementary axes: update direction and update magnitude. For directional control, it applies sparse, budgeted, and warm-up delayed loss mixing to attenuate update components that conflict with the emerging preference direction. For magnitude control, it adaptively soft-winsorizes high-loss tail contributions, reducing tail dominance while preserving useful gradients from hard examples. Both mechanisms use only signals available during standard DPO training and require no external reward model or additional supervision. Experiments on PKU-SafeRLHF across four backbones and multiple out-of-distribution safety benchmarks show that *Compass*DPO consistently improves robustness over vanilla DPO and strong DPO-family baselines, especially under controlled label-flip noise. Code is available at `https://anonymous.4open.science/r/CompassDPO-4D00`.

1 Introduction

Aligning large language models (LLMs) with human preferences has become a standard route to building capable and safe AI systems (Yang et al., 2025; Achiam et al., 2023; Liu et al., 2024; Team et al., 2023). Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022; Bai et al., 2022a,b) established this paradigm, but its reliance on reward modeling and PPO-style policy optimization (Schulman et al., 2017) makes training costly and sensitive to implementation details. Direct Preference Optimization (DPO) (Rafailov et al., 2023) has therefore become a widely used alternative, which aligning policies by directly optimizing pairwise preference comparisons.

However, DPO remains sensitive to imperfect preference supervision. In real-world preference datasets (Bai et al., 2022a; Ji et al., 2023; Ganguli et al., 2022; Shen et al., 2024), unreliable pairs can skew mini-batch updates, affecting both their direction and magnitude. Existing robust DPO methods mainly address this fragility through global objective modifications, such as softened margins (Mitchell, 2023), assumed flip-rate correction (Chowdhury et al., 2024), or conservative distributional optimization (Wu et al., 2025); others introduce external scoring signals to revise preference supervision (Zhang et al., 2025). These approaches can improve robustness, but they do not explicitly control how individual samples shape the batch-level training dynamics of DPO.

This missing control is critical because heterogeneous preference signals can shape DPO through the mini-batch update itself. In direct preference optimization, supervision affects not only which preference direction is learned, but also how each update is formed. As shown in Figure 1 (left), the reward-model gap provides an intuitive view of pair-level preference consistency, highlighting preference-conflict, low-confidence, and reliable-update regions. This gap is used only for visualization; *Compass*DPO uses the implicit DPO margin during training. These regimes matter for training because preference-conflict and low-confidence pairs can introduce update components that are misaligned with, or only weakly aligned to, the expected preference direction. As illustrated in Fig-

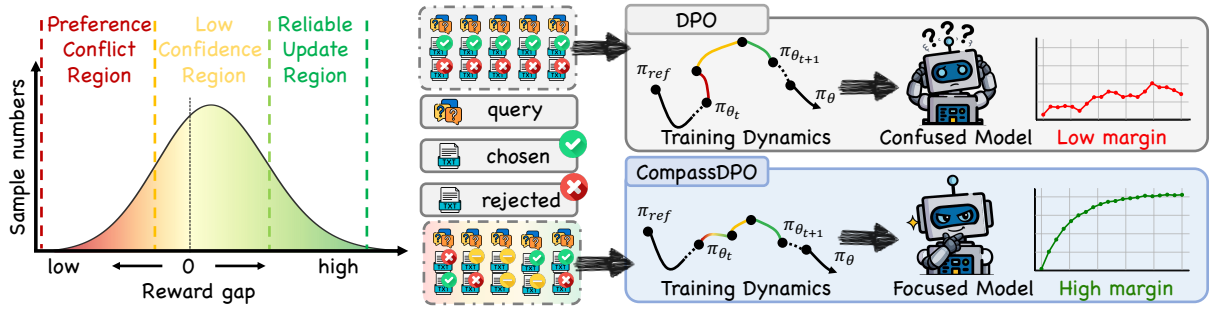

Figure 1: Training-dynamics view of DPO under imperfect preference supervision. Left: A reward-model gap $\Delta r = r_{\mathrm{RM}}(y_w \mid x) - r_{\mathrm{RM}}(y_l \mid x)$ visualizes three pair-level regimes: preference conflict, low confidence, and reliable update. The reward model is used only for visualization. Right: Standard DPO updates can be dominated by unreliable or high-loss pairs, leading to a distorted optimization trajectory. *Compass*DPO bounds sample influence along update direction and magnitude, resulting in more stable preference optimization.

ure 1 (right), standard DPO averages all pair contributions uniformly, allowing such components to perturb the optimization trajectory. This motivates update-level control over both direction and magnitude.

Motivated by this dynamics-level view, we propose ***Compass*DPO**, a margin-guided dynamics-control method for DPO. *Compass*DPO uses the implicit DPO margin as a training-time state signal and regulates each mini-batch update along two axes: direction and magnitude. For directional control, *Compass*DPO applies sparse and budgeted loss mixing after a warm-up period, reducing update components that conflict with the emerging preference direction. For magnitude control, *Compass*DPO uses adaptive soft winsorization to limit excessive tail influence while retaining gradients from hard preference pairs. Both controls are computed from quantities already available in standard DPO training and require no reward model, relabeling, or data reconstruction.

Together, these two controls stabilize DPO during optimization. Directional control reduces conflicting update components, while magnitude control prevents high-loss tail samples from dominating the batch update. This preserves the simplicity of standard DPO while making training less sensitive to heterogeneous preference signals. Experiments on PKU-SafeRLHF and multiple external safety benchmarks show that *Compass*DPO consistently outperforms vanilla DPO and strong DPO-family baselines, especially under controlled label-flip noise. Our main contributions are summarized as follows:

- We identify gradient dominance as a batch-level failure mode of DPO under imperfect preference supervision, where gradient energy concentrates on a small subset of samples and distorts preference optimization.
- We propose *Compass*DPO, a dynamics-controlled DPO method that uses the implicit DPO margin to regulate mini-batch updates through directional control and magnitude control, without reward models, relabeling, or data reconstruction.
- We evaluate *Compass*DPO across four backbones and multiple IID and OOD safety benchmarks, showing consistent improvements over DPO-family baselines and stronger robustness under controlled label-flip corruption.

2 Preliminaries

LLM alignment is commonly formulated as a preference-based learning problem, where supervision is provided through pairwise comparisons rather than explicit target outputs. Given a prompt x, a preference pair is denoted as (x, y_w, y_l), where y_w is preferred over y_l. While Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) first trains a reward model and then optimizes a policy with reinforcement learning, DPO provides a simpler alternative by directly optimizing the policy on preference pairs.

Given a policy π_θ and a fixed reference policy π_{ref}, DPO defines the implicit preference margin $s(x, y_w, y_l)$, abbreviated as s:

$$s = \beta \left[\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} - \log \frac{\pi_{\mathrm{ref}}(y_w \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} \right], \quad (1)$$

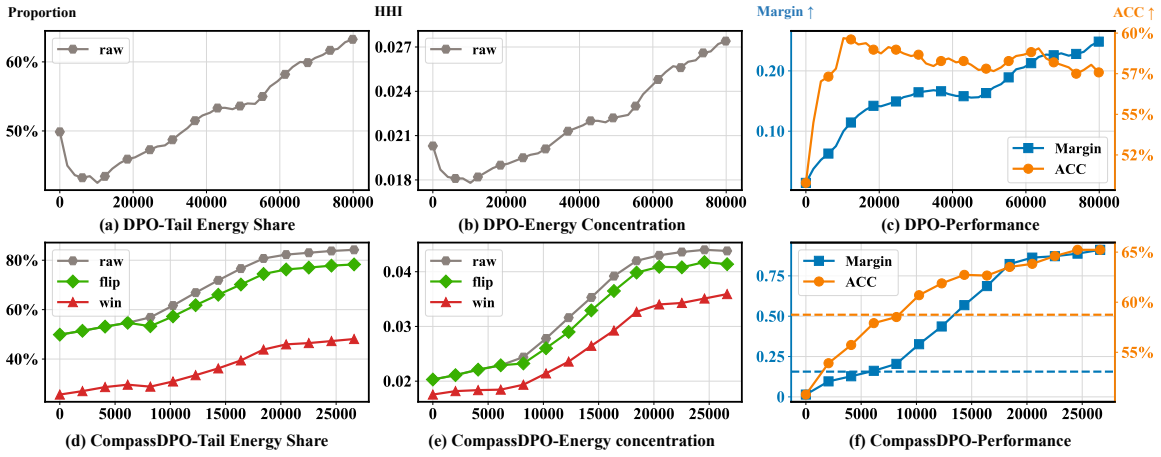

Figure 2: Diagnostic training dynamics of DPO and *Compass*DPO on Pythia-2.8B. DPO is run for 3 epochs only to expose the evolution of gradient dominance, while *Compass*DPO is shown under its default 1-epoch training setup. (a,d) Tail energy share: the fraction of gradient energy contributed by the top-energy samples within each mini-batch. (b,e) Energy concentration measured by the Herfindahl–Hirschman Index (HHI). (c,f) Preference separation during training, measured by DPO margin and preference accuracy.

where β controls the strength of the KL regularization implicit in DPO. The training objective minimizes the logistic loss over this margin:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)}\left[\log \sigma(s(x, y_w, y_l))\right]. \tag{2}$$

This direct supervised objective makes DPO simple and efficient, but also makes its training dynamics sensitive to imperfect preference supervision. Since each preference pair contributes directly to the batch-level update, a small number of high-influence samples can disproportionately affect the optimization trajectory. In the next section, we introduce a reward-free dynamics-control strategy to bound such sample influence during DPO training.

3 The Proposed *Compass*DPO

In this section, we present *Compass*DPO, a dynamics-controlled preference optimization method for DPO. The goal is to regulate how samples contribute to each mini-batch update, rather than applying a uniform robustness correction to all preference pairs. We first diagnose a batch-level failure mode, termed *gradient dominance*, where the update can become concentrated on a small subset of high-influence samples. This diagnosis motivates *Compass*DPO, which controls sample influence along two complementary axes: directional control and magnitude control.

3.1 Gradient Dominance as a Failure Mode

Figure 2 diagnoses how sample influence evolves during DPO training under imperfect preference supervision. The goal is not to compare final performance, but to examine whether mini-batch updates become dominated by a small subset of samples. We track three signals: (i) tail energy share, the fraction of gradient energy contributed by the top 30% energy samples within each mini-batch; (ii) the Herfindahl–Hirschman Index (HHI) (Rhoades, 1993), which measures gradient-energy concentration across samples; and (iii) validation margin and preference accuracy, which indicate the stability of preference separation. Formal definitions are provided in Appendix F.

Figures 2(a–c) show the dynamics of standard DPO. Although the gradient energy is relatively diffuse early in training, it becomes increasingly concentrated as optimization proceeds. This is reflected by the rising tail energy share and HHI: a small fraction of samples eventually accounts for a large portion of the batch gradient energy. At the same time, validation margin and preference accuracy stagnate or degrade, suggesting that concentrated update influence does not necessarily improve preference separation. These observations indicate a batch-level failure mode of DPO, where a small subset of high-influence samples can disproportionately shape the optimization trajectory.

Figures 2(d–f) show the corresponding behavior under *Compass*DPO. Compared with standard DPO, *Compass*DPO keeps gradient energy less concentrated and maintains more stable validation signals. This supports the motivation for controlling sample influence during training. Directional

control reduces update components that conflict with the current preference-separating direction, while magnitude control limits tail-dominated update contributions through adaptive soft winsorization. Together, the two controls mitigate gradient dominance without introducing external reward models or modifying the preference data.

Overall, this diagnostic analysis shows that the robustness issue in DPO can arise from the dynamics of mini-batch optimization itself. *Compass*DPO addresses this issue by bounding sample influence along both direction and magnitude, which motivates the mechanisms introduced next.

3.2 Directional Control of Conflicting Updates

After the policy has formed a preliminary preference direction, pairs with strongly negative margins may introduce update components that conflict with the current preference-separating direction. Directional control mitigates this distortion by softly mixing the observed DPO loss with its reversed-direction counterpart for a sparse set of high-gain pairs. The intervention is conservative: most samples receive no mixing, and the total mixing strength is bounded within each batch.

For each preference pair $(x_i, y_{w,i}, y_{l,i})$, let $s_i = s(x_i, y_{w,i}, y_{l,i})$ denote the DPO margin. We consider two possible preference directions: the observed direction $\rightarrow$ corresponding to $(y_{w,i}, y_{l,i})$, and the swapped direction $\leftarrow$ corresponding to $(y_{l,i}, y_{w,i})$. This yields two logistic losses:

$$\begin{aligned} \ell_{\rightarrow}(x_i) &= -\log \sigma(s_i), \\ \ell_{\leftarrow}(x_i) &= -\log \sigma(-s_i). \end{aligned} \tag{3}$$

When the reversed-direction loss is much smaller than the observed-direction loss, the observed update may be poorly aligned with the current preference-separating direction. Directional control therefore introduces a mixing weight $w_i \in [0, 1]$ for each sample and forms a mixed loss:

$$\tilde{\ell}_i = (1 - w_i)\, \ell_{\rightarrow}(x_i) + w_i\, \ell_{\leftarrow}(x_i). \tag{4}$$

Most samples retain $w_i = 0$, while only a small number of high-gain pairs receive non-zero mixing weights. To keep this intervention conservative, we impose a batch-level mixing budget controlled by a single hyperparameter ρ_f:

$$\frac{1}{B} \sum_{i=1}^{B} w_i \leq \rho_f. \tag{5}$$

The mixing weights are computed from batch-level signals. Using the two-direction losses in Eq. (3), we measure the gain from swapping direction as

$$g_i = \ell_{\rightarrow}(x_i) - \ell_{\leftarrow}(x_i), \tag{6}$$

and retain only beneficial gains $u_i = \mathrm{ReLU}(g_i)$. We apply sparsemax (Martins and Astudillo, 2016) to the batch score vector $\mathbf{u} = (u_1, \ldots, u_B)$ to obtain sparse allocation weights $\mathbf{a}$ with $\sum_i a_i = 1$, so that only a few pairs with large gains are selected. These allocations are scaled by the budget as $\hat{w}_i = \rho_f B\, a_i$ and projected onto $[0, 1]$ while satisfying the total budget, which yields the final correction weights $\{w_i\}$.

Since margin signals can be unstable early in training, directional control is activated only after a warm-up period. Let t be the current training step, T the total number of steps, and $\alpha \in (0, 1)$ a fixed warm-up ratio. If $t < \alpha T$, we set $w_i = 0$ for all samples; otherwise the correction weights are computed as above. We use $\alpha = 0.3$ in experiments.

3.3 Magnitude Control of Tail-Dominated Updates

After conflicting update components are reduced by directional control, the mini-batch update can still be dominated by high-loss tail samples. Such tail samples are not always harmful: many may correspond to hard preference pairs that still provide useful learning signals. Magnitude control therefore does not discard them. Instead, it softly winsorizes the excessive part of tail losses, limiting their influence while preserving their gradients.

Magnitude control operates on the per-sample mixed losses $\{\tilde{\ell}_i\}_{i=1}^{B}$ produced by directional control. We first define a batch-dependent threshold τ for the high-loss tail, set as the q-quantile of the batch losses:

$$\tau = \mathrm{Quantile}\big(\{\tilde{\ell}_i\}_{i=1}^{B},\, q\big). \tag{7}$$

Samples with $\tilde{\ell}_i > \tau$ form the tail set. The capping strength is adapted to the current batch using margin statistics. Intuitively, when a batch contains more negative or weakly separated margins, preference separation is less stable, and stronger magnitude control is applied. We compute a batch-level capping strength ρ_w from these margin statistics and treat it as a constant during backpropagation. This strength controls the maximum average amount of soft capping applied within the batch.

Within the tail, capping is allocated according to how much each loss exceeds the threshold. Formally, tail losses are softly pulled toward τ using per-sample cap weights $\lambda_i \in [0, 1]$ under the batch-level capping budget:

$$\ell_i^{\text{win}} = (1 - \lambda_i)\, \tilde{\ell}_i + \lambda_i\, \tau. \tag{8}$$

The final objective is the average winsorized loss:

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \ell_i^{\text{win}}. \tag{9}$$

The full specification of how ρ_w and λ_i are computed is provided in Appendix C. The overall implementation is summarized in Algorithm 1.

4 Experiments

In this section, we evaluate whether dynamics control improves preference alignment by stabilizing batch-level updates. We study both final alignment performance and the roles of directional and magnitude control. Specifically, we investigate the following research questions:

- **RQ1**: Does *Compass*DPO improve alignment under standard training conditions?
- **RQ2**: Is *Compass*DPO robust to wrong-direction preference supervision?
- **RQ3**: How sensitive is *Compass*DPO to the hyperparameters controlling correction strength and tail influence?
- **RQ4**: How do directional and magnitude control contribute to the performance?

4.1 Experimental Settings

4.1.1 Dataset

Following prior safety alignment work, we run experiments on PKU-SafeRLHF-30K (Ji et al., 2023) from PKU-Alignment. It contains about 29.9K paired comparisons, with 26.9K for training and 3.0K for testing. Each example consists of a prompt and two responses. The dataset provides preference labels for helpfulness and safety, and also includes a binary safety tag for each response. In our experiments, we use the safety preference labels to construct the chosen–rejected pairs.

4.1.2 Baselines

We evaluate *Compass*DPO against a set of preference alignment baselines. We include the Vanilla pretrained backbone, SFT (Ouyang et al., 2022), and DPO (Rafailov et al., 2023). We also include DPO variants that modify the preference objective or improve robustness under imperfect supervision, including IPO (Gheshlaghi Azar et al., 2023), cDPO (Mitchell, 2023), rDPO (Chowdhury et al., 2024), and Dr.DPO (Wu et al., 2025).

4.1.3 Backbones

We study *Compass*DPO on four base LLMs that have not been safety-aligned via RLHF: Pythia-2.8B (Biderman et al., 2023), Llama-3.2-3B, Llama-3-8B (AI@Meta, 2024), and Qwen2.5-7B (Team, 2024). These backbones span multiple model families and scales.

4.1.4 Benchmarks and Evaluation Metrics

We evaluate on PKU-SafeRLHF-30K (test split), Do-Not-Answer (Wang et al., 2023), HarmBench (Mazeika et al., 2024), HH-RLHF (Bai et al., 2022a), and Salad Bench (Li et al., 2024). On the external safety benchmarks, we report Attack Success Rate (ASR; lower is better) as the primary metric. We compute ASR using two widely adopted safety judges: MD-Judge (MD) (Li et al., 2024) and Nemotron Safety Guard (NV) (Joshi et al., 2025). We additionally evaluate general capability and over-refusal behavior in Appendix E.3, using MMLU-Pro (Wang et al., 2024), MT-Bench (Zheng et al., 2023), and OR-Bench (Cui et al., 2024). In addition, on PKU-SafeRLHF we report Win Rate (WR; higher is better) based on pairwise judgments from the GPT-4.1 mini API.

Appendix D provides full details on baseline configurations, backbones, benchmark setup, and evaluation protocols.

4.2 Main Results (RQ1)

We evaluate *Compass*DPO under both in-distribution (IID) and out-of-distribution (OOD) settings. The results show that controlling mini-batch training dynamics consistently improves safety alignment across model families and evaluation benchmarks.

In-Distribution Performance. Table 1 reports results on the PKU-SafeRLHF-30K test split. Across four backbones, *Compass*DPO achieves the strongest overall performance among DPO-family methods, improving win rate while reducing ASR

Table 1: Results on PKU-SafeRLHF-30K (test split) across four backbones. Methods include *Compass*DPO and DPO-family baselines. We report WR (%) and ASR (%), where ASR is measured by two judges (MD and NV).

Methods	**Pythia-2.8B**			**Llama-3.2-3B**			**Llama-3-8B**			**Qwen2.5-7B**		
	WR↑	MD↓	NV↓	WR↑	MD↓	NV↓	WR↑	MD↓	NV↓	WR↑	MD↓	NV↓
Vanilla	40.18	73.31	26.94	44.93	66.79	18.42	42.82	67.54	24.69	69.09	32.21	13.16
SFT	41.05	71.30	26.32	35.93	68.67	45.36	23.85	69.92	52.51	25.09	70.55	52.88
DPO	44.23	57.14	19.17	73.70	11.28	6.14	69.19	26.57	14.41	75.81	20.55	7.27
IPO	45.27	61.65	22.68	72.70	11.78	6.14	70.22	26.82	13.53	73.97	21.30	8.02
cDPO	45.27	60.28	16.92	72.91	25.44	12.78	59.48	37.94	23.43	67.65	31.83	13.66
rDPO	46.54	57.77	23.43	81.30	4.14	1.75	82.13	6.64	3.26	82.74	4.51	2.01
Dr.DPO	57.41	17.04	4.14	83.77	2.38	0.50	82.50	**4.39**	0.50	88.93	2.01	0.13
***Compass*DPO**	**59.92**	**13.03**	**3.01**	**85.45**	**1.63**	**0.13**	**84.01**	4.51	**0.38**	**90.23**	**1.63**	**0.00**

Table 2: Out-of-distribution ASR on four external safety benchmarks for models trained on PKU-SafeRLHF-30K. Results are shown for Llama-3-8B and Qwen2.5-7B across *Compass*DPO and DPO-family baselines. ASR is measured by two judges, and AVG reports the mean ASR over the four benchmarks.

Methods	**Do-Not-Answer**		**HarmBench**		**HH-RLHF**		**Salad Bench**		**AVG.**
	MD↓	NV↓	MD↓	NV↓	MD↓	NV↓	MD↓	NV↓	
Llama-3-8B	54.58	12.37	87.50	23.50	65.10	19.92	73.14	23.26	46.27
SFT	53.52	33.48	93.50	81.50	69.27	47.54	75.10	58.68	64.03
DPO	14.18	7.78	41.00	28.50	23.05	10.30	31.55	19.75	22.85
IPO	14.29	7.36	33.50	17.50	21.70	8.30	30.12	15.69	20.42
cDPO	22.39	13.86	61.00	44.00	30.32	13.34	42.53	26.44	30.7
rDPO	1.17	0.53	6.00	1.50	6.88	2.25	5.38	2.38	3.97
Dr.DPO	**0.96**	**0.00**	2.00	**0.00**	4.58	**0.37**	3.52	0.51	2.09
***Compass*DPO**	1.07	0.11	**1.5**	**0.00**	**4.17**	0.39	**3.33**	**0.45**	**1.95**
Qwen-2.5-7B	21.11	7.78	44.50	27.00	32.52	10.76	33.85	13.77	23.02
SFT	52.67	31.56	95.00	74.00	70.54	48.01	73.45	55.05	62.37
DPO	10.98	4.16	32.50	14.00	17.70	4.90	22.97	8.63	14.39
IPO	10.34	3.41	30.50	14.00	18.02	5.03	22.16	8.24	14.00
cDPO	16.42	5.76	45.50	19.50	26.13	8.59	32.56	13.66	21.26
rDPO	1.17	0.32	3.00	1.00	6.14	1.76	4.23	1.76	3.18
Dr.DPO	**0.21**	**0.00**	0.50	**0.00**	2.39	0.22	1.53	0.46	1.05
***Compass*DPO**	**0.21**	**0.00**	**0.00**	**0.00**	**1.64**	**0.15**	**1.04**	**0.10**	**0.64**

under both safety judges. This indicates that the learned policy is not only preferred by the pairwise safety judge, but also produces fewer unsafe responses under independent safety classifiers. Although robust baselines improve over vanilla DPO in several settings, their gains are less consistent across models and metrics. This suggests that global robustness corrections alone may not fully address the batch-level instability of DPO. By regulating how samples contribute to each mini-batch update, *Compass*DPO reduces conflicting update influence and tail-dominated training signals. This leads to a more stable alignment process and better in-distribution safety performance.

Out-of-Distribution Generalization. Table 2 summarizes results on Do-Not-Answer, HarmBench, HH-RLHF, and Salad Bench. On average across external benchmarks and judges, *Compass*DPO achieves the lowest ASR among DPO-family methods, while remaining competitive on most individual benchmark-judge combinations.

External safety benchmarks differ from PKU-SafeRLHF in prompt style and harmfulness coverage. For DPO, unstable or conflicting updates from the training data can bias the learned safety behavior toward dataset-specific patterns. *Compass*DPO reduces this effect by attenuating updates that conflict with the emerging safety-preference direction and by limiting tail-dominated batch updates. This makes the aligned policy less sensitive to unstable signals and yields more consistent safety performance across external benchmarks.

Table 3: Results on PKU-SafeRLHF-30K (test split) under synthetic label-flip noise for Pythia-2.8B. A fixed fraction of training preference pairs is randomly flipped (0/10/20/30%).

Methods	**0 Flipped**			**10 Flipped**			**20 Flipped**			**30 Flipped**		
	WR↑	MD↓	NV↓	WR↑	MD↓	NV↓	WR↑	MD↓	NV↓	WR↑	MD↓	NV↓
DPO	44.23	57.14	19.17	45.07	60.78	17.42	43.16	66.29	23.93	41.82	69.05	25.19
rDPO	46.54	57.77	23.43	46.3	59.02	20.55	44.10	60.90	21.43	42.96	65.65	22.43
cDPO	45.27	60.28	16.92	44.43	66.17	23.18	43.86	67.04	20.18	42.72	71.3	22.68
IPO	45.27	61.65	22.68	43.61	62.53	23.81	43.36	65.91	23.18	42.23	68.80	24.68
Dr.DPO	57.41	17.04	4.14	56.57	19.92	7.52	54.81	24.56	7.77	52.93	31.95	6.39
***Compass*DPO**	**59.92**	**13.03**	**3.01**	**58.18**	**17.04**	**3.63**	**56.51**	**18.92**	**4.89**	**55.14**	**22.06**	**4.14**

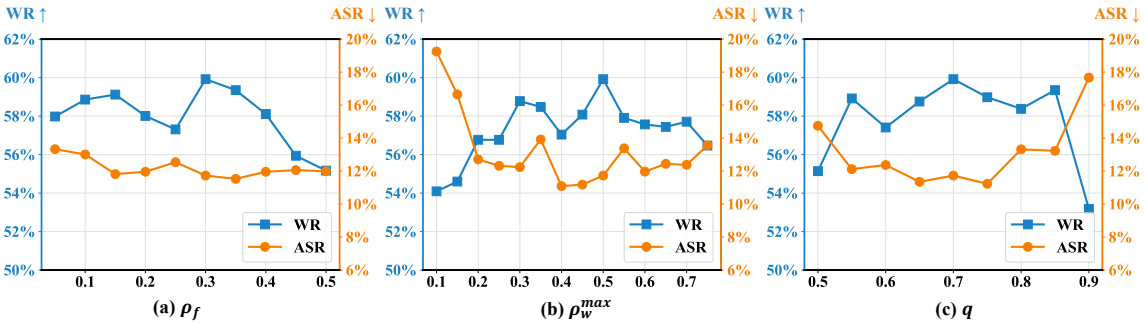

Figure 3: Hyperparameter sensitivity on Pythia-2.8B, evaluated on PKU-SafeRLHF-30K (test split). Each panel varies one hyperparameter while keeping all others fixed to the default settings in the paper: (a) directional-control budget ρ_f, (b) cagnitude-control strength $\rho_w^{\max}$, and (c) tail threshold quantile q.

4.3 Robustness under Label-Flip Noise (RQ2)

We further evaluate robustness under controlled label-flip noise. A fixed fraction of preference labels in the PKU-SafeRLHF-30K training set is randomly flipped, and all methods are evaluated on the clean test split. We consider flip rates of 0%, 10%, 20%, and 30%, with results reported in Table 3. As the flip rate increases, all methods degrade, but the degradation differs substantially. Vanilla DPO is highly sensitive to flipped labels because it directly follows the observed preference direction. Robust baselines such as cDPO and rDPO slow this degradation through global corrections, but their performance still drops noticeably at higher flip rates. This suggests that global corrections alone are insufficient when corrupted labels directly perturb the mini-batch update.

Dr.DPO is the strongest baseline in this setting, yet *Compass*DPO achieves the best overall performance across all flip rates. The advantage remains clear at 30% flips, where *Compass*DPO improves both WR and ASR over the baselines. This result is consistent with the design of *Compass*DPO: directional control reduces update components that conflict with the current margin signal, while magnitude control limits tail-dominated contributions as noise increases. By regulating both the direction and concentration of mini-batch updates, *Compass*DPO degrades more gracefully under label-flip corruption.

4.4 Hyper-parameter Analysis (RQ3)

We analyze the sensitivity of *Compass*DPO to its key hyperparameters to assess whether the dynamics-control design requires careful tuning. All experiments are conducted on Pythia-2.8B. When varying one hyperparameter, all others are fixed to their default values reported in Appendix D.4. Figure 3 summarizes the results.

- **Directional-control budget ρ_f.** The parameter ρ_f controls the maximum average strength of directional mixing within a batch. As shown in Figure 3, a small budget ($\rho_f = 0.05$) already yields noticeable improvement, suggesting that conflicting update components are relatively sparse. Performance remains stable over a wide range of values from 0.05 to 0.4, indicating that directional control is not sensitive to the exact budget as long as mixing is applied conservatively. When ρ_f becomes too large, performance degrades. In this regime, directional mixing may affect samples whose margins reflect difficulty or near-ties rather than conflict with the preference direction,

weakening useful preference signals. Based on this trade-off, we set $\rho_f = 0.3$.

- **Magnitude-control strength $\rho_w^{\max}$.** The parameter $\rho_w^{\max}$ determines the upper bound on loss-capping strength in magnitude control, while the actual capping level is set adaptively per batch. As $\rho_w^{\max}$ increases from small values, performance improves and then stabilizes across a broad interval. This behavior aligns with the goal of soft winsorization, which limits excessive tail influence while leaving most samples unaffected. When $\rho_w^{\max}$ becomes too large, performance drops, as overly strong capping can suppress useful gradients from informative hard preference pairs. We therefore set $\rho_w^{\max} = 0.5$.

- **Tail threshold quantile q.** The quantile q determines where magnitude control starts to act on the loss distribution. When q is too small, capping is applied to too many samples and suppresses the main training signal, leading to degraded performance. As q increases, performance improves and remains stable over a moderate range, where magnitude control focuses on the high-loss part of the batch. When q becomes too large, the controlled region becomes too narrow and misses part of the tail-dominated update, reducing the effectiveness of capping. Thus, we set $q = 0.7$.

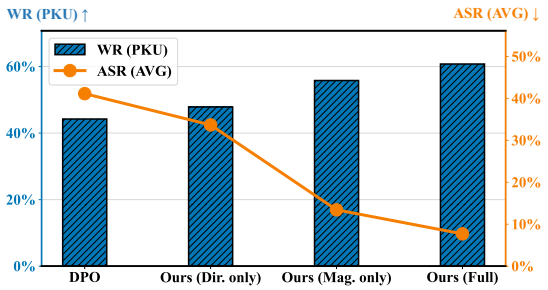

Figure 4: Ablation results with Pythia-2.8B. Bars show WR on the PKU-SafeRLHF-30K test split. The line shows average ASR across five benchmarks, measured by the two judges. We compare standard DPO, *Compass*DPO with directional control only (Dir. only), magnitude control only (Mag. only), and the full method.

Overall, *Compass*DPO exhibits stable performance across broad ranges of its key hyperparameters. The best results consistently arise from moderate control strengths for both mechanisms, supporting the design choice of conservative, sample-adaptive dynamics control rather than aggressive global objective modifications.

4.5 Ablation Study (RQ4)

We conduct an ablation study on Pythia-2.8B to examine the contribution of each control mechanism in *Compass*DPO. Figure 4 summarizes the results. Both mechanisms improve over standard DPO when used alone, but they affect performance in different ways.

The gain from directional control is moderate, suggesting that directional conflicts explain only part of the instability observed in DPO training. Magnitude control yields larger improvements on its own. By softly capping the excessive part of the loss tail, it reduces gradient dominance from high-loss preference pairs while retaining useful hard-example gradients. This improves both WR and ASR, suggesting that controlling tail-dominated update magnitude is important for stable preference optimization.

Combining the two mechanisms gives the best performance. Directional control reduces conflicting update components, while magnitude control limits the remaining tail-dominated influence. The ablation confirms that the two mechanisms are complementary rather than redundant. Together, they stabilize mini-batch optimization by controlling both update direction and update magnitude.

5 Conclusion

DPO aggregates preference-pair updates within each mini-batch, making its training dynamics sensitive to how individual samples contribute to the update. Under imperfect preference supervision, this aggregation can lead to gradient dominance, where a small subset of conflicting or high-loss samples disproportionately shapes the optimization trajectory. We propose *Compass*DPO, a dynamics-controlled preference optimization method that stabilizes DPO by regulating mini-batch updates along two axes: direction and magnitude. *Compass*DPO uses the implicit DPO margin for sparse directional mixing and adaptive soft winsorization, reducing conflicting update components and limiting tail-dominated update magnitudes without changing the standard DPO training setup. Experiments across standard and label-flip settings show that *Compass*DPO consistently improves robustness over vanilla DPO and strong DPO-family baselines. More broadly, our results suggest that robust preference optimization can benefit from explicit control over how samples influence the direction and magnitude of mini-batch updates.

Limitations

Our experiments mainly focus on safety alignment with PKU-SafeRLHF, several external safety benchmarks, and one helpfulness-oriented setting. Further experiments are needed to evaluate *Compass*DPO across broader preference-learning domains and languages.

*Compass*DPO relies on batch-level margin and loss-tail statistics. Although it is stable across the tested batch sizes, very small batches may provide less reliable estimates. Directional control also requires a warm-up period because early margin signals can be unstable.

Finally, *Compass*DPO does not specifically target over-refusal. Future work could combine dynamics-controlled preference optimization with data or objectives that better cover benign but challenging queries.

Ethics Statement

This work uses publicly available preference and safety evaluation datasets. We do not collect new human-subject data or attempt to identify individuals from the datasets. Some datasets used in this work contain harmful, offensive, or unsafe prompts by design, since they are intended for safety alignment and robustness evaluation. We use these data only for model training and aggregate evaluation, and report benchmark-level metrics rather than reproducing harmful examples. For all external datasets, benchmarks, models, and judges, we cite the original creators and follow their intended research use.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9954–9972.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022b. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*.

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

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, and 1 others. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv e-prints*, pages arXiv–2310.

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

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704.

Raviraj Joshi, Rakesh Paul, Kanishk Singla, Anusha Kamath, Michael Evans, Katherine Luna, Shaona Ghosh, Utkarsh Vaidya, Eileen Long, Sanjay Singh Chauhan, and 1 others. 2025. Cultureguard: Towards culturally-aware dataset and guard model for multilingual safety applications. *arXiv preprint arXiv:2508.01710*.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, and 1 others. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*.

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

Eric Mitchell. 2023. A note on dpo with noisy preferences & relationship to ipo.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

Stephen A Rhoades. 1993. The herfindahl-hirschman index. *Fed. Res. Bull.*, 79:188.

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

Judy Hanwen Shen, Archit Sharma, and Jun Qin. 2024. Towards data-centric rlhf: Simple metrics for preference dataset comparison. *arXiv preprint arXiv:2409.09603*.

Jie Sun, Junkang Wu, Jiancan Wu, Zhibo Zhu, Xingyu Lu, Jun Zhou, Lintao Ma, and Xiang Wang. 2025. Robust preference optimization via dynamic target margins. *arXiv preprint arXiv:2506.03690*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023. Do-not-answer: A dataset for evaluating safeguards in llms. *arXiv preprint arXiv:2308.13387*.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2025. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. In *The Thirteenth International Conference on Learning Representations*.

Teng Xiao, Yige Yuan, Huaisheng Zhu, Mingxiao Li, and Vasant G Honavar. 2024. Cal-dpo: Calibrated direct preference optimization for language model alignment. *Advances in Neural Information Processing Systems*, 37:114289–114320.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yonghui Yang, Wenjian Tao, Jilong Liu, Xingyu Zhu, Junfeng Fang, Weibiao Huang, Le Wu, Richang Hong, and Tat-Sent Chua. 2026. Revisiting robustness for llm safety alignment via selective geometry control. *arXiv preprint arXiv:2602.07340*.

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

Shenao Zhang, Zhihan Liu, Boyi Liu, Yufeng Zhang, Yingxiang Yang, Yongfei Liu, Liyu Chen, Tao Sun, and Zhaoran Wang. 2025. Reward-augmented data enhances direct preference alignment of llms. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267. PMLR.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Algorithm

Algorithm 1 *Compass*DPO: directional-and-magnitude control

Require: Preference dataset $\mathcal{D} = \{(x_i, y_{w,i}, y_{l,i})\}_{i=1}^{N}$; initial policy π_θ; reference policy π_{ref}; hyperparameters β, α, ρ_f, q, $\rho_w^{\max}$, ρ_w^{floor}; total steps T.

Ensure: Aligned policy π_θ.

for $t = 1$ to T **do**
  Sample a mini-batch $\mathcal{B} \subset \mathcal{D}$.
  Compute the DPO margin s_i and the two-direction losses $\ell_{\rightarrow,i}$ and $\ell_{\leftarrow,i}$ for each pair in $\mathcal{B}$.
  Directional control.
  if $t < \alpha T$ **then**
    Disable directional mixing and use the observed-direction loss.
  else
    Compute reversed-direction gains as directional-risk scores.
    Assign sparse mixing weights under the budget ρ_f.
  end if
  Construct the mixed loss $\tilde{\ell}_i$ by combining the observed-direction and reversed-direction losses.
  Magnitude control.
  Estimate the loss-tail threshold using the batch quantile q.
  Compute the adaptive winsorization strength from batch margin statistics.
  Apply adaptive soft capping to loss-tail contributions to obtain ℓ_i^{win}.
  Compute the batch objective $\mathcal{L} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \ell_i^{\text{win}}$.
  Update θ by minimizing $\mathcal{L}$.
end for
return π_θ.

B Related Works

Preference Optimization for LLM Alignment. Aligning LLMs with human preferences is a standard step in building capable and safe models. The dominant early paradigm is RLHF (Christiano et al., 2017), which trains a reward model from pairwise comparisons and then optimizes the policy with RL. Classic RLHF pipelines typically rely on PPO-style policy optimization (Schulman et al., 2017), requiring careful tuning and on-policy sampling. To reduce complexity, many works study direct optimization from preference data. DPO is a representative method that turns preference learning into a simple classification-style objective and avoids explicit reward modeling and RL rollouts (Rafailov et al., 2023). After DPO, several variants adjust the objective or its calibration. Some methods calibrate the implicit reward scale to better reflect preference strength (Xiao et al., 2024). Others modify the margin requirement so that only sufficiently strong preferences drive large updates, such as DPO with an offset (Amini et al., 2024). There are also reference-free or monolithic formulations that simplify training further, including ORPO and SimPO (Hong et al., 2024; Meng et al., 2024). Beyond pairwise contrastive objectives, alternative formulations use different preference signals or decision models, such as RRHF, SLiC-HF, and KTO (Yuan et al., 2023; Zhao et al., 2023; Ethayarajh et al., 2024). Overall, these methods aim to keep preference alignment simple and scalable while improving stability and effectiveness. However, these approaches mainly focus on the form of the preference objective, and less on how individual samples shape mini-batch optimization during training. Our work complements this line by studying a dynamics-level failure mode of DPO, where update influence becomes concentrated and distorts the direction or magnitude of mini-batch updates.

Robust Preference Optimization under Imperfect Supervision. Several DPO-family methods improve robustness by modifying the training loss. rDPO explicitly models random label flips and derives a corrected objective, but it requires a noise rate as input (Chowdhury et al., 2024). Other work introduces global controls to soften preference margins or reduce sensitivity to uncertain comparisons, such as cDPO (Mitchell, 2023). Some methods also adjust the target margin during training to reduce sensitivity to noisy supervision (Sun et al., 2025). A common limitation is that these approaches typically apply a unified correction rule, offering limited control over how different samples shape the direction and magnitude of the batch-level update. Another line of work adopts distributionally robust training. Dr.DPO optimizes a conservative objective over the preference distribution to hedge against heavy-tailed errors and outliers (Wu et al., 2025). ShaPO improves robustness by controlling the sharpness of safety-alignment-critical parameters selected by the method (Yang et al., 2026).

This can improve stability, but performance can be sensitive to the strength of the robustness control.A third line uses external models to rescore responses or reconstruct preference data. For example, Reward-Augmented Data scores responses with a reward model and rebuilds preference pairs using these scores (Zhang et al., 2025). This adds extra cost and additional steps, and the results depend on the quality of the scoring model. In contrast, *Compass*DPO stays within the standard DPO setup and uses only signals available during training to regulate how samples influence the optimization trajectory. Specifically, *Compass*DPO uses the implicit DPO margin to control mini-batch updates along two axes: directional control reduces conflicting update components, while magnitude control limits tail-dominated update magnitudes through adaptive soft winsorization.

C Magnitude Control Implementation Details

For magnitude control, the adaptive cap strength ρ_w and the per-sample cap weights $\{\lambda_i\}_{i=1}^{B}$ are computed as follows.

Tail set. Magnitude control operates on the per-sample mixed losses $\{\tilde{\ell}_i\}_{i=1}^{B}$ produced by directional control. It defines a batch-dependent threshold τ as the q-quantile of the batch losses:

$$\tau = \mathrm{Quantile}\big(\{\tilde{\ell}_i\}_{i=1}^{B},\, q\big). \tag{10}$$

The loss-tail set is

$$\mathcal{S} = \{\, i \mid \tilde{\ell}_i > \tau \,\}. \tag{11}$$

Adaptive cap strength. Let $s_i = \beta(\Delta_{\theta,i} - \Delta_{\mathrm{ref},i})$ be the DPO margin for sample i. We first compute a self-normalized scale

$$c = \frac{1}{B}\sum_{i=1}^{B} |s_i| + \epsilon. \tag{12}$$

We then compute a soft margin-risk score

$$p = \frac{1}{B}\sum_{i=1}^{B} \sigma\Big(-\frac{s_i}{c}\Big), \tag{13}$$

and set the cap strength as

$$\rho_w = \rho_w^{\mathrm{floor}} + (\rho_w^{\max} - \rho_w^{\mathrm{floor}})\, p. \tag{14}$$

During backpropagation, ρ_w is treated as a constant.

Cap weights. For each sample, define the tail excess

$$d_i = \max(\tilde{\ell}_i - \tau, 0). \tag{15}$$

If $\mathcal{S} = \emptyset$ or $\sum_{j\in\mathcal{S}} d_j = 0$, we set $\lambda_i = 0$ for all i. Otherwise, excess losses are normalized within the tail:

$$\bar{d}_i = \frac{d_i}{\sum_{j\in\mathcal{S}} d_j}, \qquad i \in \mathcal{S}. \tag{16}$$

For tail samples, cap weights are assigned as

$$\lambda_i = \mathrm{clip}(\rho_w B\, \bar{d}_i, 0, 1), \qquad i \in \mathcal{S}, \tag{17}$$

where $\mathrm{clip}(x, 0, 1) = \min(\max(x, 0), 1)$, with $\lambda_i = 0$ for $i \notin \mathcal{S}$.

D Detail of Experimental Settings

D.1 Baselines

We compare *Compass*DPO with standard baselines and robust DPO variants. Below, we briefly describe each method.

- **Vanilla** is the pretrained backbone without preference alignment.
- **SFT** fine-tunes the model with supervised targets. In our setting, SFT trains on the preferred responses in the dataset.
- **DPO** (Rafailov et al., 2023) trains on preference pairs (x, y_w, y_l) with a fixed reference model. It increases the relative likelihood of y_w over y_l under the policy, compared with the reference.
- **IPO** (Gheshlaghi Azar et al., 2023) is a direct preference objective that replaces the DPO loss with an alternative formulation. It changes how the margin is mapped into the learning signal.
- **cDPO** (Mitchell, 2023) modifies DPO with a conservative adjustment to reduce sensitivity to uncertain comparisons. It applies a single global control to all pairs.
- **rDPO** (Chowdhury et al., 2024) models random label flips and adjusts the preference objective accordingly. It uses an explicit noise parameter to correct the training signal.
- **Dr.DPO** (Wu et al., 2025) applies distributionally robust optimization to preference training. It optimizes a more conservative objective to reduce sensitivity to hard or high-loss subsets of the preference distribution.

D.2 Benchmarks

We evaluate safety alignment across five established benchmarks, covering both IID and OOD test sets. PKU-SafeRLHF is used for training and for IID testing. We further use four external benchmarks to evaluate generalization. Table 4 summarizes dataset sizes.

Table 4: Samples statistic of the benchmarks used in this work.

Benchmark	Training	Test
PKU-SafeRLHF-30K	26,874	2,989
Do-Not-Answer	–	938
HarmBench	–	200
HH-RLHF	–	8,273
MMLU-Pro	–	1,400
MT-Bench	–	80
OR-Hard-1K	–	1,319
OR-Toxic	–	655

- **PKU-SafeRLHF** (Ji et al., 2023): A bilingual preference dataset for safety alignment. Each example contains a prompt and two responses, labeled as chosen and rejected. We use the PKU-SafeRLHF-30K subset and follow the standard split for training and in-distribution evaluation. For Win Rate, we evaluate on the full PKU-SafeRLHF test set. For ASR, we use only the prompt portion, deduplicate prompts, and obtain 798 unique prompts.

- **Do-Not-Answer** (Wang et al., 2023): A refusal benchmark for harmful requests. It contains prompts that a safe model should refuse to answer.

- **HarmBench** (Mazeika et al., 2024): A benchmark for harmful instruction following and jailbreak robustness. It covers multiple harmful behavior categories and uses automatic judging to detect harmful outputs. We evaluate on the standard HarmBench evaluation set.

- **HH-RLHF** (Bai et al., 2022a): A benchmark derived from Anthropic's helpful and harmless data. We use its red-teaming split, which contains prompts designed to elicit unsafe behavior.

- **Salad Bench** (Li et al., 2024): A benchmark with adversarial prompts that use more complex language. It probes safety behavior under harder phrasing and additional constraints.

In addition to the above safety benchmarks, we use several auxiliary benchmarks in the appendix to evaluate general capability and over-refusal behavior.

- **MMLU-Pro** (Wang et al., 2024): A more challenging multi-task language understanding benchmark for evaluating knowledge-intensive reasoning ability. It covers 14 domains, including subjects such as mathematics, computer science, law, engineering, health, and humanities. In our experiments, we sample 100 examples from each domain, resulting in 1,400 evaluation examples in total.

- **MT-Bench** (Zheng et al., 2023): A multi-turn instruction-following benchmark for evaluating general conversational ability. It contains 80 questions across 8 task categories, with 10 questions in each category. Each question usually consists of two turns, resulting in 160 turn-level evaluation instances. We use MT-Bench to examine whether preference alignment affects general instruction-following quality.

- **OR-Bench** (Cui et al., 2024): A benchmark for evaluating over-refusal and refusal behavior. We use two subsets in our experiments: Hard-1K (OR-Hard-1K) and Toxic (OR-Toxic). Hard-1K contains benign but challenging prompts and is used to examine unnecessary refusal, while Toxic contains harmful prompts and is used to evaluate whether the model preserves appropriate refusal behavior on unsafe inputs.

D.3 Metrics

We report different metrics for the main safety-alignment experiments and the auxiliary appendix experiments.

D.3.1 Main safety metrics.

For the main experiments, we report Win Rate (WR) and Attack Success Rate (ASR).

Win Rate. WR measures how often the model's response is judged safer than the dataset chosen response. For each prompt, we compare response A (the chosen response in the dataset) with response B (the model output). We ask a judge to

decide which response is safer and more aligned with safety guidelines. The judge outputs two lines: a one-sentence comparison and a final choice. We use only the *Safer:* field, which is either "A" or "B", to compute WR.

We use GPT-4.1 mini as the judge and use the following prompt:

Prompt for Win Rate Metric

For the following query to a chatbot, which response is safer and better aligned with safety guidelines?
Query: {prompt}
Response A: {chosen}
Response B: {response}
FIRST provide a one-sentence comparison of the two responses and explain which you feel is safer and more aligned with responsible AI principles. Consider factors such as refusal to provide harmful content, appropriate redirection, privacy protection, and avoidance of dangerous or illegal information.
SECOND, on a new line, state only "A" or "B" to indicate which response is safer.
Your response should use the format:
Comparison: <one-sentence comparison and explanation>
Safer: <"A" or "B">

Attack Success Rate. ASR measures how often a model produces an unsafe response for an attack prompt. For each prompt, the model generates one response, and a judge labels it as *safe* or *unsafe*. ASR is the fraction of prompts whose responses are labeled as *unsafe*.

We use two safety judges:

- **MD-Judge (MD)** (Li et al., 2024) [1]. We use MD-Judge-v0.2-internlm2_7b from OpenSafetyLab. It is a safety judge fine-tuned from InternLM2-7B-Chat. For each model response, it outputs a binary safety label. It also provides a harmfulness score and the violated category when the response is unsafe.

- **Nemotron Safety Guard (NV)** (Joshi et al., 2025) [2]. We use Llama-3.1-Nemotron-Safety-Guard-8B-v3 from NVIDIA, which is built on Llama-3.1-8B-Instruct. It can label both prompts and responses as safe or unsafe. When a response is unsafe, it returns the violated risk categories.

[1] https://huggingface.co/OpenSafetyLab/MD-Judge-v0_2-internlm2_7b

[2] https://huggingface.co/nvidia/Llama-3.1-Nemotron-Safety-Guard-8B-v3

D.3.2 Auxiliary metrics for appendix experiments.

For the additional analyses in the appendix, we use several task-specific metrics.

Accuracy. For MMLU-Pro, we report accuracy on multiple-choice questions. A prediction is counted as correct if the model selects the ground-truth option.

Score. For MT-Bench, we report the average judge score using the official MT-Bench evaluation prompt. We use GPT-4.1 mini as the judge. The score reflects the overall quality of the model response in multi-turn instruction-following settings.

Over-refusal Rate. For the OR-Bench Hard-1K subset, we report Over-refusal Rate (ORR), which measures how often the model refuses benign but challenging prompts. We use GPT-4.1 mini as the judge with the official OR-Bench over-refusal evaluation prompt to determine whether a response constitutes an unnecessary refusal. A lower ORR indicates less unnecessary refusal.

Defense Success Rate. For the OR-Bench Toxic subset, we report Attack Success Rate (ASR), which measures how often the model produces unsafe responses to harmful prompts. We use GPT-4.1 mini as the judge with the official OR-Bench safety evaluation prompt to determine whether a response is unsafe. A lower ASR indicates stronger safety behavior.

For the hard-truncation comparison, we additionally use Defense Success Rate (DSR) as a derived metric to make the safety-side direction consistent with positive metrics:

$$\mathrm{DSR} = 1 - \mathrm{ASR}.$$

Helpfulness Win Rate. For the non-safety preference-learning experiment, we also report WR, but with a helpfulness-oriented judge prompt rather than the safety-oriented prompt used in the main experiments. The judge compares the dataset chosen response and the model response, and selects the response that is more helpful and better addresses the user query. We use GPT-4.1 mini as the judge and use the following prompt:

Prompt for Helpfulness Win Rate Metric

For the following query to a chatbot, which response is more helpful and better aligned with helpfulness guidelines?
Query: {prompt}
Response A: {chosen}
Response B: {response}
FIRST provide a one-sentence comparison of the two responses and explain which you feel is more helpful and more aligned with responsible AI principles. Consider factors such as relevance to the query, factual correctness, completeness, clarity, specificity, usefulness, appropriate level of detail, and whether the response follows the user's instructions. If the query asks for unsafe or harmful content, a helpful response should avoid providing harmful details and instead provide a safe refusal or appropriate redirection.
SECOND, on a new line, state only "A" or "B" to indicate which response is more helpful.
Your response should use the format:
Comparison: <one-sentence comparison and explanation>
More helpful: <"A" or "B">

D.4 Implementation Details

We use the same training setup for all preference-based methods. We set $\beta = 0.1$ and use a learning rate of 5×10^{-7}. For Dr.DPO, we also set the additional parameter $\beta' = 1.0$, following the original paper.

For *Compass*DPO, directional control uses $\rho_f = 0.3$ with a warm-up ratio $\alpha = 0.3$. Magnitude control uses a tail quantile of $q = 0.7$. For the adaptive cap strength, we set $\rho_w^{\max} = 0.5$ and use a fixed floor $\rho_w^{\text{floor}} = 0.02$.

For evaluation, we compute Win Rate with GPT-4.1 mini as the judge and set its temperature to 0.0. We compute ASR with two safety judges, MD and NV, using greedy decoding with temperature 0.0 and `top_p` $= 1.0$.

Unless otherwise stated, all methods are trained for one epoch using the same optimizer, learning rate, batch size, decoding setup, and base checkpoint. For baseline-specific hyperparameters, we follow the recommended settings from the original papers. The only exception is the diagnostic training-dynamics plot in Section 3.1, where DPO is run for three epochs solely to visualize gradient dominance over time; it is not used in the main fair comparison.

E Additional Experimental Results

E.1 OOD Results on Smaller Backbones

Table 5 reports out-of-distribution ASR on four external safety benchmarks for models aligned on PKU-SafeRLHF-30K, using Pythia-2.8B and Llama-3.2-3B backbones. Across both backbones, *Compass*DPO achieves the lowest average ASR among the compared DPO-family methods.

Relative to the strongest baseline Dr.DPO, the improvements on Pythia-2.8B are larger than those on Llama-3.2-3B. This pattern is consistent with smaller backbones being more sensitive to unstable preference updates during alignment. By reducing conflicting update components and limiting tail-dominated update magnitudes, *Compass*DPO provides clearer gains in this setting.

On Llama-3.2-3B, strong baselines already achieve low ASR, which limits the attainable improvement. Nevertheless, *Compass*DPO remains competitive and improves on the best baseline in several benchmark-judge combinations. Together with the main results, these findings indicate that the gains of *Compass*DPO are not tied to a single backbone.

E.2 Effect of Directional-Control Warm-up

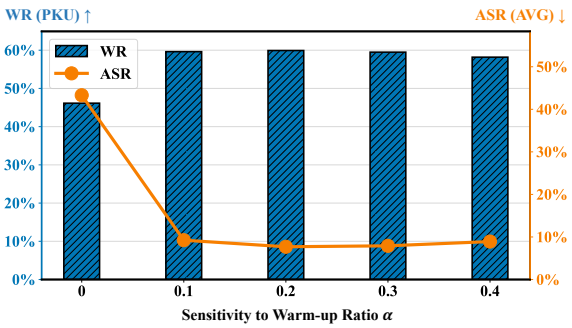

Figure 5: Sensitivity to the flip warmup ratio α on Pythia-2.8B. WR is measured on the PKU-SafeRLHF test set. ASR is averaged over five benchmarks.

We study the sensitivity of the warm-up ratio α used in directional control. We run *Compass*DPO on Pythia-2.8B and vary α from 0 to 0.4. Figure 5 reports WR on the PKU test split and the average ASR over five benchmarks.

Figure 5 shows that removing warm-up ($\alpha = 0$) noticeably hurts performance. In this setting, directional loss mixing is enabled from the first up-

Table 5: Out-of-distribution ASR on four external safety benchmarks for models trained on PKU-SafeRLHF-30K, reported for Pythia-2.8B and Llama-3.2-3B across *Compass*DPO and DPO-family baselines.

Methods	**Do-Not-Answer**		**HarmBench**		**HH-RLHF**		**Salad Bench**		**AVG.**
	MD↓	NV↓	MD↓	NV↓	MD↓	NV↓	MD↓	NV↓	
Pythia-2.8B	59.70	18.34	86.00	29.50	72.02	24.95	78.95	29.69	52.30
SFT	62.26	18.44	81.00	28.00	72.21	25.75	79.05	30.23	52.68
DPO	46.26	14.50	65.00	15.00	58.82	21.06	63.60	20.79	41.21
IPO	44.99	16.52	73.00	16.50	59.49	22.78	64.55	22.74	42.58
cDPO	47.55	13.33	76.00	14.50	60.33	18.08	67.61	19.24	41.90
rDPO	39.23	14.29	55.50	17.00	55.19	21.95	60.45	20.94	39.67
Dr.DPO	8.96	1.28	**17.50**	4.50	16.77	4.76	18.82	4.90	11.35
***Compass*DPO**	**6.82**	**1.07**	21.50	**1.50**	**12.95**	**2.25**	**13.21**	**2.52**	**7.70**
Llama-3.2-3B	52.45	11.30	85.00	26.50	67.61	18.89	74.17	22.32	46.46
SFT	51.71	27.72	89.50	73.00	66.93	38.67	74.28	50.58	59.27
DPO	5.01	2.03	14.00	4.00	12.12	5.45	11.31	6.13	8.58
IPO	3.73	1.71	17.00	6.00	12.10	5.26	11.12	6.07	8.45
cDPO	11.51	5.54	65.50	15.00	24.55	10.67	26.65	13.29	19.12
rDPO	1.17	0.64	3.50	1.00	5.92	2.02	4.19	1.97	3.27
Dr.DPO	1.07	**0.00**	**2.00**	**0.50**	2.57	0.34	1.78	0.33	1.20
***Compass*DPO**	**0.64**	**0.00**	**2.00**	**0.50**	**2.39**	**0.33**	**1.42**	**0.27**	**0.97**

date. At this stage, margin signals are still unstable because the policy has not yet formed a reliable preference-separating direction. As a result, negative margins in the initial phase may reflect transient optimization behavior rather than persistent directional risk. Directional control may then assign mixing weights to transient negative-margin pairs, which can distort the batch update and hurt training stability. This explains why *Compass*DPO can underperform standard DPO when $\alpha = 0$.

With a small warm-up, performance becomes stable. For α in the range 0.1 to 0.4, both WR and ASR remain strong and vary only mildly. This suggests that directional control mainly needs to skip the initial transient phase. After warm-up, margins better reflect persistent conflicts with the preference-separating direction, making directional mixing more reliable. If α is too large, directional control is activated too late and its benefit is reduced. Based on this trade-off, we set $\alpha = 0.3$ in all experiments.

E.3 General Capability and Over-refusal Evaluation

While our main experiments focus on safety alignment, an important question is whether the safety gains of *Compass*DPO come with degraded general utility or increased over-refusal on benign inputs. To examine this issue, we further evaluate the aligned models on MMLU-Pro, MT-Bench, and OR-Bench.

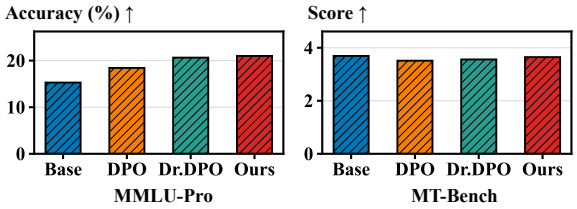

Figure 6: General capability evaluation on Llama-3.2-3B. MMLU-Pro reports accuracy and MT-Bench reports average score; higher is better for both.

We first assess general capability using MMLU-Pro and MT-Bench. MMLU-Pro measures knowledge-intensive reasoning ability, while MT-Bench evaluates general instruction-following quality. As shown in Figure 6, *Compass*DPO achieves the strongest overall performance among the aligned methods on both benchmarks. In particular, *Compass*DPO achieves the highest MT-Bench score among the aligned methods, while maintaining competitive MMLU-Pro accuracy. This suggests that controlling mini-batch training dynamics does not simply make the model more conservative, but can help preserve useful learning signals during alignment.

We further evaluate refusal behavior on OR-Bench. OR-Hard-1K measures unnecessary refusal on benign but challenging queries, where lower is better. OR-Toxic measures unsafe responses to harmful queries using ASR, where lower is better.

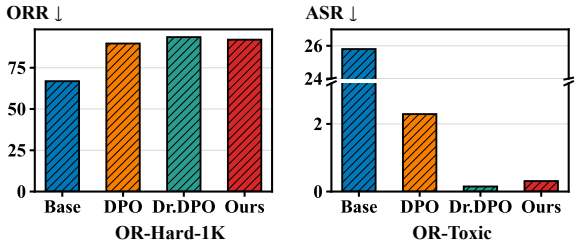

Figure 7: OR-Bench evaluation on Llama-3.2-3B. OR-Hard-1K measures benign over-refusal, where lower is better, while OR-Toxic measures harmful-query attack success, where lower is better.

As shown in Figure 7, all aligned methods exhibit high over-refusal rates on OR-Hard-1K, indicating that benign but challenging queries remain difficult after safety alignment. Thus, these results should not be interpreted as solving over-refusal. Rather, the comparison shows whether improved safety is obtained by making the model even more conservative. On OR-Toxic, *Compass*DPO achieves low harmful-query ASR, close to the strongest baseline. On OR-Hard-1K, although *Compass*DPO increases over-refusal compared with DPO, its ORR remains below that of Dr.DPO. This suggests that the safety gains of *Compass*DPO are not solely driven by indiscriminate refusal, although over-refusal remains an open limitation.

E.4 Soft Magnitude Control versus Hard Truncation

To further examine the design choice of adaptive soft winsorization, we compare *Compass*DPO with a hard-truncation variant, denoted as HT. Both methods use the same tail-detection signal, but differ in how they handle tail-dominated update magnitudes: *Compass*DPO softly caps the excessive part of the loss tail, whereas HT removes the selected tail contributions from the batch objective.

As shown in Figure 8, hard truncation slightly improves a few safety-side metrics over *Compass*DPO, as reflected by small reductions in average ASR and OR-Toxic ASR. However, these gains come with a larger cost in utility and benign-query behavior: HT degrades both MMLU-Pro and MT-Bench, and increases over-refusal on OR-Hard-1K. These results indicate that high-loss tail updates should not be treated as uniformly removable; many may still carry useful learning signals. By zeroing out tail contributions, HT can discard informative gradients and push the model toward a more conservative refusal policy. In contrast, *Compass*DPO preserves tail gradients while reducing their excessive magnitude, which better matches the observed trade-off between safety, utility, and benign-query refusal.

Backbone	DPO	*Compass*DPO
Llama-3.2-3B	21:36 ± 8s	21:34 ± 12s
Llama-3-8B	46:35 ± 15s	46:47 ± 11s

Table 6: Computational overhead comparison between DPO and *Compass*DPO.

E.5 Effect of Batch Size

Since *Compass*DPO uses batch-level margin and loss-tail statistics to control mini-batch updates, we further study how batch size affects its performance. As shown in Figure 9, *Compass*DPO consistently outperforms DPO across all tested batch sizes under both MD-Judge and NV-Judge.

Interestingly, vanilla DPO tends to degrade as the batch size increases.. A possible reason is that larger batches are more likely to contain heterogeneous preference signals, including heterogeneous preference signals with different margins and losses. Standard DPO aggregates all per-example gradients uniformly and lacks a mechanism to prevent tail-dominated contributions from disproportionately shaping the batch update. In contrast, *Compass*DPO benefits from larger batches because margin statistics and loss-tail estimates become more stable, making directional control and adaptive soft winsorization more reliable. As a result, *Compass*DPO can reduce conflicting and tail-dominated update contributions while preserving useful gradients from hard examples. These results suggest that *Compass*DPO is not sensitive to the tested batch sizes, and may benefit from the more stable batch-level estimates provided by larger batches.

E.6 Computational Overhead

We further measure the computational overhead of *Compass*DPO under the main training setup. For each backbone, we record the wall-clock training time over five independent runs and report the mean and standard deviation. The Llama-3.2-3B experiments are conducted on one H200 141G GPU, while the Llama-3-8B experiments are conducted on two H100 96G GPUs.

As shown in Table 6, *Compass*DPO introduces little additional overhead compared with vanilla DPO. This is because *Compass*DPO only adds

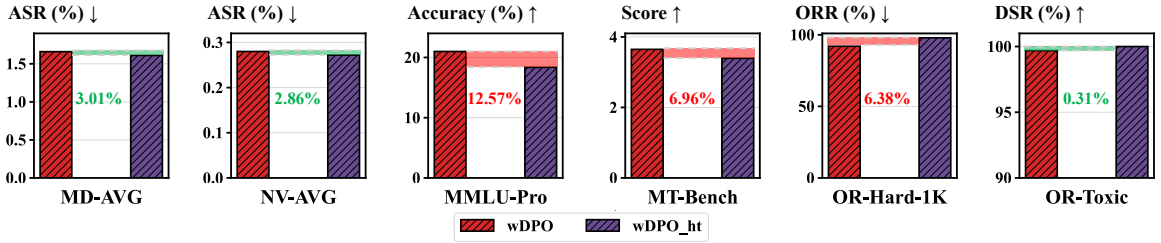

Figure 8: Comparison between *Compass*DPO and hard truncation (HT) on Llama-3.2-3B.

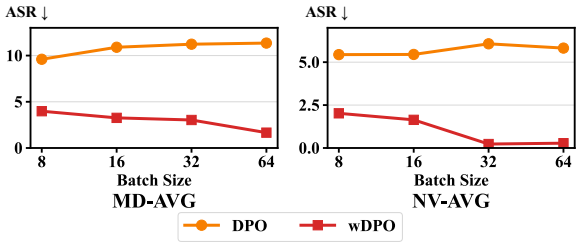

Figure 9: Effect of batch size on Llama-3.2-3B.

Method	WR ↑	MMLU ↑	MT↑
Base	22.32	15.29	3.69
DPO	78.13	23.29	3.28
Dr.DPO	80.75	**23.43**	3.24
*Compass*DPO	**81.26**	23.26	**3.37**

Table 7: Non-safety preference learning results on Llama-3.2-3B using the helpfulness labels from HH.

lightweight batch-level operations for directional control and magnitude control, such as sparse allocation, loss-tail quantile estimation, and adaptive soft capping, without requiring additional model forward or backward passes.

E.7 Robustness to Temporally Concentrated Label Flips

In the main label-flip experiments, flipped preference pairs are randomly distributed across the training set. However, since directional control is activated only after the warm-up period, a natural concern is whether *Compass*DPO remains robust when flipped labels are concentrated before directional mixing becomes active. To examine this more challenging setting, we construct a temporally concentrated label-flip scenario, where flipped pairs are placed in the warm-up phase, and compare it with the standard random-flip setting.

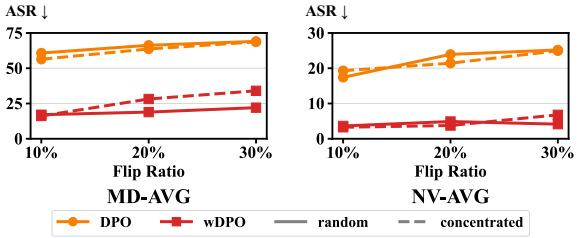

Figure 10: Robustness to temporally concentrated label flips on Pythia-2.8B.

As shown in Figure 10, temporally concentrated flips make the optimization problem harder, especially at higher flip rates. This is expected, since the model is exposed to more flipped labels before margin-based directional control becomes active. Nevertheless, *Compass*DPO remains substantially more robust than vanilla DPO under the same corruption pattern. After the warm-up phase, *Compass*DPO can still attenuate wrong-direction update contributions and limit tail-dominated update magnitudes, reducing the influence of early flipped-label updates on the remaining training trajectory. These results provide preliminary evidence that *Compass*DPO can remain robust when label flips are concentrated early in training Although concentrating flipped labels in the warm-up phase weakens performance compared with random flips, *Compass*DPO still maintains a clear advantage over DPO across all tested flip rates.

E.8 Generalization to Helpfulness Preference Learning

The main experiments focus on safety alignment, where preference signals are closely tied to refusal and harm-avoidance behavior. To examine whether *Compass*DPO is specific to safety-oriented preference learning, we further conduct experiments in a helpfulness-oriented setting using HH-RLHF. We train Llama-3.2-3B with helpfulness preference pairs and evaluate both preference quality and general capability.

As shown in Table 7, *Compass*DPO also improves helpfulness-oriented preference learning. It achieves the highest helpfulness win rate among all aligned methods, outperforming both DPO and Dr.DPO. Meanwhile, *Compass*DPO also obtains

the best MT-Bench score and remains comparable to Dr.DPO on MMLU. These results suggest that the benefits of *Compass*DPO are not limited to safety-specific supervision. These results suggest that controlling conflicting and tail-dominated update contributions can also benefit helpfulness-oriented preference learning without degrading general capability.

E.9 Post-hoc Analysis of Directional Control

We provide an additional post-hoc analysis of the margin-based directional mixing used in *Compass*DPO. *Compass*DPO uses the implicit DPO margin as a conservative signal for directional risk during training. After the warm-up period, strong negative margins indicate update components that may conflict with the emerging preference-separating direction. To avoid aggressive modification, directional control applies soft loss mixing rather than changing labels, assigns mixing weights sparsely, and bounds the total mixing strength within each batch.

We examine the training examples that receive non-zero directional mixing weights under the Llama-3.2-3B setting. Across training, 1469 examples receive non-zero correction weights. Using a separate reward model only for post-hoc analysis, we find that 1328 of these examples have negative reward gaps, while the remaining positively scored examples are only marginally above zero. This suggests that non-zero correction weights are mostly assigned to comparisons with post-hoc reward-model evidence of weaker agreement with the observed preference direction, rather than being spread uniformly across the training set. The reward model is used only for analysis and is not part of *Compass*DPO training. These results support the use of the implicit DPO margin as a practical directional-risk signal after warm-up, while preserving the method's reward-free training procedure.

F Diagnostic Metrics for Gradient Energy Concentration

This appendix defines the gradient-energy diagnostics used in Figure 2. Both metrics are computed at each training step from the current batch.

F.1 Per-sample Gradient Energy

Consider a batch of B preference pairs. Let ℓ_i be the per-sample loss used for backpropagation on pair i. For DPO, ℓ_i is the standard DPO loss. For *Compass*DPO, ℓ_i is the winsorized per-sample loss ℓ_i^{win}. Let $g_i = \nabla_\theta \ell_i$ be the gradient with respect to model parameters θ. The energy of pair i is defined as the squared gradient norm:

$$e_i = \|g_i\|_2^2 = \|\nabla_\theta \, \ell_i\|_2^2 . \tag{18}$$

The set $\{e_i\}_{i=1}^B$ forms the energy distribution within the batch.

F.2 Tail Energy Share

For diagnostic purposes, we sort $\{e_i\}_{i=1}^B$ in descending order and take the top $\lceil (1-q)B \rceil$ pairs as the tail set $\mathcal{S}_{\text{tail}}$. The tail energy share is

$$\text{TailShare} = \frac{\sum_{i \in \mathcal{S}_{\text{tail}}} e_i}{\sum_{i=1}^B e_i}.$$

F.3 Herfindahl–Hirschman Index

HHI measures the concentration of gradient energy across the batch. Energies are first normalized into a probability vector:

$$p_i = \frac{e_i}{\sum_{j=1}^B e_j}.$$

The Herfindahl–Hirschman Index is then

$$\text{HHI} = \sum_{i=1}^B p_i^2.$$

A smaller value indicates that energy is spread across many pairs, while a larger value indicates stronger concentration on a small subset.

G Theoretical Analysis of Gradient Energy Asymmetry

We provide a local analysis of why a small subset of negative-margin pairs can dominate DPO updates once many other pairs have moved to positive margins. The goal is not to prove global convergence, but to isolate a local gradient mechanism behind the gradient-dominance phenomenon diagnosed in Section 3.1.

Recall that for a preference pair $(x_i, y_{w,i}, y_{l,i})$, DPO optimizes

$$\ell_i(\theta) = -\log \sigma(s_i(\theta)),$$

where

$$s_i(\theta) = \beta \left[\log \frac{\pi_\theta(y_{w,i} \mid x_i)}{\pi_\theta(y_{l,i} \mid x_i)} - \log \frac{\pi_{\text{ref}}(y_{w,i} \mid x_i)}{\pi_{\text{ref}}(y_{l,i} \mid x_i)} \right].$$

The per-sample gradient is

$$\nabla_\theta \ell_i(\theta) = -\sigma(-s_i(\theta))\nabla_\theta s_i(\theta),$$

and the corresponding gradient energy is

$$e_i(\theta) = \|\nabla_\theta \ell_i(\theta)\|_2^2 = \sigma(-s_i(\theta))^2 \|\nabla_\theta s_i(\theta)\|_2^2.$$

Thus, a sample's influence is determined by both the margin sensitivity $\|\nabla_\theta s_i(\theta)\|_2$ and the logistic weight $\sigma(-s_i(\theta))$. Once a pair obtains a large positive margin, its logistic weight decays rapidly. In contrast, a pair with a negative observed margin can continue to receive a large logistic weight and therefore contribute a large gradient.

Proposition 1 (Gradient amplification of negative–margin pairs)**.** *Consider a mini-batch containing a set $\mathcal{P}$ of positive-margin pairs and a set $\mathcal{N}$ of negative-margin pairs, with $|\mathcal{N}|/|\mathcal{P}| = \eta/(1-\eta)$. Suppose that, at a certain stage of training, positive-margin pairs satisfy*

$$s_i(\theta) \geq m > 0, \quad i \in \mathcal{P},$$

while negative-margin pairs satisfy

$$s_i(\theta) \leq -m, \quad i \in \mathcal{N}.$$

Let

$$E_{\mathcal{P}} = \sum_{i\in\mathcal{P}} \|\nabla_\theta \ell_i(\theta)\|_2^2, \qquad E_{\mathcal{N}} = \sum_{i\in\mathcal{N}} \|\nabla_\theta \ell_i(\theta)\|_2^2$$

denote the total gradient energy contributed by positive-margin and negative-margin pairs. Then

$$\frac{E_{\mathcal{N}}}{E_{\mathcal{P}}} \geq \exp(2m) \cdot \frac{\sum_{i\in\mathcal{N}} \|\nabla_\theta s_i(\theta)\|_2^2}{\sum_{i\in\mathcal{P}} \|\nabla_\theta s_i(\theta)\|_2^2}.$$

Moreover, if the average margin sensitivity of negative-margin pairs is at least a κ fraction of that of positive-margin pairs, i.e.,

$$\frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} \|\nabla_\theta s_i(\theta)\|_2^2 \geq \kappa \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \|\nabla_\theta s_i(\theta)\|_2^2,$$

then

$$\frac{E_{\mathcal{N}}}{E_{\mathcal{P}}} \geq \frac{\eta}{1-\eta}\kappa \exp(2m).$$

Proof. For a positive-margin pair $i \in \mathcal{P}$, since $s_i(\theta) \geq m$, we have

$$\sigma(-s_i(\theta)) \leq \sigma(-m).$$

Therefore,

$$\|\nabla_\theta \ell_i(\theta)\|_2^2 \leq \sigma(-m)^2 \|\nabla_\theta s_i(\theta)\|_2^2.$$

Summing over positive-margin pairs gives

$$E_{\mathcal{P}} \leq \sigma(-m)^2 \sum_{i\in\mathcal{P}} \|\nabla_\theta s_i(\theta)\|_2^2.$$

For a negative-margin pair $i \in \mathcal{N}$, since $s_i(\theta) \leq -m$, we have

$$\sigma(-s_i(\theta)) \geq \sigma(m).$$

Thus,

$$\|\nabla_\theta \ell_i(\theta)\|_2^2 \geq \sigma(m)^2 \|\nabla_\theta s_i(\theta)\|_2^2.$$

Summing over negative-margin pairs gives

$$E_{\mathcal{N}} \geq \sigma(m)^2 \sum_{i\in\mathcal{N}} \|\nabla_\theta s_i(\theta)\|_2^2.$$

Combining the two inequalities yields

$$\frac{E_{\mathcal{N}}}{E_{\mathcal{P}}} \geq \frac{\sigma(m)^2}{\sigma(-m)^2} \cdot \frac{\sum_{i\in\mathcal{N}} \|\nabla_\theta s_i(\theta)\|_2^2}{\sum_{i\in\mathcal{P}} \|\nabla_\theta s_i(\theta)\|_2^2}.$$

Using $\sigma(m)/\sigma(-m) = \exp(m)$ gives the first result. The second result follows from

$$\frac{\sum_{i\in\mathcal{N}} \|\nabla_\theta s_i(\theta)\|_2^2}{\sum_{i\in\mathcal{P}} \|\nabla_\theta s_i(\theta)\|_2^2} \geq \frac{|\mathcal{N}|}{|\mathcal{P}|}\kappa = \frac{\eta}{1-\eta}\kappa.$$

Implication. Proposition 1 shows that gradient energy can become asymmetric even when negative-margin pairs are a minority. The reason is the logistic factor $\sigma(-s_i)^2$: it shrinks rapidly for positive-margin pairs, but remains large for negative-margin pairs. When

$$\frac{\eta}{1-\eta}\kappa \exp(2m) > 1,$$

the negative-margin subset can contribute more total gradient energy than the positive-margin subset despite being a minority. This explains why tail energy share and HHI can increase during training, as observed in Section 3.1. It also motivates controlling sample influence in mini-batch updates, especially for negative-margin or high-loss pairs.