What Do LLMs Learn First? Asymmetric Learning Dynamics of Input Complexity and Output Ambiguity in Preference Alignment

Mengyang Li[1], Jingwen Wang[1], Pinlong Zhao[2]*
[1]Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University
[2]School of Cyberspace, Hangzhou Dianzi University
limengyang@tjnu.edu.cn, wangjingwen@stu.tjnu.edu.cn, pinlongzhao@hdu.edu.cn

Abstract

Direct Preference Optimization (DPO) has become a standard approach for aligning large language models with human preferences, yet existing methods treat all preference pairs uniformly during training. We identify two distinct sources of learning difficulty: Input Complexity (IC), capturing prompt understanding challenges, and Output Ambiguity (OA), measuring preference discrimination difficulty. Through systematic analysis, we demonstrate that these dimensions induce asymmetric learning dynamics, with IC-related competencies developing rapidly in early training while OA-related competencies emerge more gradually. Building on this observation, we propose DECOPO, a training framework that maintains separate, adaptive pacing schedules for each dimension. Experiments on UltraFeedback show that DECOPO achieves 42.3% length-controlled win rate on AlpacaEval 2.0 and 7.66 on MT-Bench, outperforming curriculum baselines by 2.1% and 0.21 points respectively, while matching full-data baseline performance with only 75% of training samples.

1 Introduction

Large language models aligned through Direct Preference Optimization (DPO) have achieved impressive results across diverse tasks (Rafailov et al., 2023; Tunstall et al., 2024; Ivison et al., 2023). Given a prompt and a pair of responses labeled as preferred and rejected, DPO directly optimizes the policy to favor the preferred response without training a separate reward model. Despite its empirical success, a fundamental aspect of DPO training remains underexplored: not all preference pairs are equally informative, and the sources of learning difficulty vary across samples.

Consider two preference pairs. In the first, a user asks a multi-step reasoning question; the preferred response correctly chains the logic while the rejected response makes an obvious error. In the second, a user asks a simple factual question; both responses are largely correct, but the preferred one is slightly more concise. These two cases present qualitatively different challenges. The first requires the model to understand a complex prompt before it can even assess response quality. The second requires fine-grained discrimination between similar outputs. Current DPO implementations make no distinction between these cases, processing them with identical learning schedules (Bai et al., 2022; Stiennon et al., 2020).

This uniform treatment stands in contrast to how difficulty is handled in other machine learning contexts. Curriculum learning has demonstrated that strategic data organization improves convergence across various domains (Bengio et al., 2009; Kumar et al., 2010; Xu et al., 2020). Data selection methods have shown that filtering preference data by quality metrics improves alignment (Deng et al., 2025; Gao et al., 2025), and recent work on training dynamics reveals that different samples influence model behavior through distinct mechanisms (Ren and Sutherland, 2025; Paul et al., 2021; Zhou et al., 2023). Yet these insights have not been systematically applied to understand what makes individual preference pairs difficult and how different difficulty sources interact during training.

We hypothesize that alignment difficulty can be meaningfully decomposed along two axes: IC, which captures how challenging it is to understand and represent the prompt, and OA, which measures how subtle the preference signal is between response pairs. This decomposition reflects a natural partition of the information flow in preference learning: one component concerns the input side (prompt understanding), and the other concerns the output side (response comparison). These two dimensions capture substantial variation in sample difficulty and enable principled training strategies.

To validate this hypothesis, we conduct gradient

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17373–17388

flow and Hessian analyses on the UltraFeedback dataset, revealing that IC and OA affect distinct model layers and exhibit sharply different convergence rates. Based on this finding, we propose DECOPO (Decoupled Pacing Optimization), a training framework that maintains two independent, mastery-based pacing schedules for IC and OA respectively. At each epoch, DECOPO evaluates dimension-specific mastery on held-out subsets and expands the active training region accordingly, allowing the model to build prompt understanding before tackling subtle preference distinctions.

Extensive experiments show that DECOPO achieves state-of-the-art results on AlpacaEval 2.0 (42.3% LC win rate) and MT-Bench (7.66) while significantly improving reasoning and coding capabilities. Our method matches full-data baseline performance using only 75% of training samples and demonstrates superior robustness to label noise. Furthermore, analysis reveals a natural two-phase learning trajectory where the model prioritizes prompt understanding before shifting focus to preference discrimination. Our contributions are summarized as follows:

- We identify IC and OA as distinct factors characterizing sample difficulty and provide empirical evidence that they induce asymmetric learning dynamics.
- We propose DECOPO, a principled framework that operationalizes this insight through independent, mastery-based pacing for each dimension.
- We demonstrate that DECOPO achieves superior performance, data efficiency, and noise robustness across multiple model families and datasets.

2 Related Work

Preference-Based Alignment. Aligning language models with human preferences has evolved from multi-stage RLHF pipelines (Christiano et al., 2017; Ouyang et al., 2022; Stiennon et al., 2020) to more streamlined approaches. DPO (Rafailov et al., 2023) reformulates preference learning as a classification problem, enabling direct policy optimization without explicit reward modeling. Subsequent extensions address the reference model (Meng et al., 2024; Hong et al., 2024), loss formulation (Ethayarajh et al., 2024; Azar et al., 2024), listwise generalization (Liu et al., 2025), and meta-learned preference fusion (Li et al., 2026). These methods focus on improving the optimization objective, while our work addresses the orthogonal question of how training data should be organized temporally.

Data Selection for Alignment. The importance of data quality in preference alignment has received increasing attention. Recent studies show that margin-based filtering (Deng et al., 2025), removing overly difficult examples (Gao et al., 2025), and careful subset selection (Zhou et al., 2023) can match or exceed full-data performance. Morimura et al. (2024) integrate reward models to filter samples online, and active learning approaches (Muldrew et al., 2024) dynamically select informative samples. These approaches address which data to use, whereas our work addresses when to use different types of data during training. The two perspectives are complementary: data selection identifies valuable samples, while our approach determines optimal presentation order.

Curriculum Learning. Curriculum learning (Bengio et al., 2009) and self-paced learning (Kumar et al., 2010) demonstrate that strategic data ordering improves convergence, with extensions to joint paradigms (Jiang et al., 2015) and cyclical schedules (Kesgin and Amasyali, 2023). For language models specifically, competence-based approaches in machine translation (Platanios et al., 2019; Zhang et al., 2019) and curriculum methods for NLU (Xu et al., 2020; Sachan and Xing, 2016) show consistent benefits. Teacher-student frameworks (Matiisen et al., 2019) enable automatic curriculum discovery. Most relevant to our work, Pattnaik et al. (2024) apply curriculum learning to DPO by ordering samples based on response distinguishability scores. However, this approach treats difficulty as one-dimensional, focusing only on the output side. While Li and Zhao (2026) introduce difficulty-aware modeling for learning curves, the multi-dimensional nature of alignment difficulty remains unaddressed. Our analysis reveals that input-side and output-side challenges are distinct and require separate treatment, motivating our decoupled pacing framework.

Training Dynamics Analysis. Recent work has made progress in understanding how LLMs evolve during fine-tuning. Ren and Sutherland (2025) propose a learning dynamics framework that decom-

poses how training samples influence model predictions, explaining phenomena like the squeezing effect in DPO. Loss landscape analysis (Springer et al., 2025) reveals that fine-tuning operates within basin-like stability regions. Data diet studies (Paul et al., 2021; Li et al., 2025) show that identifying important examples early in training can substantially improve efficiency. These analyses provide theoretical grounding for understanding why different samples may require different treatment, supporting our empirical finding that IC and OA samples influence distinct model components.

3 Understanding Alignment Difficulty

This section presents our analysis of what makes preference alignment difficult. We begin by formalizing Input Complexity and Output Ambiguity, then provide empirical evidence for their asymmetric learning dynamics, and conclude with gradient-based analysis that reveals the underlying mechanism.

3.1 Characterizing Sample Difficulty

We propose that the difficulty of learning from a preference triple (x, y_w, y_l) can be characterized along two axes: the complexity of understanding the input prompt and the ambiguity of discriminating between output responses.

Input Complexity (IC). Input Complexity quantifies the semantic, reasoning, and linguistic challenges posed by a prompt x, independent of any specific response pair. To our knowledge, this is the first formalization of input-side difficulty for preference alignment that is decoupled from the specific response pair. Our formulation is motivated by the observation that complex prompts elicit more diverse and uncertain response distributions: a prompt that is difficult to understand will lead to inconsistent responses, as the model lacks a stable representation of what is being asked.

Formally, given a reference policy π_{ref} (typically the supervised fine-tuned model), we sample N candidate responses via temperature sampling:

$$y^{(i)} \sim \pi_{\text{ref}}(\cdot|x), \quad i = 1, \ldots, N \tag{1}$$

For each sampled response, we compute its perplexity using a fixed external language model p_{LM}:

$$\text{PPL}(y^{(i)}|x) = \exp\left(-\frac{1}{L_i}\sum_{t=1}^{L_i} \log p_{\text{LM}}(y_t^{(i)}|x, y_{<t}^{(i)})\right) \tag{2}$$

where L_i denotes the length of response $y^{(i)}$. We then define Input Complexity as the variability of these perplexity scores:

$$\text{IC}(x) = \text{StdDev}_{i=1,\ldots,N}\left[\text{PPL}(y^{(i)}|x)\right] \tag{3}$$

Higher IC values indicate greater response variability, signaling that the prompt requires sophisticated understanding to address consistently.

To validate that IC captures meaningful difficulty rather than merely open-endedness, we conducted a human annotation study on 300 UltraFeedback samples. Three annotators rated each prompt on domain knowledge, reasoning depth, and instruction-following precision (inter-annotator agreement $\kappa = 0.71$). The Spearman correlation between IC and average human-rated difficulty is $\rho = 0.58$ ($p < 0.001$), substantially above prompt length ($\rho = 0.21$) and vocabulary complexity ($\rho = 0.29$). We further categorized all 60K prompts into Reasoning/Code ($n = 8{,}420$), Creative Writing ($n = 6{,}180$), and Factual/Simple ($n = 11{,}350$) via GPT-4 classification. Mean IC values are 12.8, 9.7, and 5.3 respectively. While creative prompts show elevated IC, reasoning and coding prompts exhibit the highest IC on average, confirming that IC reflects a meaningful spectrum of prompt difficulty. From a training dynamics perspective, both reasoning complexity and creative open-endedness cause high-variance internal representations in early training, and the Hessian analysis (Table 1) confirms similar curvature profiles for both cases. Therefore, regardless of the specific source, the curriculum implication remains the same: such prompts benefit from gradual introduction after the model has built stable representations.

Output Ambiguity (OA). While IC captures input-side challenges, Output Ambiguity measures the subtlety of preference between chosen response y_w and rejected response y_l for a given prompt x. We leverage an external judge model (e.g., GPT-4) to assign quality scores $S_{\text{judge}}(y|x)$, and define:

$$\text{OA}(y_w, y_l|x) = \frac{1}{|S_{\text{judge}}(y_w|x) - S_{\text{judge}}(y_l|x)| + \epsilon} \tag{4}$$

where ϵ is a small constant for numerical stability. Higher OA indicates greater ambiguity (smaller quality gap between responses).

The IC-OA Difficulty Space. For each preference sample in dataset $\mathcal{D}$, we compute its IC and

OA values. To create a structured representation, we discretize each dimension into quantile-based bins: K bins for IC and M bins for OA. Each sample is assigned ranks $k \in \{1, \ldots, K\}$ and $m \in \{1, \ldots, M\}$, where lower ranks indicate easier samples. This yields a $K \times M$ difficulty grid, with each cell $\mathcal{C}_{k,m}$ containing samples of comparable difficulty profile.

3.2 Empirical Evidence for Asymmetric Dynamics

Having formalized our difficulty characterization, we investigate whether IC and OA truly exhibit independent and asymmetric effects on learning.

Experimental Setup. We train four strategies on the UltraFeedback dataset (Cui et al., 2024) using Zephyr-7B-SFT (Tunstall et al., 2024) as initialization: (1) **IC-First**, which progresses from simple to complex prompts regardless of OA; (2) **OA-First**, which orders by increasing output ambiguity regardless of IC; (3) **Diagonal**, which jointly orders by $k + m$ as a natural baseline representing coupled difficulty progression; and (4) **Random**, which samples uniformly. All strategies use three equal-duration stages.

Asymmetric Performance Patterns. Figure 1a presents a stratified analysis of final model performance across the IC-OA grid. We compute win rates against the SFT baseline for each difficulty cell and compare IC-First versus OA-First strategies. The results reveal striking asymmetries: in high-IC regions (bottom row of the grid), IC-First achieves substantially higher win rates, with advantages reaching +8.2%. Conversely, in high-OA regions (leftmost column), OA-First demonstrates clear superiority, outperforming IC-First by up to 4.8%.

This pattern has important implications: no single-dimensional ordering can be universally optimal. The relative effectiveness of IC-First versus OA-First depends critically on the difficulty profile of test samples.

Divergent Learning Trajectories. To understand the dynamics underlying these performance asymmetries, we track validation loss throughout training for different difficulty regions. Figure 1b plots loss curves for high-IC and high-OA sample subsets under IC-First and OA-First strategies.

For high-IC samples, IC-First exhibits rapid initial loss reduction followed by persistent oscillations. This suggests that early focus on prompt understanding enables quick progress, but volatile later-stage behavior indicates difficulties in fine-tuning preference discrimination without adequate OA-focused training. In contrast, OA-First on the same samples shows slower initial descent but eventually achieves more stable convergence.

The pattern reverses for high-OA samples. Here, OA-First demonstrates steady, monotonic improvement, while IC-First shows initial strength followed by stagnation. These divergent trajectories provide mechanistic insight: IC and OA dimensions progress at fundamentally different rates, and forcing them to advance in lockstep creates suboptimal compromises.

3.3 Gradient Flow Analysis

To deepen our understanding of the mechanisms underlying asymmetric dynamics, we analyze how IC and OA samples influence different parts of the model.

Layer-wise Gradient Distribution. We partition model parameters into two groups: early layers (layers 1–16, primarily responsible for input representation) and late layers (layers 17–32, primarily responsible for generation and preference discrimination). This partition follows prior work on representation formation in decoder-only transformers (Ren and Sutherland, 2025), and sensitivity checks with 40%/60% and 60%/40% splits yield r_{early} values within ± 0.03, confirming robustness to the boundary choice. For each training sample, we compute the gradient $\nabla_\theta \mathcal{L}_{\text{DPO}}$ and measure the fraction concentrated in early layers:

$$r_{\text{early}} = \frac{\|\text{Proj}_{\text{early}}(\nabla_\theta \mathcal{L})\|}{\|\nabla_\theta \mathcal{L}\|} \tag{5}$$

Figure 1c shows the average r_{early} across IC-OA regions. High-IC samples exhibit substantially larger early-layer gradient contributions (0.62–0.68), indicating that learning from these samples primarily updates prompt understanding mechanisms. When IC is controlled at low levels, high-OA samples direct gradients predominantly to late layers ($r_{\text{early}} = 0.32$–0.35). When IC is high, it dominates gradient distribution regardless of OA level ($r_{\text{early}} \geq 0.65$). A two-way ANOVA on r_{early} values confirms this pattern: IC accounts for 72.3% of the variance ($F = 48.7$, $p < 0.001$), OA for 8.1% ($F = 5.5$, $p < 0.01$), and the interaction for 4.2%.

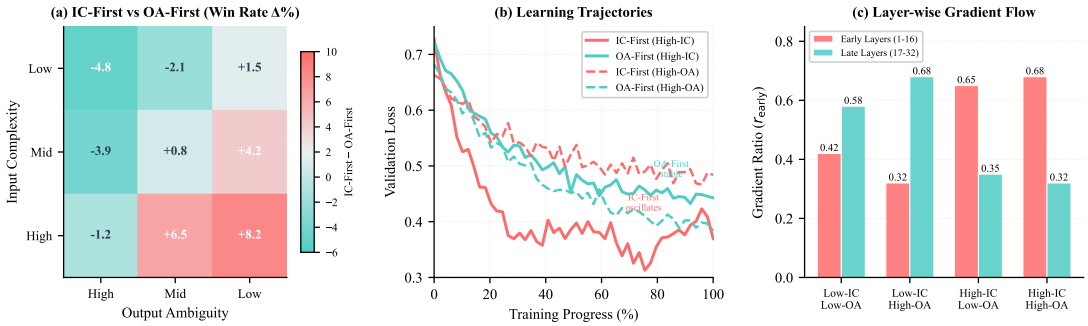


Figure 1: Empirical evidence for asymmetric learning dynamics. (a) Win rate differences (IC-First minus OA-First) across the IC-OA grid. (b) Validation loss trajectories for high-IC and high-OA subsets under different strategies. (c) Early-layer gradient ratio r_{early} across IC-OA regions; IC dominates gradient distribution regardless of OA level.

Region	Condition Number	Spectral Norm
Low-IC, Low-OA	42.3 ± 5.1	0.83 ± 0.07
High-IC, Low-OA	78.6 ± 8.2	1.42 ± 0.12
Low-IC, High-OA	51.2 ± 6.3	0.91 ± 0.08
High-IC, High-OA	89.4 ± 9.7	1.58 ± 0.15

Table 1: Hessian eigenspectrum analysis.

This corrected characterization actually sharpens our core argument. IC's dominance in gradient distribution provides the mechanistic explanation for why DECOPO's emergent trajectory (Figure 3a) prioritizes IC mastery first: the model must resolve input-side challenges before OA signals can meaningfully influence late-layer parameters.

Hessian Eigenspectrum Analysis. To further characterize the optimization landscape, we compute the Hessian eigenspectrum for sample subsets stratified by IC and OA. We approximate the top eigenvalues using the Lanczos algorithm on mini-batches from each difficulty region.

Table 1 reports the condition number and spectral norm for different regions. High-IC samples exhibit larger condition numbers (indicating more ill-conditioned optimization) and higher spectral norms (indicating sharper curvature). This suggests that IC-focused learning requires more cautious progression to avoid instability. High-OA samples show relatively flatter landscapes, allowing for larger effective step sizes once sufficient IC foundation is established.

These findings converge on a clear conclusion: effective preference alignment requires treating IC and OA as independent learning axes with dimension-specific pacing strategies.

4 Decoupled Pacing Optimization

Motivated by the asymmetric learning dynamics identified in Section 3, we now present DECOPO (Decoupled Pacing Optimization), a training framework that maintains independent, adaptive pacing for the IC and OA dimensions.

4.1 Design Principles

DECOPO is guided by three core principles: **(1) Independent Pacing:** IC and OA should have separate learning schedules rather than being coupled; **(2) Mastery-Based Adaptation:** pacing should adapt based on dimension-specific mastery; and **(3) Rectangular Activation:** active samples should form a rectangular region via independent thresholds rather than a diagonal wedge.

4.2 Algorithm Framework

Decoupled Pace Parameters. DECOPO maintains two independent scalar paces: $\lambda_{\text{IC}}(t)$ for Input Complexity and $\lambda_{\text{OA}}(t)$ for Output Ambiguity, where t indexes training epochs. At epoch t, the active training set is defined as:

$$\mathcal{D}_{\text{train}}^{(t)} = \{(x, y_w, y_l) \in \mathcal{D} \mid \text{IC}(x) \leq \lambda_{\text{IC}}(t) \wedge \text{OA} \leq \lambda_{\text{OA}}(t)\} \quad (6)$$

This formulation creates a rectangular activation frontier in the discretized IC-OA grid: a cell $\mathcal{C}_{k,m}$ is active if and only if $k \leq \lambda_{\text{IC}}(t)$ and $m \leq \lambda_{\text{OA}}(t)$.

Mastery-Driven Pace Updates. The key challenge in adaptive training is determining when the model is ready to progress to harder samples. DECOPO addresses this through dimension-specific mastery functions.

For Input Complexity, we measure mastery by evaluating performance on samples where output

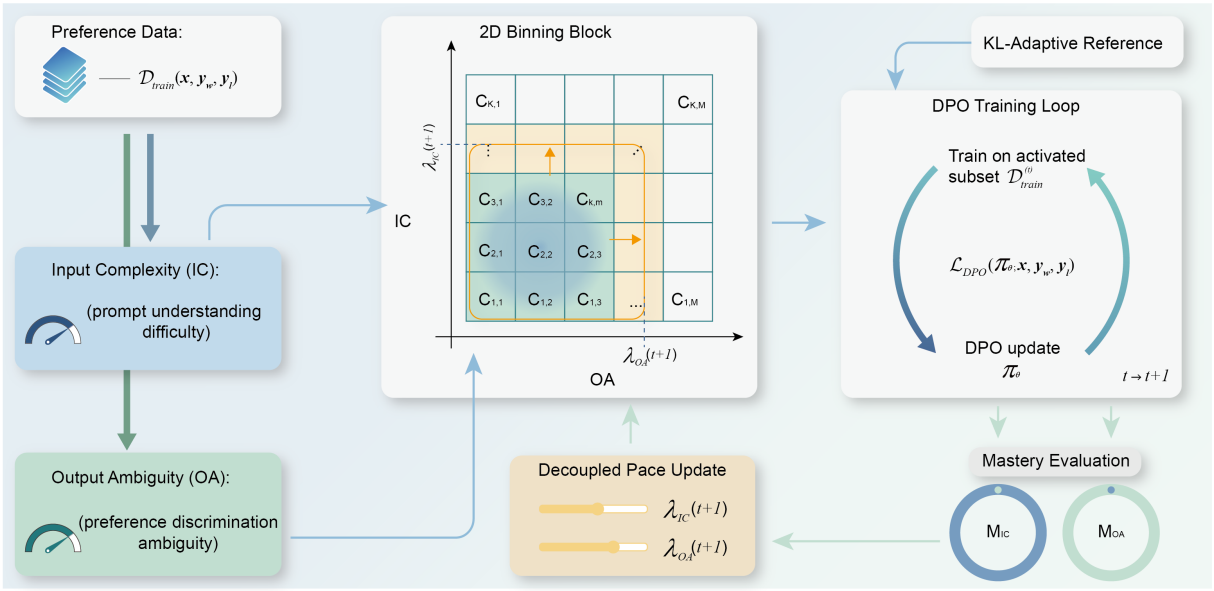


Figure 2: DECOPO algorithm overview. The framework maintains two independent pace parameters λ_{IC} and λ_{OA} that expand based on dimension-specific mastery scores. At each epoch, samples within the rectangular activation region are used for training, and paces are updated according to mastery assessments on held-out subsets.

ambiguity is minimal, thereby isolating IC-related challenges:

$$M_{\text{IC}} = \frac{1}{1 + \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{low-OA}}}[\mathcal{L}_{\text{DPO}}(\pi_\theta; x, y_w, y_l)]} \tag{7}$$

where $\mathcal{D}_{\text{low-OA}} = \{(x, y_w, y_l) \in \mathcal{D} \mid m = 1\}$ contains samples with the clearest preference signals.

Symmetrically, Output Ambiguity mastery is assessed on samples with minimal input complexity:

$$M_{\text{OA}} = \frac{1}{1 + \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{low-IC}}}[\mathcal{L}_{\text{DPO}}(\pi_\theta; x, y_w, y_l)]} \tag{8}$$

These definitions embody our core insight: to fairly assess competency in one dimension, we must control for the other.

At the end of each epoch, paces are updated proportionally to their respective mastery scores:

$$\lambda_{\text{IC}}(t+1) = \min(K, \lambda_{\text{IC}}(t) + \eta_{\text{base}} \cdot (1 + M_{\text{IC}})) \tag{9}$$

$$\lambda_{\text{OA}}(t+1) = \min(M, \lambda_{\text{OA}}(t) + \eta_{\text{base}} \cdot (1 + M_{\text{OA}})) \tag{10}$$

where η_{base} is a base step size and K, M are maximum ranks. The multiplicative factor $(1 + M)$ ensures that dimensions with higher mastery advance more rapidly.

KL-Adaptive Reference. The reference model π_{ref} in DPO serves as a regularization anchor. However, in multi-stage training, a fixed reference can become stale. We adopt a KL-divergence-based update strategy: when $\hat{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) > \delta$, we update $\pi_{\text{ref}} \leftarrow \pi_\theta$.

4.3 Complete Algorithm

Algorithm 1 presents the complete DECOPO training procedure. The algorithm alternates between three phases: (1) activating samples based on current paces, (2) performing standard DPO training on the active set, and (3) updating paces based on mastery assessments.

4.4 Implementation Details

Initialization. We initialize both paces to $\lambda_{\text{IC}}(0) = \lambda_{\text{OA}}(0) = 1.0$, ensuring training begins with the easiest samples along both dimensions.

Computational Overhead. DECOPO introduces modest computational overhead compared to standard DPO. The primary additional cost is computing mastery scores, which requires forward passes on $\mathcal{D}_{\text{low-OA}}$ and $\mathcal{D}_{\text{low-IC}}$ at each epoch boundary. For a 3×3 grid, these subsets constitute roughly 33% of data collectively. Since mastery evaluation needs only forward passes (no gradients), it adds less than 5% to per-epoch time.

5 Experiments

We conduct comprehensive experiments to evaluate DECOPO across three dimensions: alignment quality on standard benchmarks, analysis of emergent

Algorithm 1 DECOPO: Decoupled Pacing Optimization

Require: Dataset $\mathcal{D}$ with IC-OA labels, SFT model π_{SFT}, base step η_{base}, KL threshold δ, total epochs E

Initialize $\pi_\theta \leftarrow \pi_{\text{SFT}}$, $\pi_{\text{ref}} \leftarrow \pi_{\text{SFT}}$
Initialize $\lambda_{\text{IC}} \leftarrow 1.0$, $\lambda_{\text{OA}} \leftarrow 1.0$
for $t = 1$ to E **do**
  $\mathcal{D}^{(t)}_{\text{train}} \leftarrow \{(x, y_w, y_l) \in \mathcal{D} \mid \text{IC}(x) \leq \lambda_{\text{IC}} \wedge \text{OA} \leq \lambda_{\text{OA}}\}$
  for mini-batch $\mathcal{B} \sim \mathcal{D}^{(t)}_{\text{train}}$ **do**
    Compute $\mathcal{L}_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}, \mathcal{B})$
    Update π_θ via gradient descent
  end for
  if $\hat{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) > \delta$ **then**
    $\pi_{\text{ref}} \leftarrow \pi_\theta$
  end if
  Compute M_{IC}, M_{OA} via Equations (7)–(8)
  Update λ_{IC}, λ_{OA} via Equations (9)–(10)
end for
return π_θ

learning trajectories, and robustness properties.

5.1 Experimental Setup

Datasets and Preprocessing. Our primary experiments use the UltraFeedback dataset (Cui et al., 2024), containing approximately 60,000 prompts with GPT-4-scored preference pairs. We compute IC using Llama-2-7B as the external language model, sampling $N = 10$ responses per prompt. OA is derived from existing GPT-4 quality scores. We discretize both dimensions into tertiles ($K = M = 3$), creating a 3×3 difficulty grid.

Models and Baselines. We initialize training from Zephyr-7B-SFT (Tunstall et al., 2024), a supervised fine-tuned version of Mistral-7B (Jiang et al., 2023). Our baselines include: **Standard DPO**, which trains without any ordering; **IC-Only**, which orders by increasing IC; **OA-Only**, which follows the curriculum strategy of Pattnaik et al. (2024) by ordering samples based on response distinguishability; **Diagonal**, which orders by $k + m$, representing a natural baseline for jointly considering both dimensions in a coupled manner; and **SimPO** (Meng et al., 2024), which provides comparison against recent optimization-focused improvements. All curriculum methods use the KL-adaptive reference mechanism for fair comparison.

To validate generalization, we also conduct experiments on Llama-3-8B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2025).

Evaluation Metrics. We assess alignment quality through two complementary benchmarks: MT-Bench (Zheng et al., 2023), which evaluates eight categories via GPT-4-as-judge on 80 multi-turn conversations, and AlpacaEval 2.0 (Li et al., 2023), which measures length-controlled (LC) win rate against GPT-4-Turbo on 805 instructions. Win rates against the SFT baseline are also computed on held-out test sets from UltraFeedback (in-domain), Vicuna (Chiang et al., 2023), and WizardLM (Xu et al., 2024) (out-of-domain).

Training Configuration. All models are trained for three epochs on eight NVIDIA A40 GPUs with bf16 mixed precision. We use AdamW with learning rate 5×10^{-7}, linear warmup over 10% of steps, and cosine decay. Effective batch size is 64, sequence length is 2048, and DPO $\beta = 0.1$. For DECOPO, we set $\eta_{\text{base}} = 0.3$ and $\delta = 0.05$.

5.2 Main Results

Table 2 presents our primary findings. DECOPO achieves 42.3% length-controlled win rate on AlpacaEval 2.0 and 7.66 on MT-Bench (averaged over 3 random seeds), establishing state-of-the-art performance among the compared methods. On AlpacaEval 2.0, DECOPO outperforms Diagonal by 2.1% and OA-Only by 3.7%. This significant margin over Diagonal suggests that the rigid coupling assumption (where IC and OA must progress simultaneously) forces a suboptimal compromise, potentially holding back progress in one dimension while rushing the other. Notably, DECOPO achieves these gains using the standard DPO loss, demonstrating that data organization is a critical, independent factor alongside objective design. We note that SimPO and DECOPO address orthogonal aspects of preference learning, namely optimization objective and data organization, making their combination a promising direction for future work.

Category-wise Analysis. Table 3 provides MT-Bench scores across selected categories. DECOPO's advantage is most pronounced on tasks requiring complex reasoning: Reasoning (7.55 vs. 7.14 for Diagonal), Coding (7.71 vs. 7.45), and Math (6.44 vs. 6.16). These tasks typically involve high Input Complexity, where the underlying logic must be understood before preferences can be

Method	AE2-LC	MT-B	UF	Vic	Wiz
DPO	$35.8_{\pm 0.7}$	$7.08_{\pm 0.05}$	$82.9_{\pm 0.6}$	$83.5_{\pm 0.7}$	$81.2_{\pm 0.8}$
SimPO	$38.2_{\pm 0.6}$	$7.21_{\pm 0.04}$	$84.6_{\pm 0.5}$	$85.8_{\pm 0.6}$	$83.7_{\pm 0.7}$
IC-Only	$36.9_{\pm 0.7}$	$7.15_{\pm 0.05}$	$84.2_{\pm 0.6}$	$85.2_{\pm 0.7}$	$83.4_{\pm 0.7}$
OA-Only	$38.6_{\pm 0.6}$	$7.32_{\pm 0.04}$	$85.8_{\pm 0.5}$	$87.5_{\pm 0.6}$	$85.2_{\pm 0.6}$
Diagonal	$40.2_{\pm 0.6}$	$7.45_{\pm 0.04}$	$86.5_{\pm 0.5}$	$88.8_{\pm 0.5}$	$86.5_{\pm 0.6}$
DECOPO	$\mathbf{42.3}_{\pm 0.5}$	$\mathbf{7.66}_{\pm 0.04}$	$\mathbf{88.1}_{\pm 0.4}$	$\mathbf{90.3}_{\pm 0.5}$	$\mathbf{88.4}_{\pm 0.5}$

Table 2: Main alignment results on Zephyr-7B (mean $\pm$ std over 3 seeds). AE2-LC: AlpacaEval 2.0 length-controlled win rate (%). MT-B: MT-Bench score. UF/Vic/Wiz: win rates (%) against SFT baseline on UltraFeedback, Vicuna, and WizardLM test sets.

Method	Write	Role	Reason	Code	Math
Standard DPO	7.21	7.00	6.50	6.85	5.55
OA-Only	7.47	7.24	6.74	7.22	6.10
Diagonal	7.51	7.33	7.14	7.45	6.16
DECOPO	**7.87**	**7.65**	**7.55**	**7.71**	**6.44**

Table 3: MT-Bench category breakdown showing strong gains on reasoning-intensive tasks.

Model	Method	AE2-LC	MT-B	ΔMT-B
Zephyr-7B	Diagonal	40.2	7.45	–
	DECOPO	42.3	7.66	+0.21
Llama-3-8B	Diagonal	42.0	7.55	–
	DECOPO	44.5	7.78	+0.23
Qwen2.5-7B	Diagonal	42.1	7.65	–
	DECOPO	44.8	7.79	+0.13

Table 4: Cross-model validation showing consistent improvements across different model families.

meaningfully distinguished. By prioritizing input comprehension in the early phase, DECOPO likely prevents the model from overfitting to superficial preference cues on prompts it does not yet fully grasp, thereby building a more robust foundation for STEM-related domains.

Cross-Model Validation. To assess generalization, we evaluate DECOPO on Llama-3-8B and Qwen2.5-7B. Table 4 shows that DECOPO consistently outperforms baselines across model families, confirming that the benefits of decoupled pacing are not model-specific.

Complementarity with Data Selection. As discussed in Section 2, data selection methods address *which* data to use, while DECOPO addresses *when* to present data. Table 5 compares DECOPO with two representative selection methods, margin-based filtering (Deng et al., 2025) and difficulty-based filtering (Gao et al., 2025), as well as their combination with DECOPO. DECOPO substantially outperforms selection-only approaches, and combining difficulty filtering with DECOPO yields the best overall results, confirming that presentation order provides complementary value beyond sample filtering.

Method	AE2-LC	MT-B
Standard DPO (100% data)	35.8	7.08
Margin Filtering (10% data)	37.5	7.22
Difficulty Filtering (50% data)	38.1	7.28
DECOPO (100% data)	42.3	7.66
Difficulty Filtering + DECOPO	**43.1**	**7.72**

Table 5: Complementarity between data selection and DECOPO. Combining difficulty filtering with decoupled pacing yields the best performance.

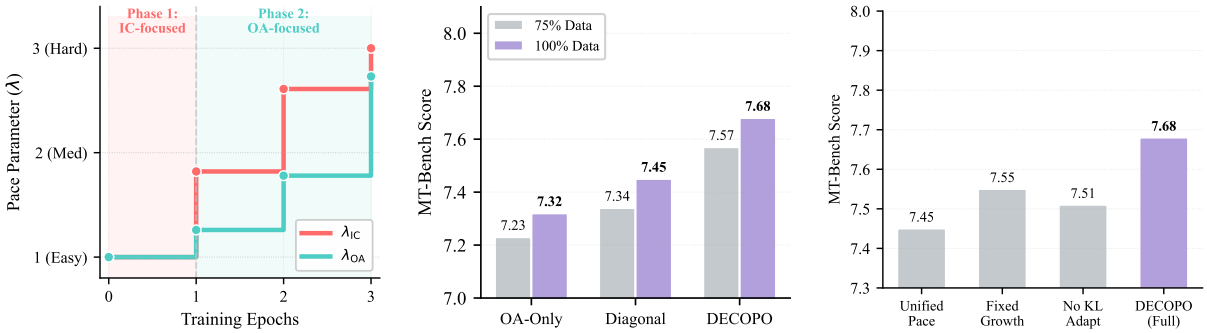
5.3 Emergent Learning Trajectories

A central claim of our work is that DECOPO automatically discovers effective learning trajectories through mastery-based adaptation. Figure 3a tracks the evolution of pace parameters $\lambda_{\text{IC}}(t)$ and $\lambda_{\text{OA}}(t)$.

A distinct two-phase pattern emerges spontaneously. During the first 30% of training, λ_{IC} grows rapidly from 1.0 to 2.4, prioritizing input comprehension. In contrast, λ_{OA} increases slowly. In the second phase, this trend reverses: λ_{OA} accelerates while λ_{IC} plateaus. This trajectory is not hand-crafted but arises naturally from the model's performance, suggesting that establishing prompt understanding creates a necessary foundation for fine-grained preference discrimination.

5.4 Data Efficiency and Robustness

Data Efficiency. We evaluate how effectively DECOPO utilizes training data by subsampling 75% of UltraFeedback. As shown in Figure 3b, DECOPO trained on only 75% of the data achieves an MT-Bench score of 7.57, remarkably surpassing the full-data OA-Only baseline (7.32) and outper-



(a) **Emergent Trajectory** (b) **Data Efficiency** (c) **Ablation Study**

Figure 3: Analysis of learning dynamics and performance drivers. **(a)** The mastery-based schedule naturally discovers a two-phase pattern: rapid λ_{IC} growth followed by accelerated λ_{OA} expansion. **(b)** DECOPO using only 75% data outperforms full-data baselines like OA-Only and matches Diagonal. **(c)** Removing decoupled pacing or adaptivity significantly drops performance, verifying the core design principles.

Method	**0%**	**10%**	**20%**	**30%**
Standard DPO	7.08	6.92	6.83	6.61
OA-Only	7.32	7.21	7.12	6.88
Diagonal	7.45	7.36	7.28	7.05
DECOPO	**7.66**	**7.58**	**7.49**	**7.29**

Table 6: MT-Bench scores under label noise injection in high-OA samples. DECOPO exhibits the smallest degradation across all noise levels.

forming the Diagonal baseline (7.45). On AlpacaEval 2.0 (see Appendix), this partial-data model achieves 40.8%, beating most full-data competitors. This confirms that structured exploration yields higher information gain per sample.

Noise Robustness. Real-world preferences often contain noise. We inject label noise at varying rates into high-OA samples and measure MT-Bench degradation. Table 6 reports the results. DECOPO exhibits the smallest degradation at every noise level because during the early IC-focused phase, training operates primarily on low-OA samples where noise was not injected, building a solid foundation before encountering potentially noisy high-OA data.

5.5 Ablation Studies

Figure 3c quantifies the contribution of each component. The most significant drop occurs when replacing independent pacing with a unified schedule ($\lambda_{\text{IC}} = \lambda_{\text{OA}}$), lowering the score to 7.45. This validates our core hypothesis that input and output difficulties are orthogonal. Using a fixed linear growth schedule instead of mastery-based updates results in a score of 7.55, showing that adaptivity adds meaningful value. Finally, removing the KL-adaptive reference mechanism reduces stability, dropping the score to 7.51.

6 Conclusion

We identify Input Complexity and Output Ambiguity as distinct sources of alignment difficulty that exhibit asymmetric learning dynamics. To leverage this insight, we introduce DECOPO, a framework employing independent, mastery-based pacing for each dimension. Our method achieves state-of-the-art performance on AlpacaEval 2.0 and MT-Bench while demonstrating superior data efficiency and noise robustness. The naturally emerging two-phase learning trajectory validates the effectiveness of our structured difficulty treatment. Future work will extend these principles to safety alignment and uncertainty estimation.

Limitations

Our experiments focus primarily on 7B-parameter models due to computational constraints. While cross-model validation shows consistent benefits, experiments on larger models (70B+) would strengthen claims about scalability. Additionally, our IC and OA metrics rely on external models (Llama-2 for IC, GPT-4 for OA); the framework's effectiveness may vary with different metric implementations. Finally, GPT-4 is used both for computing OA and for MT-Bench evaluation, which could introduce systematic biases. We partially address this by including AlpacaEval 2.0 as an additional evaluation benchmark, and Appendix F reports robustness checks using Claude as an alternative judge.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, pages 4447–4455.

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

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*, pages 41–48.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90% ChatGPT quality. *See https://vicuna.lmsys.org*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *NeurIPS*, volume 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, and 1 others. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *ICML*, pages 9722–9744. PMLR.

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

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

Chengqian Gao, Haonan Li, Liu Liu, Zeke Xie, Peilin Zhao, and Zhiqiang Xu. 2025. Principled data selection for alignment: The hidden risks of difficult examples. In *ICML*, pages 18386–18409.

Aaron Grattafiori and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. In *EMNLP*, pages 11170–11189.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, and 1 others. 2023. Camels in a changing climate: Enhancing LM adaptation with Tulu 2. *arXiv preprint arXiv:2311.10702*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander Hauptmann. 2015. Self-paced curriculum learning. In *AAAI*, volume 29.

H Toprak Kesgin and M Fatih Amasyali. 2023. Cyclical curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):12864–12872.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *NeurIPS*, pages 1189–1197.

Mengyang Li and Pinlong Zhao. 2026. Difficulty-aware learning curve extrapolation. In *AAAI*, pages 23021–23029.

Mengyang Li, Pinlong Zhao, and Zhong Zhang. 2026. Aligner, diagnose thyself: A meta-learning paradigm for fusing intrinsic feedback in preference alignment. In *ICLR*.

Mengyang Li, Xiaoling Zhou, and Ou Wu. 2025. Delving into the training dynamics for image classification. *IEEE Transactions on Image Processing*, 34:6783–6798.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, and 1 others. 2025. Lipo: Listwise preference optimization through learning-to-rank. In *NAACL*, pages 2404–2420.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2019. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *NeurIPS*, volume 37, pages 124198–124235.

Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. 2024. Filtered direct preference optimization. In *EMNLP*, pages 22729–22770.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. 2024. Active preference learning for large language models. In *ICML*, pages 36577–36590. PMLR.

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. In *NeurIPS*, volume 35, pages 27730–27744.

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Enhancing alignment using curriculum learning & ranked preferences. In *EMNLP*, pages 12891–12907.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, volume 34, pages 20596–20607.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL*, pages 1162–1172.

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. In *NeurIPS*, volume 36, pages 53728–53741.

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

Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *ACL*, pages 453–463.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. 2025. Overtrained language models are harder to fine-tune. In *ICML*, pages 56719–56789. PMLR.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*, volume 33, pages 3008–3021.

Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2024. Zephyr: Direct distillation of LM alignment. In *COLM*.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *ACL*, pages 6095–6104.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *ICLR*.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, and 1 others. 2025. Qwen2. 5-1m technical report. *arXiv preprint arXiv:2501.15383*.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *NAACL*, pages 1903–1915.

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. In *NeurIPS*, volume 36, pages 46595–46623.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023. Lima: Less is more for alignment. In *NeurIPS*, volume 36, pages 55006–55021.

A Additional Implementation Details

This section provides supplementary details regarding metric computation, data processing, and the training infrastructure used in our experiments.

A.1 IC Computation Specifics

To quantify Input Complexity (IC), we utilize Llama-2-7B-base as the external estimator p_{LM}. We employ temperature sampling ($T = 0.9$) combined with nucleus sampling (top-p $= 0.95$) to generate diverse responses. For each prompt, we sample $N = 10$ completions. This number was chosen empirically to balance estimation stability with computational efficiency. Perplexity is calculated at the token level using the model's native tokenizer. To prevent numerical instability from outliers, we clip perplexity values to the range $[1.0, 1000.0]$ prior to computing the standard deviation across samples.

A.2 OA Computation from GPT-4 Scores

Output Ambiguity (OA) is derived from the continuous quality scores provided in the UltraFeedback dataset. We define OA as the inverse of the score margin between the preferred and rejected responses. Formally, given a judge model S_{judge}, the ambiguity is computed as:

$$\text{OA}(y_w, y_l|x) = \frac{1}{|S_{\text{judge}}(y_w|x) - S_{\text{judge}}(y_l|x)| + \epsilon} \tag{11}$$

where $\epsilon = 0.01$ ensures numerical stability. This formulation maps small score differences (high uncertainty/similarity) to large ambiguity values, consistent with our definition of discrimination difficulty.

A.3 Discretization and Binning

To construct the curriculum grid, we employ quantile-based binning rather than absolute thresholds. We compute the K-quantiles for IC and M-quantiles for OA across the entire training corpus. This ensures a uniform data distribution where each cell in our 3×3 grid contains approximately 11.1% of the total samples. This balance prevents any single difficulty region from becoming a bottleneck during the pacing progression.

A.4 Training Infrastructure

All models were trained on a cluster of eight NVIDIA A40 GPUs (48GB VRAM) interconnected via NVLink. The software environment includes PyTorch 2.1.0, CUDA 12.1, and Transformers 4.36.0. We utilize DeepSpeed ZeRO Stage 2 to optimize memory usage during distributed training. The effective batch size is maintained at 64 via gradient accumulation. With automatic mixed precision (bf16), a single epoch of DECOPO on the UltraFeedback dataset requires approximately 2.5 hours.

B Optimization Landscape Analysis via Hessian Spectrum

To investigate the theoretical basis for asymmetric learning, we analyze the local geometry of the loss landscape. Specifically, we approximate the Hessian matrix of the loss function, which describes the curvature of the optimization surface. High curvature implies a difficult optimization landscape prone to instability.

B.1 Methodology

Since computing the full Hessian for a 7B-parameter model is intractable, we approximate its top eigenvalues using the Lanczos algorithm. These principal eigenvalues capture the sharpest directions of curvature, which are the primary determinants of optimization difficulty.

Lanczos Algorithm. We estimate the spectrum of the symmetric Hessian matrix H by iteratively constructing an orthonormal basis for the Krylov subspace $\mathcal{K}_k(H, v)$. We perform $k = 100$ iterations, which yields a stable approximation of the extreme eigenvalues. To avoid explicit Hessian instantiation, we compute Hessian-vector products using automatic differentiation:

$$Hv = \nabla_\theta \left(\nabla_\theta \mathcal{L} \cdot v\right) \tag{12}$$

For each of the nine difficulty regions in our grid, we sample 500 preference pairs and compute statistics over mini-batches of 32 samples. Results are averaged over 5 random initializations.

B.2 Full Spectrum Statistics

Table 7 details the spectral properties for each grid cell. We report the maximum eigenvalue ($\lambda_{\max}$), the minimum positive eigenvalue ($\lambda^+_{\min}$), and the condition number ($\kappa = \lambda_{\max}/\lambda^+_{\min}$).

B.3 Interpretation

The data highlights a clear asymmetry: curvature is primarily driven by Input Complexity. As shown in the table, $\lambda_{\max}$ nearly doubles from 0.83 (Low

IC	OA	$\lambda_{\max}$	$\lambda^{+}_{\min}$	Cond. Num.
Low	Low	0.83 ± 0.07	0.020 ± 0.003	42.3 ± 5.1
Low	Mid	0.88 ± 0.08	0.018 ± 0.003	48.9 ± 6.2
Low	High	0.91 ± 0.08	0.018 ± 0.002	51.2 ± 6.3
Mid	Low	1.12 ± 0.10	0.019 ± 0.003	58.9 ± 7.1
Mid	Mid	1.18 ± 0.11	0.017 ± 0.002	69.4 ± 8.0
Mid	High	1.24 ± 0.12	0.016 ± 0.002	77.5 ± 8.8
High	Low	1.42 ± 0.12	0.018 ± 0.003	78.6 ± 8.2
High	Mid	1.51 ± 0.14	0.017 ± 0.002	88.8 ± 9.4
High	High	1.58 ± 0.15	0.018 ± 0.003	89.4 ± 9.7

Table 7: Hessian spectral statistics across the difficulty grid. $\lambda_{\max}$ indicates curvature sharpness, while Condition Number reflects optimization instability.

IC) to 1.42 (High IC) even when Output Ambiguity remains low. In contrast, increasing OA causes only minor increases in spectral norm. This provides a mechanistic explanation for our curriculum design: high-IC samples present a sharper, more treacherous optimization landscape, necessitating a cautious, mastery-based approach.

C Extended Experimental Results

In this section, we provide a more granular analysis of model performance, statistical reliability, and generalization capabilities. These results collectively reinforce that DECOPO's improvements are not limited to specific metrics or datasets but stem from a fundamental optimization of the learning process.

C.1 Detailed Benchmarking by Category

Table 8 presents the breakdown of MT-Bench scores across eight distinct capabilities. A closer examination reveals a clear pattern aligned with our theoretical framework.

DECOPO achieves its largest margins over the Diagonal baseline in categories that demand rigorous logic and structural understanding: Coding (+0.26), Math (+0.28), and Reasoning (+0.41). In standard DPO training, the model often attempts to learn preference signals before fully mastering the underlying logic of complex prompts, leading to superficial alignment. By prioritizing Input Complexity early in training, DECOPO ensures the model establishes a robust reasoning foundation before optimizing for stylistic preferences.

Interestingly, we also observe gains in creative tasks like Roleplay (+0.32). This suggests that even for open-ended queries, a better grasp of the prompt's nuances (IC) allows for more precise adherence to character constraints (OA).

C.2 Statistical Significance Testing

Given the variability inherent in LLM evaluation, it is crucial to verify that our improvements are statistically robust and not artifacts of random seed selection or evaluation noise. We employed paired bootstrap resampling (10,000 iterations) on the MT-Bench and AlpacaEval result sets.

Table 9 confirms that DECOPO's lead is statistically significant at the $p < 0.01$ level against all baselines. Notably, the improvement over the Diagonal baseline ($p < 0.01$) is particularly important; since both methods use curriculum learning, this significance isolates the specific contribution of *decoupling* the pacing schedules versus simply ordering data by difficulty.

C.3 Cross-Dataset Generalization

Our main experiments utilize UltraFeedback, a dataset characterized by high-quality, AI-generated preference labels. To ensure our findings generalize to human-annotated data distributions, we trained models on the Anthropic Helpful-Harmless (HH) dataset.

As shown in Table 10, DECOPO maintains its superiority, achieving an 89.1% win rate on the Vicuna benchmark. This result is significant because the HH dataset contains very different prompt distributions (often focusing on safety and refusal) compared to UltraFeedback. The consistent gains confirm that the IC-OA decomposition captures intrinsic properties of language learning difficulty, rather than overfitting to the specific quirks of a single dataset.

D IC/OA Metric Robustness

To evaluate the sensitivity of DECOPO to the specific choice of IC and OA metrics, we tested alternative formulations while keeping all other settings identical.

D.1 Alternative IC Definitions

We compared three IC formulations: the default perplexity standard deviation, an entropy-based variant (output distribution entropy from a single forward pass), and an embedding-diversity variant (response embedding variance in the model's representation space). Table 11 reports the results. All three IC definitions yield consistent improvements over baselines, with differences among IC variants being much smaller than the gap between DECOPO and any baseline. The entropy-based variant

Method	Write	Role	Reason	Code	Math	Extract	STEM	Human
Standard DPO	7.21	7.00	6.50	6.85	5.55	7.75	7.68	8.10
SimPO	7.32	7.12	6.62	7.02	5.72	7.85	7.78	8.25
IC-Only	7.35	7.15	6.70	7.10	5.85	7.62	7.55	7.88
OA-Only	7.47	7.24	6.74	7.22	6.10	7.88	7.76	8.15
Diagonal	7.51	7.33	7.14	7.45	6.16	7.95	7.85	8.21
DECOPO	**7.87**	**7.65**	**7.55**	**7.71**	**6.44**	**8.02**	**7.95**	**8.25**

Table 8: Complete MT-Bench results across all eight evaluation categories. The decoupling strategy yields the most significant benefits in logic-intensive domains (Reasoning, Coding, Math) where input understanding is a prerequisite for preference discrimination.

Comparison	Δ MT-B	Δ AE2	p-value
DECOPO vs. Diagonal	+0.21	+2.1%	< 0.01
DECOPO vs. OA-Only	+0.36	+3.7%	< 0.01
DECOPO vs. SimPO	+0.47	+4.3%	< 0.001
DECOPO vs. Standard DPO	+0.60	+6.7%	< 0.001

Table 9: Statistical significance of improvements over baselines using paired bootstrap hypothesis testing.

Method	AE2-LC	MT-B	HH-WR	Vic-WR
Standard DPO	32.5	6.92	78.3	80.1
OA-Only	35.8	7.18	82.5	85.2
Diagonal	37.2	7.25	83.1	85.8
DECOPO	**41.6**	**7.53**	**86.4**	**89.1**

Table 10: Results on Anthropic HH dataset. Consistency across diverse data sources confirms that independent pacing is a generalizable alignment principle.

is particularly noteworthy as it requires only a single forward pass per prompt, avoiding the $N = 10$ sampling cost of the default formulation.

IC Variant	MT-Bench	AE2-LC
Perplexity StdDev (default)	7.66	42.3
Entropy-based	7.61	41.8
Embedding-diversity	7.63	42.0
Diagonal baseline	7.45	40.2

Table 11: DECOPO performance with alternative IC formulations. All variants maintain substantial improvements over baselines.

D.2 Alternative OA Judge

To verify that DECOPO is not tightly coupled to GPT-4's specific scoring tendencies, we replaced the GPT-4-derived OA scores with those from Llama-3-70B-Instruct as the judge. DECOPO achieves MT-Bench 7.62 with this alternative OA estimator, confirming the framework's robustness to the choice of judge model.

D.3 IC-OA Correlation Analysis

We computed the correlation between IC and OA across the full UltraFeedback dataset. The Pearson correlation is $r = 0.14$ and the Spearman correlation is $\rho = 0.11$, supporting that the two dimensions are approximately orthogonal in practice. This low correlation holds regardless of the specific IC variant used, confirming that IC and OA capture genuinely distinct aspects of sample difficulty.

E Computational Cost Analysis

Table 12 provides a complete end-to-end cost comparison on 8×A40 GPUs. The IC preprocessing is a one-time cost amortized across all subsequent experiments using the same dataset. OA computation incurs zero additional cost since it is derived from existing GPT-4 quality scores in UltraFeedback. The DECOPO training loop itself adds only 5.3% overhead versus standard DPO due to mastery evaluation at epoch boundaries.

Stage	GPU Hours
IC Preprocessing (one-time)	4.2h
OA Computation	0h (existing scores)
Standard DPO Training (3 epochs)	7.5h
DECOPO Training (3 epochs)	7.9h (+5.3%)
Total: Standard DPO	7.5h
Total: DECOPO (incl. preprocessing)	12.1h

Table 12: End-to-end computational cost breakdown on 8×A40 GPUs. The preprocessing overhead is a fixed one-time cost that does not scale with the number of subsequent experiments.

For settings where preprocessing budget is constrained, the entropy-based IC variant (Appendix D) provides a substantially cheaper alternative, requiring only a single forward pass per prompt while achieving MT-Bench 7.61.

F Robustness to Evaluation Judge

A recurring concern in recent alignment research is self-preference bias, where models scored by GPT-4 may be favored simply because they were trained on GPT-4 preferences. Furthermore, since our OA metric is derived from GPT-4 scores, there is a risk of circularity.

To address this, we re-evaluated all models using Claude-3.5-Sonnet as an independent judge. Table 13 demonstrates that while the absolute scores differ between judges (Claude tends to be slightly stricter), the relative ranking remains identical. DECOPO outperforms the Diagonal baseline by 0.16 points under Claude's evaluation, nearly matching the 0.21 margin under GPT-4. This stability suggests that DECOPO improves the fundamental quality of responses—such as reasoning coherence and instruction following—rather than merely gaming specific judge heuristics.

Method	GPT-4 Judge	Claude Judge
Standard DPO	7.08	6.95
OA-Only	7.32	7.12
Diagonal	7.45	7.21
DECOPO	**7.66**	**7.37**
DECOPO Δ vs. Diagonal	+0.21	+0.16

Table 13: Comparison of MT-Bench scores using GPT-4-Turbo vs. Claude-3.5-Sonnet as judges. The performance advantage is robust to the choice of evaluator.

G Region-Stratified Analysis

To pinpoint the source of DECOPO's effectiveness, we analyzed performance improvements across the difficulty grid. Figure 4 and Table 14 display the win-rate gain of DECOPO over the OA-Only baseline.

The results reveal a stark contrast between regions. In the Low-IC regime (top rows), the improvements are modest (+1.5% to +3.2%). This is expected: when prompts are simple, the model already possesses sufficient understanding, so the specific curriculum order matters less.

However, in the High-IC regime (bottom row), we observe dramatic gains of up to +12.1%. This confirms our core hypothesis: when the input is complex, standard training methods (or those focusing only on output ambiguity) fail because they force the model to optimize preferences on inputs it does not yet comprehend. DECOPO's decoupled pacing delays these samples until the model achieves sufficient mastery, preventing the learning of spurious correlations and resulting in significantly better alignment on complex tasks.

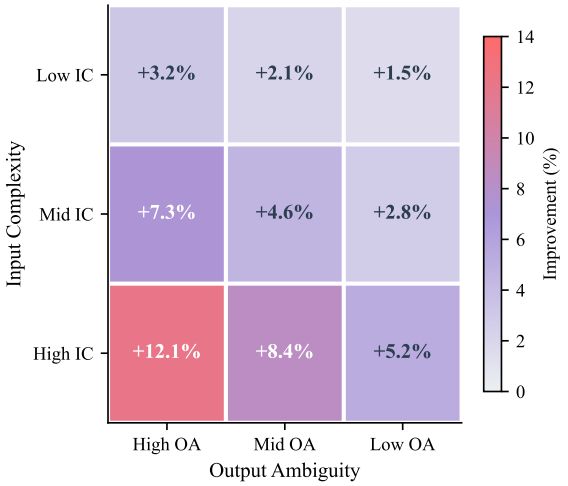


Figure 4: Win rate improvements of DECOPO over OA-Only across the IC-OA grid. The performance gap widens as Input Complexity increases, validating the necessity of mastering input understanding before preference discrimination.

	Low OA	Mid OA	High OA
Low IC	+1.5%	+2.1%	+3.2%
Mid IC	+2.8%	+4.6%	+7.3%
High IC	+5.2%	+8.4%	**+12.1%**

Table 14: Numeric breakdown of win rate improvement over OA-Only. The largest gains occur when both Input Complexity and Output Ambiguity are high.

H Hyperparameter Sensitivity Analysis

The robustness of a training framework is determined by its sensitivity to hyperparameter choices. We investigate three key parameters: the base step size (η_{base}), the KL-update threshold (δ), and the grid granularity.

Base Step Size (η_{base}). This parameter controls the minimum speed of curriculum expansion. As shown in Table 15, performance is stable within the range $[0.2, 0.4]$. When η_{base} is set below 0.2, the curriculum unfolds too slowly and effectively discards hard data samples within the fixed epoch budget, leading to underfitting. Conversely, setting it above 0.4 causes the curriculum to expand too aggressively, approaching the behavior of standard DPO where all data becomes active immediately, which negates the pacing benefits.

η_{base}	MT-Bench	AE2-LC
0.1	7.53 ± 0.04	40.5 ± 0.8
0.2	7.63 ± 0.04	41.3 ± 0.7
0.3	$\mathbf{7.66 \pm 0.04}$	$\mathbf{42.3 \pm 0.5}$
0.4	7.61 ± 0.04	41.8 ± 0.7
0.5	7.50 ± 0.05	40.2 ± 0.9

Table 15: Sensitivity to base step size. The method is robust across a reasonable range of pacing speeds (mean $\pm$ std over 3 seeds).

KL Threshold (δ). This threshold dictates when to refresh the reference model. Table 16 indicates an optimal range of $[0.03, 0.07]$. Setting δ too low triggers frequent updates, which can destabilize the implicit reward function and lead to reward hacking. Conversely, a high threshold leaves the reference model stale, reducing the regularization effect and causing the policy to drift too far from the base distribution.

δ	MT-Bench	AE2-LC
0.02	7.58 ± 0.04	40.8 ± 0.8
0.03	7.62 ± 0.03	41.5 ± 0.7
0.05	$\mathbf{7.66 \pm 0.04}$	$\mathbf{42.3 \pm 0.5}$
0.07	7.64 ± 0.03	41.9 ± 0.7
0.10	7.55 ± 0.04	41.2 ± 0.8

Table 16: Sensitivity to KL threshold. Moderate update frequencies yield the best balance between stability and plasticity.

Grid Granularity. We discretize the continuous IC-OA space into a grid. We found that a 3×3 grid offers the best trade-off (Table 17). A 2×2 grid is too coarse to separate distinct difficulty levels effectively. A 4×4 grid, while offering higher resolution, results in fewer samples per cell ($\sim$3,750), which increases the variance of the mastery estimation and leads to noisier pacing updates.

Grid	MT-B	AE2-LC	Samples/Cell
2×2	7.58 ± 0.04	42.2 ± 0.8	$\sim$15,000
3×3	$\mathbf{7.66 \pm 0.04}$	$\mathbf{42.3 \pm 0.5}$	$\sim$6,700
4×4	7.61 ± 0.04	42.1 ± 0.7	$\sim$3,750

Table 17: Impact of grid granularity. 3×3 provides sufficient resolution while maintaining statistical reliability in mastery estimation.

I Ethical Considerations and Broader Impacts

Societal Impact and Dual Use. While our work aims to align models with human values to improve helpfulness and reasoning, preference optimization methods are inherently agnostic to the specific values being aligned. There is a risk that the efficient alignment techniques proposed in DECOPO could be potentially misused to align models with malicious instructions or harmful values. We adhere to the safety guidelines of the UltraFeedback and Anthropic HH datasets and emphasize the importance of safety filtering in the data curation process.

Bias and Evaluation Limitations. Our methodology relies on Input Complexity (IC) and Output Ambiguity (OA) metrics derived from specific models (Llama-2 and GPT-4). We acknowledge that these metrics may reflect the inherent biases of the underlying estimator models. Furthermore, employing LLMs as judges (e.g., GPT-4) serves as a scalable proxy for human preference but may not fully capture the diversity of human values across different cultural backgrounds or handle subtle safety nuances perfectly.

Environmental Impact. A key contribution of DECOPO is improved data efficiency, achieving state-of-the-art performance with only 75% of training samples compared to standard baselines. This reduction in required training steps contributes to lower energy consumption and a reduced carbon footprint for model alignment, promoting more sustainable AI development practices.

AI Assistance Declaration. In accordance with the ACL AI Assistance Policy, we acknowledge the use of AI assistants for grammatical error correction and text polishing during the preparation of this manuscript. The authors have reviewed all generated content and remain fully responsible for the accuracy, integrity, and originality of the paper.