MRACL: Multi-Reward Space Guided Adaptive Curriculum Reinforcement Learning for LLMs

Wenxuan Liu*, Liangyu Huo*†, Yi Jing, Xiyuan Zhang, Jian Xie

Du Xiaoman Financial, Beijing, China
liuwenxuan@duxiaoman.com, huoliangyu@duxiaoman.com

Abstract

Reinforcement learning (RL) has recently become a powerful yet resource-intensive approach for post-training large language models (LLMs). Incorporating curriculum learning (CL) into RL has been shown to significantly improve training efficiency, particularly in reasoning tasks. However, existing CL methods face substantial challenges in multi-objective RL (MORL) settings, including: (1) difficulty in evaluating model capabilities online, (2) challenges in assessing sample importance under diverse objectives, and (3) inherent trade-offs between online training and offline inference in dynamically designing the curriculum. To address these issues, we propose a Multi-Reward space guided Adaptive Curriculum Learning framework (MRACL), which is the first to incorporate curriculum learning into multi-objective RL. MRACL first constructs a multi-dimensional reward space via offline inference to establish initial reward profiles for each training sample. During training, based on reward space, it estimates the evolving model capabilities by computing the centroid of the space and calculates the sample priority score through its capability distance, optimization direction, and historical evolution, which enables adaptive selection of the most informative training samples at each step, independent of the specific RL algorithm. After each RL training iteration, the reward space is dynamically updated to reflect the model's evolving capabilities and the shifting distribution of sample priorities. Experiments on multi-objective alignment tasks demonstrate that MRACL achieves 1.62× faster convergence compared to state-of-the-art curriculum methods and 2.55× faster than non-curriculum methods. Furthermore, it consistently outperforms all baselines in both win rate and rule-based evaluation. We further provide an in-depth analysis of the key factors contributing to MRACL's effectiveness, along with its advantages, scenarios, and generalization across diverse settings.

Introduction

Reinforcement learning (RL) has recently become a powerful method for post-training large language models (LLMs). By generating multiple rollouts for each prompt and evaluating them with reward models, RL effectively steers the actor model toward better alignment with the desired objectives. However, the exponentially large trajectory space results in sparse and high-variance gradient estimates, leading to unstable optimization and increasing the computational cost.

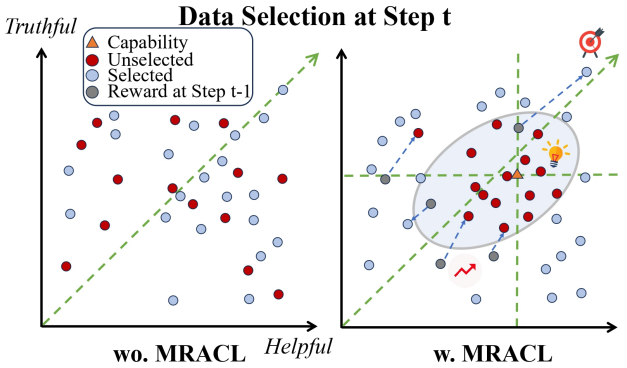

Figure 1: The data selection at step t with MRACL.

To enhance training efficiency and stability, recent studies have incorporated curriculum learning (CL) (Bengio et al. 2009) into reasoning tasks under single-dimensional reward settings. CL aims to guide models to learn from progressively more challenging examples, with the progression determined by model performance (Wang, Chen, and Zhu 2021). Existing approaches to CL can be broadly categorized into predefined (Wen et al. 2025) and adaptive curricula (Wang et al. 2025; Shi et al. 2025), both of which typically leverage external evaluation metrics (e.g., accuracy, pass rate) or internal training signals (e.g., reward, advantage) to design curriculum. However, real-world tasks such as alignment often involve multiple, and sometimes conflicting, objectives (e.g., helpfulness and harmlessness), making the single-metric heuristic curriculum insufficient.

Although recent efforts, such as Curri-DPO (Pattnaik et al. 2024), Curriculum-RLAIF (Li et al. 2025b), have applied it to DPO or reward model training by predefining the ranks of pairs, extending adaptive curriculum learning to multi-objective reinforcement learning (MORL) presents several unique challenges: 1) How to dynamically evaluate model capability using only internal training signals, without relying on external metrics; 2) How to establish reliable and interpretable sample evaluation criteria, particularly in the presence of potentially conflicting objectives; 3) How to

*These authors contributed equally.
†Corresponding Author

adaptively manage the trade-off between online training and offline inference to enable efficient curriculum design.

To address these challenges, we propose MRACL (**M**ulti **R**eward space guided **A**daptive **C**urriculum **L**earning), the first framework that enables adaptive curriculum learning in multi-objective settings by leveraging a persistently maintained multi-dimensional reward space. Specifically, MRACL constructs the reward space through offline inference, preserving the reward profile of each training sample. Based on the reward space, we estimate the evolving capability of the actor by computing the centroid of the reward space. Each sample is then prioritized by evaluating its distance from the current model capability, while considering optimization direction and historical evolution within the reward space. Subsequently, MRACL adaptively selects the most informative training samples at each step, agnostic to the specific RL algorithm. The Figure 1 illustrates the selection process. After each iteration, the space is updated with newly inferred reward vectors to capture both the model's progress and the priority distribution of the samples, which offers a stable and informative signal for curriculum design.

We evaluate MRACL on the multi-objective alignment task. Training on UltraFeedback (Cui et al. 2024) with Reinforce++ (Hu 2025), MRACL achieves 1.62× faster convergence compared to the state-of-the-art CL method DUMP and 2.55× faster than the non-CL method. Additionally, it surpasses DUMP by 5.9% in win rate evaluation and achieves best results on 6 out of 7 rule-based benchmarks for final performance. To better understand the effectiveness of MRACL, we provide a detailed data selection analysis. We further explore its strengths and applicable scenarios through rollout and sensitivity analysis. Besides, we conduct Pareto efficiency and generality analyses across different settings. Our contributions are summarized as follows:

- We propose the first curriculum learning framework MRACL for MORL, which integrates offline reward space construction with online adaptive updates. This design effectively addresses key challenges in dynamically evaluating model capabilities and assessing sample priorities.
- MRACL demonstrates faster convergence and consistently outperforms state-of-the-art methods on multi-objective alignment tasks across various evaluations.
- We perform a comprehensive analysis of MRACL, highlighting its key success factors, applicable scenarios, and generalizability in different experimental settings.

Methodology

We begin by describing the initialization of reward space, followed by the data selection process. We then detail the RL training and the mechanism for updating the reward space. The workflow is summarized in Algorithm 1.

Reward Space Initialization

In curriculum learning, it is crucial to master the model's capability distribution over the training data, especially when the data involves multiple objectives. To this end, we propose constructing a reward space to represent this distribution. Considering the high computational cost of estimating this distribution online, we instead perform offline inference before training to approximate the distribution efficiently.

According to the reward design, we define the optimization range for each reward dimension using two vectors:

$$R_{\min} = [r^1_{\min}, \ldots, r^n_{\min}],\ R_{\max} = [r^1_{\max}, \ldots, r^n_{\max}], \quad (1)$$

where $r^i_{\min}$ and $r^i_{\max}$ denote the minimum and maximum reward values for the i-th dimension. Then, for each question q_i in the dataset $\mathcal{D}$, the actor model π is instructed to generate m responses prior to training (referred to as step 0). Each response j is evaluated by a reward model RM across n distinct reward objectives, resulting in a multi-reward vector:

$$R_{i,j,0} = [r^1_{i,j,0}, r^2_{i,j,0}, \ldots, r^n_{i,j,0}], \quad (2)$$

To obtain a stable estimate of the sample's reward, we compute the average reward across the m responses:

$$R_{i,0} = \frac{1}{m} \sum_{j=1}^{m} R_{i,j,0}. \quad (3)$$

The vector $R_{i,0}$ represents the initial expected performance of the sample. By aggregating all reward vectors in dataset, we construct the initial reward space, which can be incrementally updated through training to reflect model progress.

Data Selection

Capability Estimation Based on reward space, we estimate the model capability at step t by computing the centroid of reward space. Formally, the capability is defined as:

$$R_{\text{mean},t} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} R_{i,t}, \quad (4)$$

where $R_{i,t}$ denotes the most recent reward vector of sample q_i. In practice, performing inference on the entire training dataset at every step is computationally prohibitive. To address this, $R_{\text{mean},t}$ is used as a coarse yet informative approximation of the model's evolving capability. It offers a lightweight, reward space-level tracking of training dynamics without relying on external evaluation signals, while still capturing progress across multiple objectives.

Sample Assessment To identify samples that best align with the model's current capabilities, especially in MORL, we introduce a metric termed **Selection Priority (SP)** to quantify the potential learning contribution of each sample. At each training step, our objective is to prioritize samples that are most likely to yield immediate performance improvements, thereby accelerating training efficiency. Theoretically, such samples are expected to produce informative gradients with lower variance and higher expected returns, as supported by the policy gradient theorem (Sutton et al. 2000). We assess sample priority from three perspectives:

Capability Distance The samples whose difficulty is matched to the model current capability are most suitable for learning, particularly during the early stages, when the model has not yet adapted to the task. At step t, the model

capability is defined as $R_{\text{mean},t}$. To quantify this distance, we define **capability distance**, denoted $W_{\text{Cap}}(q_i, \pi_t)$ as :

$$W_{\text{Cap}}(q_i, \pi_t) = \|R_{i,t} - R_{\text{mean},t}\|^2, \quad (5)$$

where $\|\cdot\|$ denotes the 2-norm. A smaller value indicates that the sample aligns well with the model's current competence.

Optimization Direction In multi-objective settings, it is crucial to avoid samples that deviate from the joint optimization direction, especially in the presence of conflicting objectives. To this end, we prioritize samples that both (i) align with the target optimization direction from $R_{\min}$ to $R_{\max}$, and (ii) remain under-optimized (i.e., still close to $R_{\min}$), retaining potential for further improvement. Based on this intuition, we thus define the **optimization direction** metric:

$$W_{\text{Opt}}(q_{i,t}) = \frac{\|R_{i,t} - R_{\min}\|^2}{\cos^2\theta + \epsilon}, \quad (6)$$

where θ denotes the angle between the sample's reward vector $R_{i,t} - R_{\min}$ and the target optimization direction $R_{\max} - R_{\min}$. The distance serves as a penalty term to assess whether the sample remains under-optimized.

Historical Evolution Inspired by self-paced curriculum learning (Jiang et al. 2014), which adaptively selects training samples based on the loss dynamics to estimate learning potential, we extend it to RL by leveraging reward dynamics. Empirically, we observed that the reward changes between training steps are small yet informative. If a sample exhibits an increase in reward, it is likely to continue contributing positively to subsequent updates. Conversely, samples with consistently stagnant reward are likely either over-optimized or inherently difficult, contributing little to further learning. To quantify this trend, we define the **historical evolution**, which measures the change in reward between two consecutive training steps:

$$W_{\text{Evo}}(q_{i,t}) = \sum_{j=1}^{n} sgn(r^j_{i,t} - r^j_{i,t-1}) \cdot (r^j_{i,t} - r^j_{i,t-1})^2, \quad (7)$$

where t and $t-1$ denote the current and previous steps, and j indexes the dimension. The $sgn(r^j_{i,t} - r^j_{i,t-1})$ captures whether it increases or decreases, and is denoted as λ_j.

Combining the three components described above, we define the **SP** score for each sample as:

$$\text{SP}(q_i, \pi_t) = \alpha_t W_{\text{Cap}} + \beta_t W_{\text{Opt}} - \gamma W_{\text{Evo}}, \quad (8)$$

where the coefficients α_t and β_t are defined as:

$$\alpha_t = 1 - \frac{t}{T}, \quad \beta_t = \frac{t}{T}, \quad (9)$$

with T denoting the total training steps. This design ensures that W_{Cap} is emphasized in early training, while W_{Opt} becomes more dominant as training progresses. The parameter γ is treated as a fixed hyperparameter, capturing reward changes at all time to reflect the evolution. A lower value of $\text{SP}(q_i, \pi_t)$ indicates that the sample is better aligned with the model's capabilities and likely to provide training benefit. We conduct further analysis to validate its reasonableness.

Exploitation and Exploration To strike a balance between **exploitation** (prioritizing samples that are easier to learn and consistently improve) and **exploration** (mitigating overfitting by occasionally sampling less frequently seen samples), we follow (Wang et al. 2025) and formulate the data selection process as a multi-armed bandit problem, employing the Upper Confidence Bound (UCB) strategy.

First, we normalize the priority scores across the dataset $\mathcal{D}$ using min-max scaling to $[0, C]$:

$$\widetilde{\text{SP}}(q_i, \pi_t) = C \cdot \frac{\text{SP}(q_i, \pi_t) - \min_q \text{SP}(q, \pi_t)}{\max_q \text{SP}(q, \pi_t) - \min_q \text{SP}(q, \pi_t)}, \quad (10)$$

where C is a scaling constant controlling the range of the distribution. Next, we introduce a UCB bias to encourage exploration of the unused samples:

$$\text{UCB}_{q_i} = \widetilde{\text{SP}}(q_i, \pi_t) - \sqrt{\frac{C \ln(n_{\text{total}} + 1)}{n_i + 1}}, \quad (11)$$

where n_i is the number of times sample q_i has been selected, and n_{total} is the total number of selections over all samples. Finally, we apply a softmax transformation to convert the UCB scores into a probability distribution over samples:

$$P_{q_i, \pi_t} = \frac{\exp(-\text{UCB}_{q_i})}{\sum_{q_j \in \mathcal{D}} \exp(-\text{UCB}_{q_j})}, \quad (12)$$

from which we sample the training batch B.

RL Training

Given the sampled batch B, we first obtain new reward vectors for each selected sample by querying the reward model, resulting in $R_{i,new}$. These rewards are then used to update the corresponding entries in the reward space.

$$R_{i,t+1} = \begin{cases} R_{i,new} & \text{if } i \in B \\ R_{i,t} & \text{otherwise} \end{cases} \quad (13)$$

To integrate multiple reward to derive a combined sum reward for RL optimization, we follow the standard approach in MORL (Li et al. 2025a) by introducing a weight vector $w \in R^n$, where each element w_j reflects the importance of the j^{th} dimension. This integration can be substituted with alternative methods like (Michael Gimelfarb 2025).

$$R^{\text{sum}}_{i,t+1} = w^\top R_{i,t+1}. \quad (14)$$

This scalar reward is then used to compute the advantage and update the π_t using any standard RL algorithm, such as PPO (Schulman et al. 2017), Reinforce++ (Hu 2025),GRPO (Shao et al. 2024). The whole procedure can be executed adaptively via data selection, reward space update and RL training, thereby improving sample efficiency and overall training performance.

Experiment

Task and Datasets

We validate MRACL on the multi-objective alignment task, where the actor is optimized towards multiple objectives.

Algorithm 1: The algorithm of MRACL

Initialize: Dataset $\mathcal{D}$, Actor π, Reward model RM
// Initialize the Reward Space
for *each* $q_i \in \mathcal{D}$ **do**
  $R_{i,0} = RM(q_i, \pi(q_i))$
for $t = 0$ *to* T **do**
  // Capability Estimation
  $R_{\text{mean},t} = \frac{1}{D}\sum_{i=1}^{D} R_{i,t}$
  for *each* $q_i \in \mathcal{D}$ **do**
    // Sample Assessment
    $W_{\text{Cap}}(q_i, \pi_t) = \|R_i - R_{\text{mean},t}\|^2$
    $W_{\text{Opt}}(q_{i,t}) = \frac{\|R_{i,t} - R_{\min}\|^2}{\cos^2\theta + \epsilon}$
    $W_{\text{Evo}}(q_i) = \sum_{j=1}^{n} \lambda_j (r_{i,t}^j - r_{i,t-1}^j)^2$
    $\text{SP}(q_i, \pi_t) = \alpha_t W_{\text{Cap}} + \beta_t W_{\text{Opt}} - \gamma W_{\text{Evo}}$
  for *each* $q_i \in \mathcal{D}$ **do**
    // Exploration and exploitation.
    $\widetilde{\text{SP}}(q_i, \pi_t) = C \cdot \frac{\text{SP}(q_i,\pi_t) - \min_q \text{SP}(q,\pi_t)}{\max_q \text{SP}(q,\pi_t) - \min_q \text{SP}(q,\pi_t)}$
    $\text{UCB}_{q_i} = \widetilde{\text{SP}}(q_i, \pi_t) - \sqrt{\frac{C \cdot \ln(n_{\text{total}}+1)}{n_i+1}}$
  $P_{q_i} = \frac{\exp(-\text{UCB}_{q_i})}{\sum_{q_j} \exp(-\text{UCB}_{q_j})}$
  Sample Batch B from the distribution P
  for *each* $q_i \in \mathcal{B}$ **do**
    $R_{i,t+1} = RM(q_i, \pi_t(q_i))$
  Update the Reward Space in Batch B
  Using any RL algorithm to update π_t
return π_T

Specifically, we use UltraFeedback for training, which provides four reward dimensions: Truthfulness, Helpfulness, Uncertainty, and Instruction Following. We randomly sample 15,000 instances as training set. For evaluation, we split the test set into three subsets, easy, median, and hard, based on the difficulties in the original dataset, with each subset containing 500 samples. To assess generality, we also evaluate on PKU-SAFE-10k (Ji et al. 2025), which includes Harmlessness and Helpfulness rewards.

Training Details

Following (Peng et al. 2025), we use Qwen3-235B-A22B as the well-aligned reward model, which assigns scores ranging from $[0, 10]$ across all dimensions to the model responses. Our main experimental backbone is Qwen2.5-7B-Base (Team 2025), and we additionally include Qwen2.5-3B-Instruct, LLaMA3-8B-SFT (Dong et al. 2024), and Qwen2.5-7B-Instruct in our analysis. Before training, we perform reward space initialization by sampling 4 responses for each sample, which takes approximately 2 hours and only needs to be executed once. Unless otherwise specified, we use a weight vector $w = [1, 1, 1, 1]$, $\gamma = 0.2$, $\epsilon = 0.5$, $C = 2$, a batch size of 256 and epoch of 2. The number of rollouts is set to 2, using Reinforce++. All experiments are conducted on 8×A800 GPUs. Each experiment runs for 120 RL training steps, which takes about 10 hours.

Baselines Considering MRACL is the first work for adaptive curriculum learning method for MORL, independent of any RL algorithms, we compare MRACL with the available CL methods based on Reinforce++. Besides, we explore general or specific algorithms in generality analysis:

- Standard: No curriculum learning method.
- Offline: Predefine the difficulties of each sample and train from easy to difficult samples.
- AdaRFT (Shi et al. 2025): Predefine the difficulties of each sample and select the samples according to reward.
- SpeedRL (Zhang et al. 2025): Select the samples in a certain pass rate interval at each step.
- DUMP (Wang et al. 2025): Select the sample according to its prediction of expected advantages.

Results

In-Distribution Evaluation For in-distribution evaluation, we evaluate MRACL and other methods on the three-level subsets. The results are presented in Figure 2. First, after training, MRACL surpasses the performance of Qwen2.5-7B-Instruct across all objectives, providing evidence that RL effectively facilitates multi-objective alignment. Second, compared to baselines, MRACL achieves faster convergence. It improves convergence speed by 1.62× over the state-of-the-art method DUMP, and by 2.54× over the standard baseline and exhibits a consistently strong initial growth rate across all subsets during the early stages. Third, MRACL demonstrates superior final performance, particularly on the hard subset, where it surpasses DUMP by 0.35 points average across the four rewards, which suggests that MRACL can produce more training gains through well-organized curriculum design. In contrast, offline and AdaRFT display instability during training: while they yield rapid initial gains before 40 steps, their performance deteriorates in the later stages due to the increasing task difficulty. Moreover, SpeedRL needs to conduct online inference when selecting the samples, which incurs longer computation. Besides, since DUMP designs its curriculum based on the advantage, which aggregates multiple reward signals into a single scalar, it fails to preserve the original structure of the multi-signals, thereby limiting its effectiveness. Considering that MRACL will not add further computation, it can serve as a free lunch for current RL algorithm. The results highlight the faster speed and better performance of MRACL, confirming its advantages in both efficiency and quality.

Win Rate Evaluation To further assess the quality of response generated from different methods, we conduct win rate evaluations following (Xu et al. 2025). We conduct evaluation on both the UltraFeedback and HelpSteer2 (Wang et al. 2024), using 200 unseen instances per dataset, representing in-distribution and out-of-distribution corpus, respectively. Specifically, we replace the uncertainty with harmlessness. Responses from each method are compared against the responses from the model Qwen2.5-7B-Base, and rated by DeepSeek-R1 (DeepSeek-AI 2025) to avoid reward hacking. Each instance is evaluated with three independent responses to ensure more accurate statistics. The results

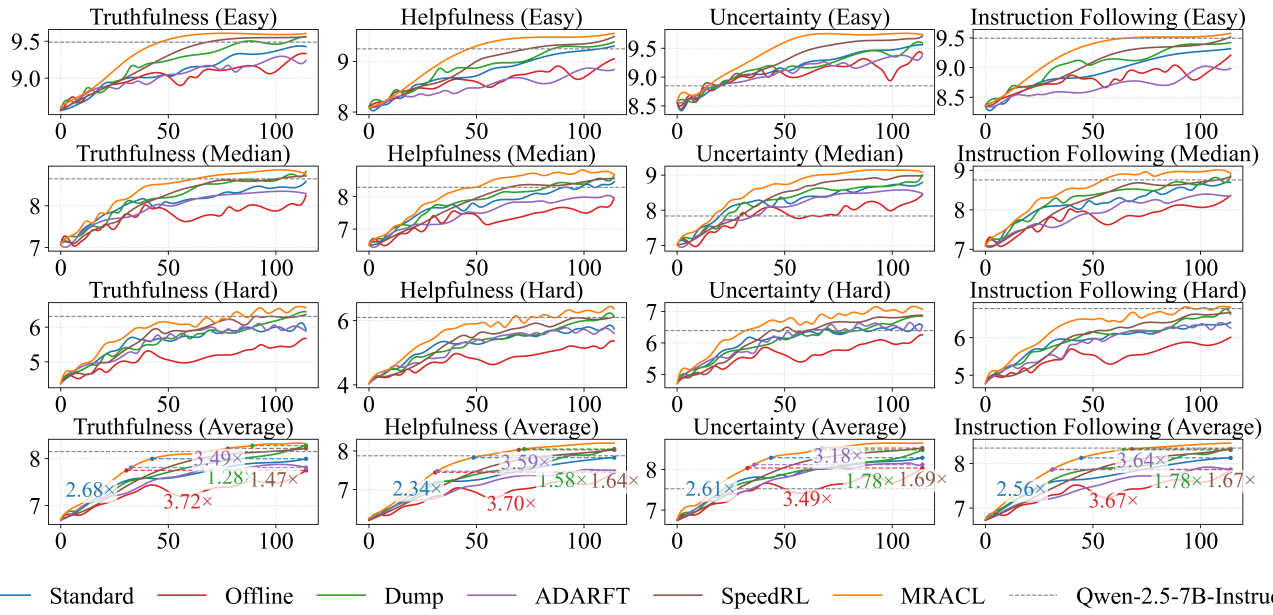

Figure 2: In-distribution evaluation results for Qwen2.5-7B-Base, with average performance calculated as the mean of the easy, median, and hard subsets. The gray horizontal line indicates the performance of Qwen2.5-7B-Instruct for reference.

are shown in Table 1, using checkpoints at step 50 and step 100, representing mid-stage and later-stage training. At both checkpoints, MRACL achieves the highest win rate on both in and out-of-distribution setting, outperforming DUMP by an average of 6.1%. Notably, at step 50, MRACL exhibits significant improvements over other baselines, particularly in instruction following and truthfulness, indicating that our curriculum design enables faster task adaptation during early training. By step 100, while the performance of all methods begins to converge, MRACL still maintains superior results, surpassing DUMP by 5.9%. Importantly, MRACL demonstrates balanced improvements across all objectives, whereas DUMP and SpeedRL exhibit a decline in harmlessness, with DUMP dropping from 55.5 to 54 and SpeedRL from 56 to 55. This contrast highlights MRACL's ability to effectively handle conflicting objectives while avoiding misalignment that leads to single-objective over-optimization.

Rule-Based Evaluation To further evaluate the model's ability in real-world tasks, we conduct experiments on several widely used rule-based benchmarks covering different dimensions: Instruction Following(denoted as I.F.) (IFEVAL (Zhou et al. 2023), CELLO (He et al. 2024), MT-Bench (Zheng et al. 2023)), Helpfulness(denoted as Helpful.) (TriviaQA (Joshi et al. 2017), MMLU (Hendrycks et al. 2020)), Truthfulness(denoted as Truth.) (TruthfulQA (Lin, Hilton, and Evans 2022)), and Harmlessness(denoted as Harm.) (ToxiGen (Hartvigsen et al. 2022)). The results are summarized in Table 2. Despite not being trained on the specific datasets, MRACL consistently outperforms all baselines on 6 of the 7 benchmarks, with particularly strong improvements on instruction-following tasks (e.g., +2.30 points on CELLO compared to the second-best method SpeedRL). Notably, MRACL achieves superior performance in safety-related dimensions, ranking first in both truthfulness (46.95) and harmlessness (58.19) while maintaining competitive results across other objectives. Given the relatively limited initial capabilities of Qwen2.5-7B-Base, these results demonstrate MRACL's superior alignment effectiveness and training efficiency. Unlike the baselines that exhibit trade-offs between objectives, MRACL achieves balanced improvements across potentially conflicting objectives, highlighting its robustness in multi-objective alignment scenarios.

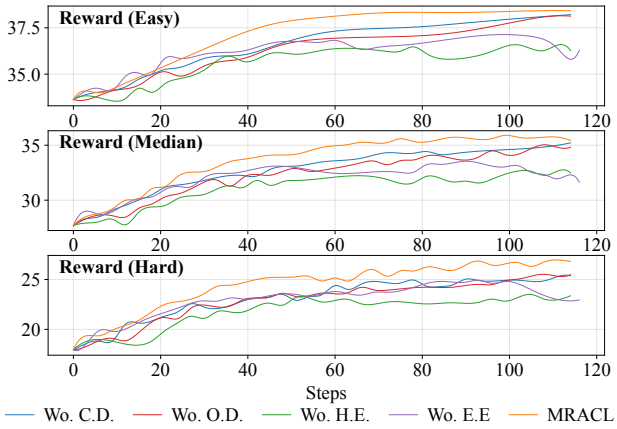

Figure 3: The results of the ablation study.

Ablation Study To assess the contribution of each component in MRACL, we conduct an ablation study by removing four key modules: capability distance (denoted as Wo.

Method	HelpSteer2				UltraFeedback				Avg
	Truthfulness	Helpfulness	Harmlessness	I.F.	Truthfulness	Helpfulness	Harmlessness	I.F.	
Step 50									
Standard	52.5	**57.0**	53.0	54.0	59.0	62.0	53.0	61.0	56.4
Offline	**55.0**	52.5	52.0	53.5	58.5	54.0	54.5	58.5	54.8
AdaRFT	52.5	56.5	50.0	55.5	58.5	61.0	54.5	61.5	56.3
SpeedRL	54.0	53.0	54.0	55.0	57.0	61.0	56.0	58.0	56.0
DUMP	51.5	53.0	55.0	53.0	55.5	61.0	55.5	58.5	55.4
MRACL	**55.0**	**57.0**	**55.5**	**57.5**	**62.0**	**65.5**	**57.5**	**62.5**	**59.1**
Step 100									
Standard	54.5	56.5	55.5	55.5	54.5	65.0	54.5	63.5	57.4
Offline	56.0	58.5	54.5	55.5	59.5	58.0	53.0	60.5	56.9
AdaRFT	55.5	59.0	52.5	54.5	61.5	56.5	58.0	62.5	57.5
SpeedRL	57.0	**63.0**	53.0	63.0	65.0	66.0	55.0	**65.0**	60.9
DUMP	56.5	60.5	54.0	57.0	66.0	68.5	54.0	63.5	60.0
MRACL	**59.0**	**62.5**	**58.0**	**66.5**	**66.5**	**69.0**	**58.5**	**64.5**	**63.1**

Table 1: The Win Rate evaluation results compared with the original training model Qwen2.5-7B-Base.

Method	I. F.			Helpful.		Truth.	Harm.
	IFEval	CELLO	MT	Trivia	MMLU	Truthful	Toxi
Standard	48.56	58.00	6.80	48.66	**73.68**	42.78	57.34
Offline	45.20	58.60	7.15	45.67	70.63	45.45	57.44
AdaRFT	47.60	59.60	7.16	44.27	72.45	46.01	57.97
SpeedRL	48.24	62.60	7.33	**51.71**	72.45	46.27	57.21
DUMP	47.48	60.80	7.35	44.64	70.63	45.73	58.09
MRACL	**48.68**	**64.90**	**7.38**	50.31	**73.68**	**46.95**	**58.19**

Table 2: The results of rule-based evaluation.

C.D.), optimization direction (denoted as Wo. O.D.), historical evolution (denoted as Wo. H.E.) and exploitation & exploration (denoted as Wo. E.E). As shown in Figure 3, performance is measured by the sum reward across all objectives. First, removing any component leads to a drop across all subsets. Specifically, the improvement of Wo. Capability is notably slower in early training, especially on the hard subset. Although the performance converges in later stages, the absence of capability distance diminishes the effectiveness of curriculum guidance during the early training. The performance of Wo. Optimization drops significantly on the median and hard subsets (from 35.9 and 26.6 to 35.0 and 25.0, respectively), as the absence of the optimization direction causes the curriculum strategy to misjudge the learning potential of samples, leading to ineffective scheduling and misaligned progression. The performance of Wo. E.E shows rapid improvement in early training. However, in later stages, the absence of exploration causes stagnation, as the model fails to discover new informative samples for continued learning. Critically, Wo. Evolution yields the worst performance across all subsets. After 50 training steps, the model's reward begins to converge, indicating a slowdown in learning efficiency, which suggests the curriculum gradually includes samples whose reward vectors exhibit limited change, typically due to being difficult to learn or offering limited optimization potential.

Analysis

We analyze the key factors contributing to MRACL in data selection analysis, validate its advantages in rollout analysis, and application in sensitivity analysis. Besides, we further conduct the Pareto and generality analysis.

Data Selection Analysis To further investigate the mechanisms behind MRACL, we analyze the distribution of selected training samples at representative stages: steps 0, 40, and 80. For visualization, we project the reward space onto two dimensions: Helpfulness and Truthfulness and use rollout=4, which is shown in Figure 4. At the early stage (step 0), the selected samples are concentrated around the current model capability, which facilitates adaptation to the training objective. As training progresses, the model's capability in the reward space shifts towards the top-right, reflecting simultaneous gains in both objectives. During this phase, the selected samples are primarily located near the optimization direction ($y = x$), representing examples that remain underoptimized and balanced across objectives, which avoids selecting samples that may cause severe reward imbalance or objective misalignment. At the later stage, the model exhibits significantly enhanced capabilities. The exploration mechanism in MRACL begins to prioritize more challenging or unaligned samples to further improve its robustness.

To further validate the rationality of this data selection, we conduct a hyper-parameter analysis of different learning schedule parameters, α and β. We design two settings, $\alpha = 0.5, \beta = 0.5$ (denoted as Fixed) and $\alpha_t = \frac{t}{T}, \beta_t = 1 - \frac{t}{T}$ (denoted as Reversed). Although Fixed maintains a comparable learning curve to MRACL in the early stages, its final performance is lower due to the lack of sufficiently challenging and optimizable samples in later training. Reversed, on the other hand, shows slower convergence and worse overall performance, as it introduces many overly difficult samples during early training, which misalign with the model's current capability and hinder further progress.

Rollout Analysis Considering that the rollouts is a key factor affecting both the training efficiency and final per-

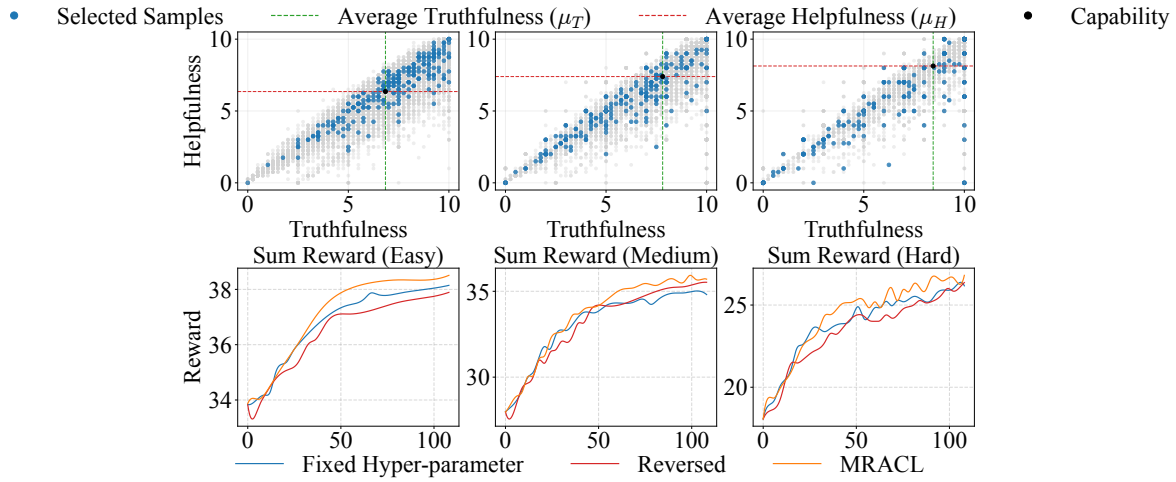

Figure 4: The upper figure shows data selection, and the lower shows the results of different learning parameters.

formance, which also plays a crucial role in curriculum design, we compare MRACL with DUMP under varying rollout settings (1, 2, and 4) in Figure 5, where we report the average sum reward across different subsets. Firstly, MRACL exhibits a consistently smooth and steady improvement across different rollout configurations, whereas DUMP's final performance degrades notably, from 33.8 to 33.0 and 32.1, as the number of rollouts decreases, which indicates that DUMP's curriculum design becomes less effective under limited rollout scenarios. We attribute the robustness of MRACL to its use of an offline reward space, which provides a stable foundation for curriculum design, particularly when informative signals are scarce in early stage. In contrast, DUMP relies on an advantage window to compute importance during training, which becomes less reliable with fewer rollouts. Notably, constructing the offline reward space in MRACL takes only 2 hours and is required only once per model, whereas training with rollout=4 takes 16 hours, which highlights the efficiency of MRACL in providing a stable curriculum even under low-rollout scenarios.

Sensitivity to Model Performance To further investigate whether initial model performance will affect the effectiveness of MRACL, we conduct additional experiments using two initially stronger instruction-tuned models: Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, as shown in Figure 6. For Qwen2.5-3B, MRACL continues to improve convergence speed and consistently outperforms all baselines across the easy, median, and hard subsets. For the more capable Qwen2.5-7B, the performance gains become smaller on the easy and median subsets, which can be attributed to the fact that when the model already performs well on simple tasks, the additional guidance from curriculum learning yields limited gains. However, notably, in the hard subset, MRACL still achieves significant advantages in both convergence and final performance, achieving a sum reward of 30, up from 25.1, which indicates that MRACL is particularly effective at facilitating the acquisition of challenging samples that lie beyond model initial competency. We hypothesize that MRACL is beneficial in scenarios where task difficulty varies widely and where the ability to focus on hard-to-learn samples is essential for pushing the boundaries.

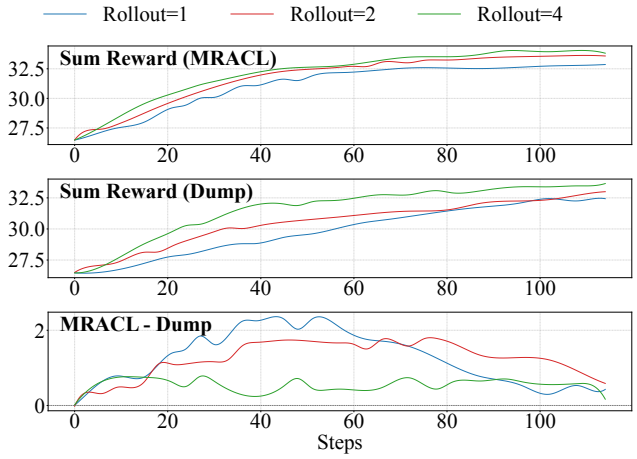

Figure 5: Performance comparison across different rollouts.

Pareto Analysis Although in most cases user preferences are relatively balanced (e.g. $w = [1, 1, 1, 1]$), we specifically evaluate the performance of MRACL when facing skewed preference configurations. Taking $w = [0.5, 2.0, 0.5, 1.0]$ as an example in Figure 7, this emphasizes Helpfulness, while assigning lower weights to Truthfulness and Uncertainty. More weight experiment are in supplementary materials. As shown in Figure 7, MRACL consistently outperforms all baselines across the four dimensions. Notably, it achieves the most significant gain on the Helpfulness objective, demonstrating its ability to adaptively adjust learning priorities in accordance with the given reward weighting. Meanwhile, MRACL maintains higher performance on the de-emphasized objectives. In contrast, Offline and SpeedRL

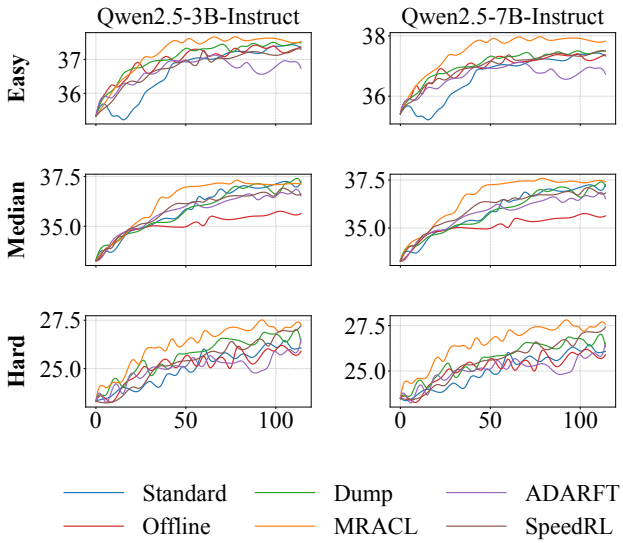

Figure 6: The results of models with varying capabilities (Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct).

show either insufficient specialization (i.e., failing to maximize the prioritized dimension) or unstable trade-offs, with visible regressions. The results validate the controllability of MRACL in aligning training behavior with user-specified preferences, even under conflicting reward settings.

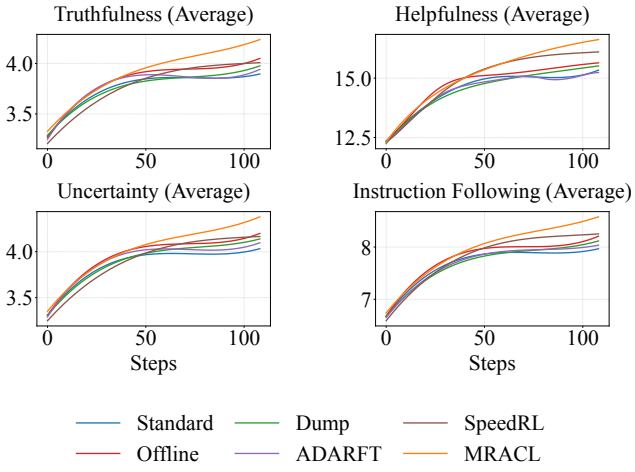

Figure 7: The pareto analysis with weight [0.5,2.0,0.5,1.0].

Generality Analysis We further validate the generality of MRACL on datasets, backbones, and RL algorithms, with more results provided in the supplementary materials.

Dataset Generalization We conduct the dataset generalization experiment on the widely used two-objective alignment dataset PKU-SAFE. We compare our method with the two most competitive methods DUMP, SpeedRL, and the standard method. As shown in Figure 8, although the reward of all methods increases smoothly, MRACL achieves the

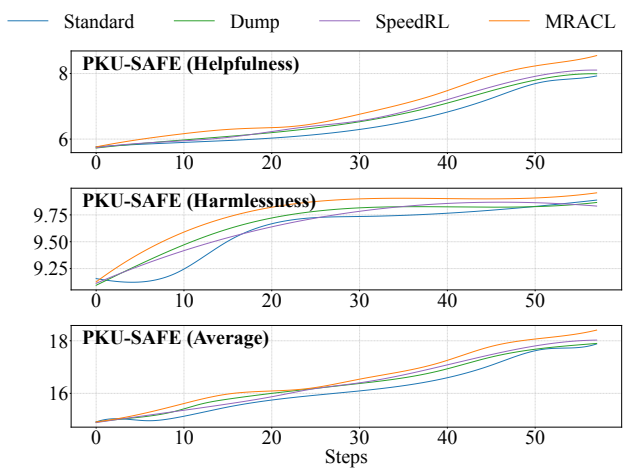

Figure 8: The results on the dataset PKU-SAFE.

best overall performance. Specifically, MRACL consistently outperforms all baselines in the Helpfulness dimension and on the average reward, establishing a clear advantage in early training and demonstrating strong generalization ability. For the Harmlessness dimension, while MRACL also shows strong initial improvement, its final score is comparable to SpeedRL and Dump, with the leading methods converging to a similar performance level in later stages. However, the relative advantage of MRACL is less pronounced on the two-objective dataset compared to results on UltraFeedback. We posit that the strength of MRACL emerges in more complex scenarios (e.g., with three or more objectives). In such settings, the finer-grained reward signals from multiple objectives become crucial for constructing reward space and curriculum design to guide the training progress.

Conclusion

In this paper, we propose MRACL, the first adaptive curriculum reinforcement learning framework for MORL. By persistently maintaining a multi-reward space, we estimate the capability of the model and assess each sample's priority, which guides the data selection in each step, independent of the specific algorithm. Experiments on the multi-objective alignment task show that MRACL accelerates convergence speed and gets higher performance than the current CL method in both win rate and rule-based evaluations. Besides, we provide various analyses of its advantages, applicable scenarios, and generality across diverse experiments.

Acknowledgements

We sincerely thank all collaborators for their substantial contributions and dedicated efforts throughout the development of this work.

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Cui, G.; Yuan, L.; Ding, N.; Yao, G.; He, B.; Zhu, W.; Ni, Y.; Xie, G.; Xie, R.; Lin, Y.; Liu, Z.; and Sun, M. 2024. ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 9722–9744. PMLR.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Dong, H.; Xiong, W.; Pang, B.; Wang, H.; Zhao, H.; Zhou, Y.; Jiang, N.; Sahoo, D.; Xiong, C.; and Zhang, T. 2024. RLHF Workflow: From Reward Modeling to Online RLHF. arXiv:2405.07863.

Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; and Kamar, E. 2022. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3309–3326. Dublin, Ireland: Association for Computational Linguistics.

He, Q.; Zeng, J.; Huang, W.; Chen, L.; Xiao, J.; He, Q.; Zhou, X.; Liang, J.; and Xiao, Y. 2024. Can Large Language Models Understand Real-World Complex Instructions? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18188–18196.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring Massive Multitask Language Understanding. *CoRR*, abs/2009.03300.

Hu, W. S., Jason Klein Liu. 2025. REINFORCE++: An Efficient RLHF Algorithm with Robustness to Both Prompt and Reward Models. arXiv:2501.03262.

Ji, J.; Hong, D.; Zhang, B.; Chen, B.; Dai, J.; Zheng, B.; Qiu, T. A.; Zhou, J.; Wang, K.; Li, B.; Han, S.; Guo, Y.; and Yang, Y. 2025. PKU-SafeRLHF: Towards Multi-Level Safety Alignment for LLMs with Human Preference. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 31983–32016. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Jiang, L.; Meng, D.; Yu, S.; Lan, Z.; and Hauptmann, A. G. 2014. Self-paced curriculum learning. In *AAAI*, volume 2, 6.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Li, C.; Zhang, H.; Xu, Y.; Xue, H.; Ao, X.; and He, Q. 2025a. Gradient-Adaptive Policy Optimization: Towards Multi-Objective Alignment of Large Language Models. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11214–11232. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Li, M.; Lin, J.; Zhao, X.; Lu, W.; Zhao, P.; Wermter, S.; and Wang, D. 2025b. Curriculum-RLAIF: Curriculum Alignment with Reinforcement Learning from AI Feedback. arXiv:2505.20075.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252. Dublin, Ireland: Association for Computational Linguistics.

Michael Gimelfarb, S. S., Ayal Taitler. 2025. Constraint-Generation Policy Optimization (CGPO): Nonlinear Programming for Policy Optimization in Mixed Discrete-Continuous MDPs. arXiv:2401.12243.

Pattnaik, P.; Maheshwary, R.; Ogueji, K.; Yadav, V.; and Madhusudhan, S. T. 2024. Enhancing Alignment using Curriculum Learning & Ranked Preferences. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 12891–12907. Miami, Florida, USA: Association for Computational Linguistics.

Peng, H.; Qi, Y.; Wang, X.; Yao, Z.; Xu, B.; Hou, L.; and Li, J. 2025. Agentic Reward Modeling: Integrating Human Preferences with Verifiable Correctness Signals for Reliable Reward Systems. arXiv:2502.19328.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Shi, T.; Wu, Y.; Song, L.; Zhou, T.; and Zhao, J. 2025. Efficient Reinforcement Finetuning via Adaptive Curriculum Learning. arXiv:2504.05520.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12: 1057–1063.

Team, Q. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Wang, X.; Chen, Y.; and Zhu, W. 2021. A Survey on Curriculum Learning. arXiv:2010.13166.

Wang, Z.; Cui, G.; Li, Y.-J.; Wan, K.; and Zhao, W. 2025. DUMP: Automated Distribution-Level Curriculum Learning for RL-based LLM Post-training. arXiv:2504.09710.

Wang, Z.; Dong, Y.; Delalleau, O.; Zeng, J.; Shen, G.; Egert, D.; Zhang, J. J.; Sreedhar, M. N.; and Kuchaiev, O. 2024. HelpSteer 2: Open-source dataset for training top-performing reward models. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 1474–1501. Curran Associates, Inc.

Wen, L.; Cai, Y.; Xiao, F.; He, X.; An, Q.; Duan, Z.; Du, Y.; Liu, J.; Tang, L.; Lv, X.; Zou, H.; Deng, Y.; Jia, S.; and Zhang, X. 2025. Light-R1: Curriculum SFT, DPO and RL for Long COT from Scratch and Beyond. arXiv:2503.10460.

Xu, Z.; Tong, Y.; Zhang, X.; Zhou, J.; and Wang, X. 2025. REWARD CONSISTENCY: Improving Multi-Objective Alignment from a Data-Centric Perspective. arXiv:2504.11337.

Zhang, R.; Arora, D.; Mei, S.; and Zanette, A. 2025. SPEED-RL: Faster Training of Reasoning Models via Online Curriculum Learning. arXiv:2506.09016.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.05685.

Zhou, J.; Lu, T.; Mishra, S.; Brahma, S.; Basu, S.; Luan, Y.; Zhou, D.; and Hou, L. 2023. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911.