Insights into Alignment: Evaluating DPO and its Variants Across Multiple Tasks

Anonymous ACL submission

Abstract

This study evaluates Direct Preference Optimization (DPO) and its variants for aligning Large Language Models (LLMs) with human preferences, testing three configurations: (1) with Supervised Fine-Tuning (SFT), (2) without SFT, and (3) without SFT but using an instruction-tuned model. We further investigate how training set size influences model performance. Our evaluation spans 13 benchmarks—covering dialogue, reasoning, mathematical problem-solving, question answering, truthfulness, MT-Bench, Big Bench, and the Open LLM Leaderboard. We find that: (1) alignment methods often achieve near-optimal performance even with smaller subsets of training data; (2) although they offer limited improvements on complex reasoning tasks, they enhance mathematical problem-solving; and (3) using an instruction-tuned model improves truthfulness. These insights highlight the conditions under which alignment methods excel, as well as their limitations.

1 Introduction

Large Language Models (LLMs) demonstrate exceptional capabilities across various tasks, but aligning them with human preferences presents challenges, including high data demands and inconsistent performance across tasks. These models excel in mathematical reasoning problem-solving (Cobbe et al., 2021a; Wei et al., 2022; Lewkowycz et al., 2022), code generation programming (Chen et al., 2021; Austin et al., 2021; Li et al., 2022), text generation (Bubeck et al., 2023; Touvron et al., 2023), summarization, and creative writing, among other tasks. Notably, LLMs have achieved significant performance with human preferences, based on alignment methods including Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF) (Sanh et al., 2022; Ouyang et al., 2022). While RLHF exhibits remarkable performance compared to just SFT, it faces limitations such as reward hacking (Liu et al., 2024). Therefore, Direct Preference Optimization (DPO) (Rafailov et al., 2023), a state-of-the-art offline reinforcement learning method, has been proposed to optimize human preferences without the need for the RL process.

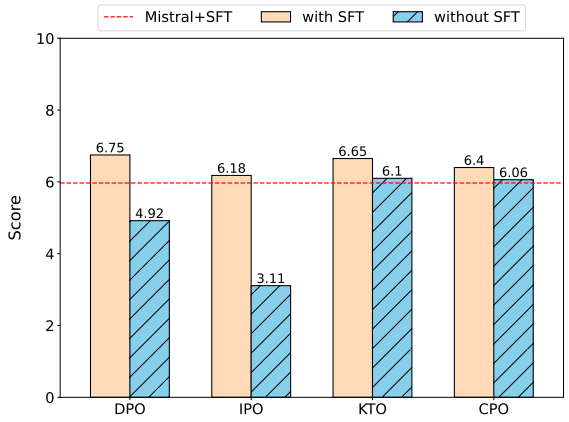

Figure 1: Performance comparison of alignment methods on MT-Bench under two scenarios: 1) fine-tuning a model with SFT (Mistral+SFT) and 2) fine-tuning a pre-trained model without SFT (Mistral). Unlike IPO and DPO, other methods like CPO and KTO demonstrate similar performance to model that undergo SFT.

Recent studies have highlighted limitations in alignment methods, including issues like overfitting, inefficient learning and memory utilization, preferences ranking, and dependence on preferences across various scenarios like dialogue systems (Tunstall et al., 2023), summarization, sentiment analysis (Wu et al., 2023), helpful and harmful question answering (Liu et al., 2024), and machine translation (Xu et al., 2024). Despite the significance of these studies, none have thoroughly examined critical ambiguities in alignment, such as (1) the learnability of emerged alignment methods without SFT, (2) fair comparison between these methods, (3) evaluating their performance post-SFT, (4) the impact of data volume on performance, and weaknesses inherent in these methods. Ad-

dressing these areas is crucial for gaining a comprehensive understanding for alignment methods.

In this study, we delve into the performance of alignment methods such as DPO (Rafailov et al., 2023), IPO(Azar et al., 2023), KTO (Ethayarajh et al., 2023), and CPO (Xu et al., 2024), which are based on RL-free algorithms. These methods typically involve two steps: 1) Supervised fine-tuning of a policy model and 2) Optimization of the SFT model with alignment algorithms such as DPO. Our exploration spans across various tasks including dialogue systems, reasoning, mathematical problem-solving, question answering, truthfulness, and multi-task understanding. We evaluate these alignment methods across 13 benchmarks such as MT-Bench (Zheng et al., 2023), Big Bench (bench authors, 2023), and Open LLM Leaderboard (Beeching et al., 2023). To assess the performance of these methods, we define three distinct scenarios: **1) Fine-tuning an SFT model**, **2) Fine-tuning a pre-trained model**, and **3) Fine-tuning an instruction model**. In scenario 1, we employ a supervised fine-tuned model on chat completion and fine-tune it with different alignment methods. In scenario 2, we omit the SFT phase and directly fine-tune a pre-trained model with alignment methods. In scenario 3, we skip the SFT phase and utilize an instruction-tuned model as the base model, fine-tuning it with alignment methods.

The results indicate that in the standard alignment process, KTO outperforms other methods across all tasks except for multi-task understanding. However, the performance of SFT and other alignment methods in reasoning tasks is relatively comparable, suggesting that RL-free algorithms do not significantly affect reasoning. Moreover, unlike DPO when skipping the SFT phase, KTO, and CPO demonstrate comparable performance on MT-Bench. Comparing the performance of methods with and without the SFT phase reveals a significant improvement in TruthfulQA (Lin et al., 2022) and GSM8K (Cobbe et al., 2021b). Additionally, an interesting finding is that alignment methods in the standard process exhibit better performance with smaller training data subsets. Lastly, it is observed that the instruction-tuned model has a notable impact only on truthfulness.

In summary, our contributions are as follows:

1. We explore the learning capabilities of alignment methods, aiming to mitigate overfitting challenges within the DPO framework. Our findings indicate that CPO and KTO show comparable performance with skipping the SFT part in MT-Bench (See Figure 1).
2. We examine the effectiveness of alignment methods across dialogue systems, reasoning, mathematical problem-solving, question answering, truthfulness, and multi-task understanding in three different scenarios.
3. A comprehensive evaluation reveals that alignment methods exhibit a lack of performance in reasoning tasks yet demonstrate impressive performance in solving mathematical problems and truthfulness.
4. We observe that in the standard alignment process, fine-tuning an SFT model with all alignment algorithms using a small subset of training data yields better performance. (See Figure 3).

2 Related Works

Recent advancements in pre-training LLMs, such as LLaMA-2 (Touvron et al., 2023), GPT-3 (Brown et al., 2020), Gopher (Rae et al., 2022), Vicunna (Chiang et al., 2023), Mistral (Jiang et al., 2023), and PaLM 2 (Anil et al., 2023), have led to impressive performance gains in zero-shot (Radford et al., 2019) and few-shot (Chowdhery et al., 2022) scenarios across various tasks. However, when applied to downstream tasks, LLMs' performance tends to degrade. While fine-tuning models using human completions aids in alignment and performance enhancement, obtaining human preferences for responses is often more feasible than collecting expert demonstrations. Consequently, recent research has shifted focus towards fine-tuning LLMs using human preferences. In this section, we present a brief review of alignment algorithms on various tasks.

RLHF (Christiano et al., 2023) proposed to optimize for maximum reward operates by engaging with a reward model trained using the Bradley-Terry (BT) model (Bong and Rinaldo, 2022) through reinforcement algorithms like Proximal Policy Optimization (PPO) (Schulman et al., 2017). While RLHF enhances model performance, it grapples with challenges such as instability, reward hacking, and scalability inherent in reinforcement learning.

Recent studies have introduced methods to address these challenges by optimizing relative preferences without depending on reinforcement learning

(RL). Optimizing a model using the BT model on preference datasets helps ensure alignment with human preferences.

Sequence Likelihood Calibration (SLiC) (Zhao et al., 2023) introduced a novel approach to ranking preferences produced by a supervised fine-tuned (SFT) model, employing calibration loss and regularization fine-tuning loss during training. Meanwhile, Rank Response with Human Feedback (RRHF) (Yuan et al., 2023) trains the SFT model utilizing a zero-margin likelihood contrastive loss, assuming multiple ranked responses for each input. Despite their efficacy, SLiC and RRHF lack theoretical underpinnings. In response, DPO proposed a method to fit an SFT model directly to human preferences using the Bradley-Terry (BT) model, offering theoretical insights into the process.

Statistical Rejection Sampling Optimization (RSO) (Liu et al., 2024) combines the methodologies of SLiC and DPO while introducing an enhanced method for gathering preference pairs through statistical rejection sampling. IPO (Azar et al., 2023), akin to DPO approaches, has mathematically demonstrated the limitations of the DPO approach regarding overfitting and generalization, proposing a comprehensive objective for learning from human preferences. Zephyr (Tunstall et al., 2023) has enhanced DPO by leveraging state-of-the-art (SOTA) models to generate responses for the same input and ranking them using teacher models like GPT-4. Additionally, they highlight the necessity of SFT as a preliminary step before employing DPO.

KTO (Ethayarajh et al., 2023), inspired by Kahneman and Tversky's seminal work on prospect theory (TVERSKY and KAHNEMAN, 1992), aims to maximize the utility of LLM generations directly rather than maximizing the log-likelihood of preferences. This approach eliminates the need for two preferences for the same input, as it focuses on discerning whether a preference is desirable or undesirable.

Self-Play fIne-tuNing (SPIN) (Chen et al., 2024) introduced a self-training approach to enhance DPO using the dataset employed in the SFT step. The key idea of this approach is to utilize synthetic data generated as the rejected response and the gold response from the SFT dataset as the chosen response. Meanwhile, Constrictive Preference Optimization (CPO) (Xu et al., 2024) proposed an efficient method for learning preferences by combining the maximum-likelihood loss and the DPO loss function, aiming to improve memory and learning efficiency.

We note that the aforementioned works lack comparative studies on alignment methods concerning both completion and preference learning. While those studies address unlearning a DPO method without the SFT step, further exploration of alternative methods is warranted. Although the significance of high-quality preferences is widely acknowledged, there remains a necessity to explore the influence of data quantity on performance of the alignment methods. Additionally, the crucial aspect of generalization remains unexplored. While aligning a model aims to enhance performance across all categories, improving alignment methods often comes at the expense of performance in other areas. Further investigation in this regard is necessary. To this end, we examine the performance of alignment methods both before and after SFT to assess the learning capabilities of IPO, KTO, and CPO. Moreover, we highlight the weaknesses of alignment methods by comparing their performance across five different domains, demonstrating the significant impact of dataset quantity on performance.

3 Exiting Alignment Methods

In this section, we explain various RL-free alignment methods and discuss the reasons behind their development. Typically, the alignment process unfolds in three phases: 1) Fine-tuning a policy model using Supervised Fine-Tuning (SFT), 2) training a reward model, and 3) further fine-tuning the initial policy model using reinforcement learning (RL), where the reward model provides the feedback mechanism. A recent development by DPO introduced an RL-free approach aimed at aligning a policy model by optimizing the likelihood of the preferred and unpreferred responses. This is implemented using a dataset labeled D, where x represents the input, y_w denotes the preferred response, and y_l indicates the unpreferred response. The DPO loss function is mathematically articulated in Equation 1 as follows:

$$\mathcal{L}_{\text{DPO}}\left(\pi_\theta; \pi_{\text{ref}}\right) = -\,\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(y_w \mid x\right)}{\pi_{\text{ref}}\left(y_w \mid x\right)} - \beta\log\frac{\pi_\theta\left(y_l \mid x\right)}{\pi_{\text{ref}}\left(y_l \mid x\right)}\right)\right] \tag{1}$$

where π_θ is the parameterized policy, σ is sigmoid function and β is a parameter controlling the deviation from the base reference policy π_{ref}.

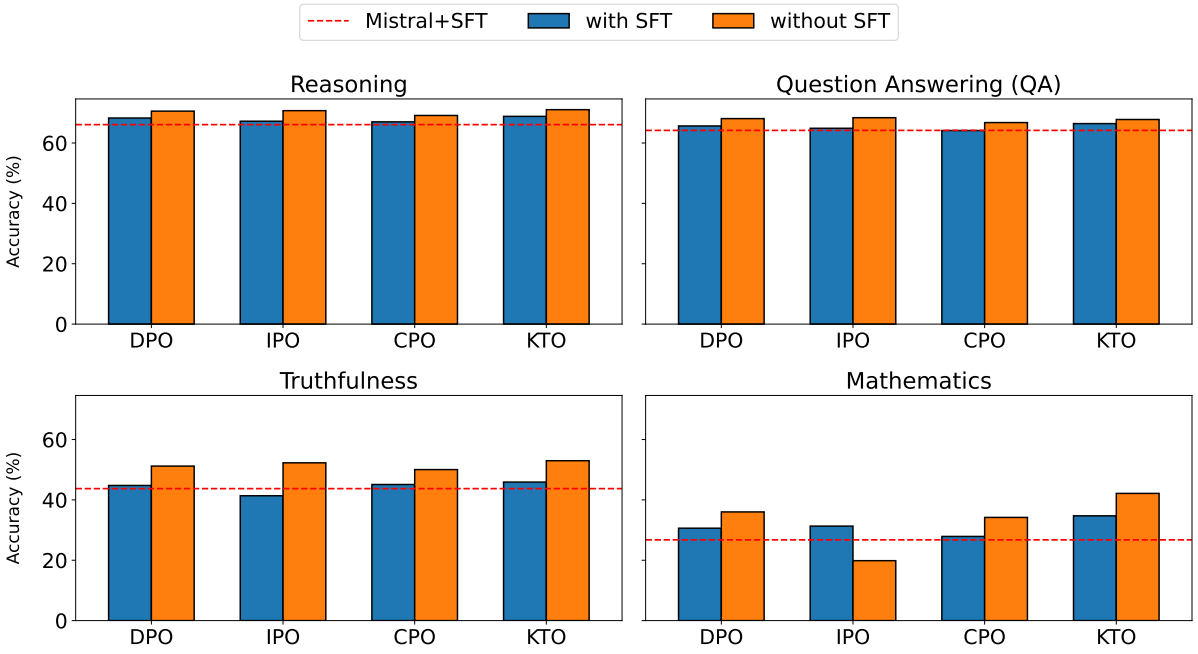

Figure 2: Comparison performance of the alignment method in different tasks based on two different scenarios: 1) fine-tuning an SFT model (Mistral+SFT) with alignment methods and 2) fine-tuning a pre-train model (Mistral) with them. For more details about reasoning and question answering, refer to Appendix B.

Despite DPO surpassing RLHF through RL-free methodology, it faces constraints like overfitting and the need for extensive regularization, which can impede the efficacy of the policy model. Addressing these limitations, in (Azar et al., 2023) introduced the IPO algorithm, which defines a general form of the DPO and reformulates it to solve the overfitting and regularization. The formulation of the IPO loss function is in Equation 2 as follows:

$$\mathcal{L}_{\text{IPO}}(\pi) = \underset{(y_w,y_l,x)\sim\mathcal{D}}{-\mathbb{E}} \left(h_\pi(y_w, y_l, x) - \frac{\tau^{-1}}{2} \right)^2 \tag{2}$$

$$h_\pi(y, y', x) = \log \left(\frac{\pi\left(y \mid x\right) \pi_{\text{ref}}\left(y' \mid x\right)}{\pi\left(y' \mid x\right) \pi_{\text{ref}}\left(y \mid x\right)} \right)$$

where x represents the input, y_w denotes the preferred response, y_l indicates the unpreferred response, π_{ref} is the reference policy and τ is a real positive regularisation parameter. Although the IPO algorithm overcomes the problems of overfitting and the need for extensive regularization present in DPO, the approach of aligning based on two preferences has different complications. The KTO study seeks to enhance the effectiveness of the DPO method by implementing a strategy that utilizes only a single preference. This method is inspired by the Kahneman & Tversky theory, which observes that humans are more acutely affected by losses than gains of comparable magnitude. In this algorithm, having a clear understanding of whether a preference is suitable or unsuitable is crucial, eliminating the necessity for an alternative preference. The KTO loss function is defined in Equation 3 as follows:

$$\mathcal{L}_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}; \beta) = \mathbb{E}_{x,y\sim\mathcal{D}} \left[1 - \hat{h}(x, y; \beta) \right] \tag{3}$$

$$\hat{h}(x, y; \beta) = \begin{cases} \sigma\left(\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \mathbb{E}_{x'\sim\mathcal{D}}\left[\beta \text{KL}(\pi_\theta \parallel \pi_{\text{ref}})\right]\right) \\ \qquad \text{if } y \sim y_{\text{desirable}}|x, \\ \sigma\left(\mathbb{E}_{x'\sim\mathcal{D}}\left[\beta \text{KL}(\pi_\theta \parallel \pi_{\text{ref}})\right] - \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right), \\ \qquad \text{if } y \sim y_{\text{undesirable}}|x \end{cases}$$

where π_θ is the model we are optimizing, β is a parameter controlling the deviation from the base reference policy π_{ref}, σ is the logistic function, KL is the KL-divergence between the two distributions and x is the input. IPO and KTO have enhanced the performance of the DPO model and addressed some of its shortcomings. However, the simultaneous loading of two models has led to inefficient learning in DPO algorithm. To improve upon this, the CPO method was developed, enhancing the efficiency of the DPO approach. Research detailed in (Xu et al., 2024) demonstrated that it is unnecessary to load a reference policy model (π_{ref}) during

training. By omitting the reference model from the memory, CPO increases operational efficiency, enabling the training of larger models at reduced costs compared to DPO. The CPO loss function is specified in Equation 4 as follows:

$$\mathcal{L}_{\text{NLL}} = -\mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\log \pi_\theta\left(y_w \mid x\right)\right]$$

$$\mathcal{L}_{\text{prefer}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log \sigma\big(\beta \log \pi_\theta(y_w|x) - \beta \log \pi_\theta(y_l|x)\big)\Big]$$

$$\mathcal{L}_{\text{CPO}} = \mathcal{L}_{\text{prefer}} + \mathcal{L}_{\text{NLL}} \tag{4}$$

where π_θ is the parameterized policy, y_w and y_l denotes the preferred and unpreferred responses, x is a set of source sentences, β is a parameter, and σ is the logistic function. In the next section, we assess the alignment methods, highlighting their strengths and weaknesses.

4 Experiments

Description. In this section, we assess the alignment methods across three scenarios: 1) fine-tuning an SFT model with alignment methods, 2) fine-tuning a pre-trained model with alignment methods, and 3) fine-tuning an instruction-tuned model with alignment methods. Subsequently, within each scenario, we examine their performance across reasoning, mathematical problem-solving, truthfulness, question-answering, and multi-task understanding. Details regarding these scenarios are provided in the following section.

Evaluation Metrics. To evaluate the methods for reasoning, we utilize benchmarks such as ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2019), Big Bench Sports Understanding (BB-sports), Big Bench Causal Judgment (BB-casual), Big Bench Formal Fallacies (BB-formal), and PIQA (Bisk et al., 2019). To evaluate their mathematical problem-solving abilities, we employ the GSM8K (Cobbe et al., 2021b) benchmark. Truthfulness is evaluated using the TruthfulQA (Lin et al., 2022) benchmark. Additionally, we gauge their performance in multitask understanding using the MMLU (Hendrycks et al., 2021) benchmark. OpenBookQA (Mihaylov et al., 2018) and BoolQ (Clark et al., 2019) benchmarks are employed to assess their performance in question-answering tasks. Finally, to evaluate their effectiveness in dialog systems, we utilize MT-Bench benchmarks, which consist of 160 questions across eight knowledge domains, with GPT-4 scoring the model-generated answers on a scale from 0 to 10.

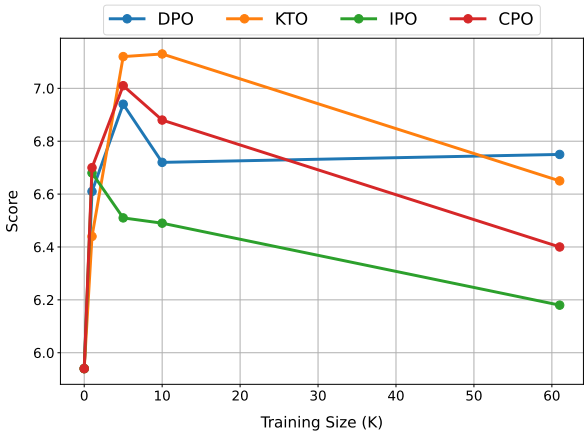

Figure 3: Comparison of performance for KTO, IPO, CPO, and DPO alignment methods on MT-Bench across various training set sizes. All methods demonstrated optimal performance with training sets ranging from 1K to 10K data points.

4.1 Scenario 1: Fine-tune an SFT Model

Motivation. In this scenario, we first train an SFT model and then refine it with the aforementioned alignment methods. These methods, designed to enhance the performance of DPO, have been applied to various tasks, such as machine translation. However, there hasn't been a comprehensive evaluation comparing them on the same task. The primary motivation behind these scenarios is to assess their performance across different benchmarks. Additionally, we aim to determine whether the performance of alignment methods improves with increasing training data, as it seems that alignment methods may not require extensive data beyond the SFT phase.

Models. We employ the `zephyr-sft-full` model as our SFT model, which underwent fine-tuning utilizing the UltraChat (Ding et al., 2023) dataset. Its baseline model is `Mistral-7B-v0.1`. We proceed by training the `zephyr-sft-full` model with DPO, IPO, KTO, and CPO. For further information regarding the training and evaluation procedures, please refer to the Appendix A.

Datasets. We utilize the UltraFeedback-binarized (Tunstall et al., 2023) dataset, akin to the UltraChat dataset, specifically designed for the chat completion task. Comprising 63k pairs of selected and rejected responses corresponding

to specific inputs, the UltraFeedback-binarized dataset is employed for training alignment models.

KTO outperforms other alignment methods. The findings depicted in Figures 2 and 3 indicate that KTO surpasses other alignment methods in MT-Bench, and across all academic benchmarks, it exhibits superior performance, with the exception of MMLU (See Table 1). Particularly noteworthy is KTO's remarkable performance on GSM8K, highlighting its strong aptitude for solving mathematical problems(Mathematics plot in Figure 2).

Model	DPO	KTO	IPO	CPO	SFT
Mistral	63.14	62.31	62.44	62.61	60.92
Mistral+SFT	59.88	59.53	59.87	59.14	-

Table 1: Performance comparison of alignment methods on MMLU across two scenarios: 1) Fine-tuning a pre-trained model (Mistral) using alignment methods, and 2) Fine-tuning an SFT model (Mistral+SFT) using alignment methods. "-" represents that there is no value for this model. We note that the MMLU score for the Mistral model fine-tuned with SFT is 60.92.

Alignment methods don't require a large training set. The results depicted in Figure 3 reveal that all alignment methods perform better with a smaller training set. We posit that in the typical alignment process, a significant portion of model alignment occurs during the SFT phase. Therefore, when aiming to enhance the performance of the SFT model with methods like KTO, DPO, IPO, and CPO, it is beneficial to utilize a smaller dataset for training. In essence, there exists a trade-off between aligning with SFT and aligning with RL-free methods to achieve optimal performance.

SFT is still enough. Another intriguing observation is that none of the alignment methods outperform SFT in MMLU (See Table 1). This suggests that SFT remains superior to other methods for multitask understanding. Additionally, apart from the KTO algorithm in reasoning, truthfulness, and question answering, SFT demonstrates comparable performance (See Reasoning, Question Answering, and Truthfulness plots in Figure 2). This indicates that alignment methods struggle to achieve notable performance improvements in these tasks.

4.2 Scenario 2: Fine-tune a Pre-Train Model

Motivation. In this scenario, we train a pre-trained model directly with alignment methods on the UltraFeedback dataset. Several motivations underlie this scenario. Firstly, we seek to determine whether alignment methods necessitate the SFT phase. Secondly, we aim to compare the performance of models aligned with DPO, CPO, KTO, and IPO against those trained with SFT. Lastly, we aim to illustrate the impact of the SFT phase on various tasks by comparing the performance of models with and without this component.

Models. We employ `Mistral-7B-v0.1` as the pre-trained model and fine-tune it with DPO, CPO, KTO, and IPO. Further information regarding the training and evaluation process can be found in the Appendix A.

Datasets. We train an SFT model using the UltraChat dataset, which contains 200k examples generated by `GPT-3.5-TURBO` across 30 topics and 20 text material types, providing a high-quality dataset. Additionally, for training the pre-trained model with alignment methods, we utilize the UltraFeedback dataset, as explained in Section 4.1. It is worth noting that both UltraChat and UltraFeedback were curated specifically for the chat completion task.

KTO and CPO don't require SFT. The findings presented in Figure 1 indicate that skipping the SFT phase resulted in Mistral+IPO and Mistral+DPO performing poorly in the dialogue system, as they attained lower scores compared to SFT. However, Mistral+KTO and Mistral+CPO achieved scores comparable to Mistral+SFT.

SFT significantly affects academic benchmarks. The results depicted in Figure 2 reveal several key findings. Firstly, skipping the SFT phase leads to a marginal improvement in reasoning performance without significant impact. Secondly, there is a notable and consistent improvement across all alignment methods except IPO in GSM8K and TruthfulQA benchmarks. Moreover, in the MMLU benchmark, skipping the SFT phase not only enhances performance but also results in all alignment methods outperforming the SFT baseline (See Table 1).

4.3 Scenario 3: Fine-tune an Instruction Tuned Model

Motivation. The primary motivation for this scenario is to investigate the impact of the instruction-tuned model on the performance of various alignment methods. Thus, we train an instruction-tuned

Model	ARC	HellaSwag	Winogrande	BB-sports	BB-casual	BB-formal	PIQA	Average
Mistral-Instruct+SFT	61.17	81.93	76.87	71.39	60	50.73	83.02	69.3
Mistral-Instruct+IPO	63.05	84.69	77.26	75.25	59.47	51.65	80.41	70.25
Mistral-Instruct+KTO	62.71	85.52	77.5	74.23	61.57	51.23	81.55	70.62
Mistral-Instruct+CPO	52.38	80.95	77.5	72.31	58.94	52.02	81.55	67.95
Mistral-Instruct+DPO	63.48	85.34	77.34	74.64	59.47	51.12	81.01	70.34

Table 2: Performance comparison of various alignment methods in scenario 3 on reasoning benchmarks. To assess reasoning abilities, we focused on common sense reasoning, logical reasoning, and causal reasoning (See Section 4.3).

Model	GSM8K	MMLU	TruthfulQA	OpenBookQA	BoolQ	Average
Mistral-Instruct+SFT	37.68	61.03	49.46	48.4	86.02	67.21
Mistral-Instruct+IPO	38.05	60.72	66.97	48.2	85.9	67.05
Mistral-Instruct+KTO	38.28	61.72	66.97	49.4	86.17	67.78
Mistral-Instruct+CPO	38.51	60.46	63.9	46.8	84.98	65.89
Mistral-Instruct+DPO	33.58	61.61	68.22	49.2	85.19	67.19

Table 3: Performance evaluation of alignment methods in scenario 3, focusing on solving mathematics problems, truthfulness, multi-task understanding, and question-answering tasks. For more detailed information, refer to Section 4.3.

Model	Align	First Turn (Score)	Second Turn (Score)	Average (Score)
Mistral-Instruct	SFT	7.78	7.16	7.47
Mistral-Instruct	DPO	7.61	7.42	7.51
Mistral-Instruct	KTO	7.66	7.36	7.51
Mistral-Instruct	CPO	7.18	6.98	7.08
Mistral-Instruct	IPO	7.88	7.32	7.60

Table 4: Performance comparison of alignment methods using an instruction-tuned model without SFT on MT-Bench (More details in Section 4.3).

model with KTO, IPO, DPO, and CPO and evaluate their performance across different benchmarks. To ensure a fair comparison, we assess the performance of the alignment methods alongside the SFT method to discern their effects. Consequently, in this scenario, we bypass the SFT phase and utilize the instruction-tuned model for evaluation.

Models. We utilize Mistral-instruct-7B-v0.2 as the instruction-tuned model and fine-tune it with DPO, CPO, KTO, and IPO. Further information regarding the training and evaluation process can be found in the Appendix A.

Datasets. Like Section 4.2, we train an SFT model using the UltraChat dataset. Additionally, we employ UltraFeedback to train the pre-trained model with alignment methods, as described in scenario 1. It's worth noting that both UltraChat and UltraFeedback were curated specifically for the chat completion task.

Aligning an instruction-tuned model significantly affects truthfulness. The findings presented in Table 3 indicate that KTO and IPO outperform SFT by 17.5%, whereas KTO, based on a pre-trained model, outperforms SFT by 9.5% (See Table 9 in Appendix B). This underscores the high effectiveness of an instruction-tuned model, particularly in terms of truthfulness. Additionally, it is observed that KTO surpasses other methods in MT-Bench (See Table 4).

SFT based on instruction tuning is enough. The findings presented in Tables 2 and 3 indicate that SFT demonstrates comparable performance across reasoning, mathematics, question-and-answer, and multi-task understanding benchmarks. While alignment methods exhibit better performance than SFT, the challenge of preparing the preference dataset remains significant, making SFT preferable in most cases. It is noteworthy that in MT-Bench, CPO performs even worse compared to SFT, suggesting that models fine-tuned with CPO exhibit weaker performance in the dialogue system compared to those fine-tuned with SFT (See Table 4).

Same or higher than GPT-4. We observe that while improving overall performance, there is a decrease in the model's ability in certain domains (See Figure 4). However, another intriguing discovery is that not only does KTO achieve an equal score with GPT-4 in Humanities, but CPO also outperforms GPT-4 in the STEM domain (See Figure

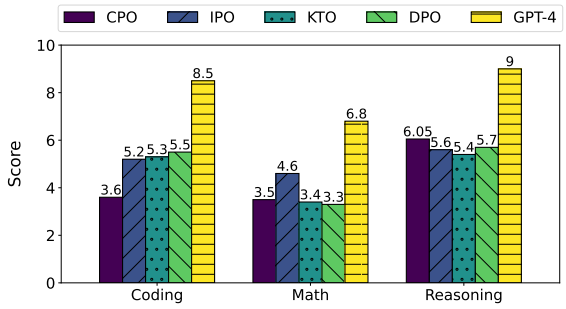

Figure 4: Performance comparison of the alignment methods based on the instruction-tuned model on MT-Bench. There exists a substantial disparity in performance between GPT-4 and alignment methods across reasoning, mathematics, and coding tasks. The score is between 0 and 10 generated by GPT-4.

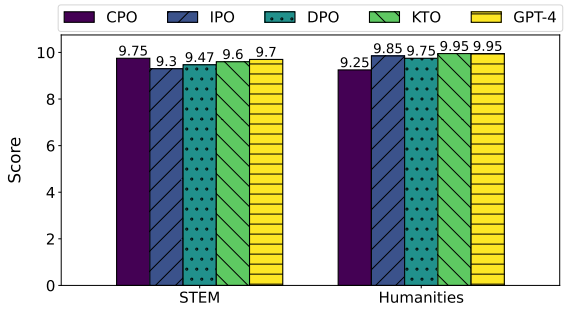

Figure 5: Alignment methods based on instruction-tuned model not only demonstrate equivalent performance to GPT-4 but can also outperform it, particularly in comparisons based on MT-Bench score. The score is between 0 and 10 generated by GPT-4.

5). This finding highlights the alignment methods' capability to rival state-of-the-art models such as GPT-4 with smaller models.

5 Conclusions

In this paper, we assessed the performance of RL-free algorithms such as DPO, KTO, IPO, and CPO across various tasks, including reasoning, mathematics problem-solving, truthfulness, question answering, and multi-task understanding in three distinct scenarios. Our findings show that KTO consistently outperforms the other alignment methods in all three scenarios. However, we noted that these techniques do not significantly enhance model performance in reasoning and question answering during regular alignment processes, though they significantly improve mathematical problem-solving. Our research also indicates that alignment methods are particularly sensitive to the volume of training data, performing best with smaller data subsets. Notably, unlike DPO, other methods, such as KTO and CPO, can bypass the SFT part and achieve comparable performance on MT-Bench. We primarily utilized an instruction-tuned model as the base for alignment, which significantly influenced truthfulness. Although this study focused on dialogue systems, we plan to extend our research to include other areas, such as safety, believing our results hold significant implications for the alignment community.

6 Limitations

A key constraint is the challenge of preparing an appropriate dataset for training alignment methods. Furthermore, ranking multiple preferences presents another limitation that can affect the quality of the research. Inefficiencies in learning and memory also hinder progress in alignment research. Additionally, using essential benchmarks like MT-Bench and AlpacaEval (Dubois et al., 2023) is costly and necessitates access to GPT-4 for evaluation.

Ethics Statement

We have used AI assistants (Grammarly and ChatGPT) to address the grammatical errors and rephrase the sentences.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem,

Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language.

Heejong Bong and Alessandro Rinaldo. 2022. Generalized results for the existence and consistency of the mle in the bradley-terry-luce model.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback.

Kawin Ethayarajh, Winnie Xu, Dan Jurafsky, and Douwe Kiela. 2023. Human-aware loss functions (halos). Technical report, Contextual AI.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

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.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024. Statistical rejection sampling improves preference optimization.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis insights from training gopher.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lewis Tunstall, Edward 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. 2023. Zephyr: Direct distillation of lm alignment.

AMOS TVERSKY and DANIEL KAHNEMAN. 1992. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence?

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Appendix

A Training and Validation Details

We utilized the Transformer Reinforcement Learning (TRL) library for fine-tuning (von Werra et al., 2020). It's noted that the notation "+" is used to indicate that a model has been fine-tuned with a specific algorithm, such as "+DPO". All models were trained using the AdamW optimizer without weight decay. Furthermore, parameter-efficient techniques such as LoRA (Hu et al., 2021) were not employed. The experiments were conducted on 6 A100 GPUs, utilizing bfloat16 precision, and typically required 5-8 hours to complete. All models are trained for one epoch, employing a linear learning rate scheduler with a peak learning rate of 5e-7 and 10% warmup steps. Additionally, the global batch size is set to 8, and $\beta = 0.1$ is used to regulate the deviation from the reference model. For every dataset used in our evaluation, we detail the count of few-shot examples utilized along with the specific metric employed for assessment (See Table 5).

B More Details for Scenarios 1 and 2

In this section, we present the details for reasoning benchmarks for scenario 1 in Table 6 and for scenario 2 in Table 7. Additionally, we provide details for other benchmarks in Tables 8 and 9.

Datasets	ARC	TruthfulQA	GSM8K	Winogrande	HellaSwag	MMLU	BB-causal	BB-sports	BB-formal	OpenBookQA	BoolQ	PIQA
# few-shot	25	0	5	5	10	5	3	3	3	1	10	0
Metric	acc_norm	mc2	acc	acc	acc_norm	acc	mc	mc	mc	acc_norm	acc	acc_norm

Table 5: Detailed information of Open LLM Leaderboard, Big Bench and other benchmarks.

Model	ARC	HellaSwag	Winogrande	BB-sports	BB-casual	BB-formal	PIQA	Average
Mistral+SFT	60.41	81.69	74.19	61.76	51.57	51.4	81.66	66.09
Mistral+SFT+DPO	61.60	82.11	77.82	72.31	51.57	51.28	81.33	65.64
Mistral+SFT+IPO	59.56	81.08	76.55	68.76	51.05	52.03	81.55	67.22
Mistral+SFT+CPO	54.52	79.24	76.4	72.21	53.68	52.18	80.9	67.1
Mistral+SFT+KTO	57.84	82.19	77.26	73.52	57.89	51.19	81.93	68.83

Table 6: Performance comparison of the various alignment methods in scenario 1 on reasoning benchmarks. To assess reasoning abilities, we focused on common sense reasoning, logical reasoning, and causal reasoning.

Model	ARC	HellaSwag	Winogrande	BB-sports	BB-casual	BB-formal	PIQA	Average
Mistral+SFT	60.41	81.69	74.19	61.76	51.57	51.4	81.66	66.09
Mistral+DPO	63.82	84.99	78.92	74.64	57.89	50.69	83.02	70.56
Mistral+IPO	68	81.7	77.03	73.93	58.94	52.3	83.18	70.72
Mistral+CPO	60.49	82.21	78.45	72	55.78	52.88	82.15	69.13
Mistral+KTO	64.5	85.31	78.68	77.68	56.84	51.05	83.35	71.05

Table 7: Performance comparison of the various alignment methods in scenario 2 on reasoning benchmarks. To assess reasoning abilities, we focused on common sense reasoning, logical reasoning, and causal reasoning.

Model	GSM8K	MMLU	TruthfulQA	OpenBookQA	BoolQ	Average
Mistral+SFT	26.76	60.92	43.73	43.2	85.16	64.18
Mistral+SFT+DPO	30.62	59.88	44.78	46	85.29	65.64
Mistral+SFT+IPO	31.31	59.87	41.37	45	84.77	64.88
Mistral+SFT+CPO	27.89	59.14	45.1	44	84.28	64.14
Mistral+SFT+KTO	34.72	59.53	45.9	47	85.87	66.43

Table 8: Evaluation of alignment methods in scenario 1, focusing on solving mathematics problems, truthfulness, multi-task understanding, and question-answering tasks.

Model	GSM8K	MMLU	TruthfulQA	OpenBookQA	BoolQ	Average
Mistral+SFT	26.76	60.92	43.73	43.2	85.16	64.18
Mistral+DPO	36.01	63.14	51.2	49.4	86.78	68.09
Mistral+IPO	19.86	62.44	52.28	50	86.78	68.39
Mistral+CPO	34.19	62.61	50.04	47.4	86.14	66.77
Mistral+KTO	42.15	62.31	52.98	48.8	86.78	67.79

Table 9: Evaluation of alignment methods in scenario 2, focusing on solving mathematics problems, truthfulness, multi-task understanding, and question-answering tasks.