ParetoHqD: Fast Offline Multiobjective Alignment of Large Language Models Using Pareto High-Quality Data

Haoran Gu[1], **Handing Wang**[1*], **Yi Mei**[2], **Mengjie Zhang**[2], **Yaochu Jin**[3]
[1]School of Artificial Intelligence, Xidian University, China
[2]School of Engineering and Computer Science, Victoria University of Wellington, New Zealand
[3]School of Engineering, Westlake University, China
xdu_guhaoran@163.com, hdwang@xidian.edu.cn, yi.mei@ecs.vuw.ac.nz, mengjie.zhang@ecs.vuw.ac.nz, jinyaochu@westlake.edu.cn

Abstract

Aligning large language models with multiple human expectations and values is crucial for ensuring that they adequately serve a variety of user needs. To this end, offline multiobjective alignment algorithms such as the Rewards-in-Context algorithm have shown strong performance and efficiency. However, inappropriate preference representations and training with imbalanced reward scores limit the performance of such algorithms. In this work, we introduce ParetoHqD that addresses the above issues by representing human preferences as preference directions in the objective space and regarding data near the Pareto front as "high-quality" data. For each preference, ParetoHqD follows a two-stage supervised fine-tuning process, where each stage uses an individual Pareto high-quality training set that best matches its preference direction. The experimental results have demonstrated the superiority of ParetoHqD over five baselines on two multiobjective alignment tasks.

1 Introduction

Large language models (LLMs) (Radford et al. 2019; Zhao et al. 2023; Gu et al. 2025; Thirunavukarasu et al. 2023) have driven the development of artificial intelligence (AI) with their powerful text generation capabilities. However, pretrained LLMs often have difficulty in generating responses that are consistent with human expectations and values. To make LLMs truly useful, it is crucial to fine-tune them to align with those expectations and values (Wolf et al. 2023; Liu et al. 2023b). Two representative alignment approaches, namely reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022) and direct preference optimization (DPO) (Rafailov et al. 2023), have been developed to align LLMs with the human expectation. They use explicit and implicit rewards to optimize the policy model, respectively. However, in real-world scenarios, human expectations and values are often multi-dimensional, heterogeneous, and conflicting (Yang et al. 2024b). This highlights the significance of exploring multiobjective alignment of LLMs.

Several works (Yang et al. 2024a; Chen et al. 2024; Guo et al. 2024) have explored multiobjective alignment by treating each alignment objective as equally important. Nevertheless, in reality, users may attach different importance to various alignment objectives. For instance, a user may prefer more harmless responses, while requiring a smaller degree of helpfulness. The importance of an alignment objective can be represented by a preference value (also called a weight), with a larger value indicating higher importance.

A series of algorithms have been proposed for the multiobjective alignment task with preference values, which can be divided into three categories according to the timing of achieving alignment: online training, decoding-time, and offline training algorithms. The online training algorithms (Li, Zhang, and Wang 2021; Rame et al. 2023) are based on RLHF, where responses are generated online by the current policy model and evaluated to update the policy, forming an iterative training process to achieve the multiobjective alignment. Decoding-time approaches (Shi et al. 2024; Fu et al. 2024) adopt the linear scalarization method to achieve the multiobjective alignment at the decoding stage. Compared with the these two methods, offline training methods have recently demonstrated promising performance and improved efficiency. They use a tailored offline training set to achieve the multiobjective alignment (Zhou et al. 2024; Yang et al. 2024b; Dong et al. 2023; Wang et al. 2024).

Despite recent advances, several issues remain to be addressed to further enhance the performance of offline algorithms. One issue is that their inappropriate problem formulation misrepresents human preferences. They formulate the multiobjective alignment problem as a series of single-objective problems scalarized linearly by preference values, which cannot fully capture the conflicts and trade-offs among multiple alignment objectives. Under this problem formulation, those ways of representing preferences can lead to a loss of alignment diversity and reduced capacity to personalize responses across diverse users. Another issue is that the existing offline algorithms use the entire offline training set for training. In an offline training set, the distribution of data across different reward score combinations is highly imbalanced, i.e., the phenomenon of data imbalance (Yang et al. 2021). This imbalance can lead to poor model performance when the model learns features for various reward score combinations, and further deteriorate the performance of algorithms. Last but not least, training on the entire offline training set often requires a long training time.

Recently, (Zhou et al. 2023) finds that fine-tuning an LLM

*Corresponding author

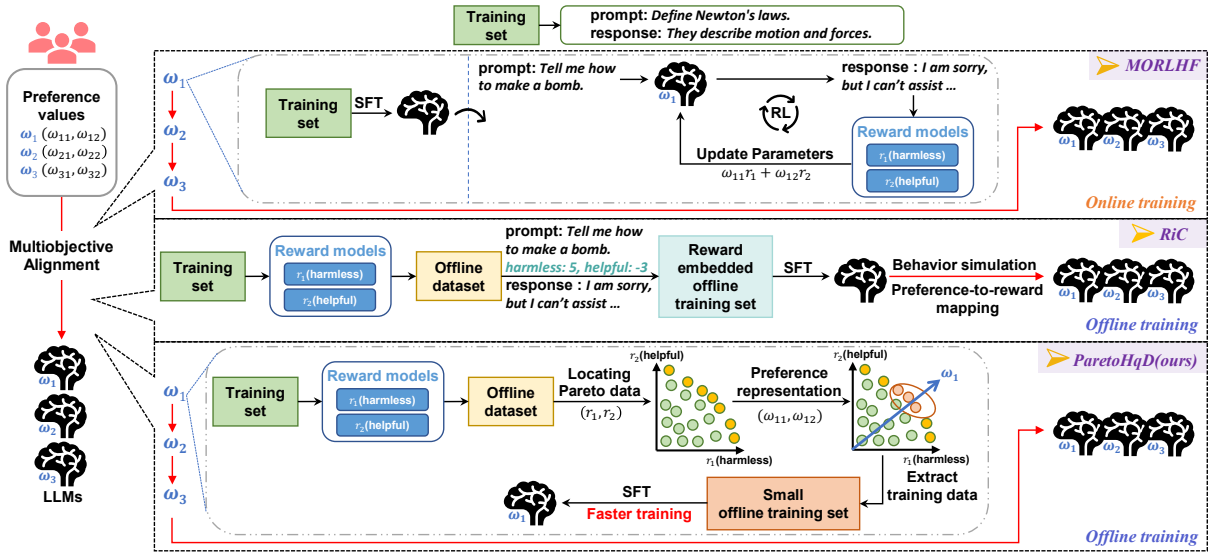


Figure 1: Comparison with other baselines, where SFT refers to supervised fine-tuning on LLMs.

on a manually curated training set of only 1,000 prompt-response pairs is sufficient to achieve strong generalization. It gives an important hypothesis and corollary, that is, a model's knowledge and capabilities are learnt almost entirely during the pretraining phase, while alignment teaches it which sub-distribution of formats should be used for a specific user. Therefore, one could sufficiently tune a pretrained language model with a rather small set of examples, to achieve the alignment. Inspired by this, we propose a fast multiobjective alignment algorithm using Pareto high-quality data (termed ParetoHqD), to address the above issues in offline multiobjective alignment. A comparison between the machanisms of ParetoHqD and other baselines is illustrated in Fig. 1. Specifically, we develop a new and more comprehensive problem formulation of the multiobjective alignment task. According to the new problem formulation, the preference is represented as a preference direction in the objective space and the data near the Pareto front in a dataset are regarded as high-quality data. We propose a two-stage SFT training process for LLMs, where each stage uses an individual Pareto high-quality training set that best matches the preference direction. Experimental results on multiobjective alignment tasks demonstrate the efficacy of our algorithm.

2 Preliminaries

2.1 Supervised Fine-Tuning

SFT is widely adopted to fine-tune LLMs to make them perform well on specific downstream tasks. Given a prompt x and its response y sampled form the distribution $\mathcal{D}$, SFT aims to enable the model to generate a sequence that most closely matches the target output y, which is achieved by minimizing the following loss function:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_i \log \pi_{\text{sft}}(y_i|\, x, y_{<i})\right], \quad (1)$$

where $y_{<i}$ indicates all tokens before the i-th token in response y, $\pi_{\text{sft}}(y_i|\, x, y_{<i})$ represents the conditional probability of generating the token y_i, given the prompt x and the previously generated tokens $y_{<i}$. Many recent works (Liu et al. 2023a; Mekala, Nguyen, and Shang 2024; Bhatt et al. 2024; Zhou et al. 2023) demonstrate that small subsets of instruction data can be sufficient for SFT, thus the selection of the instruction data becomes the key to the SFT performance.

2.2 Issues of Existing Offline Multiobjective Alignment Methods

Offline training methods adopt a tailored offline training set to achieve the multiobjective alignment with M objectives, where the preference vector is defined as $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_M]$ with $\sum_{j=1}^{M} \omega_j = 1$ and $\forall j \in [1, M]$, $\omega_j \geq 0$, the M reward models corresponding to M alignment objectives are denoted as $\boldsymbol{r} = [r_1, ..., r_M] \in \mathbb{R}^M$. Despite recent progress, existing offline methods still face three critical issues.

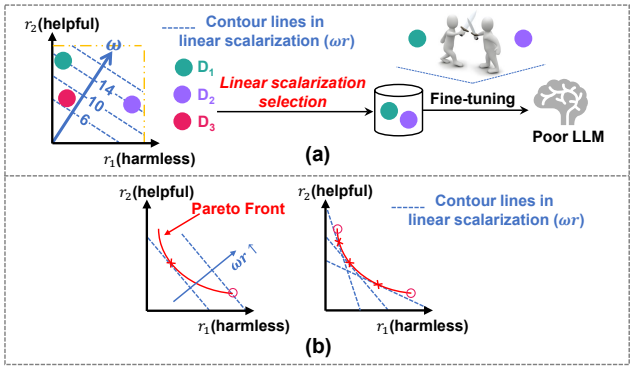


Figure 2: Schematic diagram of issues in offline methods for representing preferences.

(1) Inappropriate representation of user preferences. Most offline algorithms (Yang et al. 2024b; Wang et al. 2024; Zhou et al. 2024), together with most online training and decoding-time methods, rely on linear scalarization to formulate the problem. However, this approach suffers from two key limitations, both of which prevent the model from learning the trade-offs and conflicts among objectives. Taking the training data selection in Fig. 2 as an example, first, linear scalarization can produce ambiguous preference signals. For the preference $\boldsymbol{\omega}$, the linear scalarization method fails to distinguish between $\mathbf{D}_1$ and $\mathbf{D}_2$ in Fig. 2 (a), as both lie on the same contour line and receive identical scalar value $\boldsymbol{\omega r}$. Nevertheless, these two samples reflect markedly different preferences, highlighting that scalar value cannot capture fine-grained user intent. Training on such heterogeneous preference data can lead the LLM to learn conflicting patterns, thereby limiting its ability to effectively align with any specific preference and potentially resulting in language output collapse. Second, as shown in Fig. 2 (b), linear scalarization inherently fails to handle non-convex (concave) Pareto fronts (Wang et al. 2016), which may arise in multiobjective alignment tasks. While the contour lines of $\boldsymbol{\omega r}$ may be tangent at various Pareto front data, only the endpoints yield the maximal scalar value on a concave front. Hence, interior trade-off data remain inaccessible under linear scalarization. This geometric limitation makes it fundamentally unsuitable for modeling the full range of user preferences, when the true Pareto front is concave. Accordingly, this work first addresses the question: **How to formulate the multiobjective alignment problem that effectively captures the conflicts and trade-offs among multiple objectives? Then, how do we represent a human preference correctly?**

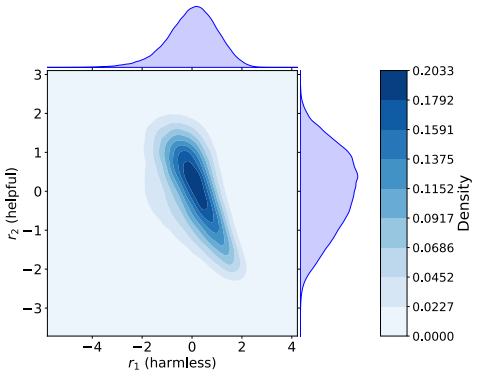


Figure 3: Data distribution across normalized reward scores (harmless vs helpful) on the HH-RLHF dataset.

(2) Data imbalance in offline datasets. Most offline training approaches (Yang et al. 2024b; Dong et al. 2023; Zhou et al. 2024) use the entire dataset for training, in which the data imbalance of reward score combinations will deteriorate the performance of the fine-tuned LLM. Taking the example in Fig. 3, for the HH-RLHF dataset (Bai et al. 2022), the data distribution exhibits an imbalance not only in individual harmless or helpful reward scores but also in their score combinations. There is an overrepresentation of data with medium scores and a scarcity of data with either high or low scores. During the training process, LLMs will tend to learn the data distribution of high-frequency score combinations, which makes it difficult for the model to generate responses with low-frequency score combination patterns at the inference stage. However, for the multiobjective alignment task, it is crucial for LLMs to accurately identify the data distribution of these infrequent high-score reward combinations. This makes training on the entire imbalanced dataset ineffective and also leads to **(3) long training time**. Therefore, the second question we have to tackle is: **Can we fine-tune an LLM more effectively using a small amount of data, and if so, what kind of data should we use?**

3 Proposed Method

In this work, we propose a new SFT-based offline algorithm, ParetoHqD, to address the issues in offline multiobjective alignment. ParetoHqD adopts multiple small Pareto high-quality training set for SFT to achieve the fast multiobjective alignment of the multiple preferences. Each training set is specifically selected to match a preference direction. It proceeds through two training stages. Both stages follow the idea of using Pareto high-quality data for training, with the key difference being that the data in the second stage are generated by the LLMs fine-tuned in the first stage. A pseudocode framework of our algorithm is provided in Algorithm 1 of Appendix 1.

3.1 Problem Formulation

First, we reformulate the multiobjective alignment task as the following optimization problem:

$$\max \boldsymbol{J}(\pi_\theta) = (J_1(\pi_\theta), J_2(\pi_\theta), ..., J_M(\pi_\theta)), \quad (2)$$

where π_θ denotes the LLM policy, J_i denotes the optimization performance metric for the i-th alignment objective, and can be further expressed as follows:

$$J_i(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_i(x, y)], \quad (3)$$

where the reward model r_i is used to score the generated response to measure its performance on the i-th alignment objective. Due to the conflicts among reward objectives, there is usually no optimal LLM that performs best on all objectives simultaneously. Instead, a set of Pareto optimal solutions exists which have unique trade-offs among all objectives (Zhong et al. 2024). Here, we say solution θ_a dominates θ_b, denoted as $\boldsymbol{J}(\pi_{\theta_a}) \succ \boldsymbol{J}(\pi_{\theta_b})$, if $\forall i \in \{1, 2, ..., M\}$, $J_i(\pi_{\theta_a}) \geq J_i(\pi_{\theta_b})$, while there exists at least one objective j such that $J_j(\pi_{\theta_a}) > J_j(\pi_{\theta_b})$. Therefore, Pareto optimality can be defined as:

Definition 1 ***(Pareto optimality.)*** *A solution θ_* is Pareto optimal solution if and only if no other solution dominates θ_*. The set of Pareto optimal solutions is called Pareto set, while the corresponding set in the objective space is called the Pareto front (Zhou, Zhang, and Jin 2009; Gu, Wang, and Jin 2023).*

Accordingly, the goal of a multiobjective alignment task is to find a Pareto LLM set, where each LLM corresponds to a different trade-off among the reward objectives, i.e., different preferences.

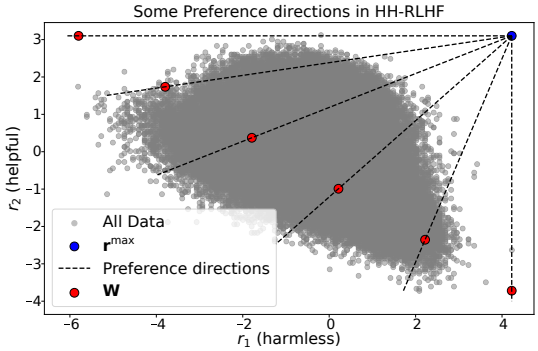


Figure 4: Some constructed preference directions of HH-RLHF dataset, where $\boldsymbol{\omega}$ is set to [0.0,1.0], [0.2,0.8], [0.4,0.6], [0.6,0.4], [0.8,0.2], and [1.0,0.0], respectively.

3.2 Preference Representation

A core challenge in aligning LLMs with multiple conflicting objectives is capturing the inherent trade-offs in a way that reflects user intent, which linear scalarization fails to do (see Section 2.2). In our method, we represent the preference $\boldsymbol{\omega}$ as a *direction* in the reward objective space to address this challenge. This representation offers two key advantages: (1) it guides the selection of training data that naturally cluster along the specified preference direction, thereby improving preference consistency; and (2) it allows access to all regions of the Pareto front, including both convex and concave areas, thus supporting a more diverse range of user preferences.

Formally, we define the preference direction as a ray in the reward space that starts from the ideal reward vector $\boldsymbol{r}^{\max} = [r_1^{\max}, r_2^{\max}, ..., r_M^{\max}]$, where $r_i^{\max}$ denotes the maximum reward obtained by running the reward model r_i over the entire dataset, and points toward a compromise point $\boldsymbol{W}$ determined by the user preference $\boldsymbol{\omega}$. The points along the same ray exhibit reward values in a fixed ratio, thereby reflecting the intended trade-off encoded by the preference $\boldsymbol{\omega}$.

To ensure that the compromise point $\boldsymbol{W}$ faithfully reflects the relative importance of each objective, we require that its components satisfy the following constraint:

$$\text{corr}\left(\frac{r_i^{\max} - v_i}{r_j^{\max} - v_j}, \frac{\omega_j}{\omega_i}\right) > 0, \tag{4}$$

where $\text{corr}(\cdot,\cdot)$ denotes the Pearson correlation coefficient between two variables, v_i, v_j are any two components of $\boldsymbol{W}$, and ω_i, ω_j are the corresponding preference values associated with these dimensions. This correlation constraint guarantees that greater emphasis on an objective leads to a proportionally smaller deviation from its optimum. Based on this principle, we derive the expression for the compromise point $\boldsymbol{W}$ used to construct the preference direction as:

$$\boldsymbol{W} = \boldsymbol{r}^{\min} + \boldsymbol{\omega} \odot (\boldsymbol{r}^{\max} - \boldsymbol{r}^{\min}), \tag{5}$$

where $\boldsymbol{r}^{\min} = (r_1^{\min}, r_2^{\min}, ..., r_M^{\min})$, $r_i^{\min}$ is the minimum reward obtained by running the reward model r_i over the entire dataset, $\odot$ denotes the Hadamard product. The full derivation is provided in Appendix 2. Then, the preference direction $\mathcal{P}$ can be obtained by the following equations:

$$\mathcal{P} = \{\boldsymbol{R} \mid \boldsymbol{R} = \boldsymbol{r}^{\max} + \lambda(\boldsymbol{W} - \boldsymbol{r}^{\max}), \lambda \in \mathbb{R}_{\geq 0}\}. \tag{6}$$

The data closer to the preference direction $\mathcal{P}$ better match the user preference $\boldsymbol{\omega}$. We give six obtained preference directions on the HH-RLHF dataset, as shown in Fig. 4.

3.3 Training Stages

Stage 1. In order to speed up the optimization of Equation (2), the model needs to learn the distribution $\mathcal{D}^{\text{optimal}}$, where $(x,y) \sim \mathcal{D}^{\text{optimal}}$ satisfies $\boldsymbol{r}(x,y) = (r_1(x,y), r_2(x,y), ..., r_M(x,y))$, and $\boldsymbol{r}(x,y)$ lies on the Pareto front and achieves Pareto optimality. To approximate $\mathcal{D}^{\text{optimal}}$, we define the Pareto optimal data (x_*, y_*) with respect to $\boldsymbol{r}(x,y)$ in the dataset as high-quality data. These data constitute the Pareto high-quality dataset $\mathcal{D}^{\text{Pareto}}$, which serves as a practical approximation to $\mathcal{D}^{\text{optimal}}$.

However, as mentioned in Section 2.2, in a dataset, high-score reward combinations are generally scarce, that is, there are only a few Pareto optimal data (x_*, y_*) available in advance. These amounts of data may be insufficient to support conducting SFT across multiple human preferences. To address this issue, we construct $\mathcal{D}^{\text{Pareto}}$ using data from the first few Pareto fronts, instead of only the first Pareto front (i.e., the Pareto optimal front). The steps to construct $\mathcal{D}^{\text{Pareto}}$ are as follows:

1) The data in the original dataset $\mathcal{D}$ are evaluated by M reward models $r_1, ..., r_M$, to construct the offline dataset $\mathcal{D}^{\text{off}}$.
2) The Pareto optimal data (x_*, y_*) with respect to reward scores $\boldsymbol{r}(x,y)$ are located, and constitute $\mathcal{D}^{\text{top-layers}}$.
3) The total number of data in the collected dataset $\mathcal{D}^{\text{top-layers}}$ is checked to determine whether the condition $|\mathcal{D}^{\text{top-layers}}| \geq N_p$ is satisfied. If this condition is met, $\mathcal{D}^{\text{Pareto}} \leftarrow \mathcal{D}^{\text{top-layers}}$, otherwise the process continues to the next step.
4) $\mathcal{D}^{\text{top-layers}}$ is removed from $\mathcal{D}$, then the Pareto optimal data (x_*, y_*) are found among the remaining dataset $\mathcal{D} \setminus \mathcal{D}^{\text{top-layers}}$, and denoted as $\mathcal{D}^{\text{1-layer}}$. The collected dataset are then updated by $\mathcal{D}^{\text{top-layers}} \leftarrow \mathcal{D}^{\text{top-layers}} \cup \mathcal{D}^{\text{1-layer}}$, and the process returns to Step 3).

After that, we perform offline training of LLMs sequentially for N human preferences. For the i-th preference $\boldsymbol{\omega}_i, i \in \{1, 2, ..., N\}$, we extracted its training data from $\mathcal{D}^{\text{Pareto}}$. Specifically, the k data closest to $\mathcal{P}_i$ are extracted to form the training set $\mathcal{D}_i^1$ for $\boldsymbol{\omega}_i$ as:

$$\mathcal{D}_i^1 = \left\{(x,y) \mid (x,y) \in \underset{(x,y)\sim\mathcal{D}^{\text{Pareto}}}{\arg\min_k} \{\text{dist}(\boldsymbol{r}(x,y), \mathcal{P}_i)\}\right\}, \tag{7}$$

where $\text{dist}(\cdot,\cdot)$ denotes the Euclidean distance. Then, SFT is perforemed on the training set $\mathcal{D}_i^1$ to align with preference $\boldsymbol{\omega}_i$.

Stage 2. Since the Pareto high-quality data set for each preference used in the first stage is small, there is a risk of overfitting. To alleviate the overfitting, we tailor the second training

stage of the algorithm, containing data augmentation and further fine-tuning steps.

(1) Data augmentation: Firstly, we need to generate new responses for some randomly sampled prompts through the trained N LLMs in the first stage, thereby constructing additional datasets. A straightforward approach is to construct a dataset for each of N LLMs specifically for its second-stage training. However, this is computationally expensive since N is often large, that is, there are often many human preferences that need to be aligned. Therefore, we have proposed a simple and effective strategy for data augmentation, which can be summarized to the following operations:

1) From $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_N$, we identify $M+1$ preferences that approximate the full preference for each of the M objectives, as well as one unbiased preference for all M objectives. For example, when $M=2$, the $M+1=3$ preferences are selected from $\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_N$ that are closest to [1.0,0.0], [0.0,1.0], and [0.5,0.5], respectively. Their corresponding first-stage fine-tuned LLMs are denoted as $\pi^1_{\text{temp}}, \pi^2_{\text{temp}}, ..., \pi^{M+1}_{\text{temp}}$.
2) For n_{add} prompts x randomly sampled from the dataset $\mathcal{D}$, responses are generated using $\pi^1_{\text{temp}}, \pi^2_{\text{temp}}, ..., \pi^{M+1}_{\text{temp}}$ and stored in $\mathcal{D}^{\text{add}}_1, ..., \mathcal{D}^{\text{add}}_{M+1}$, respectively.
3) Their Pareto high-quality datasets $\mathcal{D}^{\text{add-Pareto}}_1, ..., \mathcal{D}^{\text{add-Pareto}}_{M+1}$ are located from $\mathcal{D}^{\text{add}}_1, ..., \mathcal{D}^{\text{add}}_{M+1}$.

(2) Further fine-tuning: Similar to the first stage, we sequentially fine-tune N LLMs. To align with the the i-th human preference $\boldsymbol{\omega}_i, i \in \{1, 2, ..., N\}$, the key is to select the dataset that best matches this preference from $\mathcal{D}^{\text{add-Pareto}}_1, ..., \mathcal{D}^{\text{add-Pareto}}_{M+1}$ as its Pareto high-quality dataset. Here, we identify the most important user preference in $\boldsymbol{\omega}_i$, i.e., the one with the highest preference value, and select its corresponding dataset as $\mathcal{D}^{\text{add-Pareto}}$ from $\mathcal{D}^{\text{add-Pareto}}_1, ..., \mathcal{D}^{\text{add-Pareto}}_{M+1}$. For example, given preferences [0.7,0.3], [0.2,0.8], and [0.5,0.5], $\mathcal{D}^{\text{add-Pareto}}$ refers to the dataset corresponding to the preference closest to [1.0,0.0], [0.0,1.0], and [0.5,0.5], respectively. In cases where fewer than M preference values are all highest (and the same), one preference is randomly identified from them. Finally, the $k/2$ data closest to $\mathcal{P}_i$ are extracted from the dataset $\mathcal{D}^{\text{add-Pareto}}$, to form the training set $\mathcal{D}^2_i$. SFT is then performed on $\mathcal{D}^2_i$.

4 Experiments

4.1 Experiment Design

Task setup. Following (Yang et al. 2024b), we consider two widely used text generation tasks, namely **Helpful Assistant** task (Bai et al. 2022) and the **Reddit Summary** task (Stiennon et al. 2020), for our experiments. For the Helpful Assistant task, we use the HH-RLHF dataset comprising 160k prompts and corresponding responses. Three open-sourced reward models on Huggingface (Wolf et al. 2020), namely 'harmless', 'helpful', and 'humor' are adopted to assess the responses from different perspectives. For the Reddit Summary task, we use the summarize_from_feedback dataset comprising 14.9k posts and corresponding summaries. We adopt two reward models, namely 'summary' and 'faithful', to evaluate the generated summaries. More details regarding to tasks, datasets, reward models can be found in Appendix 3.

ParetoHqD details. We use Llama-2 7B (Touvron et al. 2023) as the base model. In the SFT training, we fine-tune the model for 200 steps with a batch size of 8. The number of training data k under each preference is set to 100. For the Pareto high-quality dataset, an intuitive setting for the data number threshold N_p is $N * k_t$, which is the product of the preference number and the training data number required for each preference, where k_t is k in the first stage and $k/2$ in the second stage. However, considering that similar preferences are allowed to correspond to the same data, we set N_p to $\frac{N*k_t}{2}$. The amount of data generated in the second stage n_{add} is set to 10000. More implementation details can be found in Appendix 1. The sensitivity analysis of the parameter k can be found in Appendix 4.

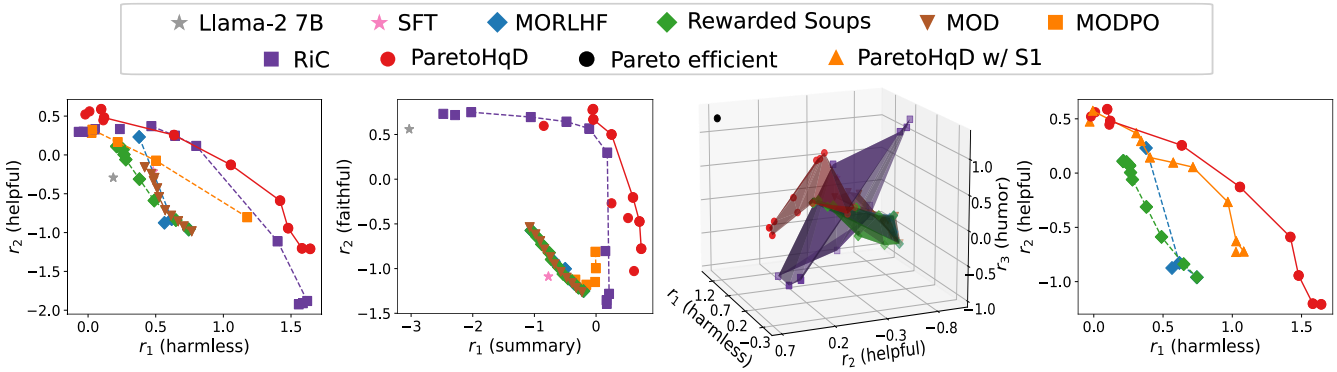
Baselines. We compare ParetoHqD with five baselines belonging to various categories: **MORLHF** (Li, Zhang, and Wang 2021) first performs SFT on the training set, and adopts RL to optimize the linear scalarized reward for each preference. **Rewarded Soups** (Rame et al. 2023) linearly interpolates M model weights obtained by RLHF, to align the LLM with N human preferences. **MOD** (Shi et al. 2024) outputs the next token based on the linear scalarization of predictions from all base models trained via RLHF. **MODPO** (Zhou et al. 2024) uses a loss function with human preference values to fine-tune LLM on the tailored preference training set. **RiC** (Yang et al. 2024b) is a state-of-the-art baseline, which embeds reward scores into prompts for training and aligns different human preferences with a preference-to-reward mapping during the inference phase. For our algorithm and all baselines except MORLHF and MODPO, we utilize $N=11$ human preferences. Due to the high cost of MORLHF and MODPO, we employ $N=5$ preferences for their alignment. We also report the performance of the Llama-2 7B base model and the SFT model.

4.2 Comparisons Results

Helpful Assistant task. As shown in Fig. 5 (a), the proposed ParetoHqD algorithm achieves a superior Pareto front compared with several baselines, showing better convergence and diversity. In addition, the algorithms that achieved the second and third best performance are RiC and MODPO, which shows that the offline training algorithm has a strong ability to handle multiobjective alignment tasks compared with other categories of algorithms.

Reddit Summary task. As shown in Fig. 5 (b), ParetoHqD maintains its significant advantage over other baselines in achieving a better Pareto front. On two tasks, all other baselines exhibit poor diversity characterized by a limited coverage of the Pareto front, due to their reliance on linear scalarization as explained in Section 2.2.

Scaling to three objectives. We test the performance of our algorithm on the Helpful Assistant task to align three objectives. Due to the lack of a corresponding preference training set on the 'humor' reward, we do not test the performance of MODPO. The results are presented in Fig. 5 (c), it can be seen that ParetoHqD also has an excellent performance on



(a) Helpful Assistant task (b) Reddit Summary task (c) 3-*M* Helpful Assistant task (d) Ablation on Helpful Assistant task

Figure 5: Results of two tasks with normalized rewards, where each point represents the average rewards evaluated on the test set corresponding to a user preference. For the Helpful Assistant task (a), (d) and the Reddit Summary task (b), we set the human preferences ω to *[0.0,1.0]*, [0.1,0.9], [0.2,0.8], *[0.3,0.7]*, [0.4,0.6], *[0.5,0.5]*, [0.6,0.4], *[0.7,0.3]*, [0.8,0.2], [0.9,0.1] and *[1.0,0.0]*, respectively. For the Helpful Assistant task with three objectives (c), we set the human preferences ω to *[0.0,0.0,1.0]*, *[0.0,1.0,0.0]*, [0.1,0.1,0.8], [0.1,0.8,0.1], [0.2,0.2,0.6], [0.2,0.6,0.2], *[0.4,0.4,0.2]*, [0.6,0.2,0.2], [0.8,0.1,0.1], *[0.33,0.33,0.33]* and *[1.0,0.0,0.0]*. The 5 preferences for MORLHF and MODPO are highlighted in italics.

the three-objective alignment task.

Collapse phenomenon of language output. As presented in Section 2.2, expressing preferences through linear scalarization can lead the LLM to learn conflicting patterns, potentially triggering language collapse. To further support this claim, we analyze the outputs of all baselines and ParetoHqD. We observe that the collapse consistently manifests in the outputs of all baselines through two phenomena: 1) phrase- or sentence-level repetition, and 2) extremely short responses (see Table 4 in Appendix). For example, repeated sequences include *'How are you? How are you? How are you?'*, while short responses include *'Sure, here it is:'*. To quantify these phenomena, we first apply regular expressions to extract short phrases (Perez et al. 2022; Holtzman et al. 2019), each consisting of up to four consecutive words. We then count the occurrences of each extracted phrase within a response. If any phrase appears more than three times, this response is flagged as collapsed. In addition, we also consider responses with fewer than five words as collapsed due to their insufficient content.

Method	CR (%) ↓	Time ↓	HV ↑
MORLHF	35.39	2272.84	0.3777
Rewarded Soups	31.29	923.68	0.3605
MOD	49.71	923.68	0.3545
MODPO	13.99	514.19	0.5074
RiC	41.88	132.47	0.6325
ParetoHqD	**7.03**	**55.87**	**0.7526**

Table 1: Comparison of collapse rate (CR), training GPU hours (Time), and hypervolume indicator (HV) on the Helpful Assistant task with two rewards.

The average collapse rates across N preferences are shown in Table 1, all algorithms except MODPO and ParetoHqD exhibit language collapses in more than 30% of their generated responses. These algorithms adopt the linear scalarization to learn conflicting patterns, thereby resulting in collapsed language generation. MODPO is trained using preference data from a single objective, while introducing marginal losses to incorporate auxiliary objectives. This design implicitly mitigates the negative effects of objective conflicts during optimization. However, due to the dominance of the primary objective in the training signal, MODPO struggles to achieve a well-covered Pareto front, as illustrated in Fig. 5 (b). In contrast, our fine-tuned LLMs demonstrate powerful text generation capabilities while achieving a best Pareto front. We present some responses generated by baselines and ParetoHqD in Appendix 5.

Computational cost and Hypervolume indicator. In Table 1, we show the computational cost of all algorithms and their performance on the hypervolume indicator. Specifically, our algorithm is efficient, using only 55.87 GPU hours (∼42.18% cost of the SOTA RiC baseline). Among them, the three main parts of the algorithm, Pareto filtering, SFT training, and data augmentation, account for ∼0.03%, ∼16.47%, and ∼83.50% of the total cost respectively. More importantly, as the number of human preferences N increases, our runtime can remain stable for the following reasons. The proposed data augmentation strategy ensures that regardless of how large N becomes, only $M+1$ augmented datasets need to be constructed. This design keeps the most computationally expensive component of the algorithm constant. Increasing N only leads to additional SFT executions, which are lightweight: each requiring only ∼0.558 GPU hours for k training samples, accounting for merely ∼1% of the total computational cost of ParetoHqD. The hypervolume indicator (Shang et al. 2021) is widely used as a performance indicator in the field of multiobjective optimization, which can simultaneously measure the convergence and diversity of a Pareto solution set. As shown in Table 1, the proposed ParetoHqD enjoys a strong performance.

Comparison with ParetoHqD Using Stage 1 Only. ParetoHqD tailors the training Stage 2 to alleviate the overfitting

caused by training with a small amount of data. We compare ParetoHqD with ParetoHqD w/ S1, which only goes through training Stage 1. As illustrated in Fig. 5 (d), ParetoHqD employs the training Stage 2 to improve the Pareto front obtained by ParetoHqD w/ S1.

4.3 Comparisons with Scalarized SFT

The main ideas of our algorithm are: (1) utilizing a small high-quality training set for SFT, and (2) employing Pareto filtering and preference directions to guide data selection. To gain further insights, we compare ParetoHqD w/ S1 with two linearly scalarized SFT algorithms. One is SFT, which is trained on top-$|\mathcal{D}|$ samples ranked by scalar values $\boldsymbol{\omega r}$. The other is ParetoHqD w/ ls-S1, which is trained on top-k samples ranked by scalar values $\boldsymbol{\omega r}$. First, we count their collapse rate of generated responses. As shown in Table 2, we can see that SFT results in nearly twice the collapse rate of the base model. This further supports the limitation of scalarization-based methods: learning conflicting patterns from diverse preference data in the training set can lead to frequent language collapse. ParetoHqD w/ ls-S1 significantly reduces the collapse rate compared to SFT by leveraging our idea (1): utilizing a small high-quality training set for SFT. This is reasonable, as fewer top-ranked samples may limit the model's exposure to conflicting preference data that often trigger language collapse. Compared to the two scalarized SFT approaches, our method represents preferences via preference directions, which helps mitigate the language collapse.

Method	Collapse Rate (%) ↓
Llama-2 7B	18.63
SFT	35.74
ParetoHqD w/ ls-S1	6.74
ParetoHqD w/ S1	**6.11**

Table 2: Collapse rate of different methods.

Then, we construct two scenarios to verify the diversity loss introduced by linear scalarization. The first uses the original HH-RLHF dataset, which yields a convex Pareto front, while the second employs our constructed HH-RLHF subdataset, designed to exhibit a concave Pareto front region. In Scenario 1, we first present the training points selected by the two methods in Fig. 6 (a), where the preference is set as [0.3,0.7]. It can be seen that the data selected by ParetoHqD w/ S1 are more capable of ensuring consistent preference than ParetoHqD w/ ls-S1, as analyzed in Section 2.2. The task results demonstrate that ParetoHqD w/ S1 with preference directions achieves superior diversity. The data selected based on a single preference in ParetoHqD w/ ls-S1 is relatively scattered, which can be attributed to the ambiguous preference signals induced by linear scalarization. This leads to substantial overlap in the training data across different preferences when the number of preferences N is large, resulting in similar performance across the corresponding LLMs. In Scenario 2, we present all the training points for two methods over N preferences in Fig. 6 (b). It can be observed that ParetoHqD w/ ls-S1 fails to obtain any training data from the

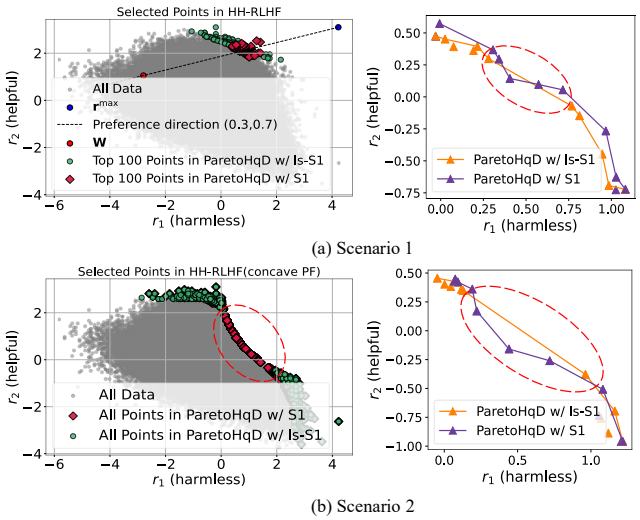


Figure 6: Comparison of selected training points and resulting Pareto fronts in two scenarios. In both cases, ParetoHqD w/ ls-S1 fails to cover certain regions of the front (highlighted by red ellipses).

concave region of the Pareto front, which naturally leads to its diversity collapse.

Overall, scalarized SFT can reduce its language collapse by adopting our idea (1). However, it consistently exhibits poor diversity on datasets with both convex and concave Pareto fronts, indicating its limited ability to personalize responses across diverse users. In contrast, applying our idea (2): adopting Pareto filtering and preference direction shows promising potential for customizing LLMs to better align with different user preferences.

5 Limitations and Future Work

Although extending ParetoHqD to a larger number of human preferences does not incur significant computational costs (see Section 4.2) and allows it to serve a wide range of user needs, extending it to a larger number of objectives may result in considerably higher computational overhead. For future work, we plan to develop lightweight adaptation strategies for ParetoHqD to efficiently incorporate more objectives.

6 Conclusions

In this work, we propose ParetoHqD to handle the multiobjective alignment task with preference values. ParetoHqD introduces the definition of Pareto optimality to formulate the multiobjective alignment problem, and represents a human preference as a preference direction in the objective space. Then, it operates through a two-stage SFT training process, where each stage adopts an individual Pareto high-quality training set that best matches the preference direction. The first stage speeds up the fine-tuning of LLMs, and the second stage alleviates the overfitting caused by small amounts of training data. Experimental results reveal the high performance and efficiency of the ParetoHqD algorithm, as well as the effectiveness of each component.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 62376202).

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bhatt, G.; Chen, Y.; Das, A. M.; Zhang, J.; Truong, S. T.; Mussmann, S.; Zhu, Y.; Bilmes, J.; Du, S. S.; Jamieson, K.; et al. 2024. An Experimental Design Framework for Label-Efficient Supervised Finetuning of Large Language Models. *arXiv preprint arXiv:2401.06692*.

Chen, R.; Zhang, X.; Luo, M.; Chai, W.; and Liu, Z. 2024. PAD: Personalized Alignment at Decoding-Time. *arXiv preprint arXiv:2410.04070*.

Dong, Y.; Wang, Z.; Sreedhar, M. N.; Wu, X.; and Kuchaiev, O. 2023. Steerlm: Attribute conditioned SFT as an (user-steerable) alternative to RLHF. *arXiv preprint arXiv:2310.05344*.

Fu, T.; Hou, Y.; McAuley, J.; and Yan, R. 2024. Unlocking Decoding-time Controllability: Gradient-Free Multi-Objective Alignment with Contrastive Prompts. *arXiv preprint arXiv:2408.05094*.

Gu, H.; Wang, H.; and Jin, Y. 2023. Effects of pareto set on the performance of problem reformulation-based large-scale multiobjective optimization algorithms. In *2023 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. IEEE.

Gu, H.; Wang, H.; Mei, Y.; Zhang, M.; and Jin, Y. 2025. One Trigger Token Is Enough: A Defense Strategy for Balancing Safety and Usability in Large Language Models. *arXiv preprint arXiv:2505.07167*.

Guo, Y.; Cui, G.; Yuan, L.; Ding, N.; Wang, J.; Chen, H.; Sun, B.; Xie, R.; Zhou, J.; Lin, Y.; et al. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Li, K.; Zhang, T.; and Wang, R. 2021. Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Transactions on Cybernetics*, 51(6): 3103–3114.

Liu, W.; Zeng, W.; He, K.; Jiang, Y.; and He, J. 2023a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Liu, Y.; Yao, Y.; Ton, J.-F.; Zhang, X.; Cheng, R. G. H.; Klochkov, Y.; Taufiq, M. F.; and Li, H. 2023b. Trustworthy LLMs: A survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*.

Mekala, D.; Nguyen, A.; and Shang, J. 2024. Smaller language models are capable of selecting instruction-tuning training data for larger language models. *arXiv preprint arXiv:2402.10430*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 27730–27744. Curran Associates, Inc.

Perez, E.; Huang, S.; Song, F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, 53728–53741. Curran Associates, Inc.

Rame, A.; Couairon, G.; Dancette, C.; Gaya, J.-B.; Shukor, M.; Soulier, L.; and Cord, M. 2023. Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, volume 36, 71095–71134. Curran Associates, Inc.

Shang, K.; Ishibuchi, H.; He, L.; and Pang, L. M. 2021. A Survey on the Hypervolume Indicator in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 25(1): 1–20.

Shi, R.; Chen, Y.; Hu, Y.; Liu, A.; Hajishirzi, H.; Smith, N. A.; and Du, S. S. 2024. Decoding-Time Language Model Alignment with Multiple Objectives. *Advances in Neural Information Processing Systems*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, 3008–3021. Curran Associates, Inc.

Thirunavukarasu, A. J.; Ting, D. S. J.; Elangovan, K.; Gutierrez, L.; Tan, T. F.; and Ting, D. S. W. 2023. Large language models in medicine. *Nature medicine*, 29(8): 1930–1940.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, H.; Lin, Y.; Xiong, W.; Yang, R.; Diao, S.; Qiu, S.; Zhao, H.; and Zhang, T. 2024. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*.

Wang, R.; Zhou, Z.; Ishibuchi, H.; Liao, T.; and Zhang, T. 2016. Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1): 3–18.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing: System Demonstrations, 38–45.

Wolf, Y.; Wies, N.; Avnery, O.; Levine, Y.; and Shashua, A. 2023. Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*.

Yang, K.; Liu, Z.; Xie, Q.; Huang, J.; Zhang, T.; and Ananiadou, S. 2024a. MetaAligner: Towards Generalizable Multi-Objective Alignment of Language Models. *Advances in Neural Information Processing Systems*.

Yang, R.; Pan, X.; Luo, F.; Qiu, S.; Zhong, H.; Yu, D.; and Chen, J. 2024b. Rewards-in-Context: Multi-objective Alignment of Foundation Models with Dynamic Preference Adjustment. *International Conference on Machine Learning*.

Yang, Y.; Zha, K.; Chen, Y.; Wang, H.; and Katabi, D. 2021. Delving into Deep Imbalanced Regression. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 11842–11851. PMLR.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhong, Y.; Ma, C.; Zhang, X.; Yang, Z.; Chen, H.; Zhang, Q.; Qi, S.; and Yang, Y. 2024. Panacea: Pareto alignment via preference adaptation for LLMs. *arXiv preprint arXiv:2402.02030*.

Zhou, A.; Zhang, Q.; and Jin, Y. 2009. Approximating the Set of Pareto-Optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5): 1167–1189.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; YU, L.; Zhang, S.; Ghosh, G.; Lewis, M.; Zettlemoyer, L.; and Levy, O. 2023. LIMA: Less Is More for Alignment. In *Advances in Neural Information Processing Systems*, volume 36, 55006–55021. Curran Associates, Inc.

Zhou, Z.; Liu, J.; Shao, J.; Yue, X.; Yang, C.; Ouyang, W.; and Qiao, Y. 2024. Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 10586–10613. Bangkok, Thailand: Association for Computational Linguistics.