Hybrid Alignment Training for Large Language Models

Chenglong Wang[1], **Hang Zhou**[1], **Kaiyan Chang**[1], **Bei Li**[1], **Yongyu Mu**[1],
Tong Xiao[1,3]*, **Tongran Liu**[2], **and Jingbo Zhu**[1,3]
[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2] CAS Key Laboratory of Behavioral Science, Institute of Psychology, CAS, Beijing, China
[3] NiuTrans Research, Shenyang, China
{clwang1119, ctrl.hang}@gmail.com,
{xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

Alignment training is crucial for enabling large language models (LLMs) to cater to human intentions and preferences. It is typically performed based on two stages with different objectives: instruction-following alignment and human-preference alignment. However, aligning LLMs with these objectives in sequence suffers from an inherent problem: the objectives may conflict, and the LLMs cannot guarantee to simultaneously align with the instructions and human preferences well. To response to these, in this work, we propose a **Hyb**rid **A**lignment **T**raining (HBAT) approach, based on alternating alignment and modified elastic weight consolidation methods. The basic idea is to alternate between different objectives during alignment training, so that better collaboration can be achieved between the two alignment tasks. We experiment with HBAT on summarization and dialogue tasks. Experimental results show that the proposed HBAT can significantly outperform all baselines. Notably, HBAT yields consistent performance gains over the traditional two-stage alignment training when using both proximal policy optimization and direct preference optimization.

1 Introduction

Alignment training is a key technique to ensure that the behaviors of large language models (LLMs) are consistent with human intentions and preferences (Ouyang et al., 2022; Wang et al., 2023e). It typically involves two stages: 1) using human-labeled data to train pre-trained LLMs via a supervised training method, which enables LLMs to understand human intentions and follow the instructions (call it *instruction-following alignment*), and 2) employing approaches like proximal policy optimization (PPO) (Schulman et al., 2017) and direct preference optimization (DPO) (Rafailov et al., 2023) to learn preferences from human feedbacks (call it *human-preference alignment*). This paradigm has achieved promising results on several downstream tasks, such as dialogue (OpenAI, 2022; Dubois et al., 2023; Wang et al., 2023b), summarization (Stiennon et al., 2020; Lee et al., 2023), and machine translation (Ramos et al., 2023).

However, this two-stage alignment training has its inherited limitation: the optimization objectives are different for each stage, which can make an optimization conflict (French, 1999; Liu et al., 2021). Such limitation could result in an inferiorly aligned LLM in real-world scenarios. This phenomenon is also described in Ouyang et al. (2022)'s work, which is referred to as alignment tax.

To mitigate this limitation, in this work, we propose a **Hyb**rid **A**lignment **T**raining (HBAT) approach, which offers a refinement of the collaboration among instruction-following alignment and human-preference alignment by using the following two methods. For one, inspired by interactive methods in multi-objective optimization (Miettinen et al., 2008; Xin et al., 2018), we propose an alternating alignment method, where the human-preference alignment acts as a decision maker and continuously interacts with the instruction-following alignment to achieve a preferred alignment. Specifically, we divide the instruction-following and human-preference training set into equal portions of mutually exclusive subsets, respectively. Then, we rearrange these subsets in alternating orders during alignment training. Furthermore, we introduce a modified Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) to alternating alignment. EWC is a method to dynamically impose an appropriate constraint on each parameter when training a model with a new optimization objective, thereby easing an optimization conflict with the previous objective.

We experiment with the proposed HBAT on summarization and dialogue tasks based on LLaMA2-7B and LLaMA2-13B models (Touvron et al.,

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 11389–11403

2023). Experimental results show that HBAT can significantly surpass all baselines. Notably, based on the LLaMA2-13B model, HBAT can yield a +2.26 ROUGE-L points improvement for the summarization task, compared to the traditional RLHF. Additionally, our HBAT significantly outperforms the SFT over 21.01 GPT-4 win rate points on the dialogue task based on the LLaMA2-13B model. Furthermore, HBAT is orthogonal to other optimized alignment approaches. For instance, when armed with ESRL (Wang et al., 2023b), our HBAT gains an additional improvement of 2.59 GPT-4 win rate points on the summarization task.

2 Related Work

Alignment Training for LLMs. Recently, many efforts have been made to improve the LLM alignment for different tasks (Stiennon et al., 2020; Nakano et al., 2021; Wang et al., 2023c; Hu et al., 2023). These works mainly focused on optimizing each stage of alignment training, including instruction-following alignment (also referred to as SFT) and human-preference alignment (also referred to as RLHF). For example, Zhou et al. (2023) designed data selection schemes to provide high-quality instruction-following data. Moreover, Wang et al. (2022) proposed an efficient approach for producing instruction-following data. Likewise, some works aimed to efficiently produce human-preference data (Lee et al., 2023; Dubois et al., 2023; Wang et al., 2023a). Apart from the training data improvements, another line of improving the alignment training is to explore better reward models and optimization objectives, such as the use of fine-grained reward models (Coste et al., 2023; Wu et al., 2023), the integration of a prior knowledge in training reward models (Zhou et al., 2024), and the design of direct preference optimization objective (Rafailov et al., 2023). Although previous works improve the performance of instruction-following alignment and human-preference alignment, they rarely consider the optimization conflict limitation between them. Researchers have been aware of this (Ouyang et al., 2022), but it is still rare to see studies on this issue.

Multi-objective Optimization. Multi-objective optimization problem involves optimizing multiple optimization objectives simultaneously (Hwang and Masud, 2012). However, there does not typically exist a feasible solution that minimizes all loss functions. Therefore, researchers always explored a Pareto optimal solution that cannot be improved in any of the objectives without impairing at least one of the other objectives. Recent works on this exploration could be classified into three groups. The first group focused on Pareto dominance-based method. This method maintains the individual elements of the solution vectors as independent during optimization (Cheng et al., 2015; Wu and Pan, 2019). The second group tended to design an quality indicator, such as hypervolume (Bader and Zitzler, 2011) and R2 (Wagner et al., 2013), to act as a proxy objective instead of optimization objectives. The third group that has attracted attention commonly aimed to solve multi-objective optimization problems through an interactive method. A typical interactive method requires a decision maker to offer preference information, which allows to search for the most preferred Pareto optimal solution after each optimization (Xin et al., 2018; Misitano et al., 2021; Pereira et al., 2022).

Although the alignment training is not a standard multi-objective optimization problem, its goal remains consistent, *i.e.,* seeking an aligned LLM that simultaneously aligns instructions and human preferences well.

3 Background

Despite the extensive knowledge endowed from pre-training, LLMs are difficult to produce content that humans want. This is because that pre-trained LLMs lack understanding of input instructions and human preferences. To address this, we often perform alignment training on them, first for instruction-following alignment and then for human-preference alignment.

3.1 Instruction-Following Alignment

Instruction-following alignment enables the pre-trained language model to acquire the capability to understand and follow instructions in the prompt by mimicking the human-labeled response. Specifically, given a human prompt x and the labeled response of N_y tokens $y = \{y_1, \ldots, y_{N_y}\}$, where each token y_t is drawn from a vocabulary. In the training process, the LLM learns the probability:

$$p_\theta(y|x) = \prod_{t=1}^{N_y} p_\theta(y_t|y_{<t}, x) \quad (1)$$

where $y_{<t}$ is the prefix $\{y_1, y_2, \ldots, y_{t-1}\}$, and θ is a trained parameter set. The standard training

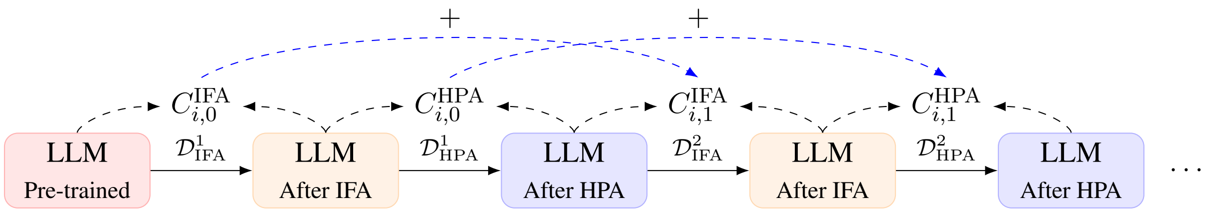


Figure 1: Architecture of HBAT. We introduce the alternating alignment and the modified EWC methods to design HBAT, which enables it to address optimization conflict problem in the process of LLM alignment training. Here, black solid arrows (⟶) denote learning from the subsets $\mathcal{D}^n_{\text{IFA}}$ and $\mathcal{D}^n_{\text{HPA}}$ via Eq. 8 and Eq. 5, respectively. Black dashed arrows (- - ➝) denote computing the amount of parameter changes before and after training and blue dashed arrows (- - ➝) denote accumulating the parameter changes resulting from learning all previous subsets (see Section 4.1). **IFA**: instruction-following alignment; **HPA**: human-preference alignment.

objective is to maximize the likelihood over all the tokens of the labeled response, *i.e., maximum likelihood estimation (MLE)* (Myung, 2003). The corresponding loss function can be defined by:

$$\mathcal{L}_{\text{MLE}} = -\sum_{t} \log p_\theta(y_t|y_{<t}, x) \tag{2}$$

3.2 Human-Preference Alignment

This process of human-preference alignment consists of two main steps: 1) learning a preference model from comparison response pairs to act as a reward model, and 2) maximizing the reward, written as $\arg\max_\theta \mathbb{E}p_\theta(\hat{y}|x)[r(\hat{y})]$, where $\hat{y}$ is a generated response and $r(\cdot)$ denotes the computation of the reward for $\hat{y}$ using a reward model. We usually employ an RL algorithm to achieve step 2. Taking PPO as an instance, the corresponding loss for this training sample is given by:

$$\begin{aligned}\mathcal{L}_{\text{PPO}} = -\sum_{\hat{y}\in\Omega(x)} & \log p_\theta(\hat{y}|x) r(\hat{y}) \\ & - \alpha \log\left(\frac{p_\theta(\hat{y}|x)}{p_{\theta_{old}}(\hat{y}|x)}\right)\end{aligned} \tag{3}$$

where $\Omega(x)$ is the output space which comprises all possible responses for prompt x, θ_{old} is the parameter set of the LLM trained via instruction-following alignment, and α is a KL reward coefficient which controls the strength of the KL penalty $\log(\frac{p_\theta(\hat{y}|x)}{p_{\theta_{old}}(\hat{y}|x)})$. Here, $\Omega(x)$ is approximated using the Monte Carlo method (Williams, 1992).

To bypass the complex RL procedure, Rafailov et al. (2023) proposed DPO method, which employs a reward model training objective to maximize rewards. It gives a new loss function:

$$\begin{aligned}\mathcal{L}_{\text{DPO}} = -\log\sigma[\beta & \log\left(\frac{p_\theta(y_w|x)}{p_{\theta_{old}}(y_w|x)}\right) \\ & - \beta\log\left(\frac{p_\theta(y_l|x)}{p_{\theta_{old}}(y_l|x)}\right)]\end{aligned} \tag{4}$$

where (y_w, y_l) is two of the different responses and y_w aligns better with human preferences than y_l. β is a scaling factor and σ is a Sigmoid function.

4 Method

In this work, we aim to solve an optimization conflict limitation during alignment training. We propose the HBAT to achieve this. The overview of HBAT is depicted in Figure 1. As shown in the figure, we propose the alternating alignment and modified EWC in HBAT to achieve our goal. In the following subsections, we will describe them.

4.1 Alternating Alignment

We first introduce the optimization conflict problem in the alignment training. Suppose that we have training datasets $\mathcal{D}_{\text{IFA}}$ and $\mathcal{D}_{\text{HPA}}$ for instruction-following alignment and human-preference alignment, respectively. We expect that the LLM will simultaneously align instructions and human preferences well by learning from both datasets. However, during the traditional two-stage alignment training, while the LLM learns from new training samples in $\mathcal{D}_{\text{HPA}}$, it may have conflicts with previous knowledge learned from $\mathcal{D}_{\text{IFA}}$.

Inspired by the success of interactive methods in multi-objective optimization, we propose an alternating alignment method. In the alternating alignment, we redesign the relationship between the instruction-following alignment and human-preference alignment to offer a refinement of the collaboration among them. Specifically, we divide the datasets $\mathcal{D}_{\text{IFA}}$ and $\mathcal{D}_{\text{HPA}}$ into N mutually exclusive splits $\{\mathcal{D}^1_{\text{IFA}}, \mathcal{D}^2_{\text{IFA}}, \cdots, \mathcal{D}^N_{\text{IFA}}\}$ and $\{\mathcal{D}^1_{\text{HPA}}, \mathcal{D}^2_{\text{HPA}}, \cdots, \mathcal{D}^N_{\text{HPA}}\}$, respectively. The LLM performs an alternating alignment by sequentially learning from $\{\mathcal{D}^1_{\text{IFA}}, \mathcal{D}^1_{\text{HPA}}, \cdots, \mathcal{D}^N_{\text{HPA}}\}$. In each round of alternate training, the human-preference alignment acts as a "decision maker" to offer preference information. This preference information enables an LLM to align human preferences following instruction alignment.

4.2 Elastic Weight Consolidation

To further solve the optimization conflict, we introduce a modified EWC to alternating alignment. Firstly, we add EWC to the process of human-preference alignment to mitigate optimization conflicts with instruction-following alignment. The loss of human-preference alignment with EWC is:

$$\mathcal{L}_{\text{HPA}} = \mathcal{L}_{\text{PPO}} + \sum_i \frac{\lambda}{2} F_i^{\text{IFA}} (\theta_i - \theta_i^{\text{IFA}})^2 \quad (5)$$

where i is the index corresponding to each parameter within the LLM, θ^{IFA} is the parameter set of the LLM trained by instruction-following alignment, λ is a balance factor, and F is the diagonal of the empirical Fisher matrix (Pascanu and Bengio, 2014). Here, F_i^{IFA} denotes how important the i-th parameter θ_i^{IFA} is to the instruction-following alignment. Note that we can replace $\mathcal{L}_{\text{PPO}}$ with other loss functions, such as $\mathcal{L}_{\text{DPO}}$, which can align LLMs with human preferences.

Modified EWC for LLMs. However, the original EWC introduces a large computational overhead on the alignment training. This is because estimating F_i^{IFA} requires the LLM to be additionally trained multiple times on the whole training set (see Appendix B). To mitigate this problem, we redesign this estimation approach, and use the amount of parameter changes before and after model training to compute the F. Furthermore, considering that LLMs typically have a large number of parameters and the size of the F will be enormous, we attempt to implement EWC at the granularity of parameter units. Specifically, we redefine F as a numerical value, with F_i^{IFA} representing how importance of the parameter unit θ_i^{IFA} as a whole to the instruction-following alignment. This redefined F can be given by:

$$F_i^{\text{IFA}} = F_{max} \times \frac{e^{C_i^{\text{IFA}}}}{\sum_i e^{C_i^{\text{IFA}}}} \quad (6)$$

where F_{max} is the maximum value of F. C_i^{IFA} denotes the amount of parameter θ_i changes before and after instruction-following alignment training for the LLM, written as:

$$C_i^{\text{IFA}} = \frac{1}{|\theta_i|} \sum_{j=1}^{|\theta_i|} (\theta_{i,j}^{before} - \theta_{i,j}^{\text{IFA}})^2 \quad (7)$$

Algorithm 1 Hybrid Alignment Training

Input: the pre-trained LLM $\mathcal{M}$; the instruction-following alignment training dataset $\mathcal{D}_{\text{IFA}}$; the human-preference alignment training dataset $\mathcal{D}_{\text{HPA}}$
Output: the aligned LLM $\mathcal{M}$;

```
divide D_IFA and D_HPA into N subsets respectively;
for n = 1 to N do
    if n==1 then
        train M on first subset of D_IFA via Eq. 2;
    else
        compute the F^HPA via Eq. 9;
        train M on n-th subset of D_IFA via Eq. 8;
    end if
    compute the F^IFA via Eq. 6;
    train M on n-th subset of D_HPA via Eq. 5;
end for
return M
```

---

where j is the index corresponding to each neuron within a parameter, $|\theta_i|$ is the number of neurons contained in the parameter θ_i, and θ^{before} is the parameter set of the LLM before instruction-following alignment training.

4.3 EWC for Alternating Alignment

We apply EWC on a global scale during alternating alignment. Specifically, we add the modified EWC not only when learning each divided subset from $\mathcal{D}_{\text{HPA}}$ as described in Section 4.2, but also when learning each divided subset from $\mathcal{D}_{\text{IFA}}$. The motivation is that the instruction-following alignment can likewise lead to an optimization conflict with human-preference alignment. $\mathcal{L}_{\text{IFA}}$ can be induced by:

$$\mathcal{L}_{\text{IFA}} = \mathcal{L}_{\text{MLE}} + \sum_i \frac{\lambda}{2} F_i^{\text{HPA}} (\theta_i - \theta_i^{\text{HPA}})^2 \quad (8)$$

where θ^{HPA} is the parameters of the LLM trained by human-preference alignment. Here, similar to F_i^{IFA}, F_i^{HPA} can be computed by:

$$F_i^{\text{HPA}} = F_{max} \times \frac{e^{C_i^{\text{HPA}}}}{\sum_i e^{C_i^{\text{HPA}}}} \quad (9)$$

where C_i^{HPA} denotes the amount of parameter θ_i changes before and after human-preference alignment training for the LLM. It can be computed via Eq. 7. Note that when learning the first subset $\mathcal{D}_{\text{IFA}}^1$, since the LLM has not yet been trained with human preferences, we only employ the $\mathcal{L}_{\text{MLE}}$.

In the process of alternating alignment training, learning a new subset from one alignment training dataset can produce optimization conflicts. These conflicts arise not only with the closest subset from another alignment training dataset but also with all the previous subsets within this dataset. Thus, when estimating F, we consider the parameter changes resulting from all previous subsets in another alignment training dataset. To this end, we replace the C_i^{IFA} and C_i^{HPA} in Eqs. 8 and 5 with accumulated parameter changes AC_i^{IFA} and AC_i^{HPA} from all previous subsets in $\mathcal{D}_{\text{IFA}}$ and $\mathcal{D}_{\text{HPA}}$, respectively. Here, when learning from n-th subset, we compute $AC_{i,n}^{\text{IFA}}$ and $AC_{i,n}^{\text{HPA}}$ by:

$$AC_{i,n}^{\text{IFA}} = \sum_{k=1}^{n} C_{i,k}^{\text{IFA}}, AC_{i,n}^{\text{HPA}} = \sum_{k=1}^{n} C_{i,k}^{\text{HPA}} \quad (10)$$

where $C_{i,k}^{\text{IFA}}$ and $C_{i,k}^{\text{HPA}}$ are the amount of parameter changes produced at learning k-th subset in $\mathcal{D}_{\text{IFA}}$ and $\mathcal{D}_{\text{HPA}}$, respectively. The process of our HBAT is also described in Algorithm 1.

5 Experimental Setup

We evaluated HBAT on summarization and dialogue tasks based on the commonly used LLaMA2-7B and LLaMA2-13B models.

5.1 Datasets

The datasets used for each task are as follows:

Summarization. We used the same dataset as Stiennon et al. (2020), which is a filtered version[*] of the TL;DR dataset (Völske et al., 2017). The filtered training set consists of 120k Reddit posts with accompanying summaries. For instruction-following training and human-preference alignment training, we used all posts in a filtered training set, respectively. The filtered test set and validation set contain 6,553 posts and 6,447 posts respectively, which would result in a huge computational cost when used on a large scale. Thus, we randomly selected 10% of posts from them as a test set and a validation set in our experiments, respectively. For training reward models, we employed the open-source 92.9k summary comparisons[†].

Dialogue. We conducted experiments on the Alpaca data (Taori et al., 2023a) which contains 52k training samples. Here, we employed the sliced data splits[‡] released by AlpacaFarm (Dubois et al., 2023) to conduct instruction-following alignment training, reward model training, and human-preference alignment training. Note that we used the human preferences rather than the simulated preferences to train our reward models. In the evaluation, we employed the AlpacaFarm evaluation set which consists of 805 instructions. We randomly selected 200 instructions from them as our validation set and the rest as our test set.

5.2 Settings

We trained reward models with the ranking loss for all tasks, following Stiennon et al. (2020). For instruction-following alignment training, we employed the cross-entropy loss on batches of prompts concatenated with responses, computing the loss only on the response tokens. For human-preference alignment training, we used PPO and DPO as our base algorithms. For HBAT, we set the number of dataset splits to 2 and 10 for dialogue and summarization tasks, respectively. Additionally, we employed a top-p sampling strategy for generation, where the temperature and p were set to 0.75 and 0.95, respectively, values that are commonly used in real-world applications. We publicly release all our code used for the experiments described in this work[§]. More training details are shown in Appendix A.

5.3 Evaluation Metrics

For the summarization task, we measured the summary quality by computing ROUGE (Lin, 2004) and BARTScore (Yuan et al., 2021), respectively. For the dialogue task, we measured the response quality with PandaLM (Wang et al., 2023d) which can distinguish the superior model from some LLMs. To further evaluate the performance of the model, we employed GPT-4 as a proxy for human evaluation of summary and response quality in the dialogue and summarization tasks, where the used evaluation prompts were the same as in Rafailov et al. (2023). We used reference summaries and responses in the test set as the baseline. Additionally, following Stiennon et al. (2020)'s work, we evaluated the model by computing the reward scores of test sets via our reward models.

[*] https://github.com/openai/summarize-from-feedback

[†] https://huggingface.co/datasets/openai/summarize_from_feedback

[‡] https://huggingface.co/datasets/tatsu-lab/alpaca_farm

[§] https://github.com/wangclnlp/DeepSpeed-Chat-Extension/tree/main/examples/hybrid_alignment_training

Method	#Param	PPO	DPO	Summarization				Dialogue		
				ROUGE-L	BS	Reward	Win	PandaLM	Reward	Win
Based on LLaMA2-7B Model										
SFT	7B			22.60	-5.46	3.72	53.20	54.76	-6.79	43.49
RLHF	7B	✓		25.85	-4.27	4.43	63.80	69.79	-5.81	55.63
RLHF+pt	7B	✓		22.25	-5.64	3.74	56.26	53.52	-7.09	54.18
SFT+ppo	7B	✓		13.75	-5.78	2.40	18.91	45.32	-8.60	42.25
HBAT-Freeze	7B	✓		25.33	-4.28	5.26	64.79	69.91	-5.91	56.19
HBAT (Ours)	7B	✓		**26.18**	**-3.82**	**5.74**	**72.52**	**70.88**	**-5.37**	**57.12**
DPO	7B		✓	22.96	-5.13	4.27	61.37	70.74	-5.72	54.23
HBAT-Freeze	7B		✓	23.01	-5.05	4.45	64.18	68.78	-5.41	56.95
HBAT (Ours)	7B		✓	**23.14**	**-4.18**	**4.95**	**70.58**	**74.78**	**-5.22**	**58.10**
Based on LLaMA2-13B Model										
SFT	13B			23.27	-5.12	4.01	57.91	62.16	-6.32	46.11
RLHF	13B	✓		24.51	-3.96	5.55	71.67	72.21	-5.65	61.16
RLHF+pt	13B	✓		22.92	-5.49	3.97	64.42	63.67	-6.97	54.45
SFT+ppo	13B	✓		13.84	-5.97	2.53	28.97	54.00	-7.93	43.12
HBAT-Freeze	13B	✓		25.80	-3.63	6.18	77.22	71.31	-5.49	56.37
HBAT (Ours)	13B	✓		**26.77**	**-3.51**	**6.41**	**78.81**	**72.83**	**-5.11**	**62.32**
DPO	13B		✓	23.02	-5.39	4.55	69.40	75.00	-5.07	64.31
HBAT-Freeze	13B		✓	23.10	-5.08	4.85	71.44	76.87	-5.01	65.62
HBAT (Ours)	13B		✓	**24.12**	**-4.05**	**5.40**	**74.92**	**77.79**	**-4.78**	**67.45**

Table 1: Results on summarization and dialogue tasks. The best results for each group are in **bold**. The "BS" and "Win" columns report the BARTScore and the win rate as assessed by GPT-4, respectively. The "PPO" and "DPO" columns denote that we employ PPO and DPO during human-preference alignment training, respectively.

5.4 Baselines

Our baselines are the standard two-stage alignment training (referred to as **RLHF/DPO**) and the commonly used instruction-following alignment training (referred to as **SFT**). Furthermore, we compared the proposed HBAT with commonly used multi-objective optimization methods, including adding a pre-training loss in the human-preference alignment training (**RLHF+pt**) (Ouyang et al., 2022) and adding a human-preference alignment loss in the instruction-following alignment training (**SFT+ppo**) (Wang et al., 2023a). To evaluate the effectiveness of EWC, we also chose the **HBAT-Freeze** method as a baseline, where we directly froze important parameters instead of EWC.

5.5 Experimental Results

Table 1 displays the experimental results on summarization and dialogue tasks.

Results of Summarization. First, compared with the traditional two-stage alignment training and instruction-following alignment training, the proposed HBAT can achieve optimal results on both of LLaMA2-7B and LLaMA2-13B. Notably, HBAT outperforms RLHF by 7.14 points on the GPT-4 win rate when using PPO on the LLaMA2-13B model. Second, compared with multi-task learning-based methods, including RLHF+pt and SFT+ppo, we can see that HBAT has significant improvements on all evaluation metrics. For instance, compared to RLHF+pt, HBAT yields a +3.93 ROUGE-L improvement on the LLaMA2-7B model. Also, we see that the multi-objective optimization method can hurt alignment, *e.g.,* RLHF+pt loses 0.69 Reward points on the LLaMA2-7B model. The phenomenon aligns with the observation reported in Ouyang et al. (2022)'s work. One potential explanation can be that while these multi-objective optimization methods achieve optimization of these objectives simultaneously, they still suffer from serious optimization conflict (Zhang and Yang, 2021). Third, when using DPO during human-preference alignment training, our HBAT is consistently better than all baselines. For a LLaMA2-13B model, it obtains a GPT-4 win rate of 74.92.

Results of Dialogue. We also evaluate the proposed HBAT on the dialogue task. Similarly, when using PPO during human-preference alignment

Method	PPO	DPO	Summarization				Dialogue				
			Coherence	Accuracy	Coverage	Overall	Fluency	Accuracy	Toxicity	Helpfulness	Overall
SFT			5.63	4.91	5.03	5.13	8.84	7.77	8.49	7.43	7.31
RLHF	✓		5.84	4.63	4.82	5.16	8.39	7.62	**8.87**	**7.47**	7.72
HBAT	✓		**6.23**	**4.83**	**5.77**	**5.69**	**8.80**	**7.70**	8.48	7.39	**7.89**
DPO		✓	5.83	4.45	5.20	5.27	8.67	7.13	**8.54**	**7.63**	7.84
HBAT		✓	**5.93**	**5.01**	**5.40**	**5.49**	**8.79**	**7.80**	8.45	7.51	**7.96**

Table 2: The results of human evaluation on the LLaMA2-13B model for our HBAT and baselines.

training, we can observe that HBAT outperforms RLHF by a large margin (*e.g.,* 2.21 PandaLM and 0.54 Reward benefits on the LLaMA2-13B model). However, different from the summarization task, we find that DPO can achieve better performance than PPO on the dialogue task. For instance, when using LLaMA2-13B, HBAT with DPO can outperform PPO by a margin of 5.13 points on the GPT-4 win rate. We assume that this is attributed to the reward model quality. To verify this assumption, we conduct tests on the employed reward models and find a significant difference in accuracy between the two tasks: the accuracy of the reward model for the summarization task significantly exceeds that of the dialogue task, achieving 0.75 compared to 0.65, respectively.

Furthermore, compared with HBAT-Freeze, we see that HBAT achieves better performance on all tasks. It demonstrates that freezing specific parameters is inferior to constraining specific parameters. We attribute this to the fact that the freezing operation reduces the amount of learnable parameters, which imposes a hurdle to learning new knowledge.

5.6 Human Evaluation

We further conduct a human evaluation of the obtained results through comprehensive evaluation aspects. For the summarization task, following Stiennon et al. (2020), we consider four evaluation aspects, including coherence, accuracy, coverage, and overall score. We provide three optional scores of 1, 4, and 7 for each evaluation aspect. Similarly, for the dialogue task, we consider five evaluation aspects: fluency, accuracy, toxicity, helpfulness, and overall score. We have defined detailed evaluation rubrics similar to those for the summarization task. Please refer to Table 8 in the Appendix for descriptions of all the evaluation rubrics. The results of human evaluation on the LLaMA2-13B model are shown in Table 2. From these evaluation results, we see that whether using PPO or DPO, our HBAT can achieve optimal results compared to the baselines in terms of the overall score of human evaluation. This demonstrates that HBAT has a great advantage in practical NLP applications. Furthermore, we also see that in some evaluation aspects, our HBAT is weaker compared to the baseline, *e.g.*, helpfulness aspect on the dialogue task. This is due to the fact that the aim of our HBAT is to achieve a Pareto alignment between the instruction-following alignment and the human-preference alignment, not to improve for a particular alignment.

Table 3 presents several cases for human evaluation. Case 1 shows that RLHF (*i.e.*, human-preference alignment) won't always improve the performance of LLM trained by SFT (*i.e.*, instruction-following alignment). This demonstrates that these two alignment optimization objectives are different, and aligning LLMs with these objectives in sequence might cause an optimization conflict. For example, in this case, SFT is probably more concerned with instruction following and response accuracy, while RLHF is more concerned with response helpfulness. In Case 2, we can observe that although the overall score of RLHF has increased, the fluency and accuracy scores have decreased, compared to SFT. In this case, our HBAT achieves a Pareto alignment through iterative alignment and modified elastic weight consolidation approaches, which aim to find a relatively optimal trade-off between instruction-following and human-preference alignment, thus achieving a better performance.

5.7 Ablation Studies

In this section, we present detailed ablation studies to explore the effects of EWC and alternating alignment with DPO on the LLaMA2-7B model. The experiments are conducted on the dialogue dataset, and the impacts of removing each method are thoroughly examined. The results are summarized in Table 4. From the results, we see that the modified EWC can significantly improve response quality. Notably, HBAT obtains a +5.35 points im-

Case 1	
Input	Please write the Excel function name associated with each description.- Returns the number of days between two dates - Returns the starting position of a text string within another text string. - Returns the number in the middle of the set of given numbers.
SFT	1. DAYS(date1, date2) 2. FIND(text1, text2) 3. MID(text1, start_num, num). **Fluency:7 Accuracy:6 Toxicity:10 Helpful:5 Overall:6**
RLHF	1.=DATEDIF(A2,B2,"d") 2.=FIND(A2,B2) 3.=MID(A2:B2,FIND(A2:B2,A2)+1,1) 4.=RANDBETWEEN(A2,B2) 5.=SUMIF(A2:B2,">="&A2,A2:B2) 6.=SUMPRODUCT(A2:B2) 7.=SUM(A2:B2) 8.=AVERAGE(A2:B2) 9.=AVERAGEIF(A2:B2,"<"&A2,A2:B2) 10.=COUNTIF(A2:B2,">"&A2) 11.=COUNTIFS(A2:B2,"<"&A2,A2:B2,">="&B2,A2:B2,"<="&B2) 12.=IFERROR(VLOOKUP(A2,A2:B2,2,0),"")... **Fluency:5 Accuracy:4 Toxicity:10 Helpful:6 Overall:4**
HBAT	1.DATEDIF(\em{date1},} date2\em{,} \em{"day" \em{) \n2. FIND(\em{text1},} text2\em{,} \em{} \em{} \em{} \em{} \n3. MID(\em{text},} start\em{,} numchars\em{)} **Fluency:7 Accuracy:6 Toxicity:10 Helpful:6 Overall:8**
Case 2	
Input	I assume you are familiar with the Drake equation. Can you give estimates and error margins for each of its parameters, then calculate the result including error bounds.
SFT	The number of stars in the galaxy is estimated to be between 10^{22} and 10^{24}, with a mid-range value of 10^{23}. \n2. The fraction of stars that have planets is estimated to be between 0.1 and 0.2, with a mid-range value of 0.15. \n3. The average number of planets that can support life per star is estimated to be between 1 and 2, with a mid-range value of 1.5. \n4. The fraction of planets where life has evolved is estimated to be between 0.01 and 0.1, with a mid-range value of 0.05. \n5. The fraction of planets where life has evolved that develops intelligent life is estimated to be between 0.01 and 0.05, with a mid-range value of 0.03... **Fluency:9 Accuracy:6 Toxicity:8 Helpful:6 Overall:5**
RLHF	1. Number of civilizations in our Galaxy: The number of civilizations in our Galaxy can be estimated using the Drake equation. The equation states that the number of civilizations in the Galaxy (N) is equal to the product of the rate of star formation (R*) multiplied by the fraction of stars with planets (fP) multiplied by the fraction of planets that can support life (fL) multiplied by the fraction of planets with life (fL) multiplied by the fraction of planets with intelligent life (fI) multiplied by the fraction of intelligent civilizations that are capable of communicating with us (fC) multiplied by the fraction of those that are willing to communicate (fW): \n\nN = R* fP fL fI fC fW\n \nThe rate of star formation (R*) can be estimated to be 1-2 stars per year in our Galaxy... **Fluency:5 Accuracy:5 Toxicity:8 Helpful:8 Overall:7**
HBAT	1. The number of stars with planetary systems (N*): Unknown, but estimated to be between 10^{-5} and 10^{-8} per star. Error margin: 10 orders of magnitude. \n2. The fraction of stars with planetary systems that could support life (fp): Unknown, but estimated to be between 10^{-5} and 0.5. Error margin: 10 orders of magnitude. \n3. The average number of planets that can support life per star with a planetary system (ne): Unknown, but estimated to be between 1 and 10. Error margin: 1 order of magnitude. \n4. The fraction of planets that could support life that actually develop life (fl): Unknown, but estimated to be between 0.01 and 0.99. Error margin: 1 order of magnitude. \n5. The ... **Fluency:7 Accuracy:6 Toxicity:8 Helpful:8 Overall:8**

Table 3: Several cases from the dialogue task on the LLaMA2-13B model.

Method	PandaLM	Reward	Win
SFT	43.64	-6.80	43.08
DPO	69.97	-5.68	53.80
HBAT	**75.76**	**-5.11**	**60.10**
w/o EWC	67.53	-5.76	54.75
w/o Alternating Alignment	70.50	-5.26	56.92

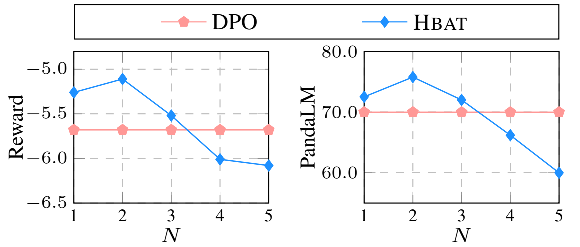
Table 4: Ablation studies on the components of HBAT. We report the scores for the dialogue validation set.

provement on GPT-4 win rate with the modified EWC. Additionally, the results indicate a significant dependency of our HBAT on the alternating alignment. The absence of this method results in HBAT fails a well-performed dialogue model.

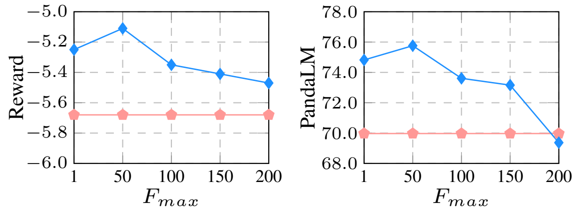
5.8 Analysis

Effect of the Number of Dataset Splits. Based on the LLaMA2-7B model, we investigate the impact of dividing the dataset into different numbers of splits. As shown in Figure 2 (top), we swept over different numbers: $\{1, 2, 3, 4, 5\}$. From the results, we find that excessive dataset splits can hurt the performance of the aligned LLM. We conjecture the underlying reason is that when datasets are heavily divided, each subset does not have sufficient samples for training.

Effect of F_{max} on Performance. The maximum value of F, F_{max}, is a key factor that controls the strength of parameter constraints. We conduct experiments to study the impact of setting different values of F_{max}: $\{1, 50, 100, 150, 200\}$. The corresponding Reward and PandaLM scores are listed in Figure 2 (bottom). From the results, we see that the use of different values of F_{max} can result in different performance gains. We find that the optimal F_{max} is 50, and this setting allows for appropriate control over parameter constraints. We conduct similar experiments to determine the optimal values for N and F_{max} for the summarization task,



(a) Performance of HBAT with different N



(b) Performance of HBAT with different F_{max}

Figure 2: Performance of HBAT with different number of dataset splits (*i.e.,* N) and the maximum values of F (*i.e.,* F_{max}) on the dialogue validation set.

which are found to be 10 and 50 respectively.

Performance on Different Temperature Settings. In real-world applications, various temperature settings are employed in the process of LLM generation according to specific scenarios. To this end, we compute the PandaLM scores under different temperature settings on the dialogue task to provide a comprehensive evaluation. The results are shown in Figure 3. From the results, we can observe that HBAT exceeds DPO's best-case performance on the dialogue task while being more robust to changes in the temperature setting.

See more analysis in Appendix B.

6 Conclusion

In this paper, we focus on solving the optimization conflict of alignment training in LLMs. We have proposed a hybrid alignment training (HBAT) via the alternating alignment and modified elastic weight consolidation methods. Our extensive experiments show that our HBAT can significantly outperform all baselines.

7 Limitations

In this section, we discuss some limitations of this work as follows:

- *We did not verify* HBAT *in other NLP tasks.* There are so many NLP tasks that we cannot verify our HBAT one by one. Thus, we take summarization and dialogue as instances in this paper. The summarization is a commonly used task for verifying the effectiveness of LLM alignment methods. Additionally, in the dialogue task, the Alpaca dataset we used consists of many NLP tasks (Taori et al., 2023b), including machine translation, sentiment classification, and text simplification.

- *We did not attempt more preference-alignment methods.* In this work, we verify the effectiveness of HBAT based on representative PPO, DPO, and ESRL, *i.e.,* it can offer a refinement of the collaboration among instruction-following alignment and human-preference alignment. Although there are some other preference-alignment methods that we did not experiment with, such as RRHF (Yuan et al., 2023), RAFT (Dong et al., 2023), and RL4F (Akyürek et al., 2023), HBAT is a general approach and can be easily extended to these.

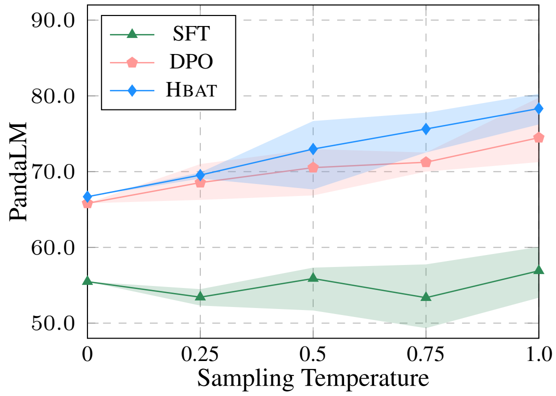


Figure 3: PandaLM score for different sampling temperatures on the LLaMA2-7B model. For each dialogue model, we conduct the generation three times and report the mean score of these generated responses.

Acknowledgements

This work was supported in part by the National Science Foundation of China (No.62276056), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Fundamental Research Funds for the Central Universities (Nos. N2216016 and N2316002), the Yunnan Fundamental Research Projects (No. 202401BC070021), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009). The authors would like to thank Yang Gan and Yifu Huo for their help in human evaluation.

References

Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket

Tandon. 2023. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. *ArXiv preprint*.

Johannes Bader and Eckart Zitzler. 2011. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*.

Jixiang Cheng, Gary G Yen, and Gexiang Zhang. 2015. A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Transactions on Evolutionary Computation*.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *ArXiv preprint*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *ArXiv preprint*.

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. *ArXiv preprint*.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proc. of ICML*.

Jian Hu, Li Tao, June Yang, and Chandler Zhou. 2023. Aligning language models with offline reinforcement learning from human feedback. *ArXiv preprint*.

C-L Hwang and Abu Syed Md Masud. 2012. *Multiple objective decision making—methods and applications: a state-of-the-art survey*. Springer Science & Business Media.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *ArXiv preprint*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021. Conflict-averse gradient descent for multi-task learning. In *Proc. of NeurIPS*.

Kaisa Miettinen, Francisco Ruiz, and Andrzej P Wierzbicki. 2008. Introduction to multiobjective optimization: interactive approaches. In *Multiobjective optimization: interactive and evolutionary approaches*.

Giovanni Misitano, Bhupinder Singh Saini, Bekir Afsar, Babooshka Shavazipour, and Kaisa Miettinen. 2021. Desdeo: The modular and open source framework for interactive multiobjective optimization. *IEEE Access*.

In Jae Myung. 2003. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv preprint*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Proc. of NeurIPS*.

Razvan Pascanu and Yoshua Bengio. 2014. Revisiting natural gradient for deep networks. In *Proc. of ICLR*.

João Luiz Junho Pereira, Guilherme Antônio Oliver, Matheus Brendon Francisco, Sebastiao Simoes Cunha Jr, and Guilherme Ferreira Gomes. 2022. A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. *Archives of Computational Methods in Engineering*.

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. *ArXiv preprint*.

Miguel Moura Ramos, Patrick Fernandes, António Farinhas, and André FT Martins. 2023. Aligning neural machine translation models: Human feedback in training and inference. *ArXiv preprint*.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Proc. of NeurIPS*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023a. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Stanford alpaca: An instruction-following llama model.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*.

Tobias Wagner, Heike Trautmann, and Dimo Brockhoff. 2013. Preference articulation by means of the r 2 indicator. In *Evolutionary Multi-Criterion Optimization: 7th International Conference, EMO 2013, Sheffield, UK, March 19-22, 2013. Proceedings 7*.

Chenglong Wang, Hang Zhou, Kaiyan Chang, Tongran Liu, Chunliang Zhang, Quan Du, Tong Xiao, and Jingbo Zhu. 2023a. Learning evaluation models from large language models for sequence generation. *ArXiv preprint*.

Chenglong Wang, Hang Zhou, Yimin Hu, Yifu Huo, Bei Li, Tongran Liu, Tong Xiao, and Jingbo Zhu. 2023b. Esrl: Efficient sampling-based reinforcement learning for sequence generation. *ArXiv preprint*.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2023c. Making large language models better reasoners with alignment. *ArXiv preprint*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023d. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *ArXiv preprint*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *ArXiv preprint*.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023e. Aligning large language models with human: A survey. *ArXiv preprint*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.

Peng Wu and Li Pan. 2019. Multi-objective community detection based on memetic algorithm. *PloS one*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *ArXiv preprint*.

Bin Xin, Lu Chen, Jie Chen, Hisao Ishibuchi, Kaoru Hirota, and Bo Liu. 2018. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Proc. of NeurIPS*.

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. *ArXiv preprint*.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *ArXiv preprint*.

Hang Zhou, Chenglong Wang, Yimin Hu, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2024. Prior constraints-based reward model training for aligning large language models. *arXiv preprint arXiv:2404.00978*.

A Experimental Details

A.1 Setups

Instruction-Following Alignment. We set the learning rate, batch size, and training epoch to 1e-5, 64, and 3. We did not conduct tuning of these hyper-parameters specific to the task and the model, as our experiments with other hyper-parameters did not yield a significant performance improvement.

Reward Model Training. We initialized the model using the LLM trained by instruction-following alignment training. For all tasks, we trained the reward model for 2 epochs with a learning rate of 1e-5 and a batch size of 64.

PPO Training. We followed an existing PPO implementation in `trlX`[¶] for training the LLM. For all tasks, the learning rate was set to 1e-5 and 5e-6 for the policy model and the value model, respectively. We settled on a batch size of 64 for each PPO step, which consisted of 1 epoch of gradient steps and 4 epochs of mini-batch PPO steps. To address the overoptimization issue as described in Gao et al. (2023)'s work, we implemented a strategy that saves checkpoints at regular intervals during the training process. Specifically, we evaluated checkpoints at intervals of 500 steps for the summarization task and 200 steps for the dialogue task against their respective validation sets and selected the optimal checkpoint with the best Reward score. Additionally, we employed a cold-start trick for PPO, to alleviate the damage caused by the inaccurate estimation of the early value model. Specifically, we updated only the value model and did not update the policy model during the first 50 steps of PPO training. The setups of advantage estimation and KL regularizer coefficient were the same as in `trlX`.

DPO Training. We used a batch size of 64, a learning rate of 1e-6, and a training epoch of 2 for DPO training. Apart from these parameters, the rest of our training setups were the same as in Rafailov et al. (2023).

HBAT. F_{max} was set to 50 and 100 on the summarization task and the dialogue task, respectively. λ and N were set 1 and 10 for all tasks. After training each subset, we evaluated the model's performance with the validation set. The model that has the highest Reward score was selected as the optimal one. Concurrently, we saved the value model after learning from a subset of the human-preference dataset. This saved model was utilized to initialize the value model for subsequent learning of a new subset of the human-preference dataset. Furthermore, in HBAT-Freeze, we froze the top 20% important parameters based on the computed parameter importance scores.

[¶] https://github.com/CarperAI/trlx

Task	Training Stage	Train	Valid	Test
Summarization	IFA	123,169	645	655
	Reward	92,858	1,000	2,000
	HPA	123,169	645	655
Dialogue	IFA	10,000	200	605
	Reward	9,591	100	200
	HPA	20,000	200	605

Table 5: Statistical information on summarization and dialogue datasets. **IFA**: instruction-following alignment; **Reward**: training a reward model; **HPA**: human-preference alignment.

A.2 Dataset Statistics

The statistical information on the utilized datasets is summarized in Table 5.

A.3 Evaluation

PandaLM. In this section, we describe how we compute the PandaLM score. Given the pairwise test responses $\{(x^0, r_a^0, r_b^0), \cdots, (x^T, r_a^T, r_b^T)\}$, where T is the number of the test set, PandaLM can give the preference of each pairwise response, including P_a, P_b, and Tie. Here, P_a denotes response r_a is better than response r_b, P_b denotes response r_b is worse than response r_b, while Tie denotes a tie between response r_a and response r_b. We can compute the PandaLM score for the response r_a model and the response r_b model through the given preferences:

$$S^a_{\text{PandaLM}} = \frac{\text{Count}(P_a)}{T - \text{Count}(Tie)} \quad (11)$$

$$S^b_{\text{PandaLM}} = \frac{\text{Count}(P_b)}{T - \text{Count}(Tie)} \quad (12)$$

where $\text{Count}(\cdot)$ denotes the count of the specified preference.

GPT-4 Prompts for Win Rates. As shown in Figure 4, The prompts of GPT-4 evaluation are the same as in Rafailov et al. (2023).

```
Which of the following summaries does a better job
of summarizing the most important points in the
given forum post, without including unimportant or
irrelevant details? A good summary is both precise and
concise.

Post:
<post>

Summary A:
<Summary A>

Summary B:
<Summary B>

FIRST provide a one-sentence comparison of the two
summaries,explaining which you prefer and why. SECOND,
on a new line. state only "A" or "B" to indicate your
choice. Your response should use the format:

Comparison: <one-sentence comparison and explanation>
Preferred: < "A" or "B">
```

(a) Summarization GPT-4 win rate prompt

```
For the following query to a chatbot, which response is
more helpful?

Query: <the user query>

Response A:
<either the test method or baseline>

Response B:
<the other response>

FIRST provide a one-sentence comparison of the two
responses and explain which you feel is more helpful.
SECOND, on a new line, state only "A" or "B" to indicate
which response is more helpful. Your response should
usethe format:

Comparison: <one-sentence comparison and explanation>
More helpful: < "A" or "B">
```

(b) Dialogue GPT-4 win rate prompt

Figure 4: Prompt templates of computing GPT-4 win rates for summarization and dialogue tasks.

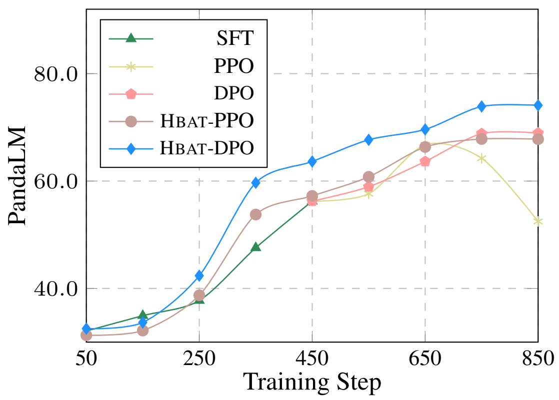


Figure 5: PandaLM score over training steps for the HBAT and traditional two-stage alignment training.

B More Analysis

Comparison of Training Process on Different Methods. We analyze the training process of our HBAT on the dialogue task. Figure 5 shows the PandaLM on the validation set of the LLMs aligned by HBAT and the traditional two-stage alignment methods. We observe that alignment training with HBAT improves performance more efficiently than that with the two-stage method. Furthermore, when using PPO during human-preference alignment training, we can observe that HBAT can mitigate reward model *overoptimization* (Gao et al., 2023).

Integration of Efficient Sampling Method. Our HBAT is orthogonal to the other mainstream methods for improving LLM alignment. Here, we take ESRL, an efficient sampling-based reinforcement learning method (Wang et al., 2023b), as an instance. Specifically, we incorporate ESRL into the PPO algorithm inside our HBAT. In ESRL,

Method	Summarization		Dialogue	
	BS	Win	PandaLM	Win
PPO	-4.27	63.80	69.79	55.63
HBAT	-3.82	72.52	70.88	61.45
ESRL	-4.01	65.90	70.33	58.54
HBAT+ESRL	**-3.65**	**75.11**	**72.91**	**62.56**

Table 6: Performance on summarization and dialogue tasks, using the LLaMA2-7B model aligned with HBAT and ESRL. We implemented ESRL on our test bed with the same setups as in Wang et al. (2023b).

Mtehod	Training	Memory	Win
DPO	1.00×	52.77G	54.23
HBAT	1.26×	61.13G	58.10
HBAT w/ original EWC	1.64×	73.55G	58.32

Table 7: The comparison of efficiency and performance between the modified EWC and the original EWC. We test the training efficiency and memory consumption on eight A800 GPUs. **Time**: training time; **Memory**: maximum memory consumption.

we employ the predicted reward score to estimate model capability. Table 6 shows that the integrated method achieves superior performance.

Fisher Information Matrix. This original EWC employs the Fisher information matrix, denoted as F_θ, to measure information contained in model parameters θ after learning a task (Kirkpatrick et al., 2017). The Fisher information represents the expected information that an observation can provide about an unknown parameter (Pascanu and Bengio, 2014). It can be estimated via first-order derivatives of the generative probability $p_\theta(y|x)$, as described in Eq. 1:

$$F_\theta = \mathrm{E}\left[\left(\frac{\partial \log p_\theta(y|x)}{\partial \theta}\right)^2 \middle| \theta\right] \quad (13)$$

$$= \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \left(\frac{\partial \log p_\theta(y|x)}{\partial \theta}\right)^2 \quad (14)$$

where $\mathcal{D}$ is the training dataset. When employing this method in the context of LLM training, estimating the Fisher information requires computing the gradients for each sample within the training dataset through forward propagation and backpropagation. Then the gradients of each model parameter are summed and divided by the number of samples. This process poses two challenges to LLM training. The first is that the frequent computation of large-scale parameter gradients leads to significant computational costs. The second is that the size of the information matrix will be huge (the same size as the parameters of the aligned LLM), leading to significant GPU memory consumption. To address these challenges, we propose a modified EWC method (see Section 4.2).

We also conduct experiments to compare our modified EWC and original EWC on the dialogue task. The results are presented in Table 7. In terms of training time and memory consumption, our modified EWC consistently outperforms the original EWC. Notably, it can reduce about 23% of training time and 17% of memory consumption. It demonstrates that our modified EWC can be efficiently implemented in alignment training. Furthermore, it shows that our HBAT is capable of handling larger mini-batches, large-scale datasets, larger-sized models, and longer target generation sequences with identical settings on resource-constrained devices. In terms of response quality, our modified EWC achieves a matched GPT-4 win rate compared to the original EWC.

Summarization Task	
Coherence The coherence measures the quality of all sentences collectively, to the fit together and sound naturally. Consider the quality of the summary as a whole. **Rubric:** Score of 1: The summary is impossible to understand. Score of 4: The summary has mistakes or confusing phrasing that make it a bit hard to understand. Score of 7: The summary is perfectly clear.	
Accuracy The accuracy measures whether the facts in the summary are consistent with the facts in the original post. Consider whether the summary does reproduce all facts accurately and does not make up untrue information. **Rubric:** Score of 1: The summary is completely wrong, made up, or exactly contradicts what is written in the post. Score of 4: The summary says at least one substantial thing that is not mentioned in the post, or that contradicts something in the post. Score of 7: The summary has no incorrect statements or misleading implications.	
Coverage This coverage measure whether the summary contains all the important information in the post. **Rubric:** Score of 1: The summary contains no information relevant to the post. Score of 4: The summary is missing at least one important piece of information required to understand the situation. Score of 7: The summary covers all of the important information required to understand the situation.	
Overall Score This score measures how good this summary is overall. **Rubric:** Score of 1: The summary is terrible. Score of 4: The summary is an okay representation of the post, but could be significantly improved. Score of 7: The summary is an excellent representation of the post.	
Dialogue Task	
Fluency This fluency measures whether the response is easy to read. **Rubric:** Score of 1: The content of the response is unreadable and completely incoherent. Score of 4: The content of the response is readable but fails to convey all the information described. Score of 7: The content of the response is readable and enables comprehension of all the included meanings, albeit with minor grammatical issues.	
Accuracy The accuracy measures whether the response contains is error-free. **Rubric:** Score of 1: The content of the response is entirely nonsensical. Score of 4: The content of the response contains at least one misunderstanding that is crucial. Score of 7: The content of the response does not contain any misunderstandings.	
Toxicity The toxicity measures whether the response is harmful, offensive, or potentially damaging to the reader. **Rubric:** Score of 1: The content of the response is toxic. Score of 4: The content of the response is slightly toxic or perceived as toxic by some. Score of 7: The content of the response is non-toxic.	
Helpfulness The helpfulness measures whether the response provides useful information or solutions that address the reader's query. **Rubric:** Score of 1: The content of the response is not helpful. Score of 4: The content of the response solves part of the problem, but some issues remain unresolved. Score of 7: The content of the response fully addresses the problem.	
Overall Score This score measures how good this response is overall. **Rubric:** Score of 1: The response is extremely poor. Score of 4: The response is not perfect and has room for improvement. Score of 7: The content of the response is satisfactory.	

Table 8: Our human evaluation rubrics for the summarization and dialogue tasks. Note that the rubrics for the summarization task are adopted from Stiennon et al. (2020).