Not All Preference Pairs Are Created Equal: A Recipe for Annotation-Efficient Iterative Preference Learning [*]

Sen Yang[1], **Leyang Cui**[2†], **Deng Cai**[2], **Xinting Huang**[2], **Shuming Shi**[2], **Wai Lam**[1]

[1]The Chinese University of Hong Kong [2]Tencent AI Lab

{senyang.stu,nealcly.nlp,thisisjcykcd}@gmail.com

timxthuang@tencent.com wlam@se.cuhk.edu.hk

Abstract

Iterative preference learning, though yielding superior performances, requires online annotated preference labels. In this work, we study strategies to select worth-annotating response pairs for cost-efficient annotation while achieving competitive or even better performances compared with the random selection baseline for iterative preference learning. Built on assumptions regarding uncertainty and distribution shifts, we propose a comparative view to rank the implicit reward margins as predicted by DPO to select the response pairs that yield more benefits. Through extensive experiments, we show that annotating those response pairs with *small* margins is generally better than *large* or *random*, under both single- and multi-iteration scenarios. Besides, our empirical results suggest allocating more annotation budgets in the earlier iterations rather than later across multiple iterations.

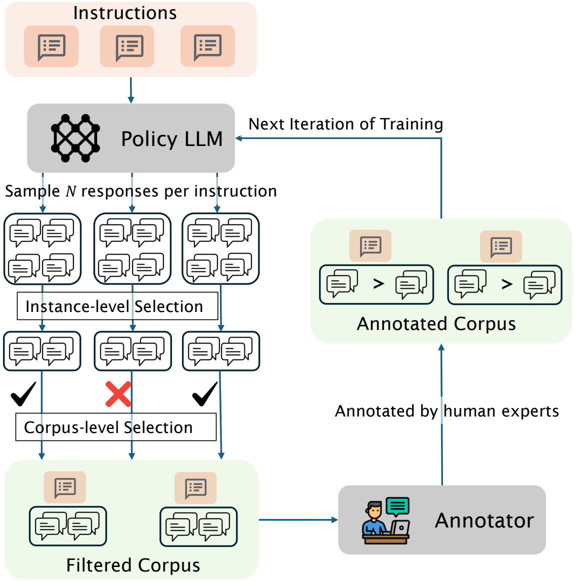


Figure 1: The workflow of online iterative preference learning, in which we apply two levels of selection before annotation.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023a; OpenAI, 2024) have shown remarkable capabilities to understand and generate human languages, supporting applications such as question answering, coding, and psychological counseling. One of the keys to such success is to align LLMs with human-desired behaviors through preference learning. This is accomplished by annotating preference datasets and employing preference learning methods such as proximal policy optimization (Schulman et al., 2017, PPO) and direct preference optimization (Rafailov et al., 2023, DPO).

To continuously improve LLMs' capability, recent work underlines the significance of iterative preference learning, which repetitively interleaves between training the model and collecting online preference annotations. For example, LLaMA-2 and Claude series benefited from iterative RLHF training on human preference annotations that were collected in batches on a weekly basis (Touvron et al., 2023b; Bai et al., 2022); while multiple papers also reported that iterative DPO brings clear performance gains (Xu et al., 2024; Yuan et al., 2024; Xiong et al., 2024; Rosset et al., 2024; Wu et al., 2024).

Despite their success, the process of collecting and annotating such online preference datasets is both time-consuming and costly. These methods normally sample multiple responses per instruction on a large new collection of instructions and simply annotate all the responses. The best and the worst responses are selected to formulate a pair per instruction to build a training corpus for the next iteration (Touvron et al., 2023b; Yuan et al., 2024; Dong et al., 2024; Wang et al., 2023). This leads

[*]The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719).

[†]Correspondence to Leyang Cui.

Findings of the Association for Computational Linguistics: EACL 2024, pages 6549–6561
November 12-16, 2024 ©2024 Association for Computational Linguistics

us to ask whether there exist alternative annotation strategies that are more cost-efficient. Besides, existing methods normally allocate annotation budgets evenly across multiple iterations, but it remains unknown whether the model benefits from training on more instances in earlier or later iterations. All in all, we are interested in a research question: *how to make better use of limited annotation budgets* to aid online iterative preference learning.

In this paper, we address this question by conducting a systematic study on iterative DPO based on LLaMA-3-8B (AI@Meta, 2024). We study the implicit reward margin as an informative indicator, which serves a key role in DPO and other direct preference learning methods (§3). Such a choice is supported by the formulation of DPO, in which the reward margin roughly represents prediction uncertainty from the discriminative perspective or distribution shift from the generative perspective. We consider two levels of granularity, namely the instance level and the corpus level, to rank reward margins from a comparative view. Instance-level selection aims to find a worth-annotating pair of responses from $\frac{N(N-1)}{2}$ pairs if N responses are sampled per instruction; while corpus-level selection considers filtering out those trios that do little help for alignment tuning from a large set of trios each consisting of an instruction and a pair of responses.

We conduct experiments in the single-iteration case, in which the policy LLM goes through one round of online training after being initially trained on an offline dataset. We find that the *smallest*-margin subset always works better than the *largest*- and *random*-subsets, on either the instance level or the corpus level. Upon further checking the ranking accuracy and KL-divergences on each selected subset, we show that our assumptions regarding uncertainty and distribution shift are partly supported by our findings. We then generalize the winning strategy, *always-smallest*, to the multi-iteration case (§4). Experimental results demonstrate that the *always-smallest* strategy yields continuous and significant improvements over multiple iterations; while the *always-random* strategy sees little to no gains upon training on more iterations. After that, we explore three strategies, namely *increase*, *constant*, and *decrease*, to allocate annotation budgets to multiple iterations. Empirical results suggest it is better to adopt *decrease* and avoid *increase*.

2 Preliminaries

In this section, we give a brief review of direct preference learning methods (§2.1) and iterative DPO (§2.2). Then we present the preliminary setup of this work (§3.2).

2.1 Direct Preference Learning

Recently, several works have bypassed the need to train a separate reward model, thus mitigating the instability issue of PPO training (Dong et al., 2023; Rafailov et al., 2023; Zhao et al., 2023). Among them, DPO gives a closed-form solution derived from the Bradley-Terry (BT) model (Bradley and Terry, 1952) to optimize a reward function from which the optimal policy is deterministically mapped:

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

where

$$\rho = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \quad (2)$$

Throughout this work, we use DPO as the default preference learning method due to its closed-form theoretical guarantee and stability during training.

2.2 Online Iterative DPO

Online iterative DPO is shown to be effective by multiple recent work (Xu et al., 2024; Yuan et al., 2024; Xiong et al., 2024; Rosset et al., 2024; Wu et al., 2024; Swamy et al., 2024; Tran et al., 2023; Ye et al., 2024; Guo et al., 2024; Tajwar et al., 2024; Calandriello et al., 2024). Most of them repetitively interleave between training the model and collecting online preference annotations.

In this work, we assume a practical scenario where one round of supervised fine-tuning (SFT) and offline DPO has been implemented before the subsequent online iterations, given the availability of many open-sourced preference learning datasets (Maas et al., 2011; Stiennon et al., 2020; Bai et al., 2022; Ethayarajh et al., 2022; Nakano et al., 2022; Lambert et al., 2023; Cui et al., 2023). Formally, we denote the SFT checkpoint as π_{ref} and the initial offline-tuned DPO checkpoint as π_θ^0. Our adopted iterative DPO framework repetitively applies the following Step-i:

Step-*i*

- Given a set of M instructions, N responses are sampled from π_θ^{i-1} for each instruction.

π_{ref} and π_θ^{i-1} are used to predict the implicit reward[*], $\log \frac{\pi_\theta^{i-1}(y|x)}{\pi_{\text{ref}}(y|x)}$.

- Some strategies are applied to select a subset of preference instances. The selected instances, each consisting of an instruction and two responses, are then annotated by an oracle preference annotator, e.g., human experts.
- The annotated instances are fed into π_θ^{i-1} and π_{ref} to train π_θ^i.

Among all these sub-steps, our work focuses on how to select a proper subset of preference instances before annotation. We present an illustration of the workflow in Figure 1.

3 Margin-based Selection within One Iteration

We begin by analyzing the data selection strategies within a single iteration. Existing methods (Touvron et al., 2023b; Yuan et al., 2024; Xiong et al., 2024; Rosset et al., 2024; Wu et al., 2024) annotate all generated instances for the next iteration. We question whether other strategies could achieve better performance with the same amount of annotation budgets. We thus explore a simple yet intuitive metric for data selection, the reward margin between the chosen and the rejected responses. The reward margin ρ serves as the key component of the DPO loss function[†]:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(\rho)\right] \quad (3)$$

where

$$\rho = \underbrace{\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}}_{\text{reward margin}} \quad (4)$$

$$= \beta \underbrace{\log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}}_{\text{policy log ratio}} - \beta \underbrace{\log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)}}_{\text{reference log ratio}} \quad (5)$$

There exist multiple interpretations for ρ. The most straightforward one is to regard ρ as the reward margin between y_w and y_l as predicted by a pairwise RM. However, ρ is much more intriguing than that because it directly models the output logits of π_θ and π_{ref} as implicit rewards, which represent the mixture of the two generative distributions.

Instead of staring at a single margin value that provides little insight for data selection, we consider ranking a set of reward margins. We assume the highest- or lowest-ranking subset might be of use to find the worth-annotating instances. We do not have a formal theory to support this assumption, but there are some intuitions from two points of view, i.e., discriminative and generative, associated with the mixed nature of DPO that optimizes a pairwise discriminative function that is built upon the output logits of generative LLMs. We will show empirical evidence to support these intuitions in §3.5.

Uncertainty Upon regarding the policy model as a pairwise discriminative model, reward margins reflect the model's confidence in the prediction, so the most or least confident instances might be of extra use (Culotta and McCallum, 2005; Schröder et al., 2022). Specifically, Bai et al. (2022) showed that the calibration curve of a preference model roughly matches the logistic function, $\text{acc} = 1/(1+e^{-\rho})$, for models ranging from 10^8 to 10^{10} parameters, demonstrating that reward margin is a good proxy of uncertainty.

Distribution Shift As shown in Eq (5), reward margins represent the difference between the log ratios of the policy and the reference model; such a difference might correlate with the degree of generative distribution shift from the reference model to the policy. For example, if y_w and y_l lie in a similar distribution to that of the dataset where π_θ was trained on, there should be a clear gap between the two log ratios since π_θ has been trained to yield a higher generative probability for y_w than y_l while π_{ref} has not. On the contrary, y_w and y_l may have not been effectively learned by π_θ if the two log ratios were almost canceled and the margin is small, in which case π_θ and π_{ref} show similar generative behaviors to distinguish y_w from y_l

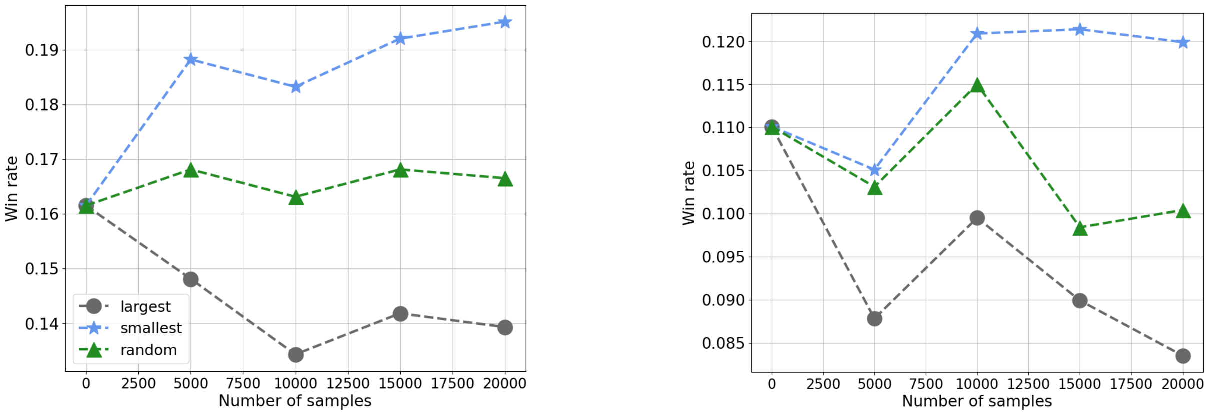
Given the above assumption that it might be useful to rank reward margins, the question then becomes how to define a good set of preference pairs on which we can get an informative ranking. In the next section, we will discuss two levels of granularity to define potentially useful sets for ranking.

3.1 Strategy Variants

We explore two levels of granularity to rank reward margins to select worth-annotating instances. The

[*] Such a definition is only valid under the case of reward margins so that the partition term is canceled.

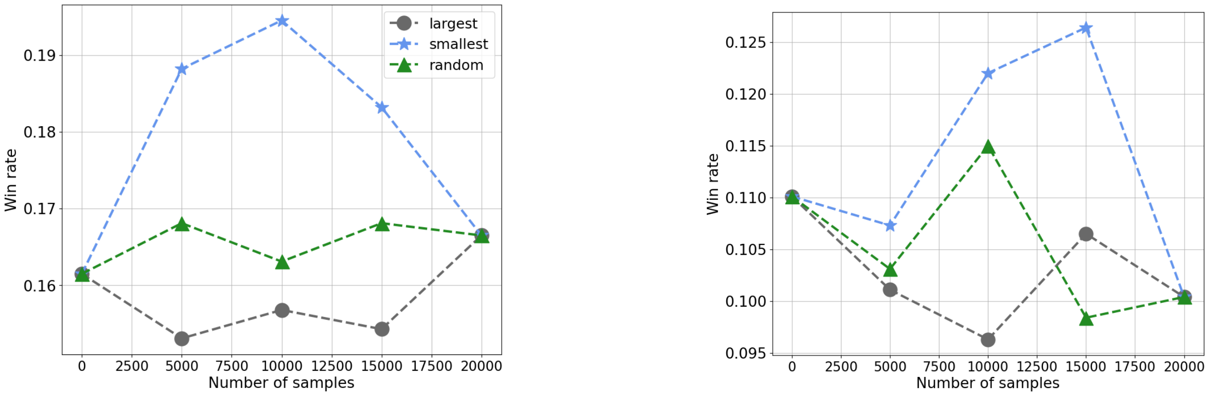
[†] ρ is also the key role in many other direct preference learning methods, such as IPO (Azar et al., 2023, $\mathcal{L}_{\text{IPO}} = (\rho - 1)^2$) and SLiC (Zhao et al., 2023, $\mathcal{L}_{\text{SLiC}} = \max\{0, 1-\rho\}$).



(a) Evaluated the gold RM.

(b) Evaluated by GPT-4.

Figure 2: Results on AlpacaEval-2.0 with different instance-level strategies and different training set sizes.



(a) Evaluated by the gold RM.

(b) Evaluated by GPT-4.

Figure 3: Results on AlpacaEval-2.0 with different corpus-level strategies and different training set sizes.

instance-level selection ranks the margins of responses sampled from the same instruction; while the corpus-level selection applies to the set of response pairs of the entire corpus.

Instance-level Existing online preference learning methods that rely on human annotations normally send multiple responses for annotating per instruction and select the best and the worst to compose a pair. However, it would be too costly to adopt this brute force strategy that asks a human expert to read and rank all N responses. We thus investigate the margins between any two responses to select a worth-annotating pair among $\frac{N(N-1)}{2}$ pairs, so that the cost remains the same as in $N = 2$ while the diversity of responses is promoted.

Corpus-level Given a set of instances, each consisting of an instruction and a pair of responses, it is intuitive to discriminate between the instances that are beneficial from those less beneficial or even counterproductive. We thus explore to rank the reward margins between pairs of responses over the entire corpus, seeking an informative partition.

The two levels of selection can be applied consecutively: One may first select the smallest-margin response pair for each instruction, then gather all such instances together to form a corpus, and finally select the largest-margin subset on the corpus level. We do not consider all the combinations of the two levels due to prohibitive computational costs. Instead, we assume one to be *random selection* when experimenting with the other.

Margin Normalization The length bias issue has been shown to prevail among RLHF methods, including DPO (Park et al., 2024; Meng et al., 2024). In our experiments, we also find the margin-based rankings of response pairs on the corpus level vary significantly between the length-normalized and un-normalized versions. We thus consider length normalization to mitigate possible length bias during ranking. Specifically, in addition to the un-normalized experiments, we also conduct experiments with the following normalized margin:

$$\hat{\rho} = \frac{1}{|y_w|} \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \frac{1}{|y_l|} \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \tag{6}$$

3.2 Experimental Setup

3.2.1 Synthetic Oracle

We aim to analyze annotation efficiencies in iterative preference learning, in which online preference annotations are collected in batches. However, obtaining the true "gold standard" preference labels from human annotators can be costly, and may be inconsistent if the group of annotators varies across batches. Inspired by Gao et al. (2022), we instead employ a synthetic setup where the ground truth is determined by the outputs of a reliable reward model, which we term as *gold RM*. This gold RM is regarded as the alternative of human experts across all our experiments in terms of both annotation and evaluation, despite it is not the real ground truth.

3.2.2 Training

We adopt LLAMA-3-8b-base (AI@Meta, 2024) to initialize the policy LLM and PairRM (Jiang et al., 2023) as the gold RM. We sample 10,000 instances from UltraFeedback (Cui et al., 2023), which are then used to train π_{ref} and π_θ^0. The instructions from the remaining UltraFeedback are kept for subsequent iterations. We use π_{ref} as the reference model for DPO training across all iterations. The training hyper-parameters are listed in Appendix A.

3.2.3 Evaluation

We evaluate the policy LLMs on AlpacaEval-2.0 (Li et al., 2023) and use the outputs generated by GPT-4 as the reference to compare against. We aim to get findings from the judgments predicted by the gold RM. Such findings are approximations for the industrial scenario where human experts take the place of our gold RM. One concern with this synthetic setup is the reward hacking issue, in which the policy overfits the preferences of the gold RM since the gold RM remains fixed after all (Gao et al., 2022; Rafailov et al., 2024). We investigate whether this issue exists in our experiments by additionally evaluating with the standard AlpacaEval-2.0 protocol, i.e., GPT-4[‡] as the evaluator. A model is deemed as "hacked" if it shows improvements when evaluated by the gold RM but degrades when evaluated by GPT-4. It should be noted that the judgments predicted by the gold RM are still considered accurate proxies under the synthetic setup upon being hacked, though they would diverge from real human preferences.

Overall, we consider two criteria: (1) How well does the model align with the gold RM which is regarded as the "ground truth" in our experiments? (2) How does the model perform under general evaluation?

3.3 Empirical Workflow

Our workflow starts from π_θ^0 and a set of 20,000 instructions sampled from UltraFeedback (Cui et al., 2023):

Step 1 $N = 8$ responses are sampled from π_θ^0 for each instruction. π_{ref} and π_θ^0 are used to predict the implicit reward (without the partition term), $\log \frac{\pi_\theta^0(y|x)}{\pi_{\text{ref}}(y|x)}$, and the normalized version, $\frac{1}{|y|} \log \frac{\pi_\theta^0(y|x)}{\pi_{\text{ref}}(y|x)}$.

Step 2 Margin-based strategies are adopted to select a subset of preference pairs. The selected pairs are then annotated using the gold RM.

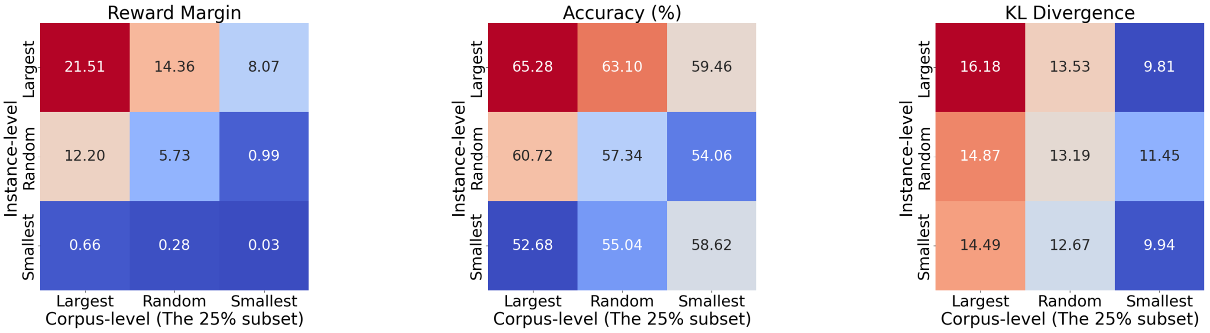
- **Instance-level**: For each prompt, the preference pairs with the {*largest* & *smallest*} margin are selected among all the response pairs. A *random* baseline is also included for comparison, in which the selected pair is simply the first two responses. All such prompt-chosen-rejected trios are collected to formulate a dataset.
- **Corpus-level**: Given a set of instances where each instance consists of a prompt and two responses, those pairs with the {*largest* & *smallest*} margins are selected. Similarly, a *random* baseline is included.
- **Margin Normalization**: Length normalization is enabled and disabled, respectively, for all the variants adopted.

Step 3 The collected subset is fed into π_θ^0 and π_{ref} to obtain the online trained policy, which then gets evaluated.

3.4 Results

smallest* $>$ *random* $>$ *largest We evaluate the single-iteration performances trained on different numbers of instances, with instance-level and

[‡] GPT-4-1106-preview



(a) (b) (c)

Figure 4: Some statistics about the selected subset using different strategies combined. The collected subsets are used to train π_θ^1 under the single-iteration setting.

Selected / Ranking	Largest	Smallest
Instance-level	$50.75_{\pm 0.51}$	$47.83_{\pm 2.10}$
Corpus-level	$49.51_{\pm 1.15}$	$49.08_{\pm 2.26}$

Table 1: Win rates of length-normalization models against the un-normalized counterparts with several variants, as evaluated by the gold RM. The win rates are averaged over multiple runs with different numbers of instances.

corpus-level ranking. The win rates against GPT-4 outputs on AlpacaEval, as evaluated by the gold RM and GPT-4, are shown in Figures 2 and 3. We first observe that annotating more data does not necessarily lead to more performance gain. This shows that it is necessary to strategically select data for annotating. Selection with *smallest* margins yields consistent improvements over the *random* baseline, regardless of the instance- or corpus-level rankings. In contrast, selection with *largest* margins shows negative effects, which may be caused by overfitting to the confident instances. Such results conform with our intuitions in §3.

Instance-level As shown in Figure 2, the three ranking schemes show consistent gaps across different numbers of samples, suggesting that it is not enough to just sample $N > 2$ responses for a prompt but one also needs to pay attention to select the proper preference pair. In our experiments, we show it is beneficial to select the preference pair with the *smallest* margin among all $\frac{N(N-1)}{2}$ pairs.

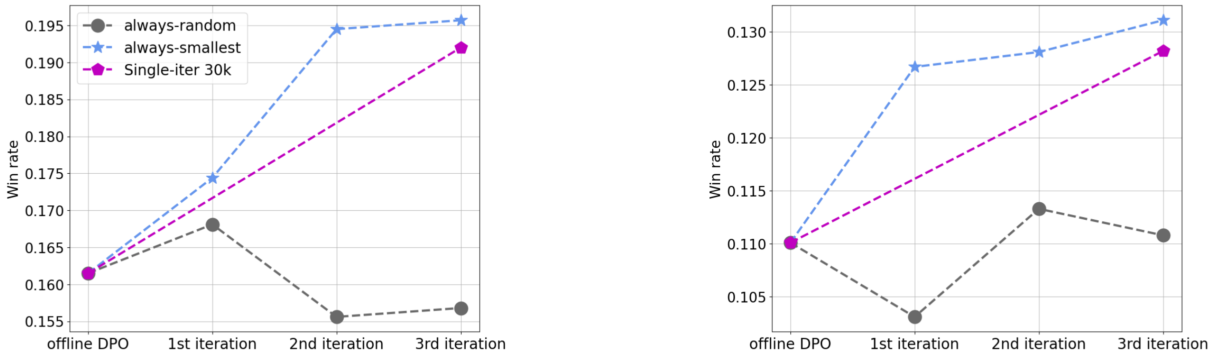
Corpus-level In Figure 3, the curve labeled as *largest* significantly improves in terms of win rate between 15,000 and 20,000 samples. The only difference between the two is adding the 5,000 samples with the *smallest* margins to the training set. In comparison, the curve labeled as *smallest* shows a significant drop from 15,000 to 20,000 samples; and the only difference is including the *largest*-margin 5,000 samples in the training set. Taking both together, our experiments suggest including the *smallest*-margin subset while discarding the *largest*-margin subset during corpus-level data selection.

Length Normalization Table 1 shows the averaged win rates of the normalized models against the un-normalized ones across multiple runs. We emphasize *smallest* since we found it to be the best strategy for both levels of selection. On the instance level, un-normalization shows clear improvement over the normalized counterpart; while on the corpus level, the superiority persists but is not significant. Overall, our experiments suggest using the un-normalized reward formulation for both levels of granularity.

3.5 Analysis

Figure 4 shows some statistics on the selected subsets, including reward margin (as predicted by π_θ^0 and π_{ref}), ranking accuracy, and KL-divergence.

Uncertainty As shown in Figure 4b, the accuracies generally follow a good calibration trend, i.e., ranking accuracy gradually increases as the reward margin increases. For example, *corpus-random instance-smallest* and *corpus-smallest instance-random*, with ranking accuracies of 55.04% and 54.06%, have been shown to yield better alignment performances than *corpus-random instance-largest* and *corpus-largest instance-random*, whose ranking accuracies are 63.10% and 60.72%. Therefore, our uncertainty assumption, that the model benefits



(a) Evaluated the gold RM.

(b) Evaluated by GPT-4.

Figure 5: Multi-iteration results on AlpacaEval-2.0 with *always-random* and *always-smallest* strategies, respectively, across three follow-up iterations, with 5k instances (originally 10k instructions) per iteration. A single-iter baseline, which is trained by using all the instructions with the *always-smallest* strategy within a single round, is also included for comparison.

more from training on the set with a higher degree of uncertainty, is partly supported.

Distribution Shift Since the reward function of DPO directly models the generative distribution of the LLMs, the reward margin measures the difference between the generative behaviors of the policy and the reference model, which may correspond to the distribution shift between the to-be-annotated instances and the already-trained ones. We measure the degree of distribution shift using KL-divergences and sketch them on different groups of data points in Figure 4c. It is observed that there is a rough trend, with smaller reward margins more likely come smaller KL-divergences, i.e., smaller distribution shifts. For example, the *corpus-largest instance-largest* strategy, with a reward margin of 21.51, yields a KL-divergence of 16.18, which is much higher than the KL-divergences of other strategies that have smaller reward margins as well.

4 Training for Multiple Iterations

Given our findings in the single-iteration case in §3, we would like to know how well the winning strategies generalize to multiple iterations (**Q4.1**). Besides, recent work reported continuous improvements with multiple iterations for iterative DPO (Wu et al., 2024; Dong et al., 2024). They all adopted an evenly distributed strategy to allocate annotation budgets across iterations. A natural question is, whether the model benefits from training on more instances in earlier or later iterations, rather than always training on the same amount (**Q4.2**). The background experimental setup follows that of §3.2.

4.1 Empirical Workflow

The multi-iteration workflow starts from π_θ^0 and a set of instructions from UltraFeedback. We divide the instructions into 3 sets for 3 rounds of iteration, where the i-th round uses M_i instructions. Specifically, for the i-th round of iteration, the following steps are implemented:

Step 1 $N = 8$ responses are sampled from π_θ^{i-1} for each instruction. π_{ref} and π_θ^{i-1} are used to predict the implicit reward, $\log \frac{\pi_\theta^{i-1}(y|x)}{\pi_{\text{ref}}(y|x)}$.

Step 2 We design experiments to answer the two questions, **Q4.1** and **Q4.2**.
Q4.1: Always-smallest *versus* Always-random:

- **Always-smallest**: We first select the response pair with the smallest reward margin among all $\frac{N(N-1)}{2} = 28$ pairs per instruction. All the instructions and selected response pairs are collected to formulate a corpus, on which a corpus-level ranking is then applied to select the $50\% \times M_i = 5,000$ instances with the smallest reward margins. The selected instances are annotated by the gold RM and then used as the training set for the current iteration. We adopt the un-normalized reward formulation.
- **Always-random**: On the instance level, we simply select the first two responses to formulate a pair for each instruction. All the instructions and selected response pairs are collected to formulate a corpus, among which $50\% \times 10,000 = 5,000$ of the instances are

Allocation \ Evaluator	Gold RM	GPT-4
Increase	19.06	12.05
Constant	**19.57**	13.11
Decrease	19.41	**13.49**

Table 2: Results on different strategies (increase, constant, and decrease) to allocate annotation budgets across multiple iterations. We evaluate the win rates against GPT-4 outputs using the gold RM and GPT-4 as evaluators. The largest numbers are bolded.

randomly sampled. The sampled instances are fed into the gold RM for annotation and then used as the training set.

Q4.2: Increase *versus* Constant *versus* Decrease: We adopt the *always-smallest* strategy in this setup. After three rounds of iteration, each allocation strategy is trained on 30,000 instances. The numbers of instructions used for the three iterations for each case are as follows ($M_1 \rightarrow M_2 \rightarrow M_3$):

- Increase: 5,000 → 10,000 → 15,000;
- Constant: 10,000 → 10,000 → 10,000;
- Decrease: 15,000 → 10,000 → 5,000.

Step 3 The collected subset is fed into π_θ^{i-1} and π_{ref} to obtain π_θ^i, which then gets evaluated.

4.2 Results

Answer to Q4.1 Figure 5 shows the win rates of *always-smallest* and *always-random*. The *always-random* baseline yields moderate improvements in the first one or two iterations but finally drops down upon being further optimized; while the *always-smallest* strategy gives consistent and significant improvements across three iterations. This suggests that the selection of response pairs for annotation plays a crucial role in facilitating continuous improvements for online iterative DPO. Besides, the single-iter-30k baseline lags behind *always-smallest*, indicating the effectiveness of corpus-level selection.

Answer to Q4.2 Table 2 shows the results with different allocation strategies. Considering the results from both evaluators, *decrease* is slightly better than *constant* and much better than *increase*. This may result from the fact that the data quality in later iterations depends on the policy trained in earlier iterations, so it is better to allocate more data in the beginning to obtain a better policy.

5 Related Work

5.1 Iterative Preference Learning

Online iterative preference learning refers to the framework in which response pairs are sampled from the policy models and are then annotated to become the training data to continuously improve the policy model. Intuitively, it could mitigate the reward hacking issue or the distribution shift issue (Gao et al., 2022; Rafailov et al., 2024) Online iterative preference learning has been verified to be effective for alignment methods with explicit rewards (Bai et al., 2022; Touvron et al., 2023b) and for direct preference learning methods (Xu et al., 2024; Yuan et al., 2024; Xiong et al., 2024; Rosset et al., 2024; Wu et al., 2024; Swamy et al., 2024; Tran et al., 2023; Ye et al., 2024; Guo et al., 2024; Tajwar et al., 2024; Calandriello et al., 2024; Chen et al., 2024). Specifically, online direct preference learning was first presented in Xu et al. (2024). Dong et al. (2024) shows a systematic training pipeline and releases a strong policy checkpoint based on LLAMA-3-8B-base. Another line of work has investigated Nash equilibrium for LLM alignment, which is shown to natively support online iterative training by theory (Wu et al., 2024; Rosset et al., 2024; Munos et al., 2024).

5.2 Active Learning for NLP

Most active learning methods for NLP consider either *informativeness*, such as prediction uncertainty (Schröder et al., 2022; Margatina et al., 2021; Zhang et al., 2022; Jiang et al., 2020) and gradient (Settles et al., 2007), or *representativeness*, such as representative of the unlabeled set (Settles and Craven, 2008) and differences from already labeled instances (Kim et al., 2006; Zhao et al., 2020; Erdmann et al., 2019; Gissin and Shalev-Shwartz, 2019). In this work, we draw intuitions from both uncertainty (from the discriminative perspective) and representativeness (from the generative perspective).

Active Learning for LLM Alignment Muldrew et al. (2024) presented an active learning approach to make better use of a limited preference labeling budget. Their methods are based on the assumption that the learning process is initialized from the base LLM and the annotated dataset is extremely small, thus the instances with large reward margins can provide greater gradients and alter the model's weights more significantly.

6 Conclusion

In this work, we investigated strategies to make better use of limited annotation budgets for iterative preference learning. Through extensive experiments, we found that it is better to select the response pairs with smaller predicted reward margins and to allocate more annotation budgets in earlier iterations. We hope our findings could benefit the community to obtain better models with limited resources.

Limitations

We observe several limitations regarding this work:

- Our experiments are conducted under a synthetic-oracle setting. Though this setting has been widely adopted as a proxy of human oracle annotations (Gao et al., 2022; Rafailov et al., 2024) and we have included an external evaluator (GPT-4) to avoid the potential reward hacking issue, it is still possible that it introduces some unknown biases to the empirical findings.
- We did not consider the levels of annotating difficulty when measuring annotation cost. Throughout this work, we consider annotation cost to linearly correlate with the number of to-be-annotated response pairs. However, in practice, different pairs come with different levels of difficulty for annotators.

References

AI@Meta. 2024. Llama 3 model card.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *Preprint*, arXiv:2310.12036.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, Rishabh Joshi, Zeyu Zheng, and Bilal Piot. 2024. Human alignment of large language models through online preference optimisation. *Preprint*, arXiv:2403.08635.

Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. 2024. Bootstrapping language models with dpo implicit rewards. *Preprint*, arXiv:2406.09760.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint*, arXiv:2310.01377.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, page 746–751. AAAI Press.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *Preprint*, arXiv:2304.06767.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *Preprint*, arXiv:2405.07863.

Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. 2019. Practical, efficient, and customizable active learning for named entity recognition in the digital humanities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2223–2234, Minneapolis, Minnesota. Association for Computational Linguistics.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Leo Gao, John Schulman, and Jacob Hilton. 2022. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *Preprint*, arXiv:1907.06347.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, Johan Ferret, and Mathieu Blondel. 2024. Direct language model alignment from online ai feedback. *Preprint*, arXiv:2402.04792.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Zhuoren Jiang, Zhe Gao, Yu Duan, Yangyang Kang, Changlong Sun, Qiong Zhang, and Xiaozhong Liu. 2020. Camouflaged Chinese spam content detection with semi-supervised generative active learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3080–3085, Online. Association for Computational Linguistics.

Seokhwan Kim, Yu Song, Kyungduk Kim, Jeong-Won Cha, and Gary Geunbae Lee. 2006. MMR-based active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 69–72, New York City, USA. Association for Computational Linguistics.

Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. 2023. Huggingface h4 stack exchange preference dataset.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. 2024. Active preference learning for large language models. In *International Conference on Machine Learning, ICML 2024*.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. 2024. Nash learning from human feedback. *Preprint*, arXiv:2312.00886.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *Preprint*, arXiv:2403.19159.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. 2024. Scaling laws for reward model overoptimization in direct alignment algorithms. *Preprint*, arXiv:2406.02900.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. 2024. Direct nash optimization: Teaching language models to self-improve with general preferences. *Preprint*, arXiv:2404.03715.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii. Association for Computational Linguistics.

Burr Settles, Mark Craven, and Soumya Ray. 2007. Multiple-instance active learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1289–1296, Red Hook, NY, USA. Curran Associates Inc.

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

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. 2024. A minimaximalist approach to reinforcement learning from human feedback. *Preprint*, arXiv:2401.04056.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *Preprint*, arXiv:2404.14367.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Hoang Tran, Chris Glaze, and Braden Hancock. 2023. Iterative dpo alignment. Technical report, Snorkel AI.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2023. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *Preprint*, arXiv:2311.09528.

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024. Self-play preference optimization for language model alignment. *Preprint*, arXiv:2405.00675.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *Preprint*, arXiv:2312.11456.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2024. Some things are more cringe than others: Iterative preference optimization with the pairwise cringe loss. *Preprint*, arXiv:2312.16682.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Nan Jiang, and Tong Zhang. 2024. Online iterative reinforcement learning from human feedback with general preference model. *Preprint*, arXiv:2402.07314.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Shujian Zhang, Chengyue Gong, Xingchao Liu, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. 2022. ALLSH: Active learning guided by local sensitivity and hardness. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1328–1342, Seattle, United States. Association for Computational Linguistics.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *Preprint*, arXiv:2305.10425.

Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. 2020. Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1796–1806, Online. Association for Computational Linguistics.

A Implementation Details

Training Setup Throughout this paper, we adopt the set of hyper-parameters shown in Table 3. We use 8× A100-40G GPUs for all the training, with BF16 enabled.

Sampling Setup For sampling the responses for online training, we adopt a temperature of 1.0 and top-k sampling with $k = 50$. Both the max-prompt-length and the max-generate-tokens are set to 512. We enable BF16 during sampling.

Evaluation Setup We use greedy decoding to generate the responses for AlpacaEval-2.0. Both the max-prompt-length and the max-generate-tokens are set to 512. We enable BF16 during generation.

Setting	β	Learning Rate	Batch Size	# Epoch
SFT	NA	2e-5	128	1.0
DPO	0.1	5e-7	128	1.0

Table 3: Training Hyper-parameters.