Self-Exploring Language Models: Active Preference Elicitation for Online Alignment

Anonymous Author(s)
Affiliation
Address
email

Abstract

Preference optimization, particularly through Reinforcement Learning from Human Feedback (RLHF), has achieved significant success in aligning Large Language Models (LLMs) to adhere to human intentions. Unlike offline alignment with a fixed dataset, online feedback collection from humans or AI on model generations typically leads to more capable reward models and better-aligned LLMs through an iterative process. However, achieving a globally accurate reward model requires systematic exploration to generate diverse responses that span the vast space of natural language. Random sampling from standard reward-maximizing LLMs alone is insufficient to fulfill this requirement. To address this issue, we propose a bilevel objective optimistically biased towards potentially high-reward responses to actively explore out-of-distribution regions. By solving the inner-level problem with the reparameterized reward function, the resulting algorithm, named *Self-Exploring Language Models* (SELM), eliminates the need for a separate RM and iteratively updates the LLM with a straightforward objective. Compared to *Direct Preference Optimization* (DPO), the SELM objective reduces indiscriminate favor of unseen extrapolations and enhances exploration efficiency. Our experimental results demonstrate that when finetuned on Zephyr-7B-SFT and Llama-3-8B-Instruct models, SELM significantly boosts the performance on instruction-following benchmarks such as MT-Bench and AlpacaEval 2.0, as well as various standard academic benchmarks in different settings.

1 Introduction

Large Language Models (LLMs) have recently achieved significant success largely due to their ability to follow instructions with human intent. As the defacto method for aligning LLMs, Reinforcement Learning from Human Feedback (RLHF) works by maximizing the reward function, either a separate model [43, 5, 18] or reparameterized by the LLM policy [48, 47, 4, 67], which is learned from the prompt-response preference data labeled by humans. The key to the success of alignment is the response *diversity* within the preference data, which prevents reward models (RMs) from getting stuck in local optima, thereby producing more capable language models.

Offline alignment methods [48, 53] attempt to manually construct diverse responses for fixed prompts [11, 24, 69], which, unfortunately, struggles to span the nearly infinite space of natural language. On the other hand, online alignment follows an *iterative* procedure: sampling responses from the LLM and receiving feedback to form new preference data for RM training [43, 21]. The former step helps explore out-of-distribution (OOD) regions through randomness in sampling. However, in standard online RLHF frameworks, maximizing the expected reward learned from the collected data is the only objective for the LLM, sampling from which often leads to responses clustered around local

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

optima. This passive exploration mechanism can suffer from overfitting and premature convergence, leaving the potentially high-reward regions unexplored.

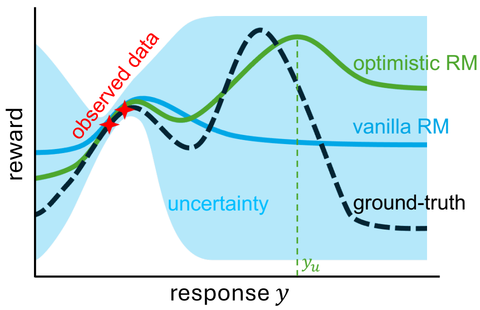

Figure 1: Intuition of our method. For a fixed prompt x, a reward model $r(x, y)$ tries to fit the ground-truth reward $r^*(x, y)$. The blue and green RMs are equally good when using standard reward-fitting loss $\mathcal{L}_{\text{lr}}$, since the observed preference data (red stars) are fitted equally well. However, the green RM has a larger $\max_y r(x, y)$ and thus a lower optimistically biased loss $\mathcal{L}_{\text{lr}} - \alpha \max_y r(x, y)$. Therefore, the response y_u at which the uncertainty is high can be elicited and then proceeded for human feedback to reduce uncertainty.

To address this issue, we propose an active exploration method for online alignment that elicits novel favorable responses. In its simplest form, an optimism term $\alpha \max_y r(x, y)$ is added to the reward-fitting objective (e.g., logistic regression on dataset $\mathcal{D}$), denoted as $-\mathcal{L}_{\text{lr}}$, resulting in a bilevel optimization objective for the *reward* model r:

$$\max_r \max_y \alpha r(x, y) - \mathcal{L}_{\text{lr}}(r; \mathcal{D}), \tag{1.1}$$

where α is a hyperparameter controlling the degree of optimism. The intuition is illustrated in Figure 1. Specifically, minimizing the vanilla reward-fitting loss $\mathcal{L}_{\text{lr}}$ is likely to give a locally accurate RM that overfits the observed data and gets stuck in local minima. Random sampling from this vanilla RM may take a long time to explore the OOD regions that contain the best response. By incorporating the optimism term, we obtain an RM that *both* fits the data well and has a large $\max_y r(x, y)$. This ensures that the greedy response y_u from it is either globally optimal when uncertainty in high-reward regions is eliminated, or potentially good in unexplored areas where $r(x, y_u)$ can be arbitrarily huge due to the relaxed reward-fitting loss. Feedback from humans on these responses y_u can then reduce uncertainty and train a more accurate RM.

In this paper, we formulate this idea within the context of online *direct* alignment, where the LLM is iteratively updated without a separate RM. We first introduce two modifications to the bilevel RM objective in 1.1, namely adding KL constraints and using relative maximum reward. Then we derive a simple LLM training objective by applying the closed-form solution of the inner-level problem and reparameterizing the reward with the LLM policy. The resulting iterative algorithm is called *Self-Exploring Language Models* (SELM). We show that the policy gradient of SELM is biased towards more rewarding areas. Furthermore, by reducing the chance of generating responses that are assigned low implicit rewards, SELM mitigates the *indiscriminate* favoring of unseen extrapolations found in DPO [48, 47] and enhances exploration efficiency.

In experiments, we implement SELM using Zephyr-7B-SFT [56] and Llama-3-8B-Instruct [37] as base models. By finetuning solely on the UltraFeedback [11] dataset and using the small-sized PairRM [25] for iterative AI feedback, SELM boosts the performance of Zephyr-7B-SFT and Llama-3-8B-Instruct by a large margin on AlpacaEval 2.0 [14] (+16.24% and +11.75% LC win rates) and MT-Bench [68] (+2.31 and +0.32). SELM also demonstrates strong performance on standard academic benchmarks and achieves higher pairwise LC win rates against the iterative DPO baseline.

2 Background

2.1 Large Language Models

A language model $\pi \in \Delta_{\mathcal{Y}}^{\mathcal{X}}$ typically takes the prompt $x \in \mathcal{X}$ as input and outputs the response $y \in \mathcal{Y}$. Here, $\mathcal{X}$ and $\mathcal{Y}$ are finite spaces of prompts and responses, respectively. Given the prompt $x \in \mathcal{X}$, a discrete probability distribution $\pi(\cdot \mid x) \in \Delta_{\mathcal{Y}}$ is generated, where $\Delta_{\mathcal{Y}}$ is the set of discrete distributions over $\mathcal{Y}$. Modern recipes for training LLMs consist of pre-training and post-training procedures, where during pre-training, LLMs learn to predict the next word on a huge and diverse dataset of text sequences in order to understand the underlying patterns and structures of natural language in an unsupervised manner. The post-training procedure aims to align better to end tasks and human preferences with two phases happening in order: Supervised Fine-Tuning (SFT) and human preference alignment. Here, SFT fine-tunes the pre-trained LLM with supervised learning on high-quality data to follow instructions on downstream tasks and obtain a model π^{SFT}. In the following of this paper, we focus mainly on preference alignment.

2.2 Reward Modeling and Preference Optimization

Reinforcement Learning from Human Feedback (RLHF). Standard RLHF frameworks consist of learning a reward model and then optimizing the LLM policy using the learned reward.

Specifically, a point-wise reward $r(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ represents the Elo score [16] of the response y given the prompt x. Then the preference distribution can be expressed by the Bradley-Terry model that distinguishes between the preferred response y_w and the dispreferred response y_l given prompt x, denoted as $y_w \succ y_l \mid x$, using the logistic function σ:

$$\begin{aligned} p(y_w \succ y_l \mid x) &:= \mathbb{E}_h\big[\mathbb{1}(h \text{ prefers } y_w \text{ over } y_l \text{ given } x)\big] \\ &= \sigma\big(r(x, y_w) - r(x, y_l)\big) = \frac{\exp\big(r(x, y_w)\big)}{\exp\big(r(x, y_w)\big) + \exp\big(r(x, y_l)\big)}, \end{aligned} \tag{2.1}$$

where h denotes the human rater and the expectation is over h to account for the randomness of the choices of human raters we ask for their preference. When provided a static dataset of N comparisons $\mathcal{D} = \{x_i, y_{w,i}, y_{l,i}\}_{i=1}^N$, the parameterized reward model can be learned by minimizing the following logistic regression loss:

$$\mathcal{L}_{\text{lr}}(r; \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\big[\log \sigma\big(r(x, y_w) - r(x, y_l)\big)\big]. \tag{2.2}$$

Using the learned reward, the LLM policy $\pi \in \Delta_{\mathcal{Y}}^{\mathcal{X}}$ is optimized with reinforcement learning (RL) to maximize the expected reward while maintaining a small deviation from some base reference policy π_{ref}, i.e., maximizing the following objective

$$\mathcal{J}(\pi; \mathcal{D}) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(\cdot|x)}\big[r(x, y)\big] - \beta \mathbb{D}_{\text{KL}}(\pi \,||\, \pi_{\text{ref}}), \tag{2.3}$$

where β is a hyperparameter and $\mathbb{D}_{\text{KL}}(\pi \,||\, \pi_{\text{ref}}) := \mathbb{E}_{x \sim \mathcal{D}}[\text{KL}(\pi(\cdot \mid x) \,||\, \pi_{\text{ref}}(\cdot \mid x))]$ is the expected Kullback-Leibler (KL) divergence. An ideal π_{ref} is the policy that helps mitigate the distribution shift issue [48, 21] between the true preference distribution and the policy π during the off-policy RL training. Since we only have access to the dataset $\mathcal{D}$ sampled from the unavailable true preference distribution, π_{ref} can be obtained by fine-tuning on the preferred responses in $\mathcal{D}$ or simply setting $\pi_{\text{ref}} = \pi^{\text{SFT}}$ and performing RLHF based on the SFT model.

Direct Alignment from Preference. With the motivation to get rid of a separate reward model, which is computationally costly to train, recent works [48, 4, 67, 56, 17] derived the preference loss as a function of the policy by changing of variables. Among them, DPO [48] shows that when the BT model in (2.1) can perfectly fit the preference, the global optimizers of the RLHF objective in (2.3) and the following loss are equivalent:

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

3 Self-Exploring Language Models

3.1 RM-Free Objective for Active Exploration

In this section, we present several modifications to the optimistically biased objective (1.1) motivated in the introduction. Then we derive an RM-free objective for the LLM policy and analyze how active exploration works by examining its gradient.

First, we consider the equivalence of (1.1): $\max_r -\mathcal{L}_{\text{lr}}(r; \mathcal{D}) + \alpha \max_\pi \mathbb{E}_{y \sim \pi}[r(x, y)]$, where the inner π is deterministic when optimal. To account for the change of π relative to the reference policy π_{ref}, we introduce two modifications: (1) replacing the optimistic bias term $\max_\pi \mathbb{E}_{y \sim \pi}[r(x, y)]$ with $\max_\pi \mathbb{E}_{y \sim \pi, y' \sim \pi_{\text{ref}}}[r(x, y) - r(x, y')]$, and (2) incorporating a KL-divergence loss term between π and π_{ref}. These changes ensure that the resulting optimistic RM elicits responses with high potential unknown to the reference policy π_{ref} while minimizing the deviation between π and π_{ref}.

Formally, for the reward function r, the bilevel optimization problem with optimism is formulated as:

$$\max_r -\mathcal{L}_{\text{lr}}(r; \mathcal{D}_t) + \alpha \max_\pi \underbrace{\left(\mathbb{E}_{\substack{x \sim \mathcal{D}_t, y \sim \pi(\cdot|x) \\ y' \sim \pi_{\text{ref}}(\cdot|x)}} \Big[r(x, y) - r(x, y') \Big] - \beta \mathbb{D}_{\text{KL}}(\pi \,||\, \pi_{\text{ref}}) \right)}_{\mathcal{F}(\pi; r)}, \tag{3.1}$$

where $\mathcal{D}_t = \{x_i, y_{w,i}^t, y_{l,i}^t\}_{i=1}^N$ is the associated dataset at iteration t and $\mathcal{L}_{\text{lr}}$ is the logistic regression loss defined in (2.2). The nested optimization in (3.1) can be handled by first solving the inner optimization $\mathcal{F}(\pi; r)$ to obtain π_r that is optimal under r. The solution is as follows and we defer all the derivations in this section to Appendix A.

$$\pi_r(y \mid x) := \operatorname*{argmax}_\pi \mathcal{F}(\pi; r) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\big(r(x, y)/\beta\big),$$

where the partition function $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp(r(x, y)/\beta)$. By substituting $\pi = \pi_r$ into $\mathcal{F}(\pi; r)$, we can rewrite the bilevel objective in (3.1) as a single-level one:

$$\max_r -\mathcal{L}_{\text{lr}}(r; \mathcal{D}_t) + \alpha \mathcal{F}(\pi_r; r).$$

Following the implicit reward formulation in DPO, we reparameterize the reward function with $\theta \in \Theta$ as $\widehat{r}_\theta(x, y) = \beta(\log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x))$, which is the optimal solution of (2.3) and can express *all* reward classes consistent with the BT model as proved in [48]. With this change of variable, we obtain the RM-free objective for direct preference alignment with optimism:

$$\max_{\pi_\theta} -\mathcal{L}_{\text{DPO}}(\pi_\theta; \mathcal{D}_t) - \alpha\beta \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\text{ref}}(\cdot|x)} \big[\log \pi_\theta(y \mid x)\big]. \tag{3.2}$$

We now analyze how this new objective encourages active exploration. Specifically, we derive the gradient of (3.2) with respect to θ as

$$\underbrace{-\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}_t} \Big[\sigma\big(\widehat{r}_\theta(x, y_l) - \widehat{r}_\theta(x, y_w)\big) \big(\nabla_\theta \log \pi_\theta(y_w \mid x) - \nabla_\theta \log \pi_\theta(y_l \mid x)\big) \Big]}_{\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \mathcal{D}_t)}$$
$$- \alpha\beta \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \big[\exp\big(-\widehat{r}_\theta(x, y)/\beta\big) \nabla_\theta \log \pi_\theta(y \mid x) \big]. \tag{3.3}$$

We note that the second line, corresponding to the gradient of the optimism term, decreases the log-likelihood of response y generated by π_θ that has a low value of $\exp(-\widehat{r}_\theta(x, y)/\beta)$. Therefore, the added optimism term biases the gradient toward parameter regions that can elicit responses y with high implicit reward $\widehat{r}_\theta$, consistent with our intuition outlined in Figure 1.

This also explains why $\mathbb{E}_{\pi_{\text{ref}}}[\log \pi_\theta]$ is minimized in our objective (3.2), which is equivalent to maximizing the KL divergence between π_{ref} and π_θ, while the reverse KL in the policy optimization objective (2.3) is minimized. For the DPO gradient $\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \mathcal{D}_t)$, the degree of deviation of policy π_θ from π_{ref} only affects the preference estimated with $\widehat{r}_\theta$. In other words, $\sigma(\widehat{r}_\theta(x, y_l) - \widehat{r}_\theta(x, y_w))$ is a scalar value and the policy deviation only determines the *step size* of the policy gradient, instead of its *direction*. On the other hand, our added exploration term directly controls the direction of the gradient toward potentially more rewarding areas while still fitting the preference data in $\mathcal{D}_t$. As more feedback data is collected iteratively, deviating from the unbiasedly fitted model incurs a higher DPO loss, which ultimately dominates our objective at convergence. This mechanism ensures that the resulting LLM effectively balances between exploring novel responses and exploiting previously observed ones, leading to a more accurate and aligned model.

3.2 Algorithm

With the optimistically biased objective derived above, the language model can actively generate OOD responses worth exploring. Human or AI feedback follows to reduce the uncertainty in these regions. These two steps are executed iteratively to get a more and more aligned model.

In practice, we split the offline preference dataset into three portions with equal sizes, one for each iteration. Besides, we use AI rankers, such as external RMs, to provide feedback on the model-generated response and the original chosen, rejected responses. The complete pseudocode of our algorithm, named *Self-Exploring Language Models* (SELM), is outlined in Algorithm 1.

Algorithm 1 Self-Exploring Language Models (SELM)

Input: Reference model π_{ref}, preference dataset $\mathcal{D}$, online iterations T, optimism coefficient α.

1: **for** iteration $t = 1, 2, \ldots, T$ **do**

2: Set $\mathcal{D}_t$ as the t-th portion of $\mathcal{D}$ and generate $y \sim \pi_{\text{ref}}(\cdot \mid x)$ for each prompt x in $\mathcal{D}_t$.

3: Rank $\{y, y_w, y_l\}$ and update $\mathcal{D}_t$ to contain the best (chosen) and worst (rejected) responses.

4: Train the LLM $\pi_{\theta_t} = \operatorname{argmax}_{\pi_\theta} -\mathcal{L}_{\text{DPO}}(\pi_\theta; \mathcal{D}_t) - \alpha \mathbb{E}_{x \sim \mathcal{D}_t}[\log \pi_\theta(y \mid x)]$ and let $\pi_{\text{ref}} = \pi_{\theta_t}$.

5: **end for**

3.3 Self-Exploration Reduces Indiscriminate Favor of Unseen Extrapolations

It has been observed recently [47, 45, 62] that DPO decreases the likelihood of responses generated by the reference policy. It is because for any prompt x, at convergence when $\pi_\theta \neq \pi_{\text{ref}}$, it holds that

$$\mathbb{E}_{y \sim \pi_{\text{ref}}}\big[\widehat{r}_\theta(x, y)/\beta\big] = \mathbb{E}_{y \sim \pi_{\text{ref}}}\big[\log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x)\big] = -\text{KL}\big(\pi_{\text{ref}}(\cdot \mid x) \,||\, \pi_\theta(\cdot \mid x)\big) < 0,$$

while at the beginning of training when $\pi_\theta = \pi_{\text{ref}}$, the above terms are zero. Thus, the expected implicit reward $\widehat{r}_\theta$ as well as the likelihood of π_θ will decrease on the reference model's responses. This indicates that DPO stimulates a biased distribution favoring unseen extrapolated responses. In the online iterative setting that we consider, the LLM policy generates responses and receives preference feedback alternately, where biasing towards OOD regions may sometimes help discover outstanding novel responses. However, DPO *indiscriminately* favors unseen extrapolations and *passively* explores based purely on the randomness inherent in sampling from the LLM. As a consequence, the vast space of natural language makes it almost impossible to exhaustively explore all the possible responses and identify those that most effectively benefit alignment.

Next, we demonstrate that SELM mitigates this issue by performing guided exploration. Specifically, consider the proposed self-exploration objective in (3.2), which, in addition to the standard DPO loss, also minimizes $\mathbb{E}_{x, y \sim \pi_{\text{ref}}}[\log \pi_\theta(y \mid x)]$. We now investigate how the probability distribution changes with this term incorporated.

Theorem 3.1. For any $\rho \in \Theta$ in the policy parameter space, let $\widehat{r}_\rho(x, y) = \beta(\log \pi_\rho(y|x) - \log \pi_{\text{ref}}(y|x))$ be the reparameterized implicit reward. Denote $\pi_\rho^{\min}$ as the policy that minimizes the expected implicit reward under the KL constraint, i.e.,

$$\pi_\rho^{\min}(\cdot \mid x) := \operatorname*{argmin}_{\pi} \mathbb{E}_{x, y \sim \pi(\cdot|x)}\big[\widehat{r}_\rho(x, y)\big] + \beta \mathbb{D}_{\text{KL}}(\pi \,||\, \pi_\rho). \tag{3.4}$$

Then minimizing $\mathbb{E}_{x, y \sim \pi_{\text{ref}}}[\log \pi_\theta(y|x)]$ decreases the likelihood of responses sampled from $\pi_\rho^{\min}$:

$$\min_{\pi_\theta} \mathbb{E}_{x, y \sim \pi_{\text{ref}}(\cdot|x)}\big[\log \pi_\theta(y \mid x)\big] = \min_{\pi_\theta} \mathbb{E}_{x, y \sim \pi_\rho^{\min}(\cdot|x)}\big[\log \pi_\theta(y \mid x)\big].$$

The above theorem states that maximizing the divergence between π_θ and π_{ref} is essentially reducing the probability of generating responses with low implicit rewards reparameterized by any policy parameter ρ during training. In other words, the policy not only exploits the existing preference data but also learns to avoid generating the text y that is assigned a low reward value. This process occurs in every iteration with updated reference models. Consequently, responses with high potential rewards are selectively preferred and many commonplace responses receive a small probability mass, thus mitigating the indiscriminate favoring of unseen responses and improving exploration efficiency.

4 Related Work

Data Synthesis for LLMs. A key challenge for fine-tuning language models to align with users' intentions lies in the collection of demonstrations, including both the SFT instruction-following expert data and the RLHF preference data. Gathering such data from human labelers is expensive, time-consuming, and sometimes suffers from variant quality [43, 29]. To address this issue, synthetic data [34] has been used for aligning LLMs. One line of work focuses on generating plausible instruction prompts for unlabeled data by regarding the target output as instruction-following responses [31, 58, 27, 54]. Besides, high-quality data can also be distilled from strong models for fine-tuning weaker ones [20, 1, 32, 12, 46]. To construct synthetic datasets for offline RLHF, a popular pipeline [11, 56, 57, 24, 69] involves selecting responses sampled from *various* LLMs on a set of prompts in the hope to increase the diversity of the data that can span the whole language space. However, data manually collected in such a passive way does not consider what improves the model most through its training, leaving the potentially high-reward regions unexplored.

Iterative Online Preference Optimization Compared to offline RLHF algorithms [48, 67, 4] that collect preference datasets ahead of training, online RLHF [43, 21], especially the iterative/batched online RLHF [5, 61, 19, 22, 60, 6, 49] has the potential to gather better and better synthetic data as the model improves. As a special case, self-alignment language models align their responses with desired behaviors, such as model-generated feedback [64, 65, 52]. Unfortunately, the above methods still passively explore by relying on the randomness during sampling and easily get stuck at local optima and overfit to the current data due to the vast space of natural language. A notable exception is [15], which proposed to use ensembles of RMs to approximately measure the uncertainty for posterior-sampling active exploration. On the contrary, our method explores based on the optimistic bias and does not estimate the uncertainty explicitly, bypassing the need to fit multiple RMs.

Active Exploration. In fact, active exploration has been widely studied beyond LLMs. Similar to [15], most existing sample-efficient RL algorithms first estimate the uncertainty of the environment using historical data and then plan with optimism [3, 50, 26], or selecting the optimal action from a statistically plausibly set of action values sampled from its posterior distribution [51, 40, 41]. The proposed self-exploration objective can be categorized as an optimism-based exploration method. However, most previous works require the estimation of the upper confidence bound, which is often intractable. Ensemble methods [42, 8, 36] can serve as approximations to the uncertainty estimation but are still computationally inefficient. MEX [35] proposed to combine estimation and planning in a single objective similar to ours and established theoretical guarantees under traditional RL setups.

5 Experiments

5.1 Experiment Setup

We select the UltraFeedback [11] dataset as our training set, which contains 61k preference pairs of single-turn conversations. For the ranker providing AI feedback during online alignment, we choose the small-sized PairRM (0.4B) [25]. All experiments are conducted on 8xA100 GPUs.

Due to the absence of performant open-source online direct alignment codebases at the time of this study, we first implement an iterative version of DPO as the baseline, adhering to the same steps as Algorithm 1 but training the LLM with the standard DPO objective. Then we conduct a grid search over hyperparameters, such as the batch size, learning rate, and iteration number, to identify the optimal settings for the iterative DPO baseline. Details on the hyperparameters and grid search results are provided in Appendix C. We follow these best settings to train SELM for a fair comparison. In addition, the top choice for the base models of SELM are LLMs that are finetuned with RLHF after SFT, since they are typically more capable than SFT-only and pertrained models. We consider two series of LLMs: Zephyr [56] and Llama-3 [37], to demonstrate the robustness of SELM. Since the official Zephyr-7B-β model is finetuned with DPO on the same UltraFeedback dataset, to avoid overoptimization, we choose Zephyr-7B-SFT[1] as the base model and perform 3 iterations of SELM after a single iteration of standard DPO training on the first portion of the training data (we refer to this model as Zephyr-7B-DPO). For Llama-3-8B-Instruct[2] that is already finetuned with RLHF, we directly apply 3 iterations of SELM training.

5.2 Experiment Results

We first report the performance of SELM and the baselines on the instruction-following chat benchmarks AlpacaEval 2.0 [14] and MT-Bench [68] in Table 1. We can observe that for AlpacaEval 2.0, SELM significantly boosts Zephyr-7B-SFT and Llama-3-8B-Instruct, achieving length-controlled (LC) win rate improvements of $+16.24\%$ and $+11.75\%$, respectively. This enhancement results in models that are competitive with or even superior to much larger LLMs, such as Yi-34B-Chat [63] and Llama-3-70B-Instruct. For the multi-turn MT-Bench, which exhibits higher variance, we report

[1]https://huggingface.co/HuggingFaceH4/mistral-7b-sft-beta

[2]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Model	AlpacaEval 2.0			MT-Bench		
	LC Win Rate	Win Rate	Avg. len	Avgerage	1st Turn	2nd Turn
Zephyr-7B-SFT	8.01	4.63	916	5.30	5.63	4.97
Zephyr-7B-DPO	15.41	14.44	1752	7.31	7.55	7.07
DPO Iter 1 (Zephyr)	20.53	16.69	1598	7.53	7.81	7.25
DPO Iter 2 (Zephyr)	22.12	19.82	1717	7.55	7.85	7.24
DPO Iter 3 (Zephyr)	22.19 (↑14.18)	19.88	1717	7.46 (↑2.16)	7.85	7.06
SELM Iter 1 (Zephyr)	20.52	17.23	1624	7.53	7.74	7.31
SELM Iter 2 (Zephyr)	21.84	18.78	1665	7.61	**7.85**	7.38
SELM Iter 3 (Zephyr)	**24.25**(↑16.24)	**21.05**	1694	**7.61** (↑2.31)	7.74	**7.49**
Llama-3-8B-Instruct	22.92	22.57	1899	7.93	8.47	7.38
DPO Iter 1 (Llama3-It)	30.89	31.60	1979	8.07	8.44	7.70
DPO Iter 2 (Llama3-It)	33.91	32.95	1939	7.99	8.39	7.60
DPO Iter 3 (Llama3-It)	33.17 (↑10.25)	32.18	1930	8.18 (↑0.25)	8.60	7.77
SELM Iter 1 (Llama3-It)	31.09	30.90	1956	8.09	8.57	7.61
SELM Iter 2 (Llama3-It)	33.53	32.61	1919	8.18	**8.69**	7.66
SELM Iter 3 (Llama3-It)	**34.67** (↑11.75)	**34.78**	1948	**8.25** (↑0.32)	8.53	**7.98**
SPIN	7.23	6.54	1426	6.54	6.94	6.14
Orca-2.5-SFT	10.76	6.99	1174	6.88	7.72	6.02
DNO (Orca-2.5-SFT)	22.59	24.97	2228	7.48	7.62	7.35
Mistral-7B-Instruct-v0.2	19.39	15.75	1565	7.51	7.78	7.25
SPPO (Mistral-it)	28.53	31.02	2163	7.59	7.84	7.34
Yi-34B-Chat	27.19	21.23	2123	7.90	-	-
Llama-3-70B-Instruct	33.17	33.18	1919	9.01	9.21	8.80
GPT-4 Turbo (04/09)	55.02	46.12	1802	9.19	9.38	9.00

Table 1: Results on AlpacaEval 2.0 and MT-Bench. Names inside the brackets are the base models that are aligned based upon. The red arrows indicate the increment or decrement from the base model. Compared to iterative DPO and other online alignment baselines, SELM gains more improvements based on the weaker Zephyr-7B-SFT model and achieves superior performance that is competitive with much larger SOTA models when finetuned from Llama-3-8B-Instruct.

the average scores of SELM and DPO baselines across 3 runs. We observe that SELM improves the scores by $+2.31$ and $+0.32$, respectively. Furthermore, the proposed method self-explores and enhances the model monotonically, with consistent performance improvements in each iteration. This validates the robustness of our algorithm. Compared to other iterative post-training algorithms, such as SPIN [7], DNO [49], and SPPO [59], SELM gains more improvements on both benchmarks when using the weaker base model (Zephyr-7B-SFT), and achieves the best performance when using Llama-3-8B-Instruct as the base model.

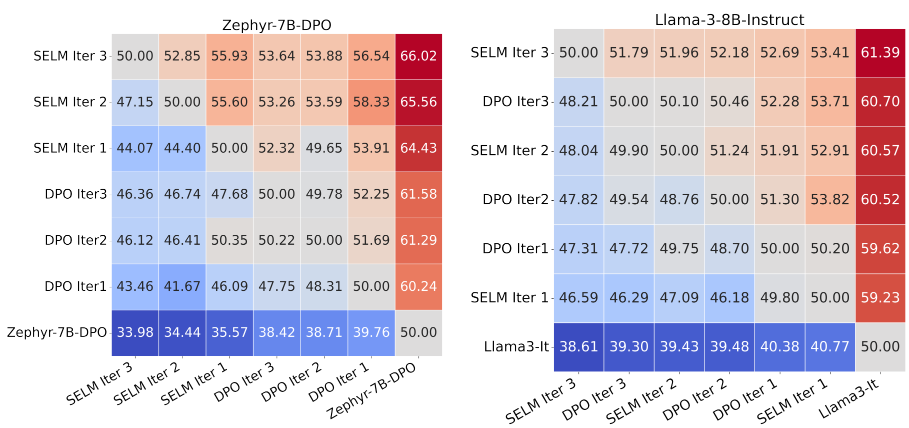

Figure 2: Pairwise length-controlled win rates comparison between SELM, iterative DPO, and base models on the AlpacaEval 2.0 benchmark. Scores represent the LC win rates of the row models against the column models. Models positioned in higher rows have higher LC win rates against the base model and thus better performance.

We also conduct pairwise comparisons between SELM, iterative DPO, and the base models to validate the effectiveness of our method. The results for AlpacaEval 2.0 are shown in Figure 2. We observe

Models	GSM8K (8-s CoT)	HellaSwag (10-s)	ARC (25-s)	TruthfulQA (0-s)	EQ (0-s)	OBQA (10-s)	Average
Zephyr-7B-SFT	43.8	82.2	57.4	43.6	39.1	35.4	50.3
Zephyr-7B-DPO	47.2	84.5	61.9	45.5	65.2	38.0	57.0
DPO Iter 1 (Zephyr)	45.5	85.2	62.1	52.4	68.4	39.0	58.8
DPO Iter 2 (Zephyr)	44.9	85.4	62.0	53.1	69.3	39.4	59.0
DPO Iter 3 (Zephyr)	43.2	85.2	60.8	52.5	69.1	39.6	58.4
SELM Iter 1 (Zephyr)	46.3	84.8	62.9	52.9	68.8	39.6	59.2
SELM Iter 2 (Zephyr)	46.2	85.4	62.1	53.1	69.3	39.6	59.3
SELM Iter 3 (Zephyr)	43.8	85.4	61.9	52.4	69.9	39.8	58.9
Llama-3-8B-Instruct	76.7	78.6	60.8	51.7	61.8	38.0	61.3
DPO Iter 1 (Llama3-It)	78.5	81.7	63.9	55.5	64.1	42.6	64.4
DPO Iter 2 (Llama3-It)	79.4	81.7	64.4	56.4	64.3	42.6	64.8
DPO Iter 3 (Llama3-It)	80.1	81.7	64.1	56.5	64.1	42.6	64.8
SELM Iter 1 (Llama3-It)	78.7	81.7	64.5	55.4	64.1	42.4	64.5
SELM Iter 2 (Llama3-It)	79.3	81.8	64.7	56.5	64.2	42.6	64.9
SELM Iter 3 (Llama3-It)	80.1	81.8	64.3	56.5	64.2	42.8	65.0
SPIN	44.7	85.9	65.9	55.6	54.4	39.6	57.7
Mistral-7B-Instruct-v0.2	43.4	85.3	63.4	67.5	65.9	41.2	61.1
SPPO (Mistral-it)	42.4	85.6	65.4	70.7	56.5	40.0	60.1

Table 2: Performance comparison between SELM and the baselines on academic multi-choice QA benchmarks in standard zero-shot, few-shot, and CoT settings. Here, n-s refers to n-shot. The red and blue texts represent the best and the second-best results.

that with the same number of training iterations and data, SELM consistently outperforms the iterative DPO counterpart. Additionally, when using Zephyr-7B-SFT as the base model, SELM outperforms iterative DPO even when the latter is trained with twice the data.

Beyond instruction-following benchmarks, we also evaluate SELM and the baselines on several academic benchmarks, including GSM8K [10], HellaSwag [66], ARC challenge [9], TruthfulQA [33], EQ-Bench [44], and OpenBookQA (OBQA) [38]. To better reflect the capabilities of LLMs, we adopt various settings for these benchmarks, including zero-shot, few-shot, and few-shot Chain-of-Thought (CoT) settings. The accuracy results for these multiple-choice QA benchmarks are provided in Table 2. It can be observed that both our method and the baselines can degrade after the RLHF phase on some benchmarks, which is known as the alignment tax [2, 39, 30]. Nevertheless, our method is still able to improve the base models on most of the benchmarks and offers the best overall performance.

We note that SELM is one of the instantiations of the proposed self-exploration objective in (1.1), with reparameterized reward functions and algorithm-specific designs described in Section 3.2, such as the dataset partition and update rule. However, this objective is not restricted to the current implementation and can also be directly applied to any other online alignment framework, with or without a separate reward model, regardless of differences in algorithm designs. Thus, the proposed method is orthogonal to and can be integrated directly into the recent online RLHF workflows [13, 60, 23] that incorporate additional delicate designs with carefully curated datasets.

5.3 Ablation Study

We first provide ablation studies to better understand the explorative optimism term. We begin by investigating the effect of the optimism coefficient α. In Figure 3 (Left), we plot the LC win rates of SELM when using Zephyr-7B-SFT as the base model for different α in the AlpacaEval 2.0 benchmark. We find that setting a small α, such as 0.0001, leads to very similar behaviors to the iterative DPO ($\alpha = 0$) baseline, while SELM with a large α may become overly optimistic and thus not very effective. These results meet our expectations, suggesting that proper values of α are essential for achieving the best trade-off between exploration and exploitation.

Next, we study the difference in reward distributions with varying α and iterations. Specifically, we greedily sample from the LLM using prompts from the holdout test set (2k in total) of UltraFeedback and generate rewards for these responses with PairRM. We then calculate the fraction of data that lies in each partition of reward values. The results for different α values of SELM Iter 2 (Zephyr) are shown in Figure 3 (Middle), which indicate that increasing α results in distributions that are more concentrated in higher-reward regions.

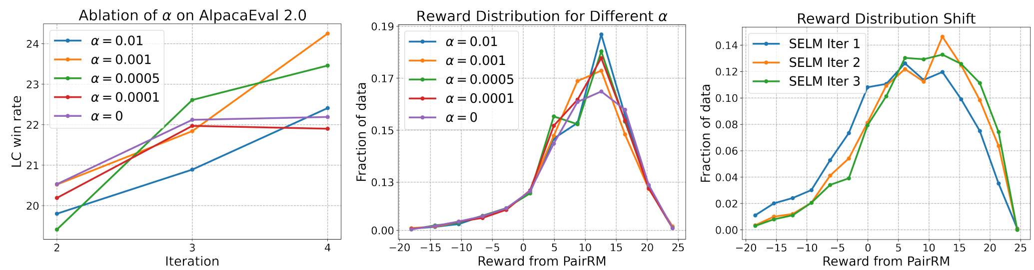

Figure 3: Ablation on the optimism coefficient α and the change of the reward distribution. **Left:** The length-controlled win rates of SELM with different α on AlpacaEval 2.0. **Middle:** Comparison of reward distributions at iteration 2 with different α. **Right:** SELM initially explores and then shifts to higher-reward regions as more training iterations are performed.

Additionally, Figure 3 (Right) demonstrates that the reward distribution shifts to the right (higher) as more training iterations are performed. This shift corresponds to an initial exploration phase, where the LLM generates uncertain responses of varying quality, followed by an exploitation phase as feedback is incorporated and more training data is collected.

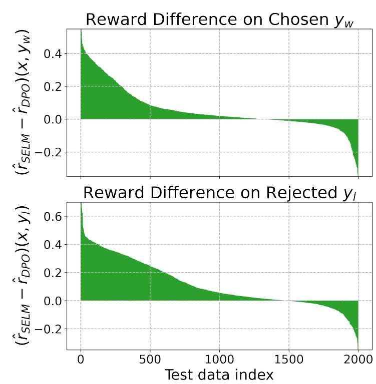

Figure 4: Difference of implicit reward between SELM and DPO on the chosen and rejected responses. SELM assigns a higher implicit reward than DPO for both responses.

We also conduct ablation studies on the implicit reward captured by the SELM and DPO models. Recall that for both SELM and DPO, the implicit reward takes the form of $\widehat{r}_\theta(x, y) = \beta(\log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x))$. We calculate the reward difference $\widehat{r}_{\text{SELM}}(x, y) - \widehat{r}_{\text{DPO}}(x, y)$ for each prompt x in the UltraFeedback holdout test set. Here, we study the implicit reward of the good (chosen) and bad (rejected) responses, so $y = y_w$ or $y = y_l$. We then sort the reward difference and plot the results for Zephyr-based models after iteration 1 in Figure 4. The plot clearly shows that for both chosen and rejected responses, SELM produces higher *implicit* rewards compared to DPO, aligning with the proposed optimistically biased self-exploration objective.

6 Conclusion & Future Work

In this paper, we introduced an active preference elicitation method for the online alignment of large language models. By incorporating an optimism term into the reward-fitting objective, the proposed bilevel self-exploring objective effectively balances between exploiting observed data and exploring potentially high-reward regions. Unlike standard online RLHF algorithms that passively explore the response space by sampling from the training LLM, whose sole objective is maximizing the expected learned reward, our method actively seeks diverse and high-quality responses. This self-exploration mechanism helps mitigate the risk of premature convergence and overfitting when the reward model is only locally accurate. To optimize this bilevel objective, we solve the inner-level problem and reparameterize the reward with the LLM policy, resulting in a simple yet novel iterative alignment algorithm called *Self-Exploring Language Models* (SELM). Compared to DPO, SELM improves the exploration efficiency by selectively favoring responses with high potential rewards rather than indiscriminately sampling unseen responses.

Our experiments, conducted with Zephyr-7B-SFT and Llama-3-8B-Instruct models, demonstrated the efficacy of SELM. Finetuning on the UltraFeedback dataset and leveraging PairRM for AI feedback, SELM achieved substantial improvements in performance on AlpacaEval 2.0, MT-Bench, and academic benchmarks. These results underscore the ability of SELM to enhance the alignment and capabilities of large language models by promoting more diverse and high-quality responses. Since the proposed technique is orthogonal to the adopted online RLHF workflow, it will be interesting to apply our method within more sophisticated alignment frameworks with advanced designs, which we would like to leave as future work.

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

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

[5] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[6] Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*, 2024.

[7] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[8] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

[12] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[13] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv e-prints*, pages arXiv–2405, 2024.

[14] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[15] Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient exploration for llms. *arXiv preprint arXiv:2402.00396*, 2024.

[16] Arpad E Elo and Sam Sloan. The rating of chessplayers: Past and present. *Ishi Press International*, 1978.

[17] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[18] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[19] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

[20] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[21] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

[22] Braden Hancock Hoang Tran, Chris Glaze. Snorkel-mistral-pairrm-dpo. 2024.

[23] Jian Hu, Xibin Wu, Weixun Wang, Xianyu, Dehao Zhang, and Yu Cao. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework, 2024.

[24] Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*, 2023.

[25] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

[26] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, pages 2137–2143. PMLR, 2020.

[27] Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. *arXiv preprint arXiv:2303.04132*, 2023.

[28] Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. sdpo: Don't use your data all at once. *arXiv preprint arXiv:2403.19270*, 2024.

[29] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] Shengzhi Li, Rongyu Lin, and Shichao Pei. Multi-modal preference alignment remedies regression of visual instruction tuning on language model. *arXiv preprint arXiv:2402.10884*, 2024.

[31] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*, 2023.

[32] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[33] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[34] Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. Best practices and lessons learned on synthetic data for language models, 2024.

[35] Zhihan Liu, Miao Lu, Wei Xiong, Han Zhong, Hao Hu, Shenao Zhang, Sirui Zheng, Zhuoran Yang, and Zhaoran Wang. Maximize to explore: One objective function fusing estimation, planning, and exploration. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. *Advances in neural information processing systems*, 30, 2017.

[37] Meta. Introducing meta llama 3: The most capable openly available llm to date. 2024.

[38] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[39] Michael Noukhovitch, Samuel Lavoie, Florian Strub, and Aaron C Courville. Language model alignment with elastic reset. *Advances in Neural Information Processing Systems*, 36, 2024.

[40] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

[41] Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Approximate thompson sampling via epistemic neural networks. In *Uncertainty in Artificial Intelligence*, pages 1586–1595. PMLR, 2023.

[42] Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[43] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[44] Samuel J Paech. Eq-bench: An emotional intelligence benchmark for large language models. *arXiv preprint arXiv:2312.06281*, 2023.

[45] Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

[46] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

[47] Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024.

[48] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[50] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

[51] Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

[52] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024.

[53] Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024.

[54] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[55] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. The alignment handbook. https://github.com/huggingface/alignment-handbook, 2023.

[56] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

[57] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*, 2023.

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

[60] Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*, 2023.

[61] Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

[62] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

[63] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

[64] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

[65] Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *arXiv e-prints*, pages arXiv–2404, 2024.

[66] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

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

[68] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[69] Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness and harmlessness with rlaif, November 2023.

A Derivations in Section 3.1

We begin by deriving (3.2). The solution for the inner-level optimization problem of (3.1) is as follows:

$$\begin{aligned}\max_{\pi} \mathcal{F}(\pi; r) &= \max_{\pi} \mathbb{E}_{\substack{x\sim\mathcal{D}_t, y\sim\pi(\cdot|x)\\ y'\sim\pi_{\text{ref}}(\cdot|x)}}\Big[r(x,y) - r(x,y')\Big] - \beta\mathbb{D}_{\text{KL}}(\pi \,||\, \pi_{\text{ref}}) \\ &= \mathbb{E}_{x\sim\mathcal{D}_t}\Big[\beta \log \mathbb{E}_{y\sim\pi_{\text{ref}}(\cdot|x)}\big[\exp(r(x,y)/\beta)\big]\Big] - \mathbb{E}_{x\sim\mathcal{D}_t, y'\sim\pi_{\text{ref}}(\cdot|x)}\big[r(x,y')\big] \end{aligned} \tag{A.1}$$

When the reward r is reparameterized by $\widehat{r}_\theta(x,y) = \beta(\log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x))$, we have that the first term in (A.1) is 0. The bilevel objective (3.1) then becomes

$$\max_{r} -\mathcal{L}_{\text{lr}}(r; \mathcal{D}_t) - \alpha \mathbb{E}_{x\sim\mathcal{D}, y'\sim\pi_{\text{ref}}(\cdot|x)}\big[r(x,y')\big].$$

By reparameterizing the reward with the LLM, we obtain the desired results in (3.2).

Then we provide the derivation of (3.3). We primarily consider the gradient of the newly incorporated term $\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_{\text{ref}}(\cdot|x)}[\log \pi_\theta(y \mid x)]$. Specifically, we have

$$\begin{aligned}\nabla_\theta \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_{\text{ref}}(\cdot|x)}\big[\log \pi_\theta(y \mid x)\big] &= \mathbb{E}_{x\sim\mathcal{D}}\Big[\sum_y \pi_{\text{ref}}(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)\Big] \\ &= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\Big[\frac{\pi_{\text{ref}}(y \mid x)}{\pi_\theta(y \mid x)} \nabla_\theta \log \pi_\theta(y \mid x)\Big] \\ &= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\Big[\exp\big(-\widehat{r}_\theta(x.y)/\beta\big) \nabla_\theta \log \pi_\theta(y \mid x)\Big].\end{aligned}$$

For the derivation of the DPO gradient $\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \mathcal{D}_t)$, we refer the readers to [48].

B Proof of Theorem 3.1

Proof. The solution to the KL-constrained reward minimization objective (3.4) is

$$\pi_\rho^{\min}(y \mid x) = \pi_\rho(y \mid x) \exp\big(-\widehat{r}_\rho(x,y)/\beta\big)/Z(x),$$

where $Z(x) = \sum_y \pi_\rho(y \mid x) \exp(-\widehat{r}_\rho(x,y)/\beta) = 1$. Then we have $\pi_\rho^{\min}(y \mid x) = \pi_{\text{ref}}(y \mid x)$, i.e., the reference policy π_{ref} achieves the lowest implicit reward reparameterized by any ρ. $\square$

C Experiment Setup

In experiments, we use the Alignment Handbook [55] framework as our codebase. We find the best hyperparameter settings by conducting a grid search over the iteration number, batch size, learning rate, and label update rule for the iterative DPO baseline. The results for the Zephyr-based models are shown in Figure 5. Specifically, we find that using the same amount of data, updating the model too many iterations can lead to instability. So we set the iteration number to 3 for Llama3-It-based and Zephyr-based models (excluding the first iteration of DPO training). Besides, we observe that choosing different batch sizes has a large effect on the models' performance and the optimal batch size heavily depends on the model architecture. In experiments, we set the batch size to 256 and 128 for the Zephyr-based and Llama3-It-based models, respectively. For the learning rate, we consider three design choices: cyclic learning rate with constant cycle amplitude, linearly decayed cycle amplitude, and decayed cycle amplitude at the last iteration. We find that a decaying cycle amplitude performs better than constant amplitudes in general. Thus, for Zephyr-based models, we set the learning to $5e-7$ for the first three iterations and $1e-7$ for the last iteration. In each iteration, the warmup ratio is 0.1. For Llama3-It-based models, we use a linearly decayed learning rate from $5e-7$ to $1e-7$ within 3 iterations with the same warmup ratio. We also test two update ways for the preference data. One is to rank y_w, y_l, y_{ref} and keep the best and worst responses in the updated dataset, which is the setting that is described in the main paper. The other is to compare y_w and y_{ref} and replace the chosen or rejected response by y_{ref} based on the comparison result. We find that the former design performs

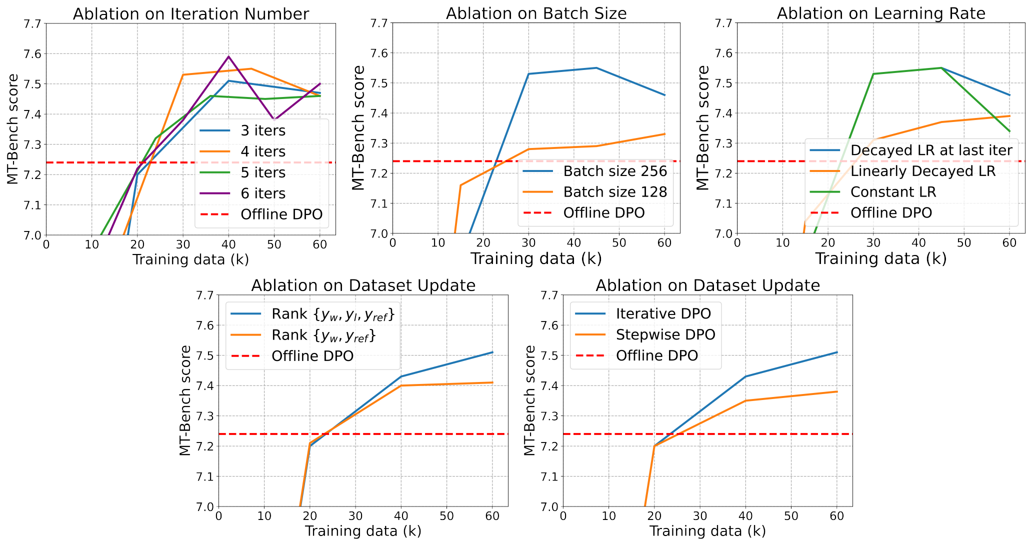

Figure 5: Ablation of the iterative DPO baseline. We conduct a grid search over the iteration number, batch size, learning rate, and designs of the dataset update rule.

better than the latter. We also compared with stepwise DPO [28], which updates the reference model at each iteration but uses the original dataset instead of the updated one. This demonstrates that exploring and collecting new data is necessary.

For the proposed SELM method, we follow the above hyperparameter settings for a fair comparison. The optimism coefficient α is searched over 0.005, 0.001, 0.0005, and 0.0001 and is selected based on the average external reward on the holdout test set of UltraFeedback. We set $\alpha = 0.001$ for Zephyr-based SELM and $\alpha = 0.0001$ for Llama3-It-based SELM. For training SELM based on other models, we recommend setting $\alpha = 0.005$ or 0.001 as it shows minimal sensitivity to variations.