ActiveDPO: Active Direct Preference Optimization for Sample-Efficient Alignment

Anonymous authors
Paper under double-blind review

Abstract

The recent success in using human preferences to align large language models (LLMs) has significantly improved their performance in various downstream tasks, such as question answering, mathematical reasoning, and code generation. However, achieving effective LLM alignment depends on high-quality human preference datasets. Collecting these datasets requires human preference annotation, which is costly and resource-intensive, necessitating efficient active data selection methods. Existing methods either lack a strong theoretical foundation or depend on restrictive reward function assumptions, such as linear latent reward functions. To this end, we propose an algorithm, `ActiveDPO`, that uses a theoretically grounded data selection criterion for non-linear reward functions while directly leveraging the LLM itself to parameterize the reward model that is used for active data selection. As a result, `ActiveDPO` explicitly accounts for the influence of the LLM on data selection, unlike methods that select the data without considering the LLM that is being aligned, thereby leading to more effective and efficient data collection. Our extensive experiments demonstrate that `ActiveDPO` outperforms existing methods across various models and real-life preference datasets.

1 Introduction

Large language models (LLMs) (Google, 2023; OpenAI, 2023; Touvron et al., 2023; Anthropic, 2023) have demonstrated impressive performance across various tasks, including question-answering (Taori et al., 2023), mathematical reasoning (Wei et al., 2022), code generation (Chen et al., 2021), and many others (Zhao et al., 2023). However, LLMs often fall short when required to produce responses that conform to specific formats or align with human values (Ji et al., 2023; Anwar et al., 2024). To address this, methods such as Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023), use binary preference feedback collected from human annotators, who indicate which of two LLM responses they prefer, to better align LLM outputs with human preferences in real-world applications. Both RLHF and DPO require high-quality human preference datasets to achieve effective LLM alignment. However, collecting these datasets requires skilled human annotators, making this process both costly and resource-intensive (Liu et al., 2024; Carvalho Melo et al., 2024; Muldrew et al., 2024).

To overcome these challenges, recent works (Mehta et al., 2023; Das et al., 2024; Liu et al., 2024; Muldrew et al., 2024) have proposed methods for actively selecting a smaller subset of preference data (i.e., triplets consisting of a prompt and two responses) for human preference annotation while maintaining alignment performance. Specifically, some existing works (Liu et al., 2024; Muldrew et al., 2024) have proposed heuristic methods for actively selecting preference data to collect human preference feedback. However, these methods lack a rigorous theoretical foundation and therefore do not guarantee reliable performance across different tasks and LLMs (see Fig. 1 in Section 4). In contrast, some works (Mehta et al., 2023; Das et al., 2024) have developed methods with theoretical guarantees to achieve sample-efficient LLM alignment. However, these methods require strong assumptions about the underlying latent reward function (e.g., linearity), which may not hold in the context of LLM alignment. Furthermore, another potential limitation of some existing works (Mehta et al., 2023; Das et al., 2024; Liu et al., 2024; Muldrew et al., 2024) is their dependence on a separate reward model or a selection method that works independently of the LLM being aligned.

These limitations naturally lead to the following question: ***How can we develop an active preference data selection algorithm that is both theoretically grounded and practically effective?*** To answer this, we propose `ActiveDPO`, a novel active preference data selection algorithm. `ActiveDPO` is built on DPO, which has shown comparable or superior empirical performance to RLHF while avoiding the complexity of reward model training and the reinforcement learning process, making it a compelling choice for aligning LLMs with human preferences (Rafailov et al., 2023). Furthermore, `ActiveDPO` uses a theoretically grounded preference data selection criterion for complex non-linear reward functions while leveraging the LLM itself as a reward model to guide preference data selection.

Specifically, we establish an upper bound on the error in estimating the reward difference between any pair of responses and their ground-truth reward for a given prompt, expressed in terms of the *gradient of the current aligned LLM* (Proposition 1 in Section 3). This result enables us to *leverage the LLM's gradient* to derive an uncertainty measure as a criterion for preference data selection, thereby *explicitly accounting for the LLM's influence* on the data selection process. To improve the efficiency and practicality of `ActiveDPO`, we introduce novel techniques, such as batch selection and random projection with LoRA gradients (more details are in Section 3.3), to reduce computational cost and storage requirements. These additional techniques make `ActiveDPO` both theoretically grounded and practically effective. Finally, extensive experiments demonstrate that `ActiveDPO` consistently outperforms existing methods across various LLMs and datasets.

The key contributions can be summarized as follows:

- In Section 3, we propose a novel algorithm, `ActiveDPO`, that uses a theoretically grounded active preference data selection criterion for LLM alignment. By leveraging an implicit reward function parameterized by the LLM itself, `ActiveDPO` ensures that the selected preference data is better suited to the specific LLM being aligned.
- In Section 3.3, we introduce techniques such as batch selection and random gradient projection to reduce the computational and storage requirements of `ActiveDPO`, making it more practical for large-scale models.
- In Section 4, we empirically demonstrate that `ActiveDPO` achieves efficient and effective active preference learning across diverse LLMs and datasets.

2 Problem Setting

In LLM alignment, we start with a preference dataset D in which each data point contains a triplet (x, y_1, y_2) where $x \in \mathcal{X}$ is a prompt and $y_1, y_2 \in \mathcal{Y}$ are two responses (which can be written by humans or generated from LLMs). The $\mathcal{X}$ and $\mathcal{Y}$ are prompt space and response space respectively. Denote n as the number of data points in D. We aim to find a k-sized subset $D^{\mathrm{s}} \subseteq D$ and ask human annotators to provide binary preference feedback on the responses denoted as $y_w \succ y_l \mid x$ where y_w and y_l denote the preferred and rejected response respectively. Note that y is not the human preference label but the corresponding response for the prompt. We train the LLM to generate responses that better align with human preference on the labeled data subset D^{l} using DPO. The objective is to obtain an LLM that gives the most desirable responses (defined by win-rate and reward score as we will discuss later) given the fixed labeling budget of k.

Direct preference optimization (DPO). We first start by discussing the DPO method, as introduced in Rafailov et al. (2023). DPO starts by training a LLM through supervised fine-tuning (SFT) on a carefully curated, high-quality dataset that is specifically tailored to a particular downstream task, resulting in a model, denoted by π_{SFT}. The objective of the SFT is to enable the LLM to effectively follow instructions for a specific downstream task. Let $\pi_\theta(y \mid x)$ denote the conditional log-likelihood of generating y given the prompt x, where the model is parameterized by θ. Within DPO, an implicit reward function is defined as follows:

$$r_\theta(x, y) = \beta \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)},$$

where π_{ref} is the reference LLM, which is usually chosen to be the SFT LLM π_{SFT} and β is the regularization hyper-parameter used in DPO. Based on this implicit reward function, DPO uses Bradley-Terry-Luce (BTL) to model the preference feedback. Specifically, BTL assumes that the probability of response y_1 being preferred over y_2, conditioned on the prompt x, is given by:

$$p(y_1 \succ y_2 \mid x) = \frac{\exp\big(r_\theta(x, y_1)\big)}{\exp\big(r_\theta(x, y_1)\big) + \exp\big(r_\theta(x, y_2)\big)} = \sigma\big(r_\theta(x, y_1) - r_\theta(x, y_2)\big), \tag{1}$$

where $\sigma(x) = {}^{1}/_{(1+\exp(-x))}$. DPO uses the following training objective to train the LLM:

$$L_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim D^l}\left[\log \sigma\big(r_\theta(y_w \mid x) - r_\theta(y_l \mid x)\big)\right]. \tag{2}$$

3 Methodology

Overview of `ActiveDPO`. `ActiveDPO` starts with generating responses from an initial data D, which consists of instructions/prompts tailored to a specific task. We use the initial LLM model (i.e., π_{SFT}) to obtain the responses to form the dataset D_t which forms the pool of selection (Section 3.1). After that, we select a batch of triplets (x, y_1, y_2) with size b according to our selection criterion (Section 3.2). Then, we ask the human annotator to provide preference feedback on the responses for the selected batch of data to obtain the labeled dataset. Finally, we train the LLM with the DPO training objective on the newly labeled dataset. We do this process for T iterations and obtain the final trained LLM which can generate responses that align with human preference.

`ActiveDPO` Active Direct Preference Optimization

Input: Initial dataset D; Reference LLM $\pi_{\text{ref}} = \pi_{\text{SFT}}$; Initial LLM $\pi_{\theta_0} = \pi_{\text{SFT}}$; parameterized by θ_0; Iteration T; Batch size B;
for $t = 1, \ldots, T$ **do**
  Generate m pairs of responses from previous LLM $y_1, y_2 \sim \pi_{\theta_{t-1}}(y \mid x)$ for each $x \in D$ to obtain the dataset D_t.
  $D_t^s = \emptyset$
  for $b = 1, \ldots, B$ **do**
    Select the $(x_b^t, y_{b,1}^t, y_{b,2}^t)$ using Eq. (3)
    $D_t^s = D_t^s \cup \{(x_b^t, y_{b,1}^t, y_{b,2}^t)\}$
    Update V_{t-1} according to Eq. (4).
  end for
  Obtain the preference feedback $y_w \succ y_l \mid x$ for each data point in D_t^s to get the labeled dataset D_t^l
  Update the LLM $\pi_{\theta_{t-1}}$ using D_t^l with the DPO training objective in Eq. (2) to obtain π_{θ_t}
end for
Return the trained LLM π_{θ_T}

3.1 Generation of the prompt-responses dataset

In each iteration of `ActiveDPO`, we regenerate the responses for each instruction/prompt in the dataset for two main reasons. Firstly, even though there are some tasks that already have responses written by humans or generated by powerful LLMs, most tasks do not have good responses for each instruction at the start. Generating responses is necessary for these tasks before asking human annotators to provide preference feedback on these responses. Secondly, even though some tasks already have responses for each instruction, these responses are not updated as the LLM improves over time. This is undesirable since the LLM will not be able to learn to generate better responses (compared to the responses provided in the original dataset) as the LLM improves. Consequently, we generate new responses for all the instructions using the latest model obtained from `ActiveDPO`, so that `ActiveDPO` training is able to further improve the LLM with higher-quality responses.

3.2 Selection of data to get human preference annotations

The selection strategy of our `ActiveDPO` is designed by drawing inspiration from the principled neural dueling bandits (Verma et al., 2025), which has derived an uncertainty quantification on the human preference for the reward function that is modeled using the neural network (NN). Inspired by this, we derived the uncertainty quantification on human preference for our LLM trained by DPO and show its empirical effectiveness in Section 4. Consequently, our selection strategy is theoretically grounded and provides empirical effectiveness instead of using the heuristic-based method (Muldrew et al., 2024).

Proposition 1 (Estimation error of the reward difference (informal version of Proposition 2))**.** *Let r_θ denote a fully connected neural network with a width of m in each layer and depth of L. Let $\delta \in (0,1)$. Assume that there is a ground true reward function r and that human preference is sampled from BTL preference modeling. As long as $m \geq M$, then with a probability of at least $1-\delta$,*

$$\left|\left[r_{\theta_{t-1}}(x,y_1) - r_{\theta_{t-1}}(x,y_2)\right] - \left[r(x,y_1) - r(x,y_2)\right]\right| \leq$$
$$\nu_T \|\frac{1}{m}(\nabla r_{\theta_{t-1}}(x,y_1) - \nabla r_{\theta_{t-1}}(x,y_2))\|_{V_{t-1}^{-1}} + \varepsilon$$

for all $x \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}, t \in [T]$ when using the DPO objective defined in Eq. (2) with an additional regularization term to train this reward function $r_{\theta_{t-1}}$. $V_{t-1} = \sum_{p=1}^{t-1}\sum_{x,y_1,y_2\sim D_p^s}\varphi_{t-1}(x,y_1,y_2)\varphi_{t-1}(x,y_1,y_2)$ and $\varphi_{t-1}(x,y_1,y_2) = \frac{1}{\sqrt{m}}(\nabla r_{\theta_{t-1}}(x,y_1) - \nabla r_{\theta_{t-1}}(x,y_2))$. The definition of M, ν_T, ε can be found in the Appendix A.

Proposition 1 is based on the theoretical results from neural dueling bandits (Verma et al., 2025). This result suggests that if $\|\frac{1}{m}(\nabla r_{\theta_{t-1}}(x,y_1) - \nabla r_{\theta_{t-1}}(x,y_2))\|_{V_{t-1}^{-1}}$ is smaller, the estimation error of the reward difference will be smaller. Note that the reward difference directly decides the human preference according to the BTL preference modeling as shown in Eq. (1). Consequently, the reward function $r_{\theta_{t-1}}$ will have a more accurate estimation of the human preference on the two responses y_1, y_2 given x. On the other hand, if $\|\frac{1}{m}(\nabla r_{\theta_{t-1}}(x,y_1) - \nabla r_{\theta_{t-1}}(x,y_2))\|_{V_{t-1}^{-1}}$ is large, this indicates that the reward model will potentially have an inaccurate estimation of the human preference for the responses and hence a higher uncertainty on the human preference. Therefore, a natural selection criterion arises with the uncertainty defined in Proposition 1. Based on this selection criterion, our selection strategy selects a triplet context and pair of arms (x, y_1, y_2) as follows:

$$x, y_1, y_2 = \text{argmax}_{x,y_1,y_2\sim D_t\setminus D_t^s}\|\nabla r_{\theta_{t-1}}(x,y_1) - \nabla r_{\theta_{t-1}}(x,y_2)\|_{V_{t-1}^{-1}} \tag{3}$$

The selection strategy in Eq. (3) uses the implicit reward function $r_{\theta_{t-1}}$ which is parameterized by the current LLM $\pi_{\theta_{t-1}}$. Note that we remove $1/\sqrt{m}$ from the selection criterion and φ_{t-1} since it only affects the scale of the gradient, and the depth m is undefined for the LLM. The selection criterion quantifies how uncertain the current implicit reward function is on the human preference of the response y_1, y_2. Specifically, a larger value of selection criterion in Eq. (3) means that the prompt-response triplet (x, y_1, y_2) is more different from the previously selected triplets. Therefore, by using this selection criterion, our selection strategy encourages the selection of responses that are very different from the previous data and hence achieves exploration of the prompt-response domain to get more informative human preference feedback. This exploration helps improve the implicit reward function as the reward function is trained on human feedback on diverse data in the domains. Although Proposition 1 is derived for a fully connected neural network, we argue in Section A that its conclusions extend to the transformer architecture used in our experiments.

Note that, in addition to being theoretically grounded, our selection strategy enjoys two other advantages. Firstly, our uncertainty criterion is defined using the LLM that we are training instead of some other external models used in the existing methods (Carvalho Melo et al., 2024; Das et al., 2024). Using uncertainty defined without the LLM implicitly assumes that different LLMs need the same data for preference alignment which does not hold practically (as we will show in the experiments). Therefore, our selection strategy is specific to the LLM used and hence is able to select data that better suits the LLM for human preference alignment. Secondly, our selection strategy selects data that directly improves the reward function defined by the LLM and hence directly improves the LLM generation, due to the use of DPO. This strategy is in contrast to prior work that focuses on selecting the data points to improve the reward function that will be used in RLHF. An additional reinforcement learning process needs to be done to obtain the final LLM. This complication makes the data points selected not necessarily helpful for the LLM alignment performance as having a better reward function does not always result in a better RL-trained LLM.

3.3 Practical consideration

Our selection criterion in Eq. (3) requires the computation of gradients of the implicit reward function with respect to the LLM parameters for each prompt-response pair, as well as updating the LLM using the DPO training in every iteration. These steps are computationally expensive and require a lot of storage for storing the gradients. To address these computational inefficiencies, we propose two accelerations to make our selection strategy efficient which we will describe in detail respectively.

Batch selection. In each iteration, we select a batch of data with size B to be labeled by the human annotators. We keep $TB = k$ to keep the annotation budget the same. The batch selection accelerates the selection in two ways: 1) We only need to recalculate the gradient for each prompt-response pair (i.e., $\nabla r_{\theta_{t-1}}(x, y)$) every B selections of data instead of every selection; 2) We only need to update the model via DPO training every B selections.

Batch selection dramatically reduces the computational cost of our selection strategy, however, at the cost of the loss of information. Specifically, the data selected in the current batch will be different from previous batches but the selection within the batch may not enjoy similar results. To remedy this, we propose to update V_{t-1} within the batch. Specifically, after a data point is selected, we update V_{t-1} using the new data point $(x_b^t, y_{b,1}^t, y_{b,2}^t)$

$$V_{t-1} = V_{t-1} + \varphi_{t-1}(x_b^t, y_{b,1}^t, y_{b,2}^t)\varphi_{t-1}(x_b^t, y_{b,1}^t, y_{b,2}^t) \ . \tag{4}$$

Consequently, the next data point to be selected will also be different from the current one even though they are in the same batch, hence further encouraging exploration.

LoRA gradient with random projection. The computation of gradients in our selection criterion is expensive and requires a large storage space. Specifically, the full gradient of the LLM is the same size as the LLM model weight and we need to calculate and store the gradients for all data points. To reduce both computational cost and storage requirement, we propose to use LoRA (Hu et al., 2022) to obtain the gradient efficiently. However, the LoRA gradient is still $1-2\%$ percent of the full model weight which still requires a lot of storage and computation for our selection criterion. Consequently, we apply random projection to further reduce the gradient to a fixed dimension. This random projection is justified by the Johnson-Lindenstrauss lemma (Dasgupta and Gupta, 2003) which shows that the inner product of the original vector can be approximated by the inner product of the projected vector via random projection. Consequently, we can reduce both the computational and storage costs dramatically without sacrificing too much on the selection performance (as shown in Section 4). Similar techniques have been used in Xia et al. (2024). The random projection also reduces the computational cost of the matrix inverse in V_{t-1} in our selection criterion.

Gradient normalization. Existing work (Xia et al., 2024) has demonstrated that the LLM gradients will have lower magnitudes in their l_2 norms when the training data are longer in their length (i.e., sentence length). This means if we use the selection criterion defined in Eq. (3), we will have a higher chance of selecting training data with shorter lengths. This is undesirable, especially for the application of question-answering in which humans may prefer medium to long answers that contain more elaboration on the response. To remedy this, we propose to normalize all the gradients to the unit norm (i.e., l_2 norm being 1) before we use these gradients to calculate the selection criterion, consequently avoiding the criterion favoring shorter sentences. We have empirically shown the effectiveness of normalization before calculating the selection criterion in Section 4.

4 Experiments

In our experiments, we show the effectiveness of our selection criterion in terms of selecting data to train an LLM that can generate responses that better align with human preference. We compare with multiple existing baselines using two widely-used LLMs across two preference alignment tasks.

Datasets. We consider two tasks that require human preference alignment: 1) TLDR summarization dataset (Liu et al., 2020; Völske et al., 2017) which contains posts from Reddit and the corresponding summarization written by humans; 2) WebGPT dataset (Nakano et al., 2021) which is a long-form question-answering dataset that is marked suitable for human preference alignment. These two datasets contain human preference feedback from human annotators and will be used later as an oracle to obtain real human preference feedback.

Models. We performance experiments using 3 different LLMs: Llama-2-7B (Touvron et al., 2023), Gemma-2B (Team et al., 2024) and Qwen3-4B (Yang et al., 2025). Using these LLMs is able to show the effectiveness of alignment on 3 different model families (i.e., Llama, Gemma, and Qwen) and 3 different model sizes (i.e., models with 7 billion parameters, 2 billion parameters, and 4 billion parameters).

Baselines. We compare 4 different selection criteria in our experiments: 1) Random: randomly select data points from the dataset to get human preference feedback; 2) APO (Das et al., 2024): a theoretically grounded method in the setting of RLHF alignment. Their theoretical results are based on the assumption of a linear reward function and is designed for RLHF training; 3) APLP (Muldrew et al., 2024), an active learning method for DPO that uses heuristic uncertainty/certainty quantification to select the data to be labeled; 4) Our proposed method `ActiveDPO`. Note that, for fair comparisons, we only vary the way to select data points to be labeled for different methods and share the same model training and data labeling pipeline among different baselines. Consequently, the only variable that leads to different performance is the way to select data among different methods.[1]

Obtaining human preference feedback. As new responses are generated by the updated model in each iteration, these responses are not part of the original preference dataset and hence do not have human preference feedback. To make our experiments feasible, we train a reward model using the original human preference feedback and use this reward model as an oracle to provide the preference feedback for newly generated responses in each iteration.[2]

Evaluation. The reward model can be used to evaluate the extent to which the LLM generates responses that align with human preference. To evaluate the performance, we use the trained LLM to generate multiple responses for 100 prompts sampled from the dataset for each task. After that, we use the reward model to obtain the average reward for all the prompt-response pairs and report the performance. Ideally, if the LLM can generate responses with higher rewards, it aligns better with human preference since the reward model is trained on real human preference.

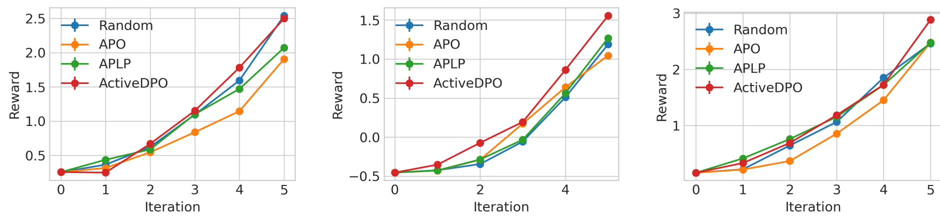
Hyper-parameters. For each task, we train the initial LLM with supervised fine-tuning with the SFT dataset provided in each task for 1 epoch with the learning rate of $2e-05$. In each iteration, we randomly select 1000 prompts from the dataset to generate 3 responses for each prompt. Consequently, each prompt will form 3 corresponding triplets (x, y_1, y_2) (i.e., $\binom{3}{2}$ number of pairwise combinations) and hence 3000 data points in the dataset D_t. We select 50 data points in each iteration using different selection strategies. We train the model using DPO objective based on the labeled dataset for 4 epochs with the learning rate of $1e-4$. As for the LoRA gradient, we use the rank of 128 with α of 512. We project all the LoRA gradients to 8192 dimensions, a dimensionality that balances performance and computational costs as we will show later.

Results. We have provided the comparison of the average reward of the responses generated by the LLM trained on the data selected by different selection strategies in Fig. 1. The LLM trained with data selected by our `ActiveDPO` consistently generates responses with higher rewards compared to other selection strategies across different LLMs and datasets. Consequently, our `ActiveDPO` outperforms all other baselines in selecting data for a fixed number of labeling budgets[3]. APLP performs well on the Gemma model, however, it performs even worse than random on Llama-2. This is likely due to the heuristic design of the uncertainty quantification method in APLP, which does not work consistently well in different settings. Specifically, APLP uses the difference of the estimated rewards for two responses given a prompt as part of the selection criterion. This criterion allows APLP to select triplets with incorrect human preference predicted by the estimated reward function in the early stage when the reward function is inaccurate, hence improving the reward function

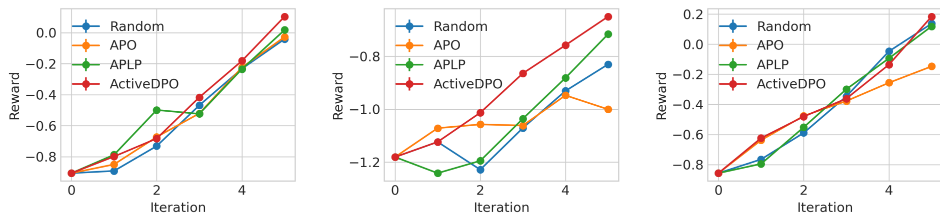
[1]Note that, for APO, we implement the original algorithm (Das et al., 2024) which does not regenerate responses using the new models.

[2]We use the reward function that is already trained and available in HuggingFace. Specifically, we use the model from OpenAssistant (2024a) for TLDR dataset and OpenAssistant (2024b) for WebGPT dataset.

[3]The result of DPO alignment training is problem-dependent (depending on the dataset and model). For datasets that are very noisy, it is expected that different active learning methods will perform similarly. For larger models, different selection methods also tend to perform more similarly than for smaller models (which explains Fig. 1 (a) where our method performs similarly to other methods in the last iteration).



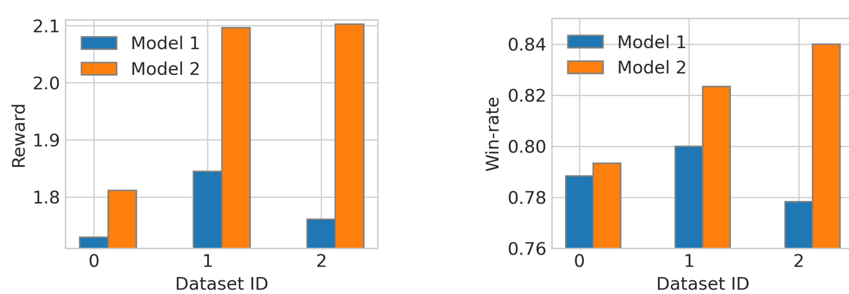
(a) TLDR with Llama-2-7B (b) TLDR with Gemma-2B (c) TLDR with Qwen3-4B



(d) WebGPT with Llama-2-7B (e) WebGPT with Gemma-2B (f) WebGPT with Qwen3-4B

Figure 1: Comparison of average rewards for responses generated by the LLM using different selection strategies.

estimation. This partially explains why APLP performs well in the first iteration for both TLDR and WebGPT on the Llama-2 model. However, as more human preference is collected, the reward function estimation is more accurate, and hence, the triplet with a large reward difference can be data points with correct human preference predicted by the estimated reward function and with a large reward margin, which do not help to improve the reward function. Consequently, APLP performs badly in the later iterations. On the other hand, APO also performs inconsistently in different settings. This is likely due to the unrealistic assumption of the reward function, which does not hold in real applications (e.g., the implicit reward function in DPO is non-linear).

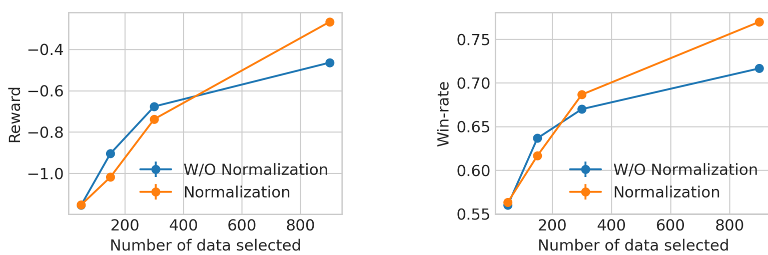


(a) Reward (b) Win-rate

Figure 2: Different models require different data to achieve good alignment performance. We train the Gemma model using two different SFT datasets to obtain Model 1 and Model 2. We construct 3 different human preference datasets and perform DPO training on these 3 datasets for these two models respectively.

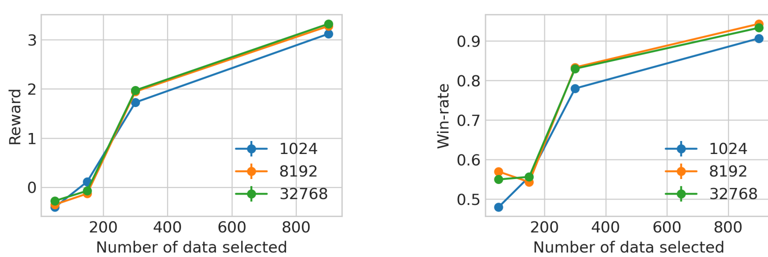
The impact of LLM on data selection. We perform additional experiments to verify that different LLMs indeed need different data to achieve better performance. Specifically, we train the Gemma-2B model on two different SFT datasets to obtain two different LLMs: Model 1 and Model 2. The way we construct the 2 SFT dataset is by using the sentence-BERT (Reimers and Gurevych, 2019) to transform each data point to embedding and use k-means to cluster the dataset into two subsets using the embeddings. We obtain 3 different DPO data subsets using the same approach. We train these two LLMs on 3 DPO data subsets respectively, and evaluate their performance. From Fig. 2, Model 1 and Model 2 achieve very different performance using these 3 DPO data subsets. Specifically, Model 2 achieves the best performance on Dataset 2, while Model 1 achieves the worst performance on the same dataset (i.e., in terms of the win-rate). Consequently, the choice of model has a substantial impact on performance and must be considered when selecting data for achieving better model performance. Intuitively, this is because Model 1 and Model 2 are trained on very different SFT dataset, and hence require different new data to make up what they missed in the previous SFT fine-tuning.

The effect of random projection on the performance. Our method uses random projection to reduce the dimensionality of the LoRA gradients, reducing the storage requirement and computational cost for our `ActiveDPO` (as described in Section 3). To further study the effect of random projection to the performance of our `ActiveDPO`, we perform experiments on using different dimensionalities for the random projection and evaluate the performance of our `ActiveDPO`. The results in Fig. 4 show that a lower dimensionality leads to poorer performance of `ActiveDPO`. However, when the dimensionality is 8192 or above, the performance of `ActiveDPO` does not improve as a larger dimensionality is used. Consequently, we use the dimensionality of 8192 across all our experiments to achieve good performance while keeping the computational cost and storage requirement low. We provide more analysis on the computational complexity and memory requirement for `ActiveDPO` in Section A.



(a) Reward (b) Win-rate

Figure 3: Effect of normalizing LoRA gradients on the performance of `ActiveDPO`.



(a) Reward (b) Win-rate

Figure 4: Effect of Random Projection Dimensionality of LoRA gradients.

The effect of the normalization of the gradient on the performance. We perform experiments to verify the effect of normalizing the gradient in our `ActiveDPO`. Specifically, as described in Section 3, we normalize LoRA gradients to unit-norm before we use them to calculate the selection criterion. We perform the selection of data using our selection strategy with gradient normalization compared to the one without gradient normalization. Fig. 3 shows the performance of our `ActiveDPO` with/without the gradient normalization on WebGPT dataset using Gemma-2B model. These results show that normalizing the LoRA gradients helps to improve the performance of our selection strategy. As described in Section 3 our method will not favor the data points with shorter responses compared to the ones without normalization. Long responses with clear reasoning may sometimes be preferred by humans instead of shorter ones. Consequently, our `ActiveDPO` with gradient normalization performs better. We have included additional results for the TLDR dataset in the Appendix in which normalization does not affect the performance by much.

5 Related Work

Learning from human preference feedback has been extensively studied for over a decade (Yue and Joachims, 2009; Fürnkranz et al., 2012; Christiano et al., 2017; Zhu et al., 2023; Verma et al., 2025). In this section, we review work on dueling bandits, active preference learning, LLM alignment, and active LLM alignment, which are most relevant to our problem.

Dueling Bandits. One of the earliest works (Yue and Joachims, 2009; 2011; Yue et al., 2012) considers finite-armed dueling bandit problem in which the learner's goal is to find the best action using available pairwise preference between two selected actions. Several follow-up works considers different settings involving different criteria for selecting the best action (Zoghi et al., 2014b;a; Ailon et al., 2014; Komiyama et al., 2015; Gajane et al., 2015) and we refer readers to Bengs et al. (2021) for a compressive survey covering these details. The standard dueling bandits has been extended to different settings, such as contextual dueling bandit setting (Saha, 2021; Bengs et al., 2022; Di et al., 2023; Li et al., 2024; Verma et al., 2025).

Reinforcement Learning with Human Feedback. Preference feedback has also been extensively studied in reinforcement learning (Fürnkranz et al., 2012; Akrour, 2014; Christiano et al., 2017; Zhu et al., 2023) introduced preference-based policy iteration, a method that relies solely on preference feedback to guide reinforcement learning, with subsequent developments by (Akrour, 2014). (Christiano et al., 2017) demonstrated the effectiveness of human preference feedback in training agents for Atari games and simulated robot locomotion. On the theoretical side, research has progressed from bandit settings to reinforcement learning (Zhu et al., 2023), providing deeper insights into the optimal use of preference feedback for decision-making and policy optimization. For a more comprehensive overview, we refer readers to a survey on preference-based reinforcement learning (Wirth et al., 2017).

LLM Alignment. Recent works have introduced methods like Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Lee et al., 2024) and Direct Preference Optimization (DPO) (Rafailov et al., 2023) to align LLMs with specific formats or human values. For a comprehensive overview of various aspects of LLM alignment, we refer readers to surveys on the topic (Ji et al., 2023; Anwar et al., 2024).

Active LLM Alignment. Actively select preference queries for a human to provide relative preferences between two queries allows efficiently learn reward functions that capture human intent. Some of works has already considered actively selecting queries in domain like autonomous (Sadigh et al., 2017; Biyik and Sadigh, 2018). Recent work on active preference data selection for LLM alignment has explored both heuristic methods (Carvalho Melo et al., 2024; Muldrew et al., 2024) and approaches with theoretical guarantees (Mehta et al., 2023; Das et al., 2024). A key distinction among these recently proposed theoretical methods lies in their data selection strategies. On the other side, existing methods with theoretical guarantees (Mehta et al., 2023; Das et al., 2024) are based on the assumption of a linear latent reward function, which may not hold in real-world applications such as LLM alignment in which reward functions are often highly non-linear and complex.

6 Conclusion

In this paper, we propose a data selection method for actively selecting data to obtain human preference feedback for LLM alignment, aiming to achieve better alignment performance with as few annotations as possible. To this end, we introduce a theoretically grounded method, `ActiveDPO`, and demonstrate that it achieves superior alignment performance under the same labeling budget across different models and datasets. Notably, the selection criterion in `ActiveDPO` requires computing the gradient of the LLM with respect to model parameters for each data point, which is computationally expensive and demands substantial storage for storing gradients. We propose several techniques to improve the efficiency of our method. Although further efforts could be made to accelerate gradient computation, this is beyond the scope of the current work and is left for future research.

References

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proc. ICML*, pages 127–135, 2013.

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proc. ICML*, pages 856–864, 2014.

Riad Akrour. *Robust Preference Learning-based Reinforcement Learning*. PhD thesis, Université Paris Sud-Paris XI, 2014.

Anthropic. Introducing claude 2.1. `https://www.anthropic.com/news/claude-2-1/`, 2023. [Online; accessed 01 February 2008].

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *arXiv:2404.09932*, 2024.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proc. ICML*, pages 263–272, 2017.

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:2204.05862*, 2022.

Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, pages 1–108, 2021.

Viktor Bengs, Aadirupa Saha, and Eyke Hüllermeier. Stochastic contextual dueling bandits under linear stochastic transitivity models. In *Proc. ICML*, pages 1764–1786, 2022.

Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Proc. CRL*, pages 519–528, 2018.

Luckeciano Carvalho Melo, Panagiotis Tigas, Alessandro Abate, and Yarin Gal. Deep bayesian active learning for preference modeling in large language models. In *Proc. NeurIPS*, pages 118052–118085, 2024.

Baiyuan Chen, Shinji Ito, and Masaaki Imaizumi. Optimal dynamic regret by transformers for non-stationary reinforcement learning. *arXiv:2508.16027*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proc. NeurIPS*, pages 4302–4310, 2017.

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. Active preference optimization for sample efficient rlhf. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Qiwei Di, Tao Jin, Yue Wu, Heyang Zhao, Farzad Farnoud, and Quanquan Gu. Variance-aware regret bounds for stochastic contextual dueling bandits. *arXiv:2310.00968*, 2023.

Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning*, pages 123–156, 2012.

Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proc. ICML*, pages 218–227, 2015.

Google. Palm 2 technical report. *arXiv:2305.10403*, 2023.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2022.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv:2310.19852*, 2023.

Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In *Proc. COLT*, pages 1141–1154, 2015.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *Proc. ICML*, pages 26874–26901, 2024.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. WWW*, pages 661–670, 2010.

Xuheng Li, Heyang Zhao, and Quanquan Gu. Feel-good thompson sampling for contextual dueling bandits. *arXiv:2404.06013*, 2024.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *Proc. ICLR*, 2024.

Fei Liu et al. Learning to summarize from human feedback. In *Proc. ACL*, 2020.

Zichen Liu, Changyu Chen, Chao Du, Wee Sun Lee, and Min Lin. Sample-efficient alignment for llms. *arXiv:2411.01493*, 2024.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *Proc. ICML*, pages 23610–23641. PMLR, 2023.

Viraj Mehta, Vikramjeet Das, Ojash Neopane, Yijia Dai, Ilija Bogunovic, Jeff Schneider, and Willie Neiswanger. Sample efficient reinforcement learning from human feedback via active exploration. *arXiv:2312.00267*, 2023.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *Proc. ICML*, pages 36577–36590, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv:2112.09332*, 2021.

OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023.

OpenAssistant. DeBERTa large summarization reward model. `https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large`, 2024a. Accessed: 2025-02-19.

OpenAssistant. DeBERTa large summarization reward model v2. `https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2`, 2024b. Accessed: 2025-02-19.

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. In *Proc. NeurIPS*, pages 27730–27744, 2022.

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. In *Proc. NeurIPS*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. EMNLP*. Association for Computational Linguistics, November 2019.

Dorsa Sadigh, Anca D. Dragan, S. Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Proc. RSS*, 2017.

Aadirupa Saha. Optimal algorithms for stochastic contextual preference bandits. In *Proc. NeurIPS*, pages 30050–30062, 2021.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Proc. NeurIPS*, pages 3008–3021, 2020.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv:2403.08295*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

Arun Verma, Zhongxiang Dai, Xiaoqiang Lin, Patrick Jaillet, and Bryan Kian Hsiang Low. Neural dueling bandits: Principled preference-based optimization with non-linear reward function. In *Proc. ICLR*, 2025.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Proc. NeurIPS*, 2022.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, pages 1–46, 2017.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proc. ICML*, pages 1201–1208, 2009.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proc. ICML*, pages 241–248, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, pages 1538–1556, 2012.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Thompson sampling. In *Proc. ICLR*, 2021.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv:2303.18223*, 2023.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *Proc. ICML*, pages 43037–43067, 2023.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proc. ICML*, pages 10–18, 2014a.

Masrour Zoghi, Shimon A Whiteson, Maarten De Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proc. WSDM*, pages 73–82, 2014b.

A Appendix

A.1 Computational resources, datasets and models

Experiments are run on a server with AMD EPYC 7763 64-Core Processor, 1008GB RAM, and 8 NVIDIA L40 GPUs.

Dataset license. TLDR dataset: MIT License; WebGPT dataset: Apache License 2.0.

Model license. Llama-2: LLAMA 2 Community License Agreement. Gemma: Gemma License. Qwen3: Apache License 2.0.

A.2 Additional discussion on Proposition 1

Although our results in Proposition 1 rely on neural tangent kernel (NTK) theory, which is primarily developed for fully connected networks, recent works (Lin et al., 2024; Chen et al., 2025) have shown promising directions for partially extending NTK theory to transformers. Furthermore, recent studies (Lin et al., 2024) have demonstrated that sufficiently large transformer models, when pre-trained on offline interaction sequences, can approximate near-optimal online reinforcement learning algorithms such as LinUCB (Li et al., 2010) and Thompson Sampling in multi-armed bandits (Agrawal and Goyal, 2013), as well as UCB value iteration for tabular Markov decision processes (Azar et al., 2017). In addition, transformers have been shown to effectively handle non-stationary RL environments, achieving near-optimal performance by minimizing dynamic regret (Chen et al., 2025).

Our analysis relies on two standard results from Neural Tangent Kernel (NTK) theory: (i) Kernel constancy: along training, the NTK remains (asymptotically) constant (i.e., it converges to a deterministic kernel independent of the training step); (ii) GP limit of the predictor: the trained predictor converges to the Gaussian process induced by that kernel. Result (i) has been established for transformer architectures via the tensor programs framework (Yang, 2020). By contrast, a general proof of (ii) for transformers is not yet available; however, extensive empirical evidence supports Gaussian process behavior in large-width networks (Malladi et al., 2023). Accordingly, the principal theoretical gap in our analysis is a formal proof of (ii) for transformers, which is a challenging problem that we leave as future work. Nonetheless, these assumptions align with existing theory and are corroborated by the strong empirical performance of our method, which together provide a credible justification for applying our theoretical insights to transformer-based LLMs. Equipped with these ideas and existing results, we could potentially extend Proposition 1 to transformer architectures; however, this is beyond the scope of the current paper and is therefore left for future work.

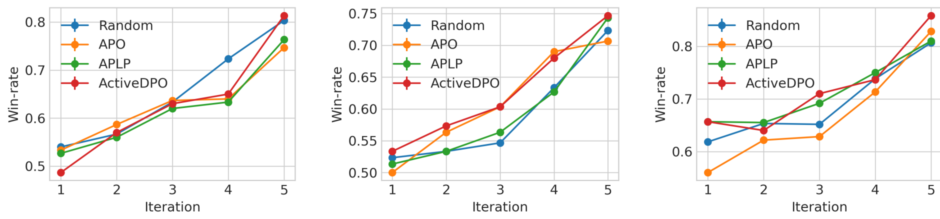
A.3 Additional Analysis on the computational complexity of `ActiveDPO`

We provide a theoretical calculation of the computational complexity and memory requirement for `ActiveDPO`.

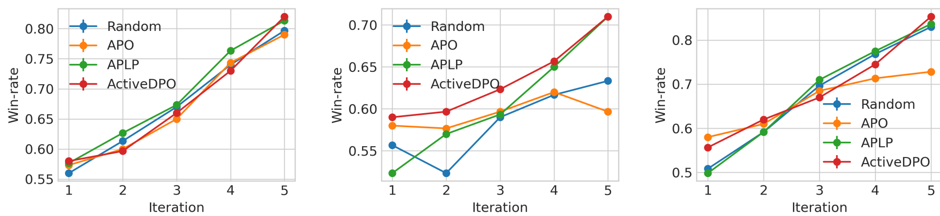
Assume that we have n number of prompts with m number of responses, the number of parameters is k and the projection dimension is d. Calculating the gradient for all the data points requires $O(nm^2k)$. Projecting all the gradients to d dimensions requires $O(nm^2kd)$, and hence a total of $O(nm^2kd)$. Assume that we have selected s number of data points. The calculation of gradient and projection for these s number of selected data points with the new model is $O(skd)$. Calculating our acquisition function for all data points is $O(nm^2d^2 + d^3)$. Therefore, for each iteration, we have the complexity of $O(nm^2d^2 + d^3 + skd + nm^2kd)$. Note that the projection dimension is controllable and a hyperparameter, and m is chosen to be small in most applications. Therefore, the overall computational complexity is small. The memory requirement is mainly dominated by storing the projected gradient, which is $O(nm^2d)$ and is again reducible by reducing d and m.

A.4 Additional experimental results

Fig. 5 shows the win-rate of different selection strategies. In general, our `ActiveDPO` still outperforms other selection strategies in the last few iterations.



(a) TLDR with Llama-2 (b) TLDR with Gemma (c) TLDR with Qwen3-4B



(d) WebGPT with Llama-2 (e) WebGPT with Gemma (f) WebGPT with Gemma

Figure 5: Comparison of the win-rate of the responses generated by the LLM trained by DPO with the responses generated by the initial LLM with different selection strategies.

A.5 Proofs for Proposition 1

Define the following objective function:

$$L(\theta) = -\frac{1}{m} \sum_{(x,y_w,y_l)\sim D^l} \left[\log \sigma\big(r_\theta(y_w \mid x) - r_\theta(y_l \mid x)\big)\right] + \frac{\lambda \|\theta - \theta_0\|}{2}\,. \tag{5}$$

Define H as the NTK matrix following the same definition in Verma et al. (2025). Define ν_T following the same definition in Verma et al. (2025). Define K as the size of the selection dataset D^s in each round.

We make the following assumption:

Assumption 1. *Assume that*

- $\kappa_\mu \doteq \inf_{x\in\mathcal{X}, y_1,y_2\in\mathcal{Y}} \sigma(r(x,y_1) - r(x,y_2)) > 0,$
- *the reward function is bounded:* $|r(x,y)| \le 1, \forall x \in \mathcal{X}, y \in \mathcal{Y},$
- *there exists* $\lambda_0 > 0$ *s.t.* $H \succeq \lambda_0 \mathbf{I}$*, and*
- *the reward function takes a vector* z *(which is the representation vector for the concatenation of* x *and* y*) as input and* z *satisfies:* $\|z\|_2 = 1$ *and* $[z]_j = [z]_{j+d/2}$ *for all* $x \in \mathcal{X}$ *and* $y \in \mathcal{Y}$.

Denote $\overline{\sigma}_{t-1}(x,y_1,y_2) = \frac{\lambda}{\kappa_\mu}\|\varphi(x,y_1,y_2)\|_{\overline{V}_{t-1}^{-1}}$ where $\varphi(x,y_1,y_2) = \frac{1}{\sqrt{m}}\big(\nabla r_{\theta_0}(x,y_1) - \nabla r_{\theta_0}(x,y_2)\big)$ and $\overline{V}_{t-1} = \sum_{p=1}^{t-1}\sum_{x,y_1,y_2\sim D_p^s} \varphi(x,y_1,y_2)\varphi(x,y_1,y_2) + \frac{\lambda}{\kappa_\mu}\mathbf{I}$. We give the following Lemma, which is a direct extension from Theorem 1 of Verma et al. (2025):

Lemma 1. *Given that Assumption 1 holds, let* $\delta \in (0,1)$, $\varepsilon_{m,t} \doteq C m^{-1/6}\sqrt{\log m}L^3(\frac{t}{\lambda})^{4/3}$ *for some absolute constant* $C > 0$*. As long as* $m \ge poly(T, L, K, 1/\kappa_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$*, then with probability of at least* $1-\delta$,

$$\left|\big[r_{\theta_{t-1}}(x,y_1) - r_{\theta_{t-1}}(x,y_2)\big] - \big[r(x,y_1) - r(x,y_2)\big]\right| \le \nu_T \overline{\sigma}_{t-1}(x,y_1,y_2) + \varepsilon_{m,t}$$

for all $x \in \mathcal{X}$ *and* $y_1, y_2 \in \mathcal{Y}, t \in [T]$ *when using the objective defined in Eq.* (5) *to train this reward function* $r_{\theta_{t-1}}$.

Proof. This Proposition is immediately true by concatenating the prompt x and response y to replace the input used in Theorem 1 of Verma et al. (2025) and instantiate the link function in Verma et al. (2025) as the sigmoid function. Specifically, we assume that the reward takes the representation vector z of the concatenation of x and y as input and assume that this z satisfies the corresponding conditions in Assumption 1. □

Denote $\sigma_{t-1}(x, y_1, y_2) = \frac{\lambda}{\kappa_\mu}\|\varphi_{t-1}(x, y_1, y_2)\|_{V_{t-1}^{-1}}$ where $\varphi_{t-1}(x, y_1, y_2) = \frac{1}{\sqrt{m}}\big(\nabla r_{\theta_{t-1}}(x, y_1) - \nabla r_{\theta_{t-1}}(x, y_2)\big)$ and $V_{t-1} = \sum_{p=1}^{t-1}\sum_{x,y_1,y_2\sim D_p^s}\varphi_{t-1}(x, y_1, y_2)\varphi_{t-1}(x, y_1, y_2) + \frac{\lambda}{\kappa_\mu}\mathbf{I}$.

Lemma 2. *Given that Assumption 1 holds, for some absolute constant $C > 0$, we have that:*

$$|\sigma_{t-1}(x, y_1, y_2) - \overline{\sigma}_{t-1}(x, y_1, y_2)| \le C\lambda^{-5/6}(t-1)^{4/3}m^{-1/6}\sqrt{\log m}L^{9/2}\ . \tag{6}$$

Proof. Following the proof of Lemma B.4 in Zhang et al. (2021), we can show that

$$\begin{aligned}
&|\sigma_{t-1}(x, y_1, y_2) - \overline{\sigma}_{t-1}(x, y_1, y_2)| \\
&\quad\le \frac{1}{\sqrt{\lambda}}\left\|\frac{r_{\theta_{t-1}}(x, y_1) - r_{\theta_{t-1}}(x, y_2)}{\sqrt{m}} - \frac{r_{\theta_0}(x, y_1) - r_{\theta_0}(x, y_2)}{\sqrt{m}}\right\|_2 \\
&\qquad + \frac{\tilde{C}^2 L}{\sqrt{\lambda}}\sum_{i=1}^{t-1}\left\|\frac{r_{\theta_{t-1}}(x_i, y_{i,1}) - r_{\theta_{t-1}}(x_i, y_{i,2})}{\sqrt{m}} - \frac{r_{\theta_0}(x_i, y_{i,1}) - r_{\theta_0}(x_i, y_{i,2})}{\sqrt{m}}\right\|_2
\end{aligned}$$

for some absolute constant $\tilde{C} > 0$. In addition, according to Lemma 3 of Verma et al. (2025), we have that

$$\|r_{\theta_0}(x, y) - r_{\theta_{t-1}}(x, y)\|_2 \le C_1 m^{1/3}\sqrt{\log m}\left(\frac{t-1}{\lambda}\right)^{1/3} L^{7/2}, \quad \forall x \in \mathcal{X}, y \in \mathcal{Y}, t \in [T]$$

Consequently, we have that

$$\begin{aligned}
&|\sigma_{t-1}(x, y_1, y_2) - \overline{\sigma}_{t-1}(x, y_1, y_2)| \\
&\quad\le \tilde{C}^2\frac{L}{\sqrt{\lambda}}(t-1) \times 2 \times \frac{1}{\sqrt{m}} \times C_1 m^{1/3}\sqrt{\log m}\left(\frac{t-1}{\lambda}\right)^{1/3} L^{7/2} \\
&\quad= C\lambda^{-5/6}(t-1)^{4/3}m^{-1/6}\sqrt{\log m}L^{9/2}
\end{aligned}$$

for some absolute constant $C > 0$. □

The result of Lemma 2 says that as long as the width m of the NN is large enough, we can ensure that the difference $|\sigma(x_{t,1}, x_{t,2}) - \overline{\sigma}(x_{t,1}, x_{t,2})|$ is upper-bounded by a small constant. Consequently, we can show the following formal version of Proposition 1.

Proposition 2 (Formal version of Proposition 1)**.** *Given that Assumption 1 holds, let $\delta \in (0, 1)$, $\varepsilon_{m,t} \doteq Cm^{-1/6}\sqrt{\log m}L^3(\frac{t}{\lambda})^{4/3} + C\lambda^{-5/6}(t-1)^{4/3}m^{-1/6}\sqrt{\log m}L^{9/2}$ for some absolute constant $C > 0$. As long as $m \ge poly(T, L, K, 1/\kappa_\mu, 1/\lambda_0, 1/\lambda, \log(1/\delta))$, then with probability of at least $1 - \delta$,*

$$\left|\left[r_{\theta_{t-1}}(x, y_1) - r_{\theta_{t-1}}(x, y_2)\right] - \left[r(x, y_1) - r(x, y_2)\right]\right| \le \nu_T\sigma_{t-1}(x, y_1, y_2) + \varepsilon_{m,t}$$

for all $x \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}, t \in [T]$ when using the objective defined in Eq. (5) *to train this reward function $r_{\theta_{t-1}}$.*

Proof. Combining Lemma 1 and Lemma 2, we get that the proposition is true. □

Remark 1. *Note that the objective function of Eq.* (5) *is almost the same as Eq.* (5)*, with Eq.* (5) *scaling the Eq.* (5) *by a constant and having an additional regularization term. The design of Eq.* (5) *is for the theoretical results. Empirically, we still use the standard Eq.* (2) *and adjust the regularization by adjusting the β in Eq.* (2)*.*