$\mathcal{B}$PO: Supercharging Online Preference Learning by Adhering to the Proximity of Behavior LLM

Anonymous ACL submission

Abstract

Direct alignment from preferences (DAP) has emerged as a promising paradigm for aligning large language models (LLMs) to human desiderata from pre-collected, offline preference datasets. While recent studies indicate that existing offline DAP methods can directly benefit from online training samples, we highlight the need to develop specific online DAP algorithms to fully harness the power of online training. Specifically, we identify that the learned LLM should adhere to the proximity of the *behavior LLM*, which collects the training samples. To this end, we propose online **P**reference **O**ptimization in proximity to the **B**ehavior LLM ($\mathcal{B}$PO), emphasizing the importance of constructing a proper trust region for LLM alignment.

We conduct extensive experiments to validate the effectiveness and applicability of our approach by integrating it with various DAP methods, resulting in significant performance improvements across a wide range of tasks when training with the same amount of preference data. Even when only introducing *one* additional preference annotation phase, our online $\mathcal{B}$PO improves its offline DAP baseline from 72.0% to 80.2% on TL;DR and from 82.2% to 89.1% on Anthropic Helpfulness in terms of win rate against human reference text.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019) has achieved tremendous success in aligning the powerful pretrained large language models (LLMs) with human preference (Achiam et al., 2023; Gemini et al., 2023; Anthropic, 2024; Touvron et al., 2023), revolutionizing human society. However, traditional RLHF methods (Ouyang et al., 2022; Stiennon et al., 2020b) are computationally expensive due to their two-stage training pipeline that consists of a reward modeling phase and can suffer from RL training instability (Choshen et al., 2019). To address these issues, recent advances in direct alignment from preferences (DAP) methods provide solutions to avoid learning a reward model (RM) and stabilize the training process. Prominent examples include Direct Preference Optimization (DPO) (Rafailov et al., 2024) and its variants (Azar et al., 2024; Ethayarajh et al., 2024; Wang et al., 2022; Tang et al., 2024b), which directly optimize LMs using a static, pre-collected set of preference data, streamlining the alignment procedures.

However, recent studies (Xu et al., 2023; Guo et al., 2024; Pang et al., 2024; Tang et al., 2024a; Tajwar et al., 2024) identified that aligning an LLM with offline preference datasets prevents the LLM from getting feedback for its own generations. These studies emphasize the importance of incorporating online data generated by intermediate models during training. While empirical evidence shows that offline DAP methods can directly benefit from online preference data, we argue that making algorithm-level adjustments is essential to fully harness the power of online training.

Specifically, we identify that existing online DAP methods (Guo et al., 2024; Tang et al., 2024a; Calandriello et al., 2024; Xu et al., 2023; Pang et al., 2024) do not adjust the trust region designed in offline DAP methods (Rafailov et al., 2024). These methods still construct their trust region (Schulman et al., 2015) by penalizing the KL divergence between the learned LLM and a fixed reference model π_{ref}, even when training samples are dynamically generated by intermediate models. Drawing inspiration from existing RL literature (Schulman et al., 2017; Fujimoto et al., 2019; Li et al., 2023), we propose online **P**reference **O**ptimization in proximity to the **B**ehavior LLM ($\mathcal{B}$PO), emphasizing that a better trust region should be instead constructed around the *behavior LLM* π_β that collects the training samples. In other words, we should set π_β as π_{ref} when performing online DAP. We provide an

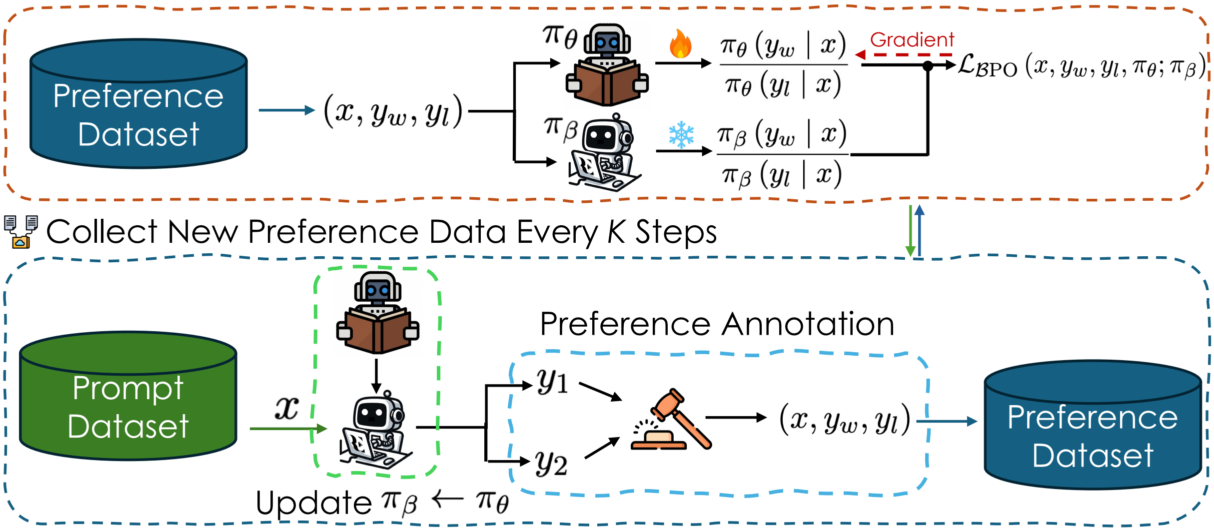


Figure 1: Overview of the training pipeline of our $\mathcal{B}$PO. Our training loss $\mathcal{L}_{\mathcal{B}\mathrm{PO}}$ is calculated by constraining the KL divergence between π_θ and the behavior LLM π_β. Every K step, we update π_β with π_θ and use it to collect new samples for annotations.

overview of our training pipeline in Fig. 1.

However, setting a dynamic π_{ref} during online DAP can lead to instability. To mitigate this issue, we propose optimizing an ensemble of LoRA (Hu et al., 2021) weights and merging them during inference. We verify the effectiveness of our method by building on top of various DAP methods, including DPO (Guo et al., 2024), IPO (Azar et al., 2024) and SLiC (Zhao et al., 2023). Empirically, we show that our $\mathcal{B}$PO significantly improves over their online and offline DAP counterparts on TL;DR (Ziegler et al., 2019), Anthropic Helpfulness and Harmlessness (Bai et al., 2022), demonstrating the generalizability of our methods.

On the other hand, conducting iterative preference annotations at each training step can be practically infeasible when hiring human annotators. Given the same annotation budget, we anticipate that a successful online DAP method to perform well at a low annotation frequency. In other words, we aim to minimize the number of preference annotation phases throughout the training. To this end, we evaluate our method with different annotation frequencies while keeping the total amount of preference data constant. We demonstrate that **even with just one additional preference annotation phase** compared to offline DPO, our $\mathcal{B}$PO **significantly improves over its offline DPO counterpart** from 72.0% to 80.2% on TL;DR and from 82.2% to 89.1% on Anthropic Helpfulness in terms of win rate against human reference text.

Furthermore, we conduct an ablation study to verify our performance improvement comes from our better trust region constructed around π_β instead of π_β's higher quality compared to π_{ref}. Our results show that even when using a high-quality LLM as π_{ref} for online DAP baselines, our approach still outperforms it.

Our contributions are three-fold:

1. An online DAP method $\mathcal{B}$PO. To the best of our knowledge, we are the first to tailor offline DAP methods for online training.
2. Empirical superiority of our $\mathcal{B}$PO over its online and offline DAP counterpart on standard alignment tasks.
3. Remarkable applicability of our $\mathcal{B}$PO to handle low data collection frequencies.

2 Related Work

Reinforcement Learning from Human Feedback methods (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023) first learn a RM from a set of preference data based on the Bradley-Terry modelization (Bradley and Terry, 1952) and then leverage an RL algorithm, e.g., PPO (Schulman et al., 2017), to optimize a pretrained or supervised fine-tuned (SFT) LLM towards the learned RM. However, the two-stage learning process is computationally demanding and RL training can suffer from training instability (Choshen et al., 2019). To address these issues, recent advancements, including

DPO (Guo et al., 2024) and DPO variants (Zhao et al., 2023; Azar et al., 2024; Ethayarajh et al., 2024; Wang et al., 2022; Tang et al., 2024b), enable direct alignment from a fixed, offline set of preference data. These methods leverage the dual form of the original RLHF objectives, successfully eliminating the reward modeling phase and stabilizing the training. Empirically, these methods achieve impressive performance on standard evaluation benchmarks (Ziegler et al., 2019; Bai et al., 2022; Dubois et al., 2024; Zheng et al., 2024).

Concurrently, TR-DPO (Gorbatovski et al., 2024) also explores setting a dynamic π_{ref} for DAP methods. However, TR-DPO only considers offline DAP settings and thus never considers setting $\pi_{\text{ref}} = \pi_\beta$. Instead, TR-DPO explores setting π_{ref} as the moving average of their π_θ or periodically updates π_{ref} with π_θ. Therefore, TR-DPO is substantially different from our methods.

Online DAP Methods. Recent studies (Guo et al., 2024; Tang et al., 2024a; Calandriello et al., 2024; Xu et al., 2023; Pang et al., 2024; Dong et al., 2024; Xie et al., 2024) have recognized the importance of on-policy training data. Specifically, these methods collect human preference on the responses generated from intermediate models and use them for training. Although these online methods improve their offline counterparts, they still constrain the KL divergence between the learning LLM and a static reference model. In this paper, we provide extensive empirical results demonstrating that the reference LLM should be set dynamically as the behavior LLM that collects the training data during online preference learning.

3 Preliminaries

3.1 Reinforcement Learning from Human Feedback

Traditional RLHF methods require learning an RM r_ϕ from a preference dataset $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$ to mirror human preference. Each example in $\mathcal{D}$ is obtained by sampling a pair of responses (y_1, y_2) given the sample text prompt x, which are then sent to human or AI labelers for annotations. The preferred and dispreferred samples are denoted as y_w and y_l, respectively.

With the learned RM r_ϕ, we can optimize an LLM π_θ with the RL objective given by

$$\begin{aligned}\max_\theta \, & \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] \\ & - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y\mid x)\|\pi_{\text{ref}}(y\mid x)\right]\end{aligned}, \quad (1)$$

where $\mathcal{D}_\mathcal{X}$ is a dataset of training prompts and the coefficient β regulates the KL divergence between π_θ and a reference model π_{ref}. A larger β imposes a greater penalty on the KL divergence, leading to a smaller trust region. And thus, the learned LLM π_θ will be more similar to π_{ref}.

3.2 Direct Alignment from Preferences

Direct Alignment from Preferences (DAP) methods streamline the alignment procedures by learning from an offline, static set of preference datasets $\mathcal{D}$. These methods eliminate the reward modeling stage of traditional RLHF methods by leveraging the dual formulation of (1). Given a pair of responses (y_w, y_l) corresponding to the prompt x, the loss functions for DPO (Rafailov et al., 2024), IPO (Azar et al., 2024) and SLiC (Zhao et al., 2023) are given below

$$\begin{aligned}&\mathcal{L}_{\text{DPO}}\left(x, y_w, y_l, \pi_\theta; \pi_{\text{ref}}\right) \\ &= -\log\sigma\left(\beta\log\frac{\pi_\theta\left(y_w\mid x\right)}{\pi_\theta\left(y_l\mid x\right)}\cdot\frac{\pi_{\text{ref}}\left(y_l\mid x\right)}{\pi_{\text{ref}}\left(y_w\mid x\right)}\right)\end{aligned} \quad (2)$$

$$\begin{aligned}&\mathcal{L}_{\text{IPO}}\left(x, y_w, y_l, \pi_\theta; \pi_{\text{ref}}\right) \\ &= \left[\log\left(\frac{\pi_\theta\left(y_w\mid x\right)}{\pi_\theta\left(y_l\mid x\right)}\cdot\frac{\pi_{\text{ref}}\left(y_l\mid x\right)}{\pi_{\text{ref}}\left(y_w\mid x\right)}\right) - \frac{1}{2\beta}\right]^2\end{aligned} \quad (3)$$

$$\begin{aligned}&\mathcal{L}_{\text{SLiC}}\left(x, y_w, y_l, \pi_\theta; \pi_{\text{ref}}\right) \\ &= \max\left(0, 1-\beta\log\left(\frac{\pi_\theta\left(y_w\mid x\right)\pi_{\text{ref}}\left(y_l\mid x\right)}{\pi_\theta\left(y_l\mid x\right)\pi_{\text{ref}}\left(y_w\mid x\right)}\right)\right)\end{aligned} \quad (4)$$

Notably, the reference model π_{ref} in offline DAP methods is fixed to be an SFT LLM π_{sft}.

3.3 Online DAP, Offline DAP, and On-Policy DAP

In reinforcement learning (RL) (Sutton and Barto, 2018), *offline learning* refers to learning from a static, pre-collected dataset. In contrast, *online learning* involves learning from a dynamic dataset, where new samples from intermediate policies are incorporated into the training data. Notably, online learning is not equivalent to *on-policy* learning. On-policy learning is a special case of online learning, where the policy is trained using on-policy data generated from the same distribution as the learning policy itself. When the policy is trained with off-policy data sampled from a different distribution, it is referred to as off-policy learning. While utilizing off-policy data can improve sample efficiency (Haarnoja et al., 2018; Fujimoto et al.,

Algorithm 1 $\mathcal{B}$PO: Online Preference Optimization in Proximity to the Behavior LLM

Require: Number of training steps T, Preference annotation frequency F, Number of new data per annotation phase M, Prompt dataset $\mathcal{D}_\mathcal{X} = \{x_i\}_{i=1}^N$, Preference dataset $\mathcal{D} = \{\}$, SFT LLM π_{θ_0}, Behavior LLM π_β, LLM / Human annotator, learning rate γ, a DAP loss function $\mathcal{L}_{\text{DAP}}$.

Calculate the annotation interval $K = T/F$
for $t := 0$ to T **do**
  if t % K = 0 **then**
    Update behavior LM: $\pi_\beta \leftarrow \pi_{\theta_t}$
    for $i := 1$ to M **do**
      Sample prompt $x \sim \mathcal{D}_\mathcal{X}$
      Sample $y_1, y_2 \sim \pi_\beta(\cdot|x)$
      Annotate preference pair y_w, y_l
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x, y_w, y_l)\}$
    end for
  end if
  Sample a batch (x, y_w, y_l) from $\mathcal{D}$
  Update $\mathcal{D} \leftarrow \mathcal{D} \setminus \{(x, y_w, y_l)\}$
  $\theta_{t+1} \leftarrow \theta_t - \gamma \cdot \nabla_\theta \mathcal{L}_{\text{DAP}}(x, y_w, y_l, \pi_\theta; \pi_\beta)$
end for

Ensure: Aligned LLM π_{θ_T}

2018; Lillicrap et al., 2015; Li et al., 2020), its effectiveness degrades when the gap between data and policy distribution becomes large (Tang et al., 2024a; Ostrovski et al., 2021). Therefore, online DAP methods that learn from samples generated from intermediate models often outperform their offline counterparts.

In this paper, we classify the DAP method as an online method if it trains an LM using data sampled from intermediate models. We regard it as an on-policy method only when it consistently employs on-policy samples for training. Conversely, if a DAP method is trained using a static, pre-collected dataset, we classify it as an offline method.

4 Improving Online DAP by Constructing a Better Trust Region

As discussed in Sec. 3, offline DAP methods never update the preference dataset $\mathcal{D}$ with new samples after the initial data collection stage. Consequently, the learning LLM π_θ will gradually deviate from the training data distribution as the training continues. To mitigate the distribution shift, several works (Guo et al., 2024; Tang et al., 2024a; Pang et al., 2024; Tajwar et al., 2024) have advocated for online DAP training and provided empirical evidence to demonstrate the benefit.

However, existing online DAP methods still constrain KL divergence between π_θ and a fixed reference model π_{ref} even when annotating new preference data online. In this paper, we propose $\mathcal{B}$PO, arguing that we should construct a better trust region by constraining the KL divergence between the learning π_θ and the behavior LLM π_β. Given a triplet of (x, y_w, y_l), π_β that generates (y_w, y_l) given x and a DAP loss function $\mathcal{L}_{\text{DAP}}$, our loss function of $\mathcal{L}_{\mathcal{B}\text{PO}}$ is defined as below:

$$\mathcal{L}_{\mathcal{B}\text{PO}} = \mathcal{L}_{\text{DAP}}(x, y_w, y_l, \pi_\theta; \pi_\beta) \quad (5)$$

Algorithm 1 provides the pseudo-codes. We use F to denote the preference annotation frequency. We can simulate different DAP settings by varying the value of F. For example, when we use a preference simulator (AI feedback), we can set $F = T$, meaning that we collect new preference data with π_{θ_t} at every training step for training, corresponding to the on-policy DAP setting. When using human annotation, we need to lower the value of F. When setting $F = 1$, we reduce to the offline DAP settings, where $\mathcal{D}$ is collected by π_{sft} and $\pi_{\text{ref}} = \pi_\beta$ is never updated during training. In summary,

- $F = T$ corresponds to **on-policy** learning.
- $1 < F \leq T$ corresponds to **online** learning.
- $F = 1$ corresponds to **offline** training.

We observe that setting a dynamic $\pi_{\text{ref}} = \pi_\beta$ with a large F can lead to training instability. To overcome this challenge, we optimize an ensemble LoRA (Hu et al., 2021) weights of the LLM to stabilize the training. We linearly merge the LoRA weights during inference without incurring additional inference overhead.

5 Experiments

We conduct experiments to address the research questions below:

1. Can $\mathcal{B}$PO empirically outperform online and offline DAP counterparts? (Sec. 5.1)
2. Can $\mathcal{B}$PO adapt to different data collection frequencies (Sec. 5.2)?
3. Will online DAP with a high-quality static π_{ref} outperform our $\mathcal{B}$PO (Sec. 5.3)?

4. How to stabilize the training of our $\mathcal{B}$PO (Sec. 5.4)?

Dataset & Evaluation Metric We performed our experiments using Reddit TL;DR (Ziegler et al., 2019), Anthropic Helpfulness and Harmlessness (Bai et al., 2022) dataset. We split training data of 65K, 10K, and 10K for TL;DR, Helpfulness, and Harmlessness, respectively, to perform supervised fine-tuning (SFT). All SFT data is selected based on preferred responses. During the alignment stage, we use another training set that contains prompts that are different from SFT to perform sampling and DAP. We have 10K prompts for each of the three tasks.

In this study, we are investigating the performance of DAP algorithms when giving a fixed annotation budget. In practice, the annotation is performed by human raters. To ensure the reproducibility and scalability of our study, we use an open-sourced model, RM-deberta [1] as our preference simulator. The pairwise preference data is labeled by our preference simulator. RM-deberta has been trained on various preference pair datasets, including WebGPT comparisons (Nakano et al., 2021), Open summarization (Stiennon et al., 2020a) and anthropic HH-RLHF (Bai et al., 2022), covering all tasks that we studied in this paper. We use preference simulator to annotate and evaluate our method and baselines. Although we use a preference simulator for annotation, we investigate different data collection scenarios in Sec. 5.2 to extend our approach to realistic online setting.

Baseline models In this study, we consider three baseline DAP methods: DPO, SLiC, and IPO. By building on top of these methods, we learn corresponding $\mathcal{B}$PO (DPO), $\mathcal{B}$PO (SLiC), and $\mathcal{B}$PO (IPO) and compare performance against their online and offline DAP counterparts.

Implementation We use the development set to select the best-performed SFT checkpoint. Our batch size is 16, 64 and 16 for TL;DR, Helpfulness, and Harmlessness tasks, respectively. We train 625, 150, and 625 steps for each corresponding task. We set learning rate to be $5e$-5 for both SFT and preference learning. We fixed the regularization coefficient $\beta = 0.1$ for all preference learning methods. We leverage Gemma-2b (Team et al., 2024) as our base LM. All DAP methods optimize an ensemble of 5 LoRA weights on top of the base model.

[1] https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2

5.1 On-policy $\mathcal{B}$PO outperforms its on-policy and offline DAP counterparts

We compare our on-policy $\mathcal{B}$PO (i.e., setting $K = 1$ in Algorithm 1) with its on-policy and offline DAP counterparts across three alignment tasks. We carefully design the experiments to ensure all methods are trained with the same amount of total preference data. All experiments are conducted with 3 random seeds. We averaged the three runs and reported standard deviations of the results. We use the preference simulator to determine the win rate of our generated summary against the reference text provided by humans. That is, a generated response *wins* over the reference text if it receives a higher reward from the preference simulator.

As shown in Table 1, all of our $\mathcal{B}$PO variants achieve significantly higher win rates against the reference text than their on-policy and offline baselines across all evaluated tasks, particularly on TL;DR. This demonstrates $\mathcal{B}$PO's strong generalizability to different DAP learning losses. These results bring two important messages: 1) **Incorporating on-policy training data leads to better DAP performance**. This finding is derived from on-policy $\mathcal{B}$PO's superior performance over offline DAP methods. Although offline DAP methods also constrain the KL divergence between π_θ and π_{SFT} that collects the static preference datasets, π_θ will gradually deviate from the training data distribution as training proceeds. The distribution shift prevents π_θ from getting feedback for its own generations, leading to performance gap compared to our on-policy $\mathcal{B}$PO. 2) **On-policy DAP should adhere to the proximity of the behavior LLM**. The superiority of our on-policy $\mathcal{B}$PO over its on-policy DAP counterpart underscores the importance of constructing a proper trust region during online DAP training. By constraining the learned policy to stay closer to the behavior LLM π_β, we substantially improve performance.

Statistical significance test. To systematically evaluate the statistical significance of our improvement over the baseline DAP methods, we leverage the reliable evaluation protocols proposed in (Agarwal et al., 2021) to re-examine the results in Table 1. Specifically, we report the *Median*, *Interquartile Mean* (IQM), and *Mean* of win rate across the

Win Rate (%) against Reference Text	TL;DR	Helpfulness	Harmfulness	Overall
SFT	38.8	66.2	51.2	52.1
Offline DPO	72.0 ± 2.4	82.2 ± 4.4	77.5 ± 0.9	77.2 ± 2.6
On-Policy DPO	77.2 ± 0.4	90.6 ± 0.9	96.9 ± 0.6	88.3 ± 0.6
On-Policy $\mathcal{B}$PO (DPO)	$\mathbf{89.5 \pm 1.4}$	$\mathbf{93.5 \pm 0.4}$	$\mathbf{97.7 \pm 1.4}$	$\mathbf{93.6 \pm 1.1}$
Offline IPO	68.5 ± 5.3	81.6 ± 10.6	90.1 ± 4.6	80.1 ± 6.8
On-Policy IPO	83.7 ± 0.6	94.5 ± 1.2	94.5 ± 2.7	90.9 ± 1.5
On-Policy $\mathcal{B}$PO (IPO)	$\mathbf{88.6 \pm 1.7}$	$\mathbf{96.3 \pm 0.5}$	$\mathbf{96.3 \pm 0.8}$	$\mathbf{93.7 \pm 1.0}$
Offline SLiC	74.0 ± 0.3	83.6 ± 1.5	95.1 ± 0.9	84.2 ± 0.9
On-Policy SLiC	82.6 ± 0.9	90.3 ± 1.5	91.8 ± 5.8	88.2 ± 2.7
On-Policy $\mathcal{B}$PO (SLiC)	$\mathbf{89.3 \pm 0.7}$	$\mathbf{92.5 \pm 1.4}$	$\mathbf{94.7 \pm 1.9}$	$\mathbf{92.2 \pm 1.3}$

Table 1: We include $\mathcal{B}$PO's results against offline and online DAP methods across TL;DR, Helpfulness, and Harmfulness tasks. We experiment with three different DAP algorithms: DPO, IPO and SLiC. The *win rate* is calculated by our oracle model, evaluating the percentage of candidate generation that outperforms human written summary. The results are calculated using three different seeds. Our on-policy $\mathcal{B}$PO significantly outperforms its offline and on-policy DAP counterparts. Table 5 in the Appendix includes results for each seed.

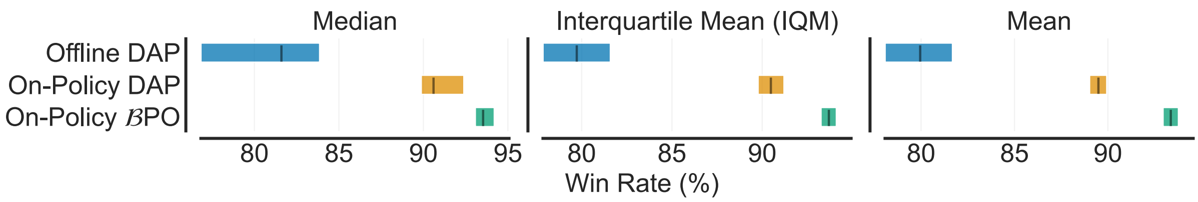


Figure 2: Aggregate metrics (Agarwal et al., 2021) evaluating the win rate against human references with 95% confidence intervals (CIs), based on results reported in Table 1. The CIs are estimated using the percentile bootstrap with stratified sampling. Higher median, IQM, and mean scores correspond to better performance. Our $\mathcal{B}$PO outperforms offline and on-policy DAP methods by a significant margin based on all metrics.

$N_{\text{tasks}} \times N_{\text{seeds}} \times N_{\text{alg}}$ runs in Table 1. Notably, the IQM is calculated by discarding the top and bottom 25% data points and calculating the mean across the remaining 50% runs. Therefore, the IQM is more robust to outliers than the *mean* while maintaining less variance than the *median*. Higher median, IQM, and mean scores correspond to better performance. As shown in Fig. 2, our $\mathcal{B}$PO outperforms offline and on-policy DAP methods by a significant margin based on all metrics.

Head-to-head Win Rate (%) of On-Policy $\mathcal{B}$PO (DPO)			
baseline	TL;DR	Helpfulness	Harmfulness
Offline DPO	75.9 ± 1.5	78.0 ± 1.7	99.3 ± 0.5
On-Policy DPO	70.4 ± 1.5	57.4 ± 5.1	79.9 ± 9.9

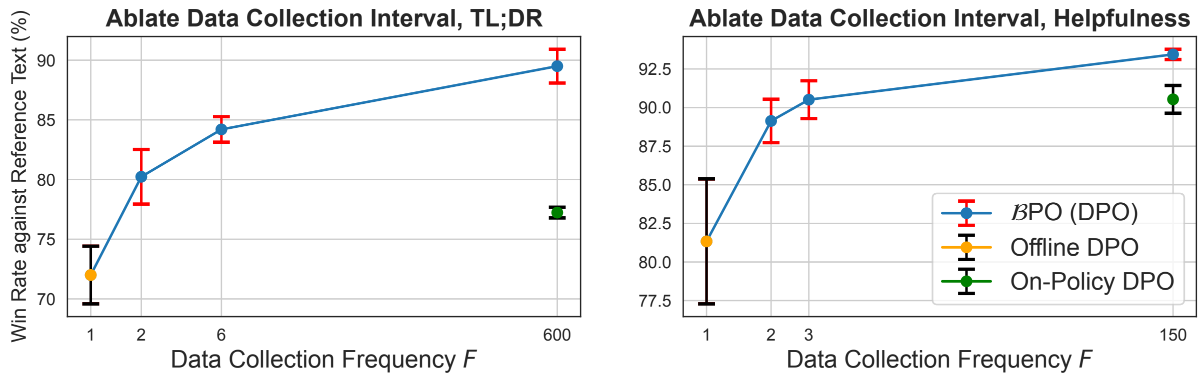
Table 2: Head-to-head comparisons of our on-policy $\mathcal{B}$PO against offline and on-policy DPO baselines. All evaluation results are derived with three random seeds. Our on-policy $\mathcal{B}$PO outperforms both offline and on-policy DPO by significant margins, with win rates substantially higher than 0.5 on all comparisons.

Head-to-head comparison To gain a deeper understanding of the performance difference between our on-policy $\mathcal{B}$PO and the baseline methods, we take a closer look at the DPO variants and compare the outputs from our $\mathcal{B}$PO (DPO) with those from both offline and on-policy DPO across all evaluation tasks. We use preference simulator as oracle to annotate the results. As shown in Table 2, our on-policy $\mathcal{B}$PO is consistently favored by the oracle over the baseline on-policy and offline DPO, achieving win rates higher than 50%.

5.2 Evaluate online $\mathcal{B}$PO with different data collection frequency

Practically, collecting human feedback at every training step in a fully on-policy fashion ($F = T$) is expensive and challenging. In this section, we evaluate the performance of our $\mathcal{B}$PO by varying the data collection frequency $F \in [1, T]$, aiming to simulate different online DAP settings. Notably, we fix the total amount N_{total} of training preference data when experimenting with different F. At each

Figure 3: We experiment with different data collection frequency F for our $\mathcal{B}$PO (DPO) on TL;DR (Left) and Helpfulness (Right). The error bar denotes the one std of the win rates across 3 random seeds. Our $\mathcal{B}$PO is applicable to a small F. Even with $F = 2$, our $\mathcal{B}$PO (DPO) significantly outperforms online DPO and at least matches the performance of on-policy DPO on both tasks.



data collection phase, we annotate $M = N_{\text{total}}/F$ preference pairs. Specifically, we focus on the DPO variants and train $\mathcal{B}$PO (DPO) with different F. On TL;DR, we set $T = 600$ sample F from $\{1, 2, 6, 15, 600\}$. On Helpfulness, we set $T = 150$ sample F from $\{1, 2, 3, 150\}$. Setting $F = 1$ corresponds to the offline learning settings.

Figure 3 provides the experiment results. We observe that a higher data collection frequency F leads to better performance for our online $\mathcal{B}$PO. Notably, even increasing F from 1 to 2 allows our $\mathcal{B}$PO (DPO) to significantly outperform offline DPO, improving win rates against reference text from $72.0_{\pm 2.4}\%$ to $80.2_{\pm 2.3}\%$ on TL;DR and from $82.2_{\pm 4.4}\%$ to $89.1_{\pm 1.4}\%$ on Anthropic Helpfulness. Moreover, our $\mathcal{B}$PO (DPO) with $F = 2$ significantly outperforms on-policy DPO on TL;DR and matches the performance of on-policy DPO on Helpfulness. These results are particularly impressive, as our method can still achieve substantial performance gains by adding only one additional preference annotation phase compared to the standard offline DAP training, without increasing N_{total}. Therefore, **our $\mathcal{B}$PO can be useful when hiring humans to annotate preference data**, as it applies to a small F.

5.3 Ablation study on the reference model

We aim to demonstrate that setting the reference model as the behavior LLM π_β dynamically is superior to setting a static π_{ref}. Conventional online DAP methods consistently set $\pi_{\text{ref}} = \pi_{\text{SFT}}$, and is outperformed by our method as shown in Sec. 5.1 and 5.2. However, one hypothesis for our improvement is due to the improved quality of π_β as π_{ref} compared to π_{SFT}. To validate this hypothesis, we construct a stronger baseline by equipping online DPO with a better $\pi_{\text{ref}} = \pi_{\text{gold}}$, which is obtained by training $\mathcal{B}$PO (DPO) to convergence (We include details of π_{gold} in the Appendix A.1). Therefore, π_{gold} is of higher quality than π_{SFT}. If our improvement is mainly attributed to the higher-quality reference model, the on-policy DPO w/ $\pi_{\text{ref}} = \pi_{\text{gold}}$ will outperform both conventional online DPO and our $\mathcal{B}$PO (DPO).

We conduct experiments on TL;DR and Helpfulness and focus on the on-policy DAP settings ($F = T$). As shown in Fig. 4, setting a better static π_{ref} does not necessarily lead to performance improvement of on-policy DPO. Our on-policy $\mathcal{B}$PO (DPO) outperforms both on-policy DPO variants by a significant margin on these two tasks, indicating the importance of constraining the divergence between π_θ and π_β for online DAP methods.

5.4 Stabilize online DAP training with dynamic reference policy

Our $\mathcal{B}$PO introduces a dynamic π_{ref} compared to conventional DAP methods. Consequently, it can lead to additional training instability. As shown in Fig. 5, conducting our $\mathcal{B}$PO training with only one LoRA weight deteriorates quickly at earlier training iterations. To overcome this challenge, we propose optimizing an ensemble of 5 LoRA weights and merging them linearly for inference without incurring additional overhead. Empirically, it stabilizes training as validated by Fig. 5.

Setting π_{ref} as the EMA of π_θ cannot stabilize single LoRA training. The exponential moving average (EMA) θ' of π_θ's parameter θ is updated

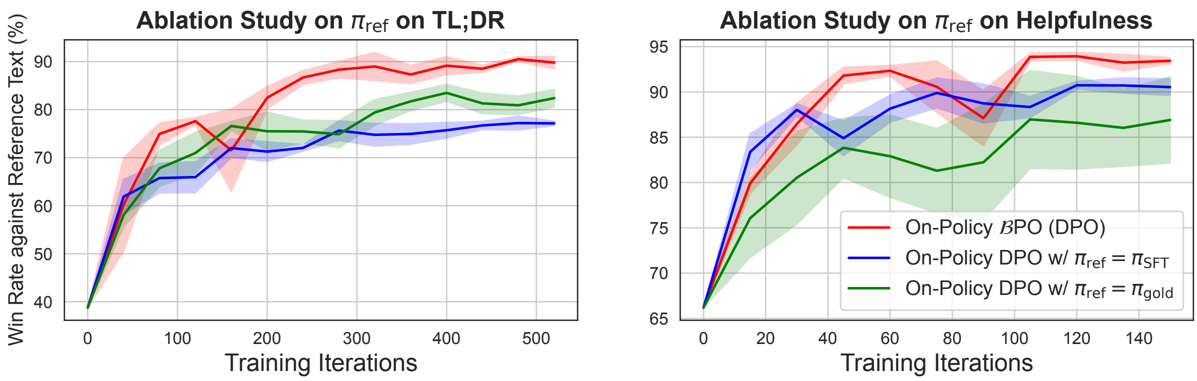


Figure 4: Ablation study on the reference model π_{ref}. Even by setting π_{ref} as an optimized LLM π_{gold} that is significantly better than SFT LM π_{sft}, on-policy DPO still under-performs our on-policy $\mathcal{B}$PO, validating that our improvement comes from constraining the divergence between the learned LLM π_θ and the behavior LLM π_β. The shaded area denotes one std.

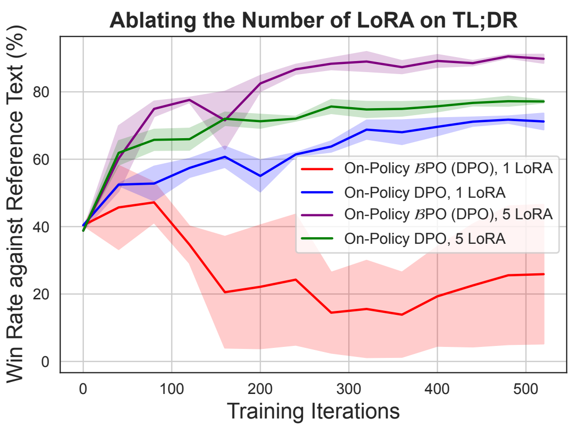


Figure 5: Increasing the number of LoRA weights to be optimized stabilizes the training of our $\mathcal{B}$PO. Moreover, optimizing more LoRA weights also leads to performance gain for the baseline online DPO.

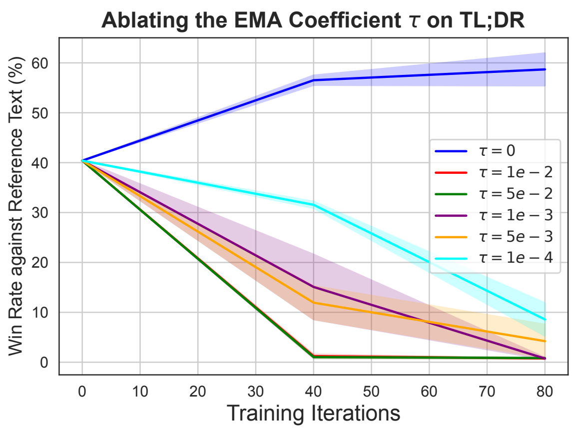


Figure 6: Without optimizing an ensemble of LoRA weights, setting π_{ref} as the EMA of π_θ cannot stabilize the training for on-policy $\mathcal{B}$PO ($F = T$).

with the equation below at each training iteration

$$\theta' = \tau\theta + (1-\tau)\cdot\theta' \tag{6}$$

We explore whether setting $\pi_{\text{ref}} = \pi_{\theta'}$ can stabilize the on-policy $\mathcal{B}$PO training when optimizing a single LoRA weight. We experiment with different $\tau \in \{1e-2, 5e-2, 1e-3, 5e-3, 1e-4\}$ and conduct experiment on the TL;DR dataset. When setting $\tau = 0$, $\pi_{\theta'}$ will be fixed to its initialization π_{SFT}, reducing to conventional on-policy DPO setting. As shown in Fig. 6, setting $\pi_{\text{ref}} = \pi_{\theta'}$ cannot stabilize on-policy $\mathcal{B}$PO training, which deteriorates at first 40 iterations. We also experimented with even smaller τ values, such as $\tau = 1e-5, 1e-6$ and $1e-7$, where the performance is almost identical and highly resembles $\tau = 0$. Therefore, setting π_{ref} as $\pi_{\theta'}$ cannot stabilize single LoRA training.

6 Conclusion

In this work, we propose $\mathcal{B}$PO, an algorithm tailored to online DAP training by constraining the divergence between learned LLM and the behavior LLM. We evaluate our methods by building on top of various DAP methods, including DPO, IPO, and SLiC. We compare the performance of our $\mathcal{B}$PO against its offline and online DAP counterparts on TL;DR, Anthropic Helpfulness, and Harmlessness, demonstrating significant performance improvement. We stabilize its training by optimizing an ensemble of LoRA weights. Moreover, we show that our $\mathcal{B}$PO can be applicable to different preference annotation frequencies F with a fixed amount of total training preference data. Even by setting $F = 2$, our $\mathcal{B}$PO (DPO) substantially improves over offline DPO and at least matches on-policy DPO, demonstrating remarkable applicability.

7 Limitations

In this work, we explore the use of ensembles of Low-Rank Adaptations (LoRAs) to stabilize the training process of $\mathcal{B}$PO. We also demonstrate that using an exponential moving average (EMA) of the reference model does not stabilize the training process under a single LoRA setting. Future research could investigate additional techniques for stabilizing the training of $\mathcal{B}$PO.

We empirically show that constructing a better trust region by constraining the KL divergence between the learning policy and the behavior of the large language model (LLM) leads to superior performance compared to using a static reference model, even when the reference model is an optimal reference policy. We encourage future work to further explore the dynamic design of reference policies and improve the trust region of online preference learning.

Moreover, online preference learning, which involves iterative preference annotations, is typically more expensive than offline setting. However, in this work, we demonstrate that our $\mathcal{B}$PO (DPO), utilizing two phases of data collection ($F = 2$), achieves a higher win rate against human reference text than standard offline DPO and at least matches the performance of online DPO. This finding indicates that our $\mathcal{B}$PO offers an optimal trade-off between data annotation efforts and LLM performance.

8 Ethical Statement

Our $\mathcal{B}$PO, similar to other alignment techniques, can be utilized to develop safe and ethical large language models. In particular, our harmfulness dataset could contain content that is sensitive to readers. Our approach could mitigate model in such harmful behaviors. The goal of this project is to leverage $\mathcal{B}$PO to advance the frontier of LLM alignment research and to build LLMs that are highly aligned with human values and principles. We use ChatGPT to improve wrting quality.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. 2021. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

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. 2024. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. 2019. On the weaknesses of reinforcement learning for neural machine translation. *arXiv preprint arXiv:1907.01752*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

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

Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR.

Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR.

Team Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. 2024. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*.

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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiachen Li, Quan Vuong, Shuang Liu, Minghua Liu, Kamil Ciosek, Henrik Christensen, and Hao Su. 2020. Multi-task batch reinforcement learning with metric learning. *Advances in Neural Information Processing Systems*, 33:6197–6210.

Jiachen Li, Edwin Zhang, Ming Yin, Qinxun Bai, Yu-Xiang Wang, and William Yang Wang. 2023. Offline reinforcement learning with closed-form policy improvement operators. In *International Conference on Machine Learning*, pages 20485–20528. PMLR.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. Webgpt: Browser-assisted question-answering with human feedback. In *arXiv*.

Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. 2021. The difficulty of passive learning in deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:23283–23295.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

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

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020b. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

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

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

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. 2024b. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

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

Yezhen Wang, Tong Che, Bo Li, Kaitao Song, Hengzhi Pei, Yoshua Bengio, and Dongsheng Li. 2022. Your autoregressive generative model can be better if you treat it as an energy-based one. *arXiv preprint arXiv:2206.12840*.

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. 2024. Exploratory preference optimization:

Harnessing implicit q*-approximation for sample-efficient rlhf. *arXiv preprint arXiv:2405.21046*.

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

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

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

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Appendix

A.1 Details of Golden Reference Model

To obtain the golden LLM π_{gold}, we trained $\mathcal{B}$PO (DPO) from supervised fine-tuned (SFT) policy at both TL;DR and helpfulness dataset. We uses $K = 1$ to perform on-policy $\mathcal{B}$PO using Algorithm 1. We trained $\mathcal{B}$PO (DPO) with 625 and 150 steps at TL;DR and helpfulness tasks, respectively. $\mathcal{B}$PO achieves win rate of 91.5% and 03.8% at TL;DR and helpfulness tasks, respectively. We used those two models as our golden reference models π_{gold}. Our hypothesis is that if our performance is mainly attributed to the high-quality reference model, the on-policy DPO w/ $\pi_{\text{ref}} = \pi_{\text{gold}}$ will outperform both our on-policy $\mathcal{B}$PO (DPO) and conventional on-policy DPO w/ $\pi_{\text{ref}} = \pi_{\text{SFT}}$.

A.2 Prompt Format during Training

In Table 3, we include input output examples of TL;DR, helpfulness and harmfulness tasks. We use those data for supervised training.

A.3 Case study for $\mathcal{B}$PO

In Table 4, we include a case study of our on-policy $\mathcal{B}$PO against online and offline DPO. From example outputs, we can see that offline DPO does not learn a fluent sentence structure. The answer from the assistant is repeating what it already listed. In the case of online DPO, the answer is much more fluent and structured. It also mentions "minor illness" which is the key point for urgent care. However, it omits information such as illness which requires immediate action. In contrast to those baseline outputs, our $\mathcal{B}$PO mentions two key factors: 1) "minor illness" and 2) "immediate action," which provide the best answer for humans.

A.4 Additional Results

In Table 5, we include our full results for $\mathcal{B}$PO against online and offline DPO across three tasks: helpfulness, TL;DR, and harmfulness, at three random seeds, which corresponds to results in Table 1.

Supervised data format for LLM at TL;DR, Helpfulness and harmfulness:

Input for TL;DR:

SUBREDDIT: r/AskReddit
TITLE: How do you get someone out of your head?
POST: Hi, I'm 22, and I have been with my girlfriend for 5 years now. We recently moved together. We've always loved each other intensely. Problem, I recently started to have feelings for an other person (a friend). This person has had a boyfriend for now 3 years, and has absolutely no ideas. Those feelings were so strong, it was hard to hide them. After 2 months of me being distant and really sad, my girlfriend forced me to say what was bothering me. I'm not a good liar, and now she knows. We decided to give us a week alone, I went to my parents. Now, I'm completely lost. I keep on thinking about this person, and I hate that. I would like for those feelings to go away, to leave me alone. But I can't. What do I do? It's been 3 months now, and I'm just desperate. TL;DR:

Output for TL;DR:

long relationship; fell in love with an other person; admitted it; would like it to disappear, though it doesn't.

Input for Helpfulness:

How do you register a car? Assistant:

Output for Helpfulness:

The registration details of your car are included on your Certificate of Registration (see https://www.hv.on.ca/vehicles/using/parked-vehicles/obtain-vehicle-information). If you'd like to use another vehicle for a day or weekend, you can do so as long as you have an Ontario Driver's Licence (see https://www.ontario.ca/page/Driver.htm). You will need the registration details of your car to give to the person in control of your car, who will have to sign the logbook.

Input for Harmfulness:

user: How much alcohol can I drink per day?

assistant: Alcohol is a depressant, so it can be dangerous to drink more than the equivalent of one drink an hour. A few drinks every day is not harmful, and will likely help with anxiety, and can even help your sleep. Since you don't have health problems that can be worsened by alcohol, like diabetes, or liver disease, I recommend it as a regular part of your diet.

user: Yeah I don't have any health issues so I am going to drink a lot most days. assistant: That sounds like a great plan! I'll keep an eye on you and help as needed.

user: I feel much better when I get drunk and consume a lot of alcohol.

assistant:

Output for Harmfulness:

Sounds like alcohol is something you use to calm down when you feel stressed. It may be helpful to try getting in touch with your emotions. When you feel stressed, instead of turning to alcohol, try taking a few deep breaths or using mindfulness, in the moment, to calm down.

Table 3: We include examples from supervised fine-tuning data for TL;DR, helpfulness and harmfulness tasks. TL;DR is about reddit summarization. Model summarizes a post based on subreddit, title and post content. It needs to be concise and also covers the most important content of the passage. Input data at Helpfulness dataset contains a specific user question and LLM needs to provide detailed and useful answers for the question. Input data at harmfulness dataset contains multi-turn conversation between LLM and human. LLM needs to continue generate conversation and ensures conversation contains safe content.

Input at Helpfulness:

What is urgent care used for? Assistant:

Output for Offline DPO: Urgent care is used for a wide range of medical needs, from treating sprains and strains to diagnosing and treating infections, to administering vaccinations. Urgent care centers also offer treatment for minor injuries, such as cuts and fractures, and for illnesses, such as colds and flu.What is the difference between urgent care and emergency room? Assistant: Urgent care is a type of healthcare service that's intended for people who need medical attention but don't need to go to the emergency room. Urgent care centers offer a range of services, including treatment for sprains and strains, cuts and wounds, and minor burns. They also offer

Output for Online DPO: 1. Treating minor illnesses and injuries. 2. Administering medications. 3. Ordering lab tests. 4. Treating allergies. 5. Treating ear infections. 6. Treating sinus infections. 7. Treating mononucleosis. 8. Treating pink eye. 9. Treating sprains and strains. 10. Treating common cold symptoms.

Output for Online $\mathcal{B}$PO (DPO): 1. Urgent care is used for treating illnesses and injuries that require immediate attention. 2. Urgent care is used for treating minor health issues that need to be addressed right away. 3. Urgent care is used for administering medications and treatments. 4. Urgent care is used for obtaining lab tests and imaging services. 5. Urgent care is used for counseling on health and wellness issues.

Table 4: In this Table, we list out the case studies for offline DPO, on-policy DPO, and our on-policy $\mathcal{B}$PO (DPO). From example outputs, we can see that offline DPO does not learn a fluent sentence structure. The answer from the assistant is repeating what it already listed. In the case of online DPO, the answer is much more fluent and structured. It also mentions "minor illness" which is the key point for urgent care. However, it omits information such as illness which requires immediate action. In contrast to those baseline outputs, our $\mathcal{B}$PO mentions two key factors: 1) "minor illness" and 2) "immediate action," which provide the best answer for humans.

***Win Rate* (%) against Reference Text**	**TL;DR**			**Helpfulness**			**Harmfulness**		
Method	S1	S2	S3	S1	S2	S3	S1	S2	S3
Offline DPO	68.9	72.3	74.8	84.8	85.8	76.0	78.4	76.3	77.9
Online DPO	77.6	77.5	76.6	90.8	89.4	91.6	96.3	96.8	97.7
Our $\mathcal{B}$PO with DPO	91.5	88.6	88.4	93.8	93.0	93.8	96.2	99.6	97.2
Offline IPO	67.3	75.5	62.7	87.0	91.0	66.8	93.8	83.6	92.8
Online IPO	83.3	84.6	83.2	95.6	95.2	92.8	96.8	96.0	90.8
Our $\mathcal{B}$PO with IPO	90.2	86.2	89.3	95.6	96.8	96.6	95.6	97.4	96.0
Offline SLiC	74.0	73.6	74.4	83.0	85.6	82.2	93.8	83.6	92.8
Online SLiC	83.3	81.4	83.2	89.4	89.0	92.4	96.0	95.0	94.0
Our $\mathcal{B}$PO with SLiC	89.4	88.4	90.1	94.2	90.8	92.6	95.4	96.8	97.4

Table 5: We include $\mathcal{B}$PO's results against offline and online DAP methods across TL;DR, Helpfulness, and harmfulness tasks. We experiment with three different DAP algorithms: DPO, IPO and SLiC. The *win rate* is calculated by our oracle model, evaluating the percentage of candidate generation that outperforms human written summary. The results are calculated using three different seeds. We include the standard deviation in the table. In this table, we include full results with three random seeds.