Extended Abstract

Motivation We implement Direct Preference Optimization (DPO) and curriculum learning methods to fine-tune the Qwen 2.5 0.5B base model on instruction-following tasks. More specifically, we study the effect of curriculum learning DPO on performance in comparison to the base approach. We are interested in the application of curriculum learning to supervised learning tasks in computer vision; previous studies [1] have examined the effect of this training approach in object detection and instance segmentation experiments and show faster convergence and more accurate results (especially for unbalanced datasets). We wanted to study its application in the fine-tuning of large language models.

Method In contrast to the standard RLHF approach of using a trained reward model to evaluate the performance of the fine-tuned policy, DPO directly optimizes over the model using a parameterized loss function in closed form. We run supervised fine-tuning on the Qwen base model using an instruction dataset and then DPO on a preference dataset of more advanced instructions. We then analyze two approaches to curriculum learning: the standard non-iterative approach (using the SFT model as the reference model) and an iterative approach, where the reference model is continuously updated after each iteration. The intuition behind curriculum learning is that preference pairs that are more dissimilar (i.e. one response is more clearly better) are easier for the model to learn and thus should be trained on first. Once the model improves on easier data, it is trained on harder examples.

Implementation In order to implement curriculum learning DPO, we process the original UltraFeedback dataset into sets of preference pairs. The original UltraFeedback dataset has 4 responses to a prompt, each response scored by GPT-4 on criteria like truthfulness and accuracy. We take the highest-scored response and create 3 preference pairs with the other lower-scoring responses, filtering out any pairs with the same quality. We then rank all pairs across the dataset by absolute score difference. The easiest pairs to learn (the pairs with the largest gap in scores) are fed into the model first.

Results We see improvement in the instruction following task for all DPO approaches in comparison the base SFT model. However, the standard DPO approach scored more highly in comparison to both the iterative and non-iterative curriculum learning approaches (with the non-iterative approach performing the worst).

Discussion We don't see any improvements in the instruction following task with a curriculum learning approach. We hypothesize this could be for several reasons. One, we only perform one epoch of training for each approach due to limited compute. It is possible that the benefits of curriculum learning require multiple epochs to become visible. Second, previous literature suggests that curriculum learning might be most effective when the dataset of examples is unbalanced and the model loss doesn't initially decrease. In this way, the curriculum approach can "kickstart" the learning process. Because we already performed supervised fine-tuning on the smol-smoltalk dataset, our model was already warm-started and simply might not have needed curriculum learning to perform well on harder examples. Additionally, the preference dataset included a wide and balanced range of preference pair difficulties. Future work in this area should test the effect of curriculum learning on non-warmstarted models, although even if proven effective here it might still be unnecessary in practical applications.

Conclusion In the SFT $\rightarrow$ DPO pipeline we used to fine-tune a large language model on instruction following tasks, the use of curriculum learning DPO was not shown to be effective at improving performance. We saw improvements over the SFT-trained model but lower scores than the standard DPO approach. We believe the use of SFT to warm-start the model, and the balanced nature of the preference dataset, negated any advantanges that curriculum learning can offer.

RL Methods on Large Language Models: A Curriculum Learning Approach

Stanford CS224R Default Project

Jack Hung, Luke Moberly
Department of Computer Science
Stanford University
{jjhung66, lmoberly}@stanford.edu

Key Information: Our TA mentor is Sergio Charles. We have no external collaborators or mentors, and we are not sharing the project with other classes. We contributed equally to this project. Luke implemented DPO and RLOO (which we did not evaluate for this report), while Jack implemented the dataloaders and curriculum learning.

Abstract

We implement Direct Preference Optimization (DPO) and curriculum learning methods to fine-tune the Qwen 2.5 0.5B base model on instruction-following tasks. More specifically, we study the effect of curriculum learning DPO on performance in comparison to the base approach. We analyze two approaches to curriculum learning: the standard non-iterative approach (using the SFT model as the reference model) and an iterative approach, where the reference model is continuously updated after each iteration. We find improvement in all three DPO approaches over the SFT baseline, but curriculum learning offered no performance benefit over the standard DPO training. We believe the use of SFT to warm-start the model, and the balanced nature of the preference dataset, negated any advantanges that curriculum learning can offer.

1 Introduction

We explore the implementation of RL algorithms to improve performance on large language models (LLMs). Although pretraining is effective at producing intelligible text, outputs to prompts often lack precision. Reinforcement learning (RL) addresses some of the short-comings of simple pretrained models. First, it aligns the model with human preferences. Reward models are used to judge responses along pre-set criteria, such as truthfulness and helpfulness, which enables the base model to align itself more closely with these values. When humans, or reward models trained to mimic human preferences, indicate one response as preferable over another, the model learns these differences. Second, the RL pipeline optimizes base models on downstream tasks performance. Rather than just imitating existing data, RL methods can surpass expert performance by optimizing for an objective reward (rather than the expert's performance as the goal).

We implement the Direct Preference Optimization (DPO) RL method to fine-tune the Qwen 2.5 0.5B base model on instruction-following tasks. The mechanism of DPO is relatively simple: the data consists of a prompt, a preferred response, and a dispreferred response, and the loss objective encourages the model to minimize distance to the preferred response and decrease likelihood of the dispreferred response. We explore two extensions to the standard DPO model: sequential curriculum learning and categorical learning. The intuition behind curriculum learning is that the model should start training with easier pairwise examples (responses that are more dissimilar) and slowly increase the difficulty in training examples. This can be done by scoring preferential data (e.g. response A is 2x worse than response B) and ordering the input data by difficulty. Sequential ordering is an objective easiest to hardest ordering of all response pairs, while categorical ordering breaks difficulties into buckets that response pairs are grouped into.

2 Related Work

We use the vanilla version of DPO as proposed in "Direct Preference Optimization: Your Language Model is Secretly a Reward Model" [2]. The DPO algorithm is an improvement over the more traditional reinforcement learning from human feedback (RLHF) approaches. RLHF is a complex mechanism and often unstable; it trains a reward model to reflect human preferences and then uses RL to fine-tune the base model to maximize this estimated reward without straying too far from the original model. The DPO method introduces a new parameterization of the RLHF reward model that extracts the optimal policy in closed form and enables the use of a simple classification loss function. The resulting algorithm is more stable, more performant, and computationally lighter.

We implemented DPO with curriculum learning as our extension to compare it to the DPO baseline, as proposed in Curri-DPO: Enhancing Alignment using Curriculum Learning & Ranked Preferences [3]. The hypothesis of Curri-DPO is two-fold:

1. Using multiple preference pairs per prompt acts as a form of data augmentation, similar to computer vision training methods using rotated or skewed images as additional training data
2. Systematically introducing these preference pairs by relative quality rating into the model improves over the base approach to training

The authors find a 7.5% improvement in performance using Curri-DPO in comparison to the base DPO approach.

3 Approach

3.1 Supervised Fine-Tuning

We first start by fine-tuning the base Qwen 2.5 0.5B model using SFT on a lighter version of the Smoltalk databsase: smol-smoltak [4]. Smoltalk is a sythentic dataset of 1M samples aimed to improve model instruction following and covers a diverse range of tasks, such as text editing, rewriting, summarization, and reasoning. The smol-smoltalk dataset reduces the length of conversations found in the original dataset, includes less task-specific data (e.g. no function calling), and does not include advanced math examples.

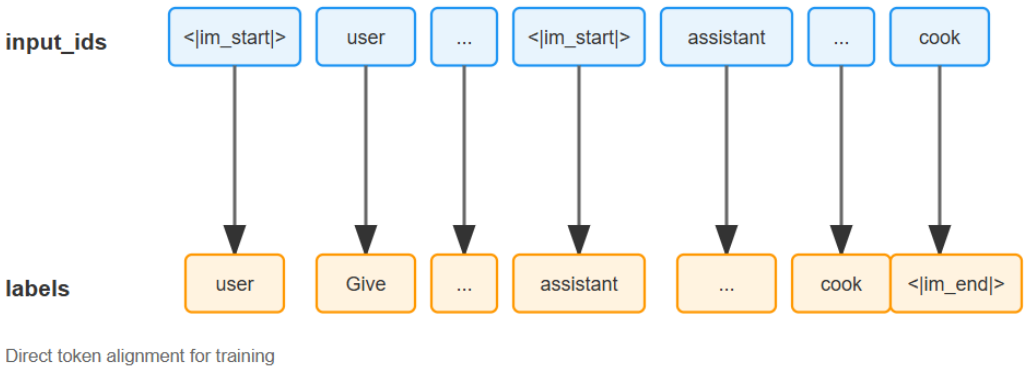

Figure 1: Token Prediction

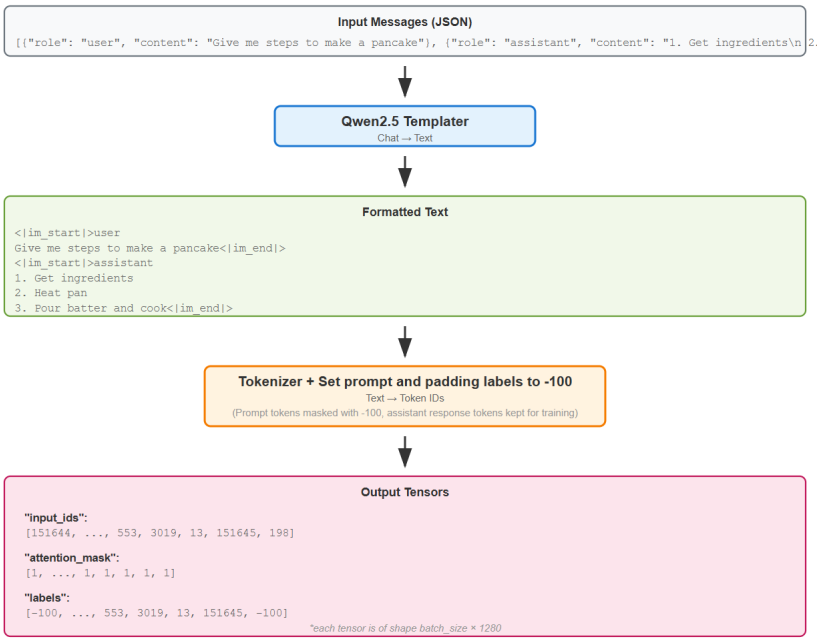

Figure 2: Tokenization

SFT uses the same next-token prediction objective that is used in the pre-training of base language models, yet masks the loss from the query tokens.

The conversations from the SmolTalk dataset are formatted as lists of dictionaries. We first convert these conversations into a templated conversation using Huggingface's Qwen apply template function. The apply template function adds additional system messages, which we trim from the templated results (since it takes up unnecessary space). Then, these messages are tokenized to a length of 1280 tokens with right padding and truncation. Since we only want the model to predict the response, we also set the labels corresponding to the prompt and padding to -100. This prevents loss calculation during the training loop.

The supervised learning objective is optimized over prompts x and completions y that are drawn from the dataset. We optimize the objective as follows:

$$\max_{\theta} \mathbb{E}_{x,y \in D} \sum_{t=1}^{|y|} \log \pi_{\theta}(y_t \mid x, y_{<t}) \tag{1}$$

3.2 Direct Preference Optimization (DPO)

After fine-tuning the base model to produce a reference policy π_{ref}, we run DPO on preference data. We use the UltraFeedback Binarized dataset [5], a pre-processed version of the original UltraFeedback dataset. The original dataset consists of 64k prompts, each with four model completions. A score for each completion was generated by GPT-4 using criteria like helpfulness and honesty. The UltraFeedback Binarized dataset takes the completion with the highest score as the "chosen" response, and one of the three remaining responses at random as the "rejected" response.

The full pipeline is shown in Figure 3.

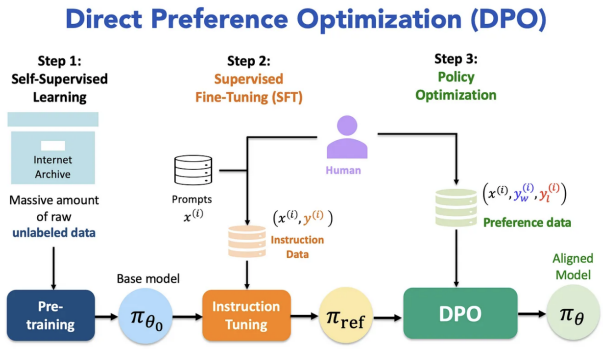

Figure 3: DPO Method [6]

Rather than train a reward model as in RLHF, DPO parameterizes it as a function of the log-likelihoods of the preferred and dispreferred responses. In Rafailov et al. [2], they reformulate the constrained RL problem as a supervised preference classification problem using the following loss objective:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right] \tag{2}$$

The gradient of this loss function can then be calculated as:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher when estimate is wrong}} \left[\underbrace{\nabla_\theta \log \pi(y_w|x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l|x)}_{\text{decrease likelihood of } y_l} \right] \right]$$

As seen, the gradient of the loss function increases the likielhood of the preferred completions y_w and decreases the likelihood of dispreferred completions y_l. In DPO, an offline dataset of preferences is construction $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$ (in our case, we use the UltraFeedback dataset) and the model π_θ is optimized by minimizing $\mathcal{L}_{\text{DPO}}$ for the SFT-trained model π_{ref} and $\mathcal{D}$.

3.3 Curriculum Learning

The intuition behind curriculum learning is that preference pairs that are more dissimilar (i.e. one response is more clearly better) are easier for the model to learn and thus should be fed in first. The model first trains with this easier data and later moves on to examples that are scored closer together as it improves, as shown below.

There are two approaches to curriculum learning that we implement: non-iterative and iterative. The non-iterative approach uses the SFT model as the reference model π_{ref} throughout the entire training process. In contrast, the iterative approach uses the model of the previous iteration as the reference model for the next iteration. The first iteration uses the base SFT model as the reference model. This can be seen in Figure 4. In our iterative curriculum learning experiments, we execute 1 epoch over the easy dataset and 5 epochs each over the medium and hard datasets in order to allow the model to converge. The loss for iterative curriculum learning is:

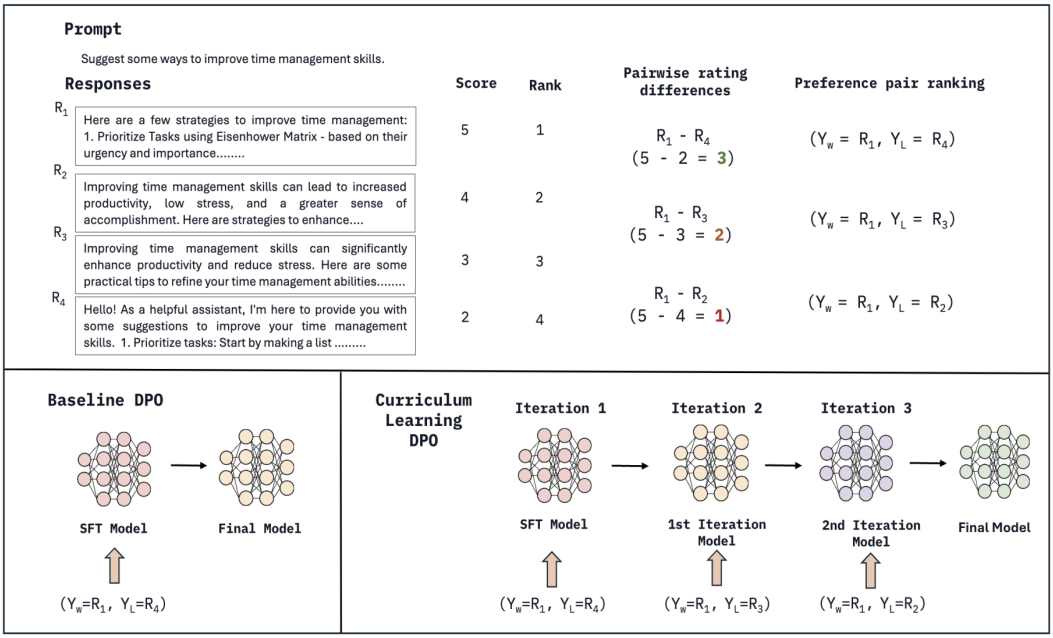

Figure 4: The top section of the figure demonstrates how preference pairs are created in curriculum learning: the highest-scored response is paired with each of the rejected responses and are ranked in descending order of pairwise rating difference. Each iteration of curriculum learning DPO is performed with slightly harder data. This figure outlines the iterative approach to curriculum learning [3].

$$\mathcal{L}_{(\pi_\theta^{i+1};\pi_\theta^i)} = -\mathbb{E}_{(x,y_w^{i+1},y_l^{i+1})\sim\mathcal{D}} \log\sigma\left(\beta\log\frac{\pi_\theta^{i+1}(y_w^{i+1}\mid x)}{\pi_\theta^i(y_w^{i+1}\mid x)} - \beta\log\frac{\pi_\theta^{i+1}(y_l^{i+1}\mid x)}{\pi_\theta^i(y_l^{i+1}\mid x)}\right) \quad (3)$$

3.4 Other Considerations

Initialization with supervised fine tuning is a crucial beginning step. The performance of the initialized model greatly determines the outcome of subsequent models. For this reason, it is important to balance SFT so that the model does not overfit on the dataset, potentially leading to catastrophic forgetting in which the initially language-capable LLM loses its ability to generate intelligible responses.

A second risk is that using general preference ratings for responses during DPO can lead to the model optimizing away from decent quality responses just because they are marginally worse than a prompt in the same example pair. This can lead to unintended behavior in which the model decreases performance by learning against generating the positive characteristics of the rejected responses.

4 Experiments

We run three experiments to study which approach to DPO learning is more performative in contrast to the standard SFT model:

1. Standard DPO vs SFT
2. Non-iterative Curriculum Learning DPO vs SFT
3. Iterative Curriculum Learning DPO vs SFT

We also look at the performance of π_{base} (the pre-trained Qwen 2.5 0.5B Instruct model) vs. π_{SFT}.

We train the base SFT model for one full epoch on the smol-smoltalk dataset, where we see convergence in both train and validation loss. We train all three DPO models on one epoch of the Ultrafeedback-Binarized dataset.

Our evaluation pipeline uses vllm, a high-through and memory efficient inference engine for LLMs [7]. We sample responses from the following models: π_{base}, π_{SFT}, π_{DPO}, $\pi_{\text{DPO-Iter-Curriculum}}$, $\pi_{\text{DPO-NonIter-Curriculum}}$. We then use a parametric reward model for scoring, specifically the Llama 3.1 Nemotron 70B Reward Model [8] [9]. Given the prompt, the reward model generates a score for each model's response. For each prompt, we calculate a per-prompt win-rate binary label, where 1 corresponds to the reward of the trained model being higher and 0 corresponds to the reward of the reference model being higher. The win-rate for each trained model is the average of the binary label over all held-out test prompts.

5 Results & Analysis

5.1 SFT Baseline

Training loss decreased rather quickly when fine-tuning the Qwen base model on the smol-smoltalk dataset using SFT. After one full epoch, we achieved a training loss of 0.585 and a validation loss of 0.483. The training was relatively stable and loss decreased exponentially. The SFT model achieved an 80.0% win-rate on the leaderboard.

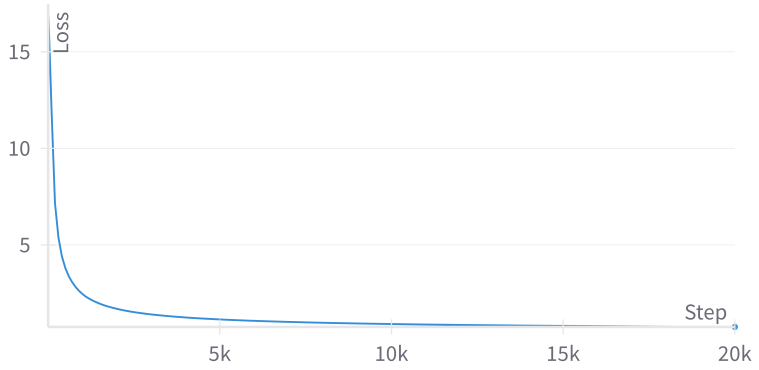

Figure 5: Training Loss of SFT on smol-smoltalk (LR = 1e-06)

5.2 DPO

Training loss also decreased on DPO throughout the epoch, as well as the reward margins and reward accuracies, achieving a final train loss of 0.66 and validation loss of 0.68. The DPO model achieved a 81.0% performance over the base Qwen 2.5 0.5B Instruct model and a 66.0% win-rate over the SFT model. See Figures 12, 13, 14.

Figure 6: Training Loss of DPO

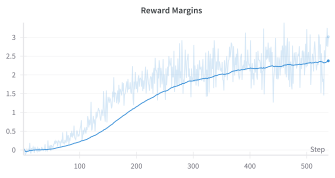

Figure 7: Training Margins (Chosen Rewards - Rejected Rewards) of DPO

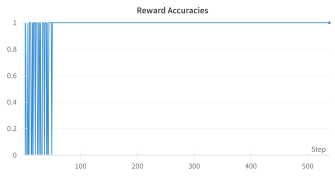

Figure 8: Training Accuracy (Probability of Chosen Response > Probability of Rejected Response) of DPO

5.3 DPO with Iterative Curriculum Learning

We see improvement over the SFT model with iterative curriculum learning, but slightly worse performance compared to the standard DPO approach.

Figure 9: Training Loss of DPO

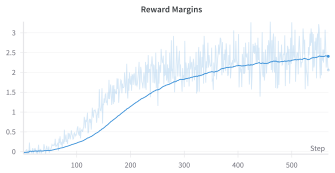

Figure 10: Training Margins (Chosen Rewards - Rejected Rewards) of DPO

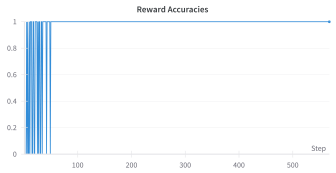

Figure 11: Training Accuracy (Probability of Chosen Response > Probability of Rejected Response) of DPO

5.4 DPO with Non-Iterative Curriculum Learning

We see improvement over the SFT model with non-iterative curriculum learning, but once again slightly worse performance compared to the standard DPO approach.

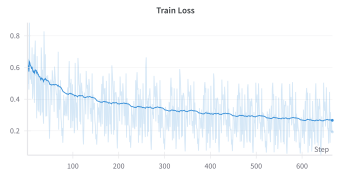

Figure 12: Training Loss of DPO

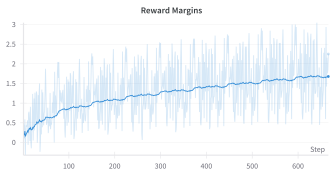

Figure 13: Training Margins (Chosen Rewards - Rejected Rewards) of DPO

Figure 14: Training Accuracy (Probability of Chosen Response > Probability of Rejected Response) of DPO

5.5 All Results

Overall, curriculum learning does not seem to offer any benefit over the standard DPO approach.

Experiment	**Trained Model Win-Rate**
SFT vs Base Qwen 2.5 0.5B Instruct	0.800
DPO vs Base Qwen 2.5 0.5B Instruct	0.810
DPO vs SFT	0.660
Non-Iter Curriculum DPO vs SFT	0.610
Iter Curriculum DPO vs SFT	0.630

Table 1: Win-Rate Results from experiments

6 Conclusion

In the SFT $\rightarrow$ DPO pipeline we used to fine-tune a large language model on instruction following tasks, the use of curriculum learning DPO was not shown to be effective at improving performance. We saw improvements over the SFT-trained model but slightly lower scores than the standard DPO approach. Previous literature suggests that curriculum learning might be most effective when the dataset of examples is unbalanced and the model loss doesn't initially decrease. In this way, the

curriculum approach can "kickstart" the learning process. Because we already performed supervised fine-tuning on the smol-smoltalk dataset, our model was already warm-started and simply might not have needed curriculum learning to perform well on harder examples. Additionally, the preference dataset included a wide and balanced range of preference pair difficulties. Because of this, we believe that the use of SFT to warm-start the model, and the balanced nature of the preference dataset, could have negated any advantanges that curriculum learning can offer.

7 Limitations and Future Work

Due to limited compute, we only perform one epoch of both DPO and SFT. For SFT, this is unlikely to be a huge issue, since we also do not want the model to overfit too much on the dataset. However, for DPO, it is possible that training the model for a longer period of time could lead to better results. In the future, it would be helpful to explore this possibility by learning on more epochs.

Additionally, in our DPO experiment, we leverage general quality ratings for responses. However, these overall ratings don't fully capture the exact characteristics in which each response excels. In this way, optimizing for general ratings can cause the model to discard beneficial behaviors observed in discarded responses and encourage disadvantageous ones in accepted responses. Could implementing ratings for a diverse set of specific characteristics and performing multi-objective optimization toward each measure lead to even better performance?

8 Ethical Considerations

Existing bias in preference data can bias the fine-tuned model when performing preference optimization. Agiza et al., in a study of how data selection impacts political biases in large language models, found that running DPO on biased preference data will successfully influence the trained model to reflect those biases [10]. More specifically, they find that right-leaning preference data will shift the model responses to the right, and same when training on left-leaning data.

However, despite the potential risks of preference optimization, it can also be used to address and reduce bias in LLM output. Ahmed Allam proposes BiasDPO, a preference optimization approach to mitigate gender, racial, and religious biases in LLM-generated English text [11]. The dataset is manually created and encompasses a diverse range of prompts with both biased and unbiased completions. They implement the loss function proposed in Identity Preference Optimization (IPO) [12] that adds a regularization term to the DPO loss function to prevent overfitting on preference data, shown below:

$$\mathcal{L}_{\text{IPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\left(\log\left(\frac{\pi_\theta(y_w \mid x)\,\pi_{\text{ref}}(y_l \mid x)}{\pi_\theta(y_l \mid x)\,\pi_{\text{ref}}(y_w \mid x)}\right) - \frac{\beta^{-1}}{2}\right)^2\right] \tag{4}$$

The BiasDPO approach beats most other existing models on bias benchmarks, specifically with regards to gender and racial bias, toxicity, and truthfulness. While we did not implement any bias safeguards for the purpose of this assignment (which only aimed to score performance on instruction-following tasks and was not meant for deployment), it is critical to consider this dimension when releasing models for general access.

References

[1] Petru Soviany. Curriculum learning with diversity for supervised computer vision tasks, 2020.

[2] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024.

[3] Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. Curry-dpo: Enhancing alignment using curriculum learning ranked preferences, 2024.

[4] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav,

Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025.

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

[6] L. M. Po. Direct preference optimization (dpo) of llms: A paradigm shift, April 2025. Medium, 29 Apr. 2025.

[7] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[8] Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. Helpsteer2-preference: Complementing ratings with preferences, 2024.

[9] Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training top-performing reward models, 2024.

[10] Ahmed Agiza, Mohamed Mostagir, and Sherief Reda. Politune: Analyzing the impact of data selection and fine-tuning on economic and political biases in large language models, 2024.

[11] Ahmed Allam. Biasdpo: Mitigating bias in language models through direct preference optimization, 2024.

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