A Comparative Analysis of Direct Preference Optimization and Proximal Policy Optimization

Rauno Arike*
TU Delft

Luna Mendez*
Independent

Luke Marks*
Independent

Amir Abdullah*
Guru Technologies

***Equal Contribution**

Abstract

Direct Preference Optimization (DPO) is an alternative to established reinforcement learning (RL) algorithms like Proximal Policy Optimization (PPO) for use in reinforcement learning from human feedback (RLHF). It allows for large language models (LLMs) to optimize for a preference dataset without constructing a reward model. Initial results suggest that DPO is more computationally efficient and performant than PPO, potentially making it a viable replacement. This research presents preliminary findings in attempting to assemble a mechanistic explanation for the benchmarked variance between the two algorithms. Success in doing so may allow for the designing of more performant algorithms or a more developed understanding of how industry standard practices like RLHF affect LLMs.

Keywords:Mechanistic interpretability, AI safety, Reinforcement learning, Reinforcement learning from human feedback

1. Introduction

RLHF has become a standard practice in aligning LLMs in recent years [4], and yet a robust understanding of how it affects neural networks is not established. Given the surprising emergent capabilities of these models [1], being able to interpret how they function may be critical in mitigating catastrophic outcomes, and also provide a better understanding of why techniques like RLHF induce negative phenomena (e.g. mode collapse) [2], allowing us to design better human feedback algorithms.

One problem with existing human feedback methods is that they require the LLM to engage in a reward modeling phase. This can be avoided by engaging in DPO in place of common algorithms like PPO, which early results suggest causes superior performance and increased computational efficiency [3]. Our primary hypothesis for why DPO was able to outperform alternatives was that optimizing for preferences directly resulted in

more efficient updates (meaning greater impact on loss relative to divergence in weights/biases) as it circumvents the problem of optimizing for an imperfectly learned reward model. This is achieved by optimizing for a higher likelihood of generating a preferred completion over a dispreferred completion, and is formalized by the following loss function:

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

Where:

1. *x is some prompt*
2. *$\pi_{\theta(\theta}(y_w|x)$ and $\pi_\theta(y_l|x)$ are the probabilities of the preferred and dispreferred completions under the current model.*
3. *$E_{(x,yw,yl)\sim D}$ denotes the expectation over the dataset of preferences D.*
4. *β is a parameter controlling the deviation from the base reference policy π_{ref}.*

We examine the weight divergence and sentiment detection capabilities of the DPO and PPO model relative to the reference model in order to falsify this hypothesis. Additionally, we begin some more fine-grained analysis in an attempt to find out whether or not DPO and PPO are in fact optimizing for the same things, which would help falsify our prior hypothesis. If their optimization targets are highly similar, it is probable that DPO is simply a more efficient algorithm for the examined task, alternatively if their optimization targets differ nontrivially, it might show that the primary benefit of DPO is in avoiding errors in defining an accurate optimization target.

Future work might turn to larger models in order to be able to engage in more general and complex tasks, and may also attempt to extract learned reward models from human feedback enforced LLMs that were required to do so in order to make this comparison more direct.

2. Methods

We worked throughout on this Google Colab.

2.1 Preliminary Training:

We sought to replicate the controlled sentiment generation task from the initial DPO paper, which aims to force positive sentiment completions to prefix elements in the IMDb dataset. This involved first fine-tuning a sentiment classifier on the labeled IMDb data, for which we used GPT-2. Following this, we trained a DPO and PPO variant of Pythia-70m. The models are tasked with generating completions to prefixes from the IMDb dataset, which are then classified by the fine-tuned GPT-2 instance such that they are graded based on sentiment. The DPO training script can be found here and the PPO script here.

The completion with the highest likelihood of being positive is designated the preferred completion, and that with the lowest is designated the dispreferred completion. These completions comprised the preference datasets on which the Pythia-70m instances were trained. Both the DPO and PPO variants were first trained for 25,000 examples (with some variance in hyperparameters), and then separately (from the base models) for

10,000 examples with equivalent hyperparameters where possible (see section 5 for hyperparameter details).

We also measured the performance of the DPO and PPO variants compared to the reference model by generating completions to 2,000 test prefixes not previously seen by any of the examined models. The sentiment of their completions was then classified and loss was computed based on the frequency of positive sentiment generations.

2.2 Weight and Activation Analysis:

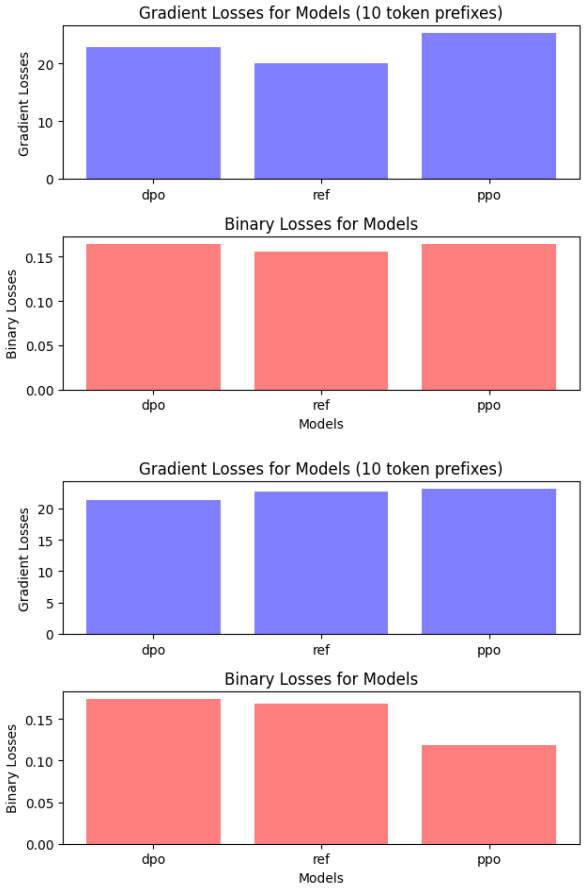
By comparing the divergence of the PPO and DPO model weights and biases to the reference weights we can make inferences about the internal differences made by PPO and DPO. We also analyze the divergence of parameter tensors. In addition, we examine the average distance between the activation vectors of all layers in the fine-tuned models and reference model over unseen prompts. We do this for two distance measures; KL divergence and cosine similarity, and again individually for prompts labeled as positive or negative in the IMDb dataset.

2.3 Head Ablation and Sentiment Detection:

Finally, we individually ablate heads to discern their contribution to loss and attempt to locate heads that contribute to the sentiment of the generation. We also attempt to identify heads with significant impact on loss in order to provide an explanation for their criticality and understand why some models may rely on them more heavily than others.

3. Results

3.1 Preliminary Training:



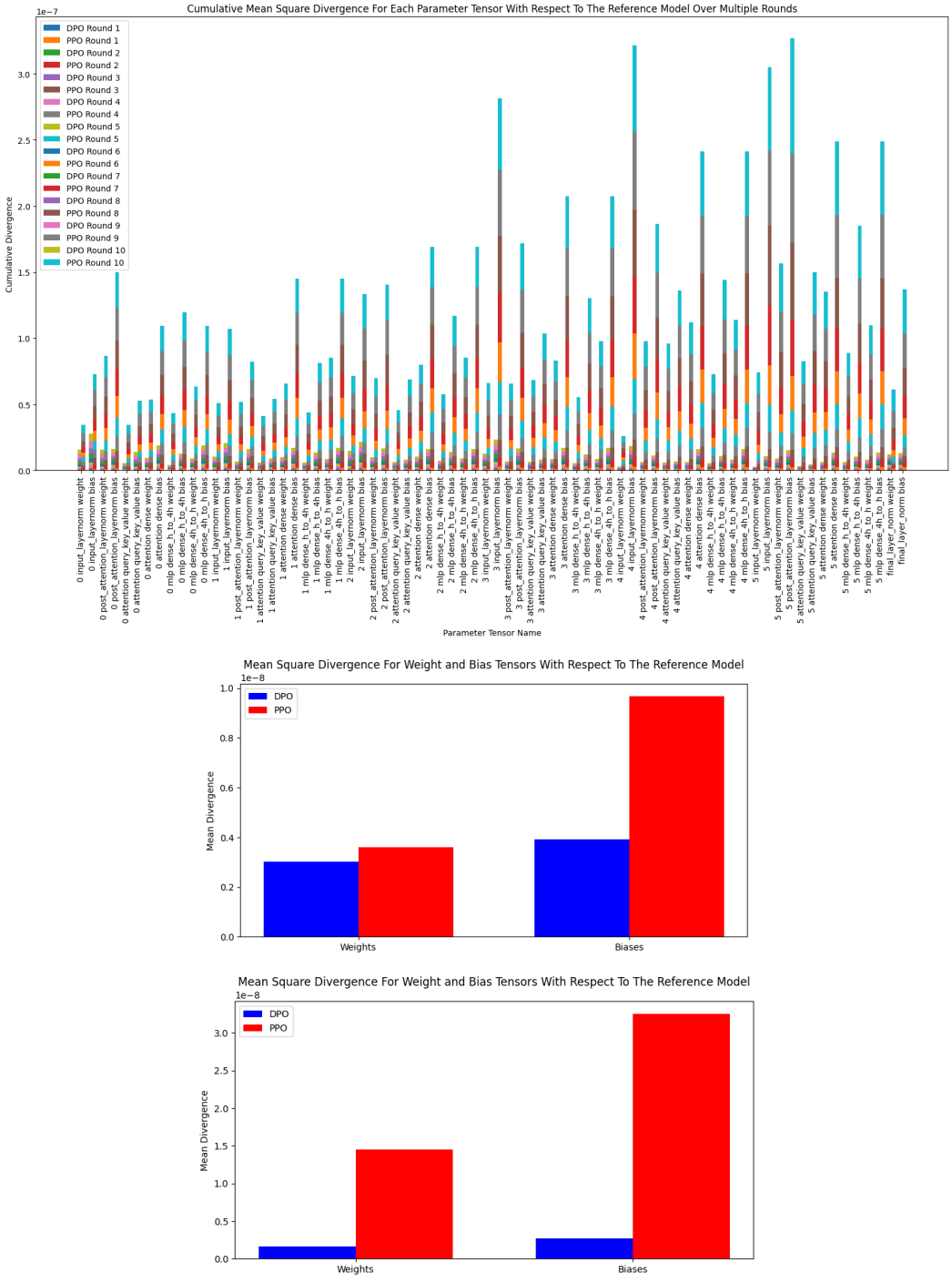
The gradient loss measure assesses the model's ability to produce an outcome close to a perfect positive sentiment grading, whilst the binary loss measure assesses the likelihood of the model in generating a positive completion at all. In both instances all models were tested for 2000 unseen prefixes. First training run models (left), second training run models (right)

In our testing the reference model had the lowest loss on all tasks followed by the fine-tuned models for the first training round. This is almost certainly caused by utilizing non-optimal hyperparameters as opposed to a failure of the algorithms themselves, as time restrictions and training delays meant we only had time to conduct two full training runs of both models, relying on the DPO paper for hyperparameter information.

In the models from the second training round, PPO considerably outperforms both the reference and DPO model on the binary task, which is marked given this was the training round with equivalent hyperparameters. This could be simply due to the unshared hyperparameter (KL divergence or the DPO beta parameter) not having equivalent impact on loss. Interestingly, whilst the DPO model is able to generate on average the most positive completions, it generates the least number of positive completions total,

suggesting completions are either strongly positive or strongly negative, at least relative to those of compared models.

3.2 Weight and Activation Analysis:



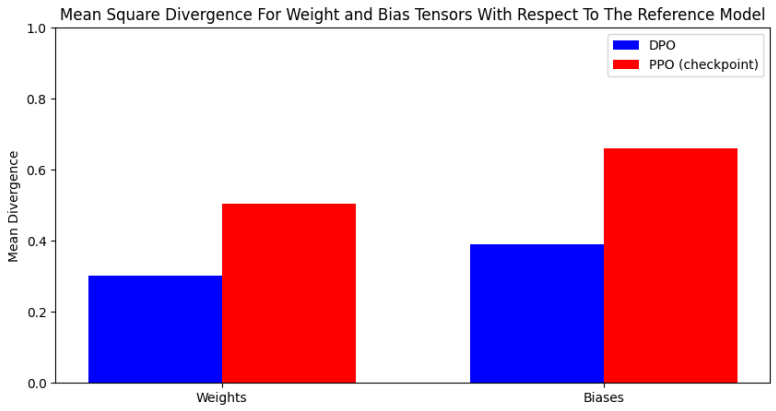
Top: Comparison of the cumulative mean square divergence for parameter tensors over the second training run. Each round represents 2,000 examples.

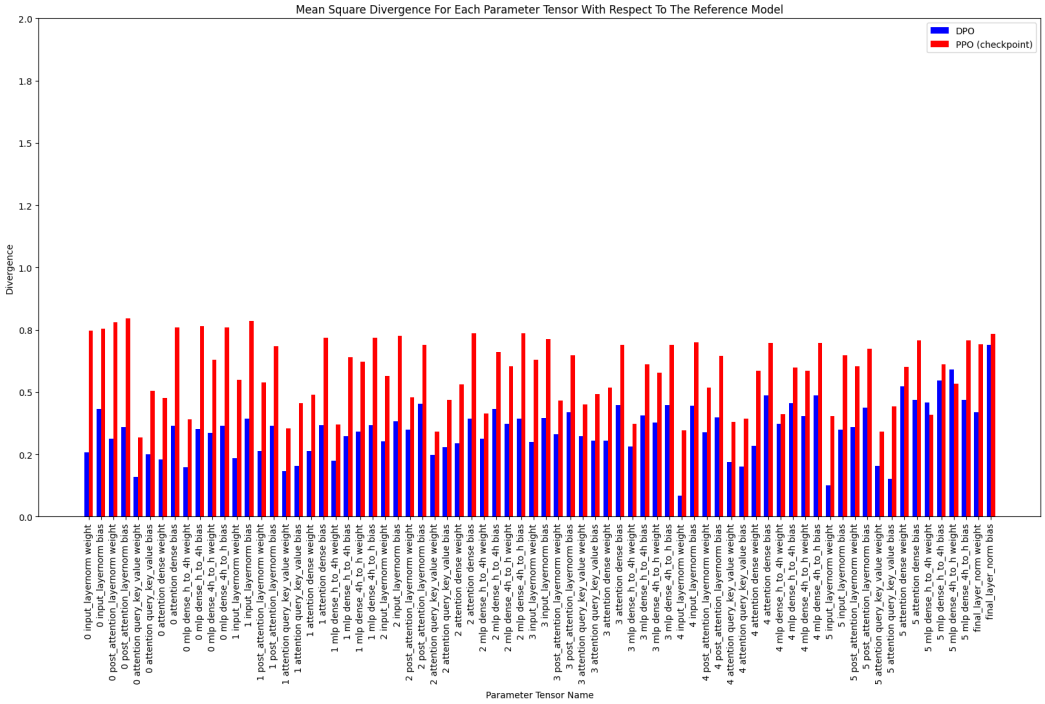
Bottom: Comparison of the mean square divergence for weights and biases to the reference model for the first training round models (left) and the second training round

models (right). Notice the y-axis in the right graph scales from 0-3 whilst the left scales from 0-1.

For the first training round models; whilst the average weight divergence is very similar across both models, the PPO average bias divergence is heavily elevated. This could suggest overfitting (which is possible due to the time constraint forcing an inability to be able to optimize hyperparameters), but we consider this hypothesis unlikely as both the DPO and PPO models were trained under very similar hyperparameters for the same number of examples, and PPO model has lower loss on unseen data. It is plausible this could be caused by a factor inherent to the PPO algorithm (e.g. KL regularization), which is challenging to falsify without more data on divergence given variance in hyperparameters.

In the second training round PPO diverges strongly from the reference model in both weights and biases. This training run lasted only 10,000 examples with the same learning rate, and so overfitting seems a less probable conclusion for the bias divergence in the first training round. As with the first training round, bias elevation is again extreme. For additional information we computed the divergence for weight and bias tensors in the first training round at example 5,000 examples (below), and see only a slight elevation in bias divergence, providing contrary evidence to the second training round model divergence, and perhaps suggesting simple overfitting.

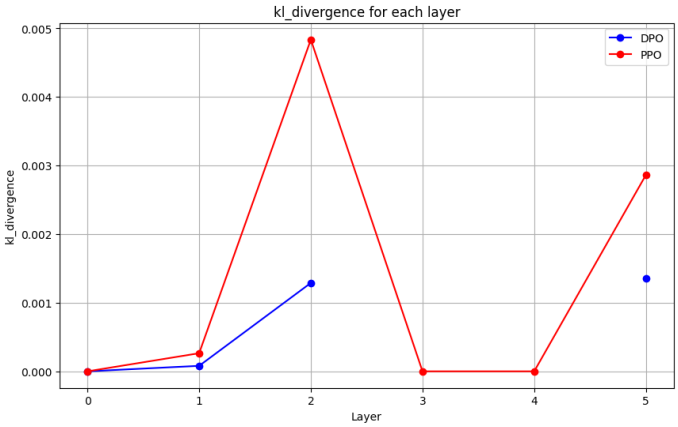


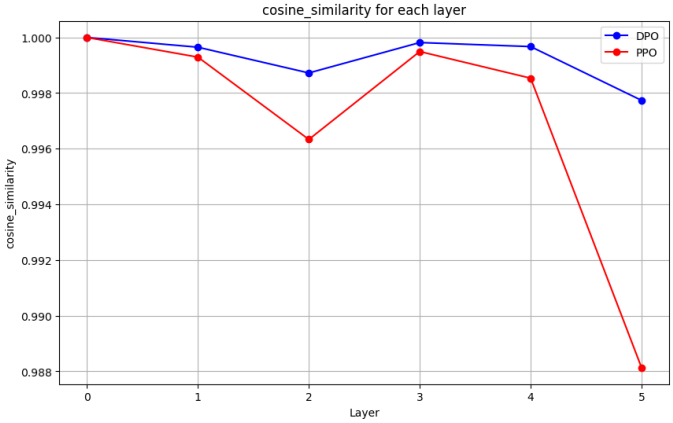


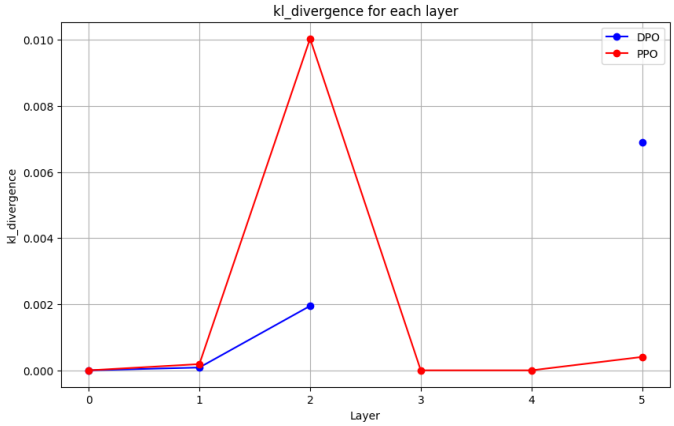
Top: Mean square divergence for weights and biases to the reference model for the first training round models, with the PPO weights biases from only 5,000 training examples.

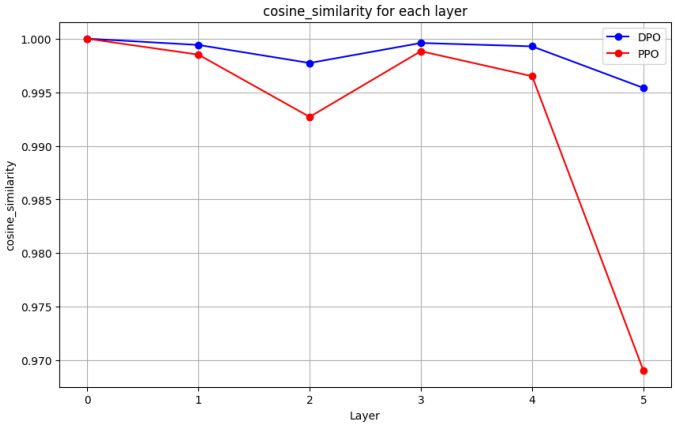
Bottom: Comparison of the mean square divergence for parameter tensors over the first training run.

In a more fine-grained analysis of the weight/bias divergence we still see nothing remarkable, noting only the general elevation in bias divergence seen in the average above. Overall, the DPO divergence was stable over both training runs, whereas the PPO divergence fluctuated wildly. This was unexpected as the hyperparameters used in the second training round were more similar to that of the PPO model's in the first than they were the DPO models.


kl_divergence for each layer
DPO
PPO
kl_divergence
Layer


cosine_similarity for each layer
DPO
PPO
cosine_similarity
Layer


kl_divergence for each layer
DPO
PPO
kl_divergence
Layer

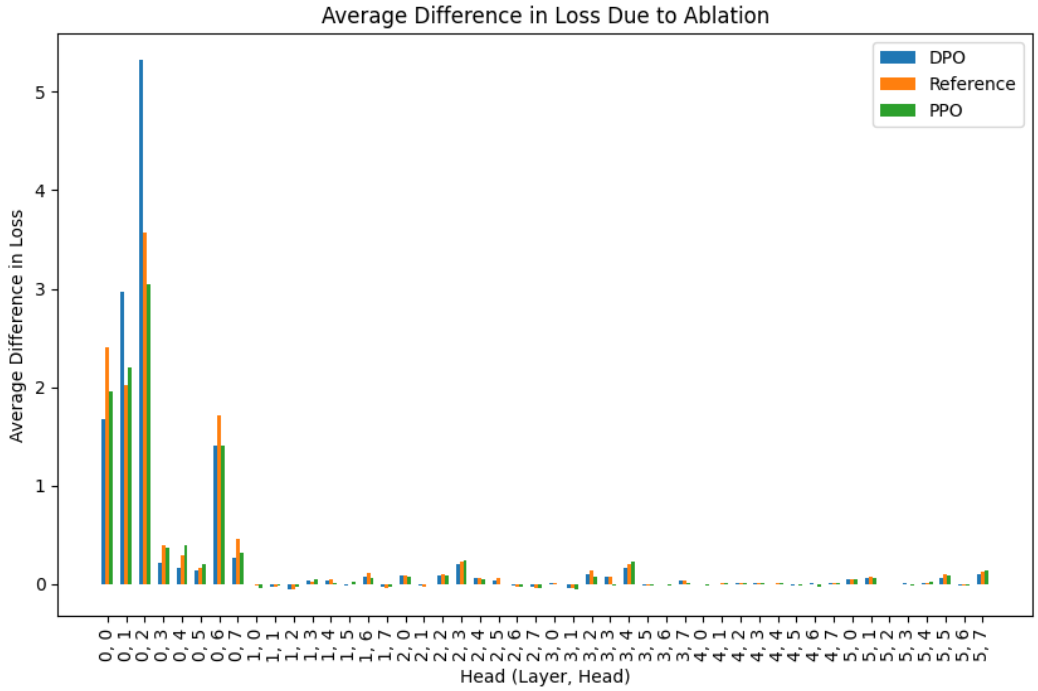

cosine_similarity for each layer
DPO
PPO
cosine_similarity
Layer

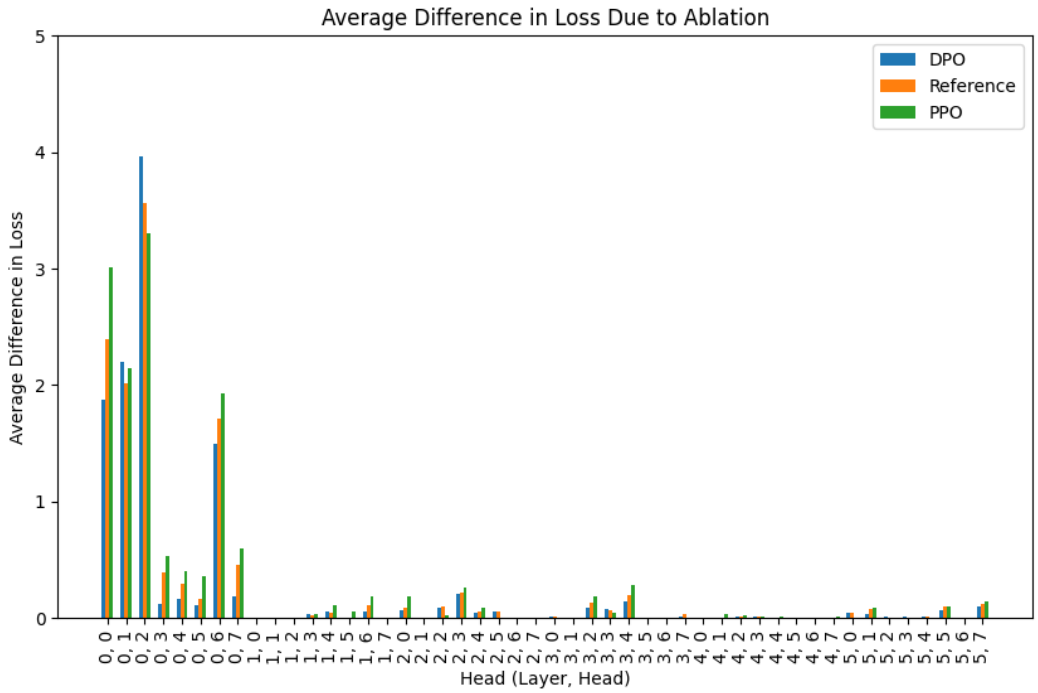
Top: Average KL divergence and cosine similarity for the first round DPO and PPO models respectively over 12,500 positive examples and 12,500 negative examples.

Bottom: Average KL divergence and cosine similarity for the first round DPO and PPO models respectively over 12,500 positive examples and 12,500 negative examples.

We observed hardly even a trivial change between the divergence in activation vectors for negative and positive sentiments, and note that all tests (only positive examples, only negative examples and the average of both) returned the same results as above at least to the 12th decimal place. The dissension between the distance measures for the DPO and PPO model is likely accounted for by the large bias divergence shown above. We observe quite similar patterns throughout both training runs, just amplified to slightly different extents.

3.3 Head Ablation and Sentiment Detection:





Top: Loss variance due to head ablation in the first training round.

Bottom: Loss variance due to head ablation in the second training round.

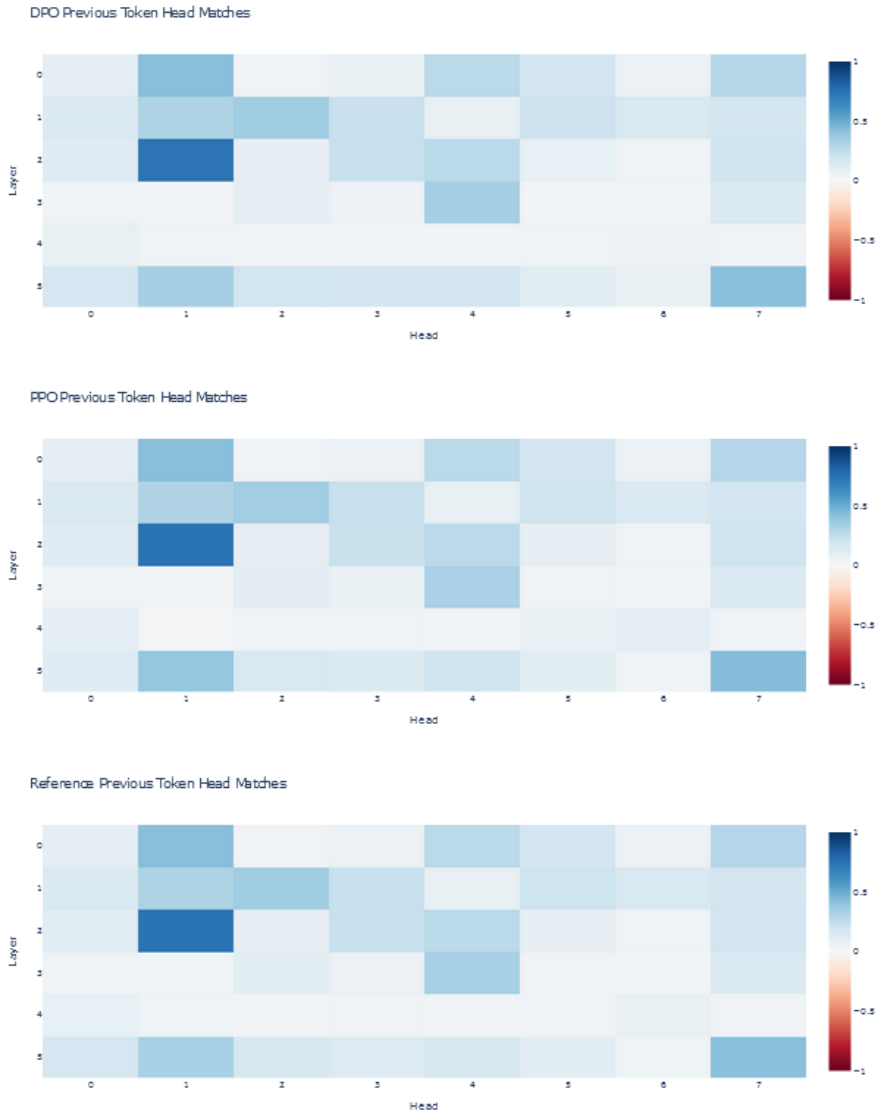
In the first training round, the PPO model is clearly considerably more robust to head ablation than the DPO model, and marginally more so than the reference model, which might be due to the steep bias divergence observed in 3.2. As mentioned earlier, this doesn't necessarily equate to the PPO algorithm being more robust to head ablation, as the bias divergence is possible to have originated from overfitting, although if this is not the case it may be preliminary evidence to suggest that. In the following section, we try to account for why the ablation of some heads has such a significant impact.

In the second training round, despite the fact that the PPO model had significantly more average bias divergence, it is not obviously more robust to ablation than the DPO model. Whilst the PPO model increased in average sensitivity to ablation, the inverse occurred for the DPO model. This could suggest that the increased generation length the DPO model was trained on using in the first round, as well as the increased prefix length caused the model to depend more heavily on heads 0-2.

3.4 Comparing the Fine-Tuned Models and the Reference Model:

Given the effectiveness of fine-tuning in making the model produce outputs with a positive sentiment, it is natural to inquire whether it is possible to find circuits that are responsible for the changes in the behavior of the models resulting from fine-tuning. As the first experiment, we tested whether the basic circuits of the model remain the same after fine-tuning, using the head detection functionality of TransformerLens. As expected from the facts that heads like previous token heads and induction heads are useful for a wide variety of tasks and that the fine-tuning objective does not provide explicit incentives for the formation or modification of such basic heads, we found that there are

no big changes in the behavior of previous token, duplicate token, and induction heads before and after fine-tuning.

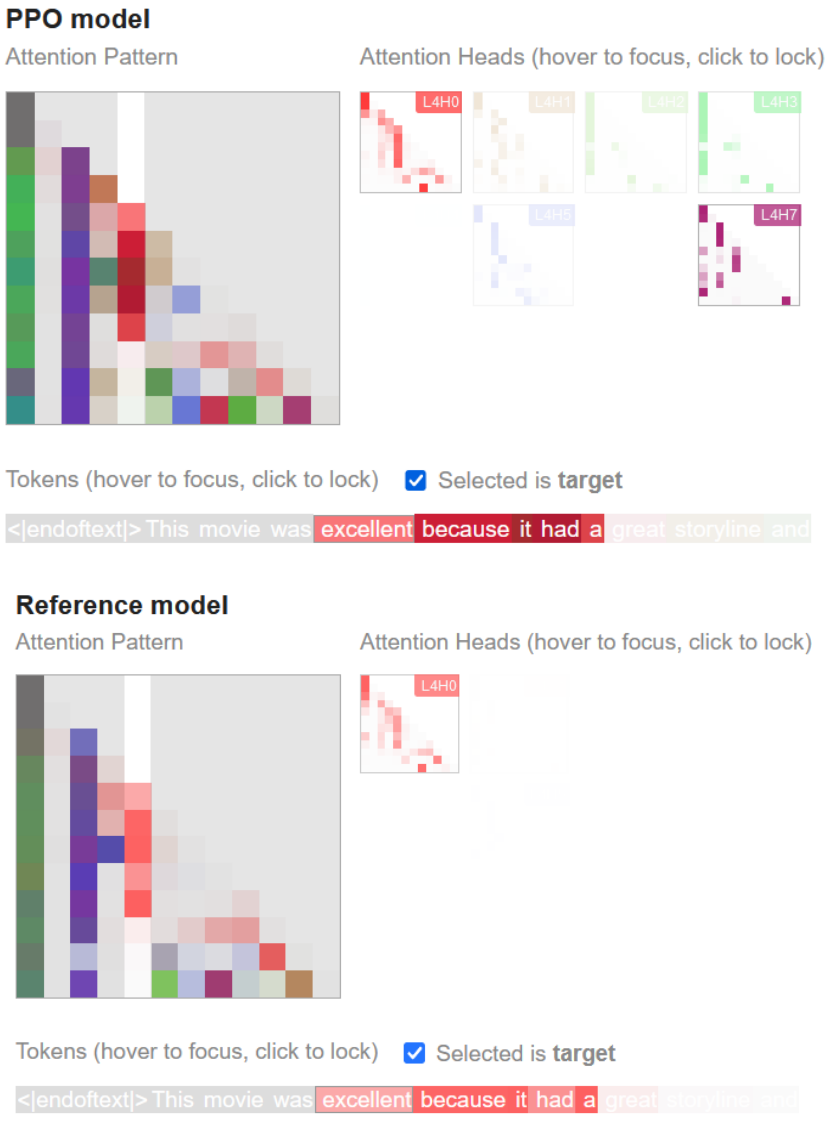


A comparison of the previous token head scores in the model fine-tuned with DPO (top), the model fine-tuned using PPO (middle), and the base model (bottom) for the first training round. While minor differences in the exact scores exist, the overall behavior of the heads in terms of attending to the previous token appears to remain the same before and after fine-tuning. The same results were observed for duplicate token and induction heads.

Though the behavior of the more universal heads remains the same before and after fine-tuning, our results indicate the possibility that fine-tuning modifies the behavior of some more specialized heads. Note, however, that these results are mostly based on qualitative analyses of the models' attention patterns and we would be excited to see a more rigorous analysis of this phenomenon being done in the future. Nevertheless, we

will provide a short overview of our findings.

Since the clearest evidence for head repurposing was found in the 4th layer of the models, we will focus on the heads of that layer. While the reference model and DPO-tuned model almost exclusively use head 0 of layer 4 for copying information from the sentiment token, the PPO-tuned model is also using head 7 of layer 4 for this purpose. This hints at some head repurposing taking place during the fine-tuning process.



The PPO-tuned model uses additional heads for attending to the sentiment token that the base model does not, particularly L4H7. However, the same result is not seen in the DPO-tuned model. Both were from the first training round. A similar difference can be seen using other prompts as well, including ones that have a negative sentiment word (e.g., "awful" instead of "excellent").

While it makes intuitive sense that the tuned model would need to use additional heads for processing the sentiment information in order to make the overall sentiment of its output positive, we remain uncertain in this hypothesis: the fact that the DPO-tuned model uses its heads in almost the exact same way as the reference model when attending to the sentiment token seems like counterevidence to it. In order to confirm or disconfirm our speculative findings, we suggest performing additional experiments assessing the behavior of attention heads at many different points over the course of the tuning process while also using a more diverse set of prompts and fine-tuned models.

4. Discussion and Conclusion

While our results do show that the DPO-tuned model and PPO-tuned model have different divergence patterns and that they may be using different heads to a different extent, our data does not strongly support our initial hypothesis that the DPO algorithm's direct optimization for preferences results in more efficient updates. In fact, we observed a higher divergence in both weights and biases for the PPO-tuned model compared to the DPO-tuned model, which seems to contradict our hypothesis. However, this might simply reflect improper hyperparameter selection, or possibly speak to the PPO algorithm being less stable to noise. More of the same data for these models trained using varying hyperparameters could assist in validating this.

Our head ablation tests from the first training round showed that the PPO-tuned model is more robust to head ablation than the DPO-tuned model. This suggests that the PPO-tuned model may be better at distributing its dependencies across different heads, resulting in a model that is less likely to have its performance significantly affected by the failure of a single head. On the contrary, the second training results suggest approximately equivalent dependency distribution due to the almost equivalence of the impact of ablation on loss.

Our preliminary observations in analyzing attention patterns hint at the repurposing of attention heads in the PPO model. While this requires further investigation, it opens up an interesting line of inquiry about how fine-tuning modifies the internal structures and dynamics of the model.

In conclusion, our findings suggest that while DPO generally results in less divergence from the base model, PPO may result in a model that is more robust to head ablation due to the repurposing of heads. These conclusions are tentative however, as they are made in light of a very small amount of data about how hyperparameters affect the functioning of these models. Future research performed over longer time frames should re-run these experiments with a much larger sampling of trained models for both algorithms for more definitive results.

5. Experiment Details

5.1 Hyperparameters

First training run

DPO

Examples - 25000
Learning rate - 1e-6
Beta DPO param - 0.1

Completions chosen from per prefix - 8
Prefix size - 100 tokens
Completion max length - 500 tokens

PPO

Examples - 25000
Learning rate - 1e-6
Target KL divergence - 0.1
Completions chosen from per prefix - 2
Prefix size - 25-75 tokens
Completion max length - 200 tokens

Second training run

DPO

Examples - 10000
Learning rate - 1e-6
Beta DPO param - 0.1
Completions chosen from per prefix - 4
Prefix size - 25 tokens
Completion max length - 100 tokens

PPO

Examples - 10000
Learning rate - 1e-6
Target KL divergence - 0.1
Completions chosen from per prefix - 4
Prefix size - 25 tokens
Completion max length - 100 tokens

All training runs for all models use the AdamW optimizer. The first training run hyperparameters were designed with convergence in mind whilst in the second the hyperparameters are kept equivalent regardless of whether or not it negatively impacts the trained models performance.

5.2 Testing and Evaluation

Parameters

Test set size - 2000 examples
Token prefix amount - 25 tokens
Completion max length - 100 tokens

6. References

1. Bubeck, S. (2023, March 22). *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. arXiv.org. https://arxiv.org/abs/2303.12712

2. Janus. (2022, November 8). *Mysteries of mode collapse [Online forum post]*.

https://www.lesswrong.com/posts/t9svvNPNmFf5Qa3TA/mysteries-of-mode-collapse

3. Rafailov, R. (2023, May 29). *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. arXiv.org. https://arxiv.org/abs/2305.18290
4. Ziegler, D. M. (2019, September 18). *Fine-Tuning Language Models from Human Preferences*. arXiv.org. https://arxiv.org/abs/1909.08593