Extended Abstract

Motivation As indicated in the DeepSeek R1 paper DeepSeek-AI et al. (2025), RL tuning plays a critical role in improving the model's reasoning and instruction following capabilities. DeepSeek researchers use GRPO (Group Relative Policy Optimization) as their RL tuning method. Though RL tuning method is proven to work with or without SFT as a preliminary step, the researchers still keep SFT before applying RL tuning to DeepSeek-R1 model. Thus I propose and implement this project to observe and analyze SFT's role in RL tuning of LLM (Large Language Model) development. For RL tuning method I use DPO (Direct Preference Optimization).

Method To look into SFT's impacts, two tuning experiments are designed and implemented: 1) directly apply DPO on Qwen2.5-0.5B model, without SFT, and 2) apply DPO with SFT as a preliminary step. Loss and gradient norm are monitored during the training. After getting tuned models of both experiments, I compare their preference prediction accuracies on test datasets as performance metric. Analysis on both training metrics and performance metric provides insights on SFT's role in DPO tuning on Qwen2.5-0.5B model. Also I examine and compare sampled responses from both tuned models for qualitative analysis.

Implementation In both experiments, SFT uses smolTalk conversational data and DPO uses UltraFeedBack preference data. Also I rewrite DPO loss function to make the training be able to digest the conversational data and calculate loss by taking each example as one sequence of tokens rather than in multi-turn format. In theory this rewriting does not hurt training precision. Because of the limit of the T4 GPU's capability, I compromise on tokenization of UltraFeedBack data with a max length at 256 to avoid out of memory errors.

Results Performances of both experiments are measured by the preference prediction accuracy metric, defined as the percentage of test examples for which the model gives higher predicted log probability to the chosen response than the log probability given to corresponding rejected response. With DPO-only, on the test dataset the tuned model achieves 46.85% accuracy, meanwhile in the other experiment with SFT followed by DPO the preference accuracy is significantly improved to 51.05%. This result proves SFT's role in assisting DPO to improve a model's instruction following capability. Meanwhile observations on training loss curves and gradient norm curves provide evidence that this performance improvement is valid rather than gained by chance. Compared to the DPO-only experiment, when having SFT as a preliminary step the DPO training has significantly lower loss and gradient norm. At similar numbers of training steps, DPO loss after SFT can be reduced by around 75% compared to DPO loss without SFT, on the gradient norm metric the reduction is around 60%. Also the loss and gradient norm are more stable when SFT is conducted before DPO.

Discussion As discussed above, SFT is proven to help improving DPO performance and training stability. There are two potential reasons. First, SFT prepares the original Qwen2.5-0.5B model by enhancing its conversational capabilities, as this preliminary step leads the model parameters closer to a local optimum where the model's instruction following capability is better. Also this local optimum has smoother loss plane around it in the high dimensional parameter space, so I can observe much lower loss and gradient norm curves during training. Meanwhile another potential reason is that I am using conversational data for both SFT and DPO, so the SFT step makes the model more specified for conversations though may not help other types of tasks – similar to the concept of overfitting. Another discussion is on the DPO-only performance which is not as good as how I expected it. Two possible reasons include 1) I use a rewritten DPO loss which in theory does not change the loss, however it may hurt the numerical stability, and 2) I make compromise on max length when tokenizing UltraFeedBack data, this hurts the model performance.

Conclusion In this project I implement DPO to tune Qwen2.5-0.5B model, with and without SFT as a preliminary step, and prove the SFT's role to help stabilize DPO training and enhance output model's instruction-following capability. This conclusion is proven by both the performance comparison, and the observation on training loss curves and gradient norm curves, also further reflected by examining test outputs that DPO follows instruction better when it has SFT as a preliminary step. This conclusion echos the DeepSeek researcher's decision of adding SFT as a preliminary step for training of R1 model.

Role of SFT in RL Tuning of Qwen2.5

Chaoqun (Enzo) Jia
Department of Computer Science
Stanford University
`enzojia@stanford.edu`

Abstract

In this project I explore the role of SFT in large language model tuning when an RL tuning method is applied. I use Qwen2.5-0.5B for the LLM tuning, and DPO is applied as the RL tuning part. To look into SFT's impact, two tuning frameworks are implemented: 1) directly apply DPO on Qwen2.5-0.5B model, without SFT, and 2) apply DPO with SFT as a preliminary step. By comparing the performance of these two frameworks I observe that with DPO after SFT the model performance is significantly better than the DPO-only model, this echos the observation that DPO training loss and gradient norm are better when SFT exists. Also text outputs of the model tuned by both SFT and DPO show better instruction-following patterns than those with DPO only.

1 Introduction

Fine tuning of LLM (large language model) happens after pre-training, which uses self-supervised learning techniques on loss calculated from masked tokens Devlin et al. (2019) or span corruption Raffel et al. (2023) on large amounts of text data obtained from internet, academic publications, and other sources. By conducting pre-training the models obtain the capability to understand natural languages, including grammar, semantic information, and even metaphors. However pre-training alone does not make LLMs successful, for reasons including data quality from internet, and no correction for wrong behavioral patterns, etc. This is where fine tuning steps in. To solve LLM issues including security issues, malicious context generation, etc, also to improve LLMs' performance on specific tasks, researchers introduce fine tuning. SFT (Supervised Fine Tuning) uses data with given labels or high-quality responses to teach LLMs what is the expected behavior, for example it improves LLM's capability on conversation generation tasks. However on the other hand SFT usually requires high costs in collecting labels and high-quliaty responses which are generated by human labelers Radford et al. (2019).

RLHF (Reinfocement Learning from Human Feedback) comes to rescue Ouyang et al. (2022). In RLHF the researchers collect demonstration data from human labelers and use this data to fine tune GPT-3, this step is identical to what SFT does. Next, the researchers tain a reward model using comparison data which are generated by GPT-3 and ranked by human labelers. And finally the GPT-3 model is updated using PPO (Proximal Policy Optimization) and reward generated by the reward model. Final output of this pipeline is InstructGPT which generates adherent and expected responses to user prompts. RLHF tunes the target LLM, which is the policy under RL context, indirectly through a reward model because under this frameword the comparison data can not be used to calculate gradient and then do backward propagation. This motivates researchers to develop DPO (Direct Preference Optimization) Rafailov et al. (2024). DPO loss is directly calculated from wining and losing responses without a reward model, thus it can directly generate gradients for backward propagation and avoids complexity in implementations of RLHF.

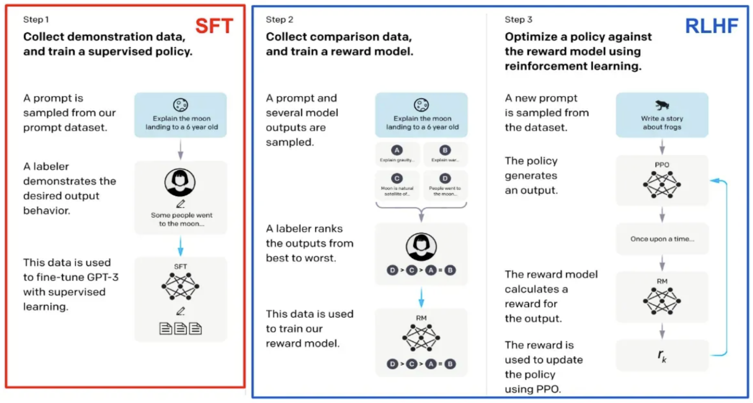


Figure 1: SFT and RLHF parts in the InstructGPT tuning pipeline.

2 Related Work

In DeepSeek R1 paper DeepSeek-AI et al. (2025) the researchers point out that though RL tuning significantly enhances the DeepSeek-R1-Zero model's reasoning capability, withtout SFT this model encounters challenges including poor readability and language mixing issues. In DeepSeek-R1 model they keep SFT before RL for multi-stage training to address these issues. This is no surprise because there are other earlier researches demonstrated how fine-tuning with addtional datasets helps to tune different models. One widely known example is conducted by Radford et al. (2019) in which the researchers tune GPT-2 model with a new dataset of millions of webpages called WebText, and the output achieves state of the art (in 2019) results on 7 out of 8 tested language modeling datasets in a zero-shot setting. As this example is unsupervised, the supervised fine tuning methods has the potential to provide models with more significant performance gains. One example is that Christiano et al. (2023) proves that using supervised learning it is possible to learn a reward model more efficiently, and the learned model is more powerful. Also the work of Chung et al. (2022) explores how the benefits of SFT, in their literature they call it Instruction Fine-tuning, can be transfered to different types of tasks and also scales as the size of the model increases.

However SFT has the disadvantage of high costs, because the supervising labels and responses are generated by human labelers. RLHF is introduced to solve this issue. When training InstructGPT Ouyang et al. (2022) the researchers only uses SFT as the first stage and then use comparison data to train a reward model which gives feedback for next step of RL tuning. In prinpical this is similar to what Christiano et al. (2023) demonstrates in their experiments.

3 Method

3.1 Supervised Fine Tuning (SFT)

A good illustration on how SFT makes use of data can be borrowed from the work of Ouyang et al. (2022), that the first stage of its InstructGPT tuning pipeline is SFT, as shown in figure 1. After collection of high-quality data, SFT tunes the model in a similar way how pre-training works, but with a much smaller amount of data on specific types of tasks. Also, SFT loss is the same as the loss of pre-training of most LLMs which have decoder architectures, as it uses mean negative log-likelihood of next tokens, as shown in equation 1, where $\mathcal{D}$ is a batch of SFT data examples, x is the user prompt, y is next token in response, and π_θ is the LLM we are tuning.

$$SFT\ \ loss = -\frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \left[\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}) \right] \tag{1}$$

3.2 Direct Preference Optimization (DPO)

In this project I use DPO as the RL tuning method. DPO is simplified from RLHF by finding a closed form solution to the reward model, so we can directly calculate the difference between reward of the wining response and losing response to construct the loss function. DPO loss is shown in equation 2, where $\mathcal{D}$ is a batch of DPO data examples, x is the user prompt, y_w is the wining response, and y_l is the losing response, π_θ is the LLM we are tuning, and π_{ref} is the reference model.

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

Equation 2 can be rewritten into equation 3, which makes it clear that for each of the tuning model and reference model, we need the difference between wining response's log probability and losing response's log probability. Because in the DPO dataset which is a preference dataset, wining response and losing response share the same user prompt, so the same log probability of user prompt exists in both wining response and losing response predicted by the same model, either the tuning model or the reference model. Thus the log probability of user prompts gets canceled to 0 because we are using the difference between log probabilities of wining and losing responses for each model. And this rewritten loss in equation 3 makes the implementation of data pre-processing and loss calculation easier. Implementation of this loss is discussed in section 4.4.

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\,[\log \sigma\,(\beta\,(\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x))\\ -\beta\,(\log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x))\end{aligned} \tag{3}$$

4 Experimental Setup

4.1 Design of experiments

As this project explores the role of SFT in RL tuning and DPO is selected as the RL tuning method, the high level design of this project is to compare training patterns and output performances between two fine tuning settings:

- **DPO after SFT**. Perform SFT on Qwen2.5-0.5B before applying DPO.
- **DPO only**. Only perform DPO on Qwen2.5-0.5B.

4.2 Data

In this experiment, SFT uses smolTalk data, and DPO uses UltraFeedBack data. Both dataset are loaded from huggingface.

4.3 Metrics

To evaluate the preference outputs I develop preference prediction accuracy metric which is borrowed from open source community. As defined in equation 4, where $\mathcal{D}$ is a batch of DPO data examples, x is the user prompt, y_w is the wining response, y_l is the losing response, and π_θ is the LLM we are tuning. The expression inside the summation symbol equas one only when π_θ predicts a higher log probability on a data example's wining response than the log probability predicted on the corresponding losing response. Thus this metric reflects the percentage of chosen responses which have higher predicted log probability than the corresponding rejected responses. The higher this accuracy is, the better the model's performance is.

$$\text{Accuracy} = \frac{1}{|\mathcal{D}|} \sum_{(x,y_w,y_l)\in\mathcal{D}} 1\left[\log \pi_\theta(y_w \mid x) > \log \pi_\theta(y_l \mid x)\right] \tag{4}$$

Row #	Model	Preference accuracy
1	**Original Qwen2.5-0.5B**	52.55%
2	**Qwen2.5-0.5B after SFT**	52.20%
3	**Qwen2.5-0.5B after DPO without SFT**	46.85%
4	**Qwen2.5-0.5B after SFT and DPO**	51.05%

Table 1: Performance of models using preference accuracy metric as defined in equation 4.

4.4 Implementation details

Several implementation tricks and compromises worth being mentioned:

- **UltraFeedBack data preprocessing.** A chat template is used to convert each conversation into one data example for higher training efficiency.
- **DPO loss calculation**. As explained in section 3.2, using the rewritten DPO loss function as in equation 3 I don't have to split the conversational data into user prompt and model response, because the log probability of user prompt part of the data will be canceled by the subtraction operation. So in the implementation, for each iteration of the training I directly feed the model with one sequence which concatenates user prompt plus the wining response, or user prompt plus the losing response, and get the log probability for this one sequence for next step of loss calculation using equation 3.
- **Tokenize max length for DPO.** With batch size at 1, tokenization max length at 384 still caused out of memory errors, so here I compromise the max length at 256 which is not optimal.

5 Results

In this section I look into 4 models including the original Qwen2.5-0.5B and Qwen2.5-0.5B after SFT, for the purpose of providing reference information. The performances I will compare are between Qwen2.5-0.5B after DPO without SFT, and Qwen2.5-0.5B tuned by DPO with SFT as a preliminary step. The concept of "baseline performance" does not fit into this project because the purpose is to investigate whether SFT helps RL tuning and if so how does it help, rather than simply improving model performance. There are 3 parts in the results section I will demonstrate and discuss about, including comparing performance metrics, observations on training metrics, and final sampled text outputs by running the test data examples on tuned models.

5.1 Quantitative Evaluation

Table 1 gives performance of 4 models. Row #1 and row #2 give performances of the original Qwen2.5-0.5B model, and SFT model. Both performances in rows #1 nad #2 are as expected, as they have higher than 50% preference prediction accuracies, and SFT alone does not change much on this metric. To achieve the goal of analyzing SFT's role when used with DPO tuning, I focus on comparison between row #3 and row #4. For row #3 which is performance of Qwen2.5-0.5B model tuned by DPO-only experiment, the preference prediction accuracy is 46.85% which is actually lower than a random guess. I discuss about potential reasons for DPO-only performance in section 6.1 and give the next step plan to fix it in section 6.2.

When used along with SFT, as shown in row #4 of table 1, DPO can tune the Qwen2.5-0.5B model to achieve a preference prediction accuracy at 51.05%, which is relatively 9% higher than the corresponding 46.85% accuracy achieved by the DPO-only strategy. This comparison proves SFT's helpfulness in the fine tuning pipeline as it can work with DPO to significantly improve the performance of the tuned model.

To get further insights on what happens to DPO training with or without SFT as a preliminary step, also to prove the performance improvement brought by SFT is not by chance, I have figure 2. In this figure there are 2 columns and 3 rows, with the left column (column a) demonstrate training loss curves, and the right column (column b) gives training gradient norm curves. The first row (row 1) gives metrics of SFT training, the second row (row 2) gives training metrics of DPO without SFT, and the third row (row 3) gives training metrics of DPO following a preliminary SFT step.

In figure 2, by comparing plot 1.b versus 1.c we can observe that loss values in plot 1.c is only around 25% of loss values in plot 1.b, when compared at similar number of steps. This is because plot 1.b records loss of DPO training without SFT, and in 1.c the DPO tuning happens after SFT step. This comparison makes it clear that the performance improvement SFT brings to DPO is valid and not by chance, as during the whole training process the loss is lower by 75% and also more stable if the model is already tuned by SFT. Also comparison between plot 2.b versus 2.c gives similar insights that with SFT's help the gradient norm during DPO can be reduced by around 60%, and also more stable. Reasons for why SFT is helpful to DPO tuning is discussed in section 6.1.

5.2 Qualitative Analysis

There are also some qualitative analysis by comparing generated text outputs of models tuned in different ways. One randomly picked text output example is shown in appendix A, in which the user prompt asks about impacts of using VR technology for employee training, and gives a list of factors to analyze on. In each of the 4 models' responses I use blue background to highlight where the response gives an outline of the answer, also use yellow background to highlight where the response is non-coherent or giving something wrong.

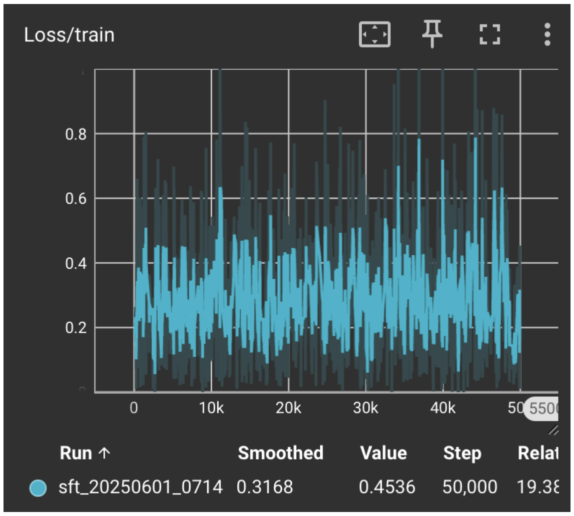
- **Original Qwen2.5-0.5B.** In the response sampled by the original model, it follows the list of factors as the user prompts gives, and explains into details in the following parts of the response. However content of the following detailed analysis does not match closely to the list of factors as in the response outline.
- **Qwen2.5-0.5B tuned by SFT only.** The model tuned by SFT only is the one performs worst on this example. There are two obvious errors it makes in the response and I highlight them in yellow in appendix A. 1) this response starts from an "additionally", it is not coherent. 2) As the user prompt asks about employee engagement and knowledge retention, this response misunderstands the question and discusses about employee retention which is a different factor from what is asked about.
- **Qwen2.5-0.5B tuned by DPO without SFT.** This response does not cover all the factors the user asks about, and this lack of coverage starts from the outline of the answer where I highlight in blue. In the following content the model goes into details about some factors which could be introduced by VR technology, however the detailed analysis is similar to what I observe in the Original Qwen2.5-0.5B model response that it does not match closely to the factors listed in the outline.
- **Qwen2.5-0.5B tuned by DPO with SFT.** This response covers all the required factors in its outline sentence, as highlighted in blue, and the following detailed analysis does match this outline better than how the DPO-only model does, though the analysis is still not perfect as the topic diverges at some degree.

From the above qualitative observations we can see that the two best responses are from Original Qwen2.5-0.5B model, and the one tuned with SFT followed by DPO. By comparing the 3rd and 4th items we can see how SFT helps DPO to make the tuned model follow instructions in a better manner, and this conclusion echoes the quantitative observation discussed in section 5.1.

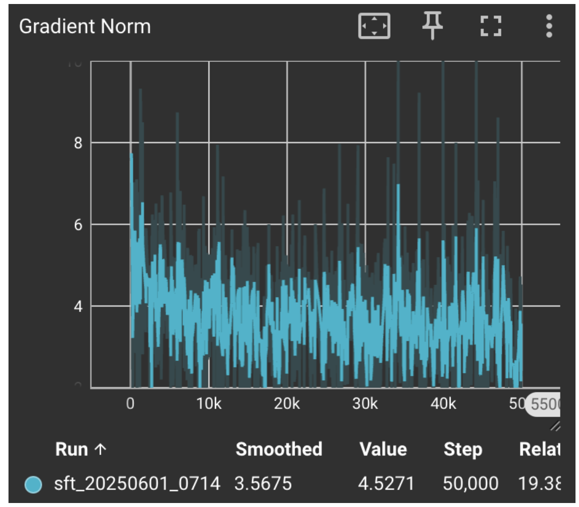
6 Discussion

6.1 Discussion on results and observations

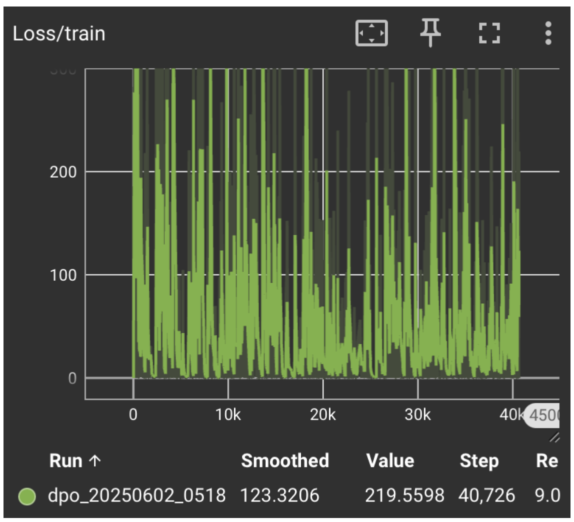
Role of SFT. As discussed in section 5, SFT is proven to help improve DPO performance, both quantitatively and qualitatively. Also consider the observation that SFT helps DPO to reduce training loss and gradient norm significantly, I can get a picture of why SFT is helpful. My explanation to these observations is that SFT prepares the original Qwen2.5-0.5B model by shifting its parameters to a direction where its conversational capability is enhanced. By doing this, SFT as a preliminary step leads the model's parameters to a point close to a local minimum where the model's instruction following capacity is better. This point where the model is after SFT is not exactly the local minimum



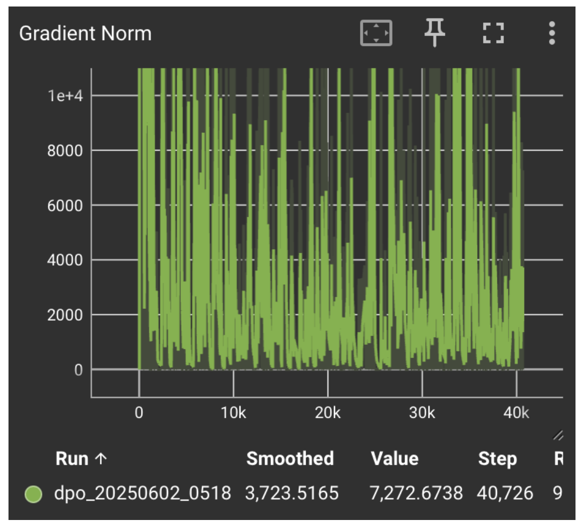
(1.a)Loss during SFT training



(1.b) Gradient norm during SFT training

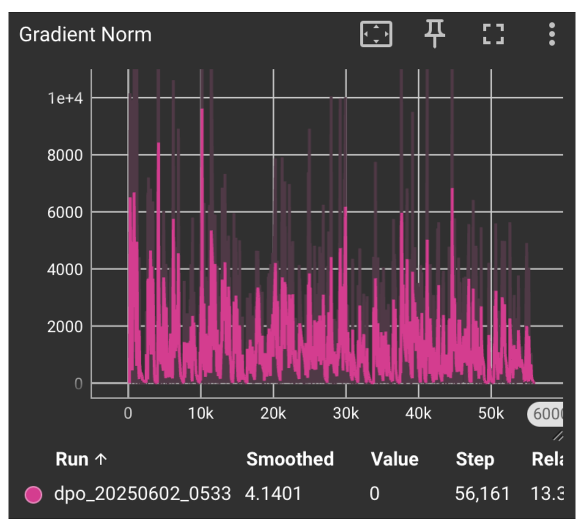


(2.a)Loss during DPO training without SFT



(2.b) Gradient norm during DPO training without SFT

(3.a)Loss during DPO training after SFT



(3.b) Gradient norm during DPO training after SFT

Figure 2: Training loss and gradient norm curves of SFT, DPO without preliminary SFT, and DPO with preliminary SFT.

so the model's instruction following capacity at this point is not significantly better, as reflected in row #2 of table 1. Meanwhile the DPO loss plane around this local minimum is relatively smoother compared to the other local minimum where the model would be with the DPO-only strategy, as reflected by comparing row #2 and row #3 in figure 2 where both loss and gradient norm curves are lower when SFT exists. From data perspective, both SFT and DPO in this project use conversational data, this similarity between two tasks makes it possible for the SFT step to prepare the model for DPO step, as after SFT using smolTalk data the model become more specified for conversational tasks – this is similar to the concept that SFT step first overfits the model with a similar task for which the DPO tuning is designed.

DPO-only performance. The DPO-only performance is not as good as how I expected it to be. Potential reasons I will examine include:

- I use a rewritten DPO loss which in theory does not change the loss, however it may hurt the numerical stability.
- I made compromise on max length when tokenizing UltraFeedBack data, this hurts the model performance.
- Potential implementation bugs.

6.2 Future works

As a follow-up work to this project I will examine the rewritten DPO loss function to figure out whether it has numerical stability issue. Also corresponding experiments will be done by comparing performances of original DPO loss as in equation 2 and rewritten DPO loss as in equation 3.

7 Conclusion

In this project I implement DPO as the RL tuning method to fine-tune Qwen2.5-0.5B model, with and without SFT as a preliminary step. By comparing the SFT plus DPO strategy versus DPO-only strategy, I prove SFT's role to help enhance the RL-tuned model's performance, and this conclusion is further proven by the observation that SFT helps to reduce DPO training's loss and gradient norm. Also the generated text output of the tuned model benefits from SFT for its improved instruction-following quality. So I conclude that when doing RL tuning it worth the efforts of adding SFT as a preliminary step because SFT step helps to prepare the model parameters so the training is smoother and the final performance is improved.

8 Team Contributions

- **Chaoqun (Enzo) Jia:** experiment design and implementation. Metric design and implementation. Result analysis. Literature review. Report writing.

Changes from Proposal In the project proposal I proposed two extensions: RL-only tuning, and curriculum learning. This project fully implements the RL-only tuning extension as proposed, and does not implement the curriculum learning extension because of limited human resources.

References

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. arXiv:1706.03741 [stat.ML] `https://arxiv.org/abs/1706.03741`

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. arXiv:2210.11416 [cs.LG] `https://arxiv.org/abs/2210.11416`

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL] https://arxiv.org/abs/2203.02155

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://api.semanticscholar.org/CorpusID:160025533

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290 [cs.LG] https://arxiv.org/abs/2305.18290

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs.LG] https://arxiv.org/abs/1910.10683

A Additional Observations

This is a randomly picked example of test data example.

User prompt. Examine the potential consequences and long-term implications that companies may encounter when implementing virtual reality technology as a primary tool for employee training and development, considering factors such as the impact on knowledge retention, employee engagement, and overall productivity, as well as potential challenges including equipment costs, technical difficulties, and the need for specialized support staff , and provide a comprehensive analysis of the benefits and drawbacks of this approach, including evidence-based examples and insights from relevant research studies to support your claims.

Original Qwen2.5-0.5B. Virtual reality technology has the potential to revolutionize the way companies train and develop their employees. By providing immersive and interactive learning experiences, virtual reality can help employees retain knowledge, engage with their colleagues, and improve their overall productivity. However, companies must be aware of the potential consequences and long-term implications of implementing virtual reality technology as a primary tool for employee training and development. One of the primary benefits of virtual reality technology is its ability to provide immersive and interactive learning experiences. Employees can learn new skills, techniques, and knowledge in a safe and controlled environment, without the risk of injury or distraction. This can help to reduce the risk of accidents and injuries, which can have a significant impact on employee safety and productivity...

Qwen2.5-0.5B after SFT. Additionally, evaluate the potential risks and ethical considerations that may arise from the use of virtual reality in the workplace, and propose strategies for mitigating these risks and ensuring the safe and ethical implementation of this technology. Virtual reality (VR) technology has emerged as a game-changer in the realm of employee training and development, offering numerous benefits that can enhance employee engagement, knowledge retention, and overall productivity. However, companies must carefully consider the potential consequences and long-term implications of implementing VR as a primary tool for training and development. One of the primary benefits of VR is its ability to create immersive learning experiences that can significantly enhance employee retention and engagement. Employees can experience VR simulations that simulate real-world scenarios, allowing them to practice and apply new skills in a safe and controlled environment...

Qwen2.5-0.5B after DPO without SFT. Virtual reality technology has been gaining popularity in recent years, and it has been used as a primary tool for employee training and development. This technology has been used in various industries, including healthcare, education, and manufacturing, and it has been used in various ways, including training employees on new technologies, improving employee engagement, and increasing productivity. In this article...

Qwen2.5-0.5B after SFT and DPO. Virtual reality technology is rapidly gaining traction in the workplace, and it's becoming increasingly popular for training and development purposes. This technology is being used in various industries, including education, healthcare, and retail, to enhance employee training and development. However, it's also raising questions about its potential consequences and long-term implications. In this analysis, we'll examine the potential consequences and long-term implications that companies may encounter when implementing virtual reality technology as a primary tool for employee training and development. We'll consider factors such as the impact on knowledge retention, employee engagement, and overall productivity, as well as potential challenges including equipment costs, technical difficulties, and the need for specialized support staff. First, let's briefly review the benefits and drawbacks of virtual reality technology...