Multi-Objective Exploration: A Novel Reinforcement Learning Method to Solve Instruction Following and Mathematical Reasoning Problems

Team Members: Hugo Nathanael Yuwono

Emails: hyuwono@stanford.edu

Abstract

With the rapid progression in large language model (LLM) technology, we have seen its applications expand to a vast number of fields. Among which are the task of instruction following and mathematical reasoning. These tasks are paramount to the development of LLMs due to different reasons - the task of instruction following is highly related to the objective of maximizing the precision, lately, and reliability of LLMs while the task of mathematical reasoning is a common method of evaluating the efficacy of a model. In this paper, we attempt to solve these two tasks by utilizing a Qwen 2.5 0.5B Base Model that is augmented with both the defalut algorithms as well as some additional algorithms that make up our extension.

1 Introduction

The rapid growth of artificial intelligence models is reflected in the present use of prevalence of large language models (LLMs). One of the most crucial capabilities offered by LLMs is instruction following, which is the ability of LLMs to interpret and follow natural language instructions. This is anything but tangential towards the overarching objective of maximizing the precision, safety, and reliability of said models Zhou et al. (2023). However, instruction following is a problem that is yet to be completely decoded, which is conventionally attributed to the varied and ambiguous expressions of real human users Hill et al. (2020). Therefore, we are of the opinion that it is paramount to exercise reinforcement learning and its extensions to attempt to solve this problem.

Another area of interest that we plan to work on is mathematical reasoning, which is often utilized as a framework to develop and test the adaptive capability of new algorithms Shehper et al. (2024). This is due to the complexity posed by difficult research-level mathematical problems. Therefore, the problem may not be dissimilar to searching for a needle in a haystack. However, these problems do not only serve as a cost-effective and risk-free method of testing problems, but also has the benefit of potentially solving difficult mathematical problems and conjectures.

In this study, we will utilize a Qwen 2.5 0.5B Base Model that is augmented with Direct Preference Optimization (DPO) and REINFORCE Leave One-Out (RLOO) algorithms. On top of said algorithms, we also plan on implementing several extensions, namely Few-Shot Preference Optimization (FSPO) and Exploratory Preference Optimization (XPO).

2 Related Work

An existing study that explores the implementation of multi-objective optimization to perform instruction optimization is done by Yang and Li (2023). In this study, the algorithm starte off by initializing a parent population of instructions to start evolving before having it be manipulated by LLM-based operators to generate offspring. It then selected a parent instruction and a random parent instruction before having them both evaluated and extracted the definition and example from. It then uses said operator to generate mutated and crossoverd definitions and examples that are then combined with their original counterparts to augment the population. This study successfully outperforms the established benchmarks, though it suffers from potential crisis of local optima in the multi-objective optimization.

An attempt to devise reinforcement learning models to solve mathematical problems is reflected in the creation of AlphaGeometry, a theorem prover for Euclidean plane geometry that synthesizes millions of theorems and proofs, by Trinh et al. (2024). It is a neuro-symbolic system that utilizes a neural language model to direct a symbolic deduction engine through infinite branching points when encountering challenging problems. They first utilized an existing symbolic engines to extract hundreds of millions of synthetic theorems, which is then used to produce millions of synthetic proof steps. Said synthetic data is then utilized to pretrain a language model and had it fine-tuned to put emphasis on auxiliary construction during proof search. Deduction proof steps, on the other hand, are left to specialized symbolic engines. AlphaGeometry managed to correctly solve 25 of the 30 olympiad-level problems it was tested on, especially excels in geometry problems, and also has the ability to produce human-readable proofs, though said proofs become increasingly lengthy for more difficult problems, such as those that require construction. Despite its impressive performances, AlphaGeometry is constrained to high school olympiad-level problems that involve synthetic geometry. Moreover, it utilizes fixed symbolic language, which constraints its flexibility when solving problems that may present novel concepts.

3 Method

The techniques that we plan to implement on top of our Qwen Base Model are the DPO and RLOO algorithms as well as FSFO and Exploratory Preference Optimization. For the last 2 algorithms, we plan on implementing them independently then having them work in tandem. In other words, we will attempt a vast number of combinations of the approaches and algorithms we have implemented to see which results in the best performance. For those approaches, we plan on collecting a set of prompts to evaluate, using an inference engine to sample responses the trained model and a reference model, from where a reward score is generated for both the trained and reference model. In this section, we aim to illustrate them in greater detail.

3.1 Direct Preference Optimization (DPO)

Direct Preference Optimization (DPO) is a simpler counterpart of RLHF algorithms - maximizing reward with a KL-divergence constant Rafailov et al. (2023). It relies on a theoretical preference model, which is used to evaluate how well a reward function aligns with the existing data. DPO produces updates that increase the relative log probability of preferred to dispreferred responses. However, unlike a naive probability ratio objective, it prevents model degeneration by implementing a dynamic, per-example importance weight.

3.2 REINFORCE Leave One-Out (RLOO)

REINFORCE Leave One-Out (RLOO) is an algorithm built on top of REINFORCE and allows for the use of multiple online samples Ahmadian et al. (2024). REINFORCE, in turn, is a policy optimization algorithm that learns directly from experience. At the end of each trial, each weight w_{ij} is increased by $\Delta w_{ij} = \alpha_{ij}(r - b_{ij})e_{ij}$ where α_{ij} is a learning rate factor, b_{ij} is a reinforcement baseline, and e_{ij} is the characteristic eligibility of w_{ij} Williams (1992).

3.3 Few-Shot Preference Optimization (FSPO)

Few-Shot Preference Optimization (FSPO) is a meta-learning framework that makes use of the in-context learning capabilities of LLMs to produce policies that reflect a unique user and a dataset of their preferences Singh et al. (2025). It does so by sampling the training user, preferences from the user, and held-out preference example before predicting the loss based on said samples, whose gradient of loss on the sampled held-out preference is then used to update the learner parameters.

3.4 Exploratory Preference Optimization (XPO)

Exploratory Preference Optimization (XPO) is a variant of DPO with active exploration built on it Xie et al. (2024). Like the an online DPO algorithm, for each step, given the current policy and initial state, XPO samples a pair of trajectories, which is used to update the preference dataset. However, unlike DPO, which only samples from the current policy, XPO also explores from the reference policy. The optimism parameter α is utilized to determine the influence of the samples from the reference policy.

4 Experimental Setup

4.1 Datasets

We utilized the following datasets for our study - the SmolTalk Dataset for SFT (HuggingFaceTB/smol-smoltalk) Allal et al. (2025), the Huggingface Binarized Dataset for DPO and RLOO (HuggingFaceH4/ultrafeedback_binarized) HuggingFaceH4 (2024), and the Elix Generation Preference Dataset (Asap7772/elix_generations_gpt4omini_pref) Singh (2025) for FSPO.

Before utilizing them to train our model, we preprocessed said datasets. For the SmolTalk dataset, we stitched the messages by the user and the assistant together before using the Qwen 2.5 0.5B Base Model's tokenizer to encode the messages, which we then appended into our preprocessed dataset.

For the Huggingface Binarized dataset, we used the tokenizer to encode the accepted and rejected content (which we put right after the prompt) and appended said encoded content along with the prompt, prompt id, and attention masks to our preprocessed dataset.

4.2 Algorithms Implemented

As stated in the method segment, we decided to implement combinations of the previously stated algorithms. Those combinations include SFT independently, DPO independently, RLOO independently, a combination of SFT, DPO, and RLOO, FSPO with SFT, FSPO with DPO, a combination of FSPO with SFT and RLOO, combination of FSPO with DPO and RLOO, a combination of SFT and XPO, a combination of DPO and XPO, a combination of RLOO and XPO, a combination of SFT, DPO, RLOO, and XPO, a combination of FSPO with SFT, RLOO, and XPO, and a combination of FSPO with DPO, RLOO, and XPO.

4.3 Algorithm Configurations

For SFT, we used the following configuration - α (optimizer learning rate): 3×10^{-5}, portion of dataset used: 0.005, evaluation steps: 100, perplexity threshold: 1.1, training epochs: 2, and batch size: 1.

For DPO, we used the following configuration - α (optimizer learning rate): 5×10^{-6}, portion of dataset used: 0.05, β (preference optimization temperature): 0.2, evaluation steps: 100, perplexity threshold: 1.1, training epochs: 1, and batch size: 1.

For RLOO, we used the following configuration for preference reward modeling (PRM) - - and the following configuration for RLOO - .

4.4 Evaluation

To evaluate the performance of the Qwen 2.5 0.5B Base Model after our augmentations, we had the default model and the augmented model answer prompts contained within the leaderboard JSON files. Those files include an Ultrafeedback file (instruction following) and Countdown (mathematical reasoning).

To determine the model performance on Ultrafeedback, we compared the reward of the responses made by the default and augmented model to see which did better on each prompt. Said rewards are graded by the Llama 3.1 Nemotron 70B Reward Model and the number of times the augmented model outperformed the default model is tallied to calculate the win rate.

To determine the model performance on Countdown, we utilized a two stage reward that involves the following metrics - whether any answer was provided and whether the responses is correct Pan et al. (2025).

5 Results

5.1 Quantitative Evaluation

5.1.1 Ultrafeedback

The win rates of our augmentation combinations can be seen below:

Table 1: Win Rates on the Ultrafeedback Dataset

Augmentation 1	**Augmentation 2**	**Augmentation 3**	**Win Rate**
SFT			0.775
DPO			0.540
SFT	DPO		0.635

5.2 Qualitative Analysis

The Ultrafeedback results shows that SFT yields the best performance for the Ultrafeedback task and DPO lags behind. Moreover, even when DPO is used in tandem with SFT, the results are not on par with that of SFT.

6 Discussion

We feel that the primary limitation of this study rests in the depth of the mathematical reasoning task. Although we believe that it is important for our augmented models to be able to solve tasks that resemble that in the Countdown dataset, we think that for it to have a wider impact within the field of academia, we will have to expand the scope of the task. We aim to rectify this concern through what is listed in our future work segment.

7 Conclusion

7.1 Summary

Our study has shown that the augmentations - be it the default algorithms (SFT, DPO, and RLOO) as well as our extensions (FSPO and XPO) have successfully improved the model performance on our test sets.

7.2 Future Work

As potential work, we are interested in combining our study in the 2 fields - instruction following and mathematical reasoning - into one study. This way, we aim to not only solve problems that are similar to that in the Countdown dataset, but also real life mathematical problems that require linguistic reasoning.

Moreover, we are also interested in involving human baselines in our study instead or merely relying on the default Qwen 2.5 0.5B Base Model as the baseline and the Llama 3.1 Nemotron 70B Reward Model as the measuring stick. The reasoning behind it is that the tasks that we attempt to solve are very human-intuitive. Therefore, we believe that it is worth exploring whether involving humans in the study would yield better results.

8 Team Contributions

Hugo is responsible for writing the proposal and the paper (both milestone and the report), writing the code for the algorithms implemented, applying the extensions, debugging all issues that arose during the duration of this project, made the poster for the poster presentation, and presented said poster during the presentation session.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740* (2024).

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, and ... Thomas Wolf. 2025. HuggingFaceTB/smol-smoltalk. `https://huggingface.co/datasets/HuggingFaceTB/smol-smoltalk`. Accessed: 2025-05-07.

Felix Hill, Sona Mokra, Nathaniel Wong, and Tim Harley. 2020. Human instruction-following with deep reinforcement learning via transfer-learning from text. *arXiv preprint arXiv:2005.09382* (2020).

HuggingFaceH4. 2024. ultrafeedback_binarized Dataset. `https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized`. mit license; Accessed: 2025-05-10.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. 2025. TinyZero. `https://github.com/Jiayi-Pan/TinyZero` Accessed: 2025-06-02.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2023), 53728–53741.

Ali Shehper, Anibal M Medina-Mardones, Lucas Fagan, Bartłomiej Lewandowski, Angus Gruen, Yang Qiu, Piotr Kucharski, Zhenghan Wang, and Sergei Gukov. 2024. What makes math problems hard for reinforcement learning: a case study. *arXiv preprint arXiv:2408.15332* (2024).

Anikait Singh. 2025. Asap7772/elix_generations_gpt4omini_pref. `https://huggingface.co/datasets/Asap7772/elix_generations_gpt4omini_pref`. Tabular/text dataset ("prompt" + pairwise preference labels), parquet format; Accessed: 2025-06-09.

Anikait Singh, Sheryl Hsu, Kyle Hsu, Eric Mitchell, Stefano Ermon, Tatsunori Hashimoto, Archit Sharma, and Chelsea Finn. 2025. FSPO: Few-Shot Preference Optimization of Synthetic Preference Data in LLMs Elicits Effective Personalization to Real Users. *arXiv preprint arXiv:2502.19312* (2025).

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature* 625, 7995 (2024), 476–482.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8 (1992), 229–256.

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* (2024).

Heng Yang and Ke Li. 2023. Instoptima: Evolutionary multi-objective instruction optimization via large language model-based instruction operators. *arXiv preprint arXiv:2310.17630* (2023).

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911* (2023).