Extended Abstract

Motivation In recent years, large language model and its usage is thriving since it was first introduced to the world. Lots of companies and research centers focused on fine-tuning them to enhance the performance or for customization. In nowadays, large vision model was introduced and the researchers are starting to study them. There are not much research on fine-tuning the vision language model so I wanted to apply the reinforcement learning technique on vision language model in this project.

Method The model used in this project is LLaVA-NeXT model Zhang et al. (2024), a vision-language model developed by Meta. It utilizes large-language model, LLaMA3, with a visual assistant. For PPO implementation, TRLvon Werra et al. (2020) library is usually used. It provides the functions that enables PPO on fine-tuning large language model. Since the objective of this project is using vision language model, I needed to implement custom PPO without TRL library. The objective was to see if PPO could enhance the performance of VQA with vision-language model.

Implementation LLaVA-NeXT model has 8,377,427,968 in total, and utilizing low-rank adaptation (LoRA) Hu et al. (2022) PEFT method, I set only 22,151,168 parameters to be trainable, which are part of text generation, critical for answer generation. So, the efficient fine-tuning was possible. The reference model used was the same LLaVA-NeXT model with all parameters frozen. This model was required for PPO training. Its log probabilities output is used in computing policy loss with the fine-tuning model. The custom PPO Trainer calculates cosine similarity score and use it as a reward. Using the generated tokens of reference model, which has all parameters frozen, and the fine-tuning model, the custom PPO trainer calculates generalized advantage estimation (GAE) to get advantages and returns. With these calculated advantages and returns, the trainer gets policy loss. For value estimation, I added value head class that has three linear layers to give value prediction from the hidden states of generated tokens. The total loss the trainer uses is the sum of value loss, policy loss, and entropy loss from log probabilities of the model output. Figure 2 shows simplified PPO steps and where the value head class and the reference model was used.

Results The final reward return for training dataset was 0.38 and the one for validation dataset was 0.39. These are below the reward returns of pretrained LLaVA-NeXT model before fine-tuned.

Discussion The experiment of this project didn't show much improvement in rewards. First possible reason was the value head. When debugging to find the reason of low performance, I found out that the value head was not giving meaningful values. Even though I added two more layers to the initially implemented one-layer value head, it didn't make difference. Possibly related to this first reason, the second reason I thought of was PPO might not be a good choice to apply with this VQA dataset. Most of the target answers were very short. The answers were mostly containing one word or a few. Therefore, it's possible that the loss was 0 most of the time, there's not much to improve from the beginning, or the model is tending to overfit. If there were VQA dataset that consists of sentence-based target answers, I think PPO could have been effective.

Conclusion Unfortunately and unexpectedly, PPO trainer didn't work effectly on Visual Genome dataset with LLaVA-NeXT model. Due to some possible reasons, there are much to research on fine-tuning VLM on VQA problems with reinforcement learning based techniques.

Fine-Tuning Vision-Language Model with Reinforcement Learning for Visual Question Answering

Dayoung Kim
Department of Computer Science
Stanford University
dkim9613@stanford.edu

Abstract

In recent years, there have been lots of research on fine-tuning large language models (LLM). Utilizing the knowledge of research in LLM, this project implements reinforcement learning technique called, proximal policy optimization(PPO) and fine-tune visual language model (VLA) built with large vision model (LVM) and large language model. This project explains the custop PPO trainer and its implementation and evaluates the fine-tuned model's performance.

1 Introduction

In recent years, large language model and its usage is thriving since it was first introduced to the world. Lots of companies and research centers focused on fine-tuning them to enhance the performance or for customization. In nowadays, large vision model was introduced and the researchers are starting to study them. There are not much research on fine-tuning the vision language model so I wanted to apply the reinforcement learning technique on vision language model in this project.

Vision language model (VLM) is considered as a multi-modal since it deals with text data and image data. In vision question-answering (VQA) problems, the model takes an input of image and a question text to generate an answer text. To fine-tune this kind of model, similar technique that can be applied to large language model (LLM) can be used. Since there are billions of parameters used in large models, a technique called parameter-efficient fine-tuning (PEFT) Mangrulkar et al. (2022) is used a lot. It freezes most of the parameters that are pre-trained well enough and just trains some designated parameters. Therefore, efficient fine-tuning is possible even with limited resource. Adding to this, in question-answering problems, reinforcement learning is applied a lot. In LLM fine-tuning human feedback reinforcement learning Long Ouyang (2022) is used widely but it requires constant feedback generated by human. Therefore, in this VQA, I utilized proximal policy optimization (PPO) Zhai et al. (2024) with dataset that already has target answers. This method is try to make the model's answer generation to be good as the target answers.

This project uses Visual Genome dataset Ranjay Krishna ([n. d.]). It consists of RGB image of a scene, text data of question, and text data of answer. Visual Genome provides additional relationship and regional data of the image.

Stanford CS224R 2025 Final Report

Figure 1: Example of dataset: {"question": "What color is the clock?", "answer": "Green."}

Dataset	Total
Train	8379
Val	3591

Table 1: Summary of the Visual Genome Dataset

It has 11970 total set of images and VQA json data, and I splited them into 8379 training dataset and 3591 test dataset. Figure 1 shows the example of an image. The question and answer of this image is {"question": "What color is the clock?", "answer": "Green."}. The detailed processed dataset statistics are shown in Table 1.

2 Related Work

There hasn't been much study conducted on large vision model training. However, the following research conducts fine-tuning VLA on VQA problems. Fine-Tuning Large Vision-Language Models as Decision-Making Agents via Reinforcement Learning Zhai et al. (2024) fine-tune VLM to enhance decision-making. It utilized chain-of-thoughts (CoT) method. The used prompt is thoughts: "I am solving task T_t , given the current status of the task, I should choose a_t", action: a_t and the model is expected to give the recommended action. The paper uses traditional CNN-based RL method and evaluates with GymCards game data. It showed 20% higher performance on average than the one without CoT. The limitation of this approach is that the used evaluation dataset doesn't require complicated understanding of objects and their relationship. Also, it only provides evaluation on game dataset, which requires less understanding of objects than real-world scene images. The main adjustment is also focused on VLM prompting, so the performance is mainly explained based on the existence of CoT, not RL. It uses traditional RL-based fine-tuning method and does not focus on novel modification of it. Therefore, the effect of RL-based fine-tuning is not considered significant.

3 Method

The model used in this project is LLaVA-NeXT model Zhang et al. (2024), a vision-language model developed by Meta. It utilizes large-language model, LLaMA3, with a visual assistant. For PPO implementation, TRLvon Werra et al. (2020) library is usually used. It provides the functions that enables PPO on fine-tuning large language model. Since the objective of this project is using vision language model, I needed to implement custom PPO without TRL library. The objective was to see if PPO could enhance the performance of VQA with vision-language model. The original reward of LLaVA-NeXT model with training and validation dataset is shown in Table 2.

LLaVA-NeXT model has 8,377,427,968 in total, and utilizing low-rank adaptation (LoRA) Hu et al. (2022) PEFT method, I set a few layers and their parameters to be trainable, which are part of text generation, critical for answer generation. The chosen layers were "q_proj", "v_proj", "k_proj", "o_proj", "gate_proj", "down_proj", and "up_proj". So, the efficient fine-tuning was possible with limited resources. The reference model was the same LLaVA-NeXT model with all parameters

Dataset	Mean Average Cosine Similarity
Train	0.41
Val	0.40

Table 2: Mean Average Cosine Similarity before Reinforcement Learning Fine-tuning

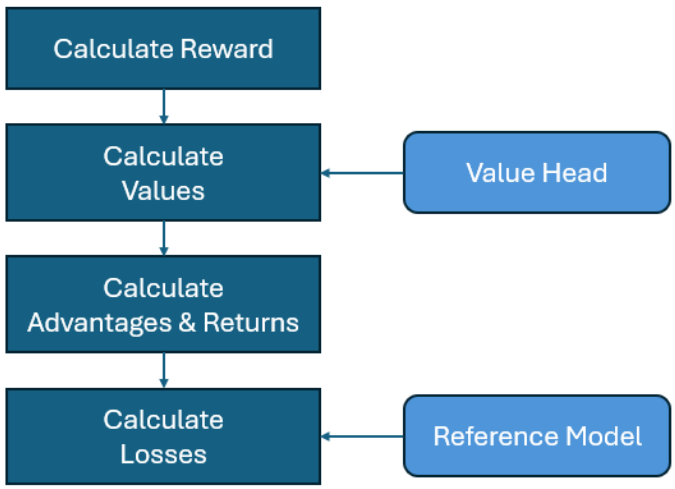

Figure 2: Custom PPO steps with additionally used model and class

frozen. This model was required for PPO training. Its log probabilities output is used in policy loss computation with the fine-tuning model with the fine-tuned model's generated logits. The custom PPO trainer calculates cosine similarity score and uses it as a reward. Additionally, from the generated tokens of reference model, which has all parameters frozen, and the fine-tuning model, the custom PPO trainer calculates generalized advantage estimation (GAE) to get advantages and returns. With all of these calculated advantages, returns, and logits, the trainer gets policy loss. For value estimation, I added value head class that has three linear layers to give value prediction from the hidden states of generated tokens. The total loss the trainer uses is the sum of value loss, policy loss, and entropy loss from log probabilities of the model output. Figure 2 shows simplified PPO steps and where the value head class and the reference model was used.

4 Experimental Setup

The learning rate for the vision language model's optimizer was set to 0.000005 and the learning rate for the value head optimizer was set to 0.00001. Vision language model's optimizer was lower since it's for fine-tuning a pre-trained model. Initially, I used 1 epoch and 1 batch size due to resource limit. I am planning to run once more before the final report submission with different epoch and batch size. The hyperparameters for PPO were set as gamma = 0.99, GAE lambda = 0.95, clip ratio = 0.2, value coefficient = 0.5, entropy coefficient = 0.01, and maximum gradient norm = 0.5. The final LoRA r was set to 8, LoRA alpha was was set to 32, and the LoRA dropout was set to 0.05. All experiments were conducted with local GPU servers, Tesla-V100.

5 Results

I expected the PPO based fine-tuned VLA will return higher average reward return than the one from pre-trained VLA before fine-tuned. However, the result was different from expected. The fine-tuned

model returned lower average reward. To find out the possible issues, I debugged the custom PPO trainer and value head implementation. I found out that value head's class doesn't seem to be learning information well from the dataset. The insight I thought of from this is that the length of the target answers in dataset. They mostly consist of one word or only a few words. I didn't realize before the training but this could have been the potential problem since there are not much information for value head to estimate value from.

5.1 Quantitative Evaluation

Dataset	Train Reward	Val Reward
LLaVA-NeXT	0.41	0.40
PPO Fine-tuned LLaVA-NeXT	0.38	0.39

Table 3: Reward Comparison of Origianl LLaVA-NeXT with Fine-tuned Model

5.2 Qualitative Analysis

The custom PPO trainer was implemented well without an issue and using the basic reward learning, the model was fine enough as well. However, the performance result is lower than expected and even lower than the base model. As mentioned above, the target answers in the dataset could be too short for PPO trainer to work effectively. The result could be low due to overfitting to the dataset or gradient vansishing/exploding problem due to lack of information to fit for.

6 Discussion

The experiment of this project didn't show much improvement in rewards. First possible reason was the value head. When debugging to find the reason of low performance, I found out that the value head was not giving meaningful values. Even though I added two more layers to the initially implemented one-layer value head, it didn't make difference. Possibily related to this first reason, the second reason I thought of was PPO might not be a good choice to apply with this VQA dataset. Most of the target answers were very short. The answers were mostly containing one word or a few. Therefore, it's possible that the loss was 0 most of the time, there's not much to improve from the beginning, or the model is tending to overfit. If there were VQA dataset that consists of sentence-based target answers, I think PPO could have been effective.

7 Conclusion

Unfortunately and unexpectedly, PPO trainer didn't work effectly on Visual Genome dataset with LLaVA-NeXT model. Due to some possible reasons, there are much to research on fine-tuning VLM on VQA problems with reinforcement learning based techniques. I couldn't find sentence-based VQA public dataset. Assuming VQA is researched based on short problems, there could be other reinforcment learning fine-tuning method that could work better on these datasets.

8 Team Contributions

- **Dayoung Kim: Implementation of custom PPO trainer. Fine-tuned LLaVA-NeXT model. Evaluated the fine-tuned model with the base model.**

Changes from Proposal **Original Hypothesis:** I planned to use a custom reward model and have it learned to figure out the weight values of α and β in equation (1).

$$Reward = \alpha \cdot cosine\_similarity(output, answer) + \beta \cdot CLIP(output, image) \quad (1)$$

Revised Hypothesis: Use cosine similarity only and apply proximal policy optimization (PPO) for fine-tuning the LVA model.

Justification: Due to the feedback from the project proposal that the original method doesn't seem feasible, I decided to use one reward. Additionally, I specified the reinforcement learning technique to use, which is PPO.

References

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=nZeVKeeFYf9

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 Ryan Lowe Long Ouyang, Jeff Wu. 2022. Training language models to follow instructions with human feedback. https://arxiv.org/abs/2203.0215.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

Oliver Groth Justin Johnson Kenji Hata Joshua Kravitz Stephanie Chen Yannis Kalantidis Li-Jia Li David A. Shamma Michael S. Bernstein Li Fei-Fei Ranjay Krishna, Yuke Zhu. [n. d.]. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. https://homes.cs.washington.edu/~ranjay/visualgenome/index.html

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. 2024. Fine-Tuning Large Vision-Language Models as Decision-Making Agents via Reinforcement Learning. https://openreview.net/forum?id=nBjmMF2IZU&referrer=%5Bthe%20profile%20of%20Yi%20Ma%5D(%2Fprofile%3Fid%3D~Yi_Ma4)

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024. LLaVA-NeXT: A Strong Zero-shot Video Understanding Model. https://llava-vl.github.io/blog/2024-04-30-llava-next-video/

A Additional Experiments

During the debugging, I modified the value head model in various ways. I started with one linear layer and added more layers due to policy optimization issues. Additionally, I tried with different activation function, tanh and leaky relu. However, they didn't improve the performance as well. Due to more debuggings, in my opinion, the most possible issue that makes PPO not effective would be the simplicity of target answers. More VQA research and created sentence-based dataset could make difference in the future.

B Implementation Details

The code for this project is uploaded https://github.com/LeannaDayoungKim/cs224r_project.