

GRPO Finetuning for VLM Mathematical Reasoning

by

Tayyib Chohan

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR'S OF APPLIED SCIENCE

in

The Faculty of Electrical and Computer Engineering
(Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

April 2025

© Tayyib Chohan 2025

GRPO Finetuning for VLM Mathematical Reasoning

submitted by **Tayyib Chohan** in partial fulfillment of the requirements for the degree of **Bachelor of Applied Science in Computer Engineering**.

Examining Committee:

Renjie Liao, Professor, Electrical and Computer Engineering, UBC
Supervisor

Muchen Li, PhD, Computer Science, UBC
Supervisory Committee Member

Tanzila Rahman, Postdoctoral Fellow, Computer Science, UBC
Supervisory Committee Member

Abstract

Vision Language Models (VLMs) like GPT-4v[1] have demonstrated remarkable capabilities but struggle with visual mathematical reasoning tasks. These tasks require numerical precision, multi-step logic, and accurate interpretation of visual elements like charts and diagrams. This thesis investigates the potential of Group Relative Policy Optimization (GRPO), a reinforcement learning technique, to enhance these specific capabilities in VLMs. We finetuned the Qwen2VL-7B model using GRPO with reward signals based on answer accuracy and response formatting. The study also explored the impact of GRPO by comparing full model finetuning against scenarios with selectively frozen vision or language components. Our findings indicate that GRPO finetuning yields observable improvements in mathematical reasoning accuracy compared to the baseline model. Notably, finetuning the complete model produced better performance than freezing the vision or language components, suggesting potential benefits to adapting both modalities. Additionally, we conducted a qualitative analysis that revealed persistent error types in evaluation. While GRPO clearly enhances visual-mathematical reasoning in VLMs, it does not solve every problem. Gaps remain around fine-grained graph interpretation and formula application. Future work should explore more sophisticated RL strategies, from carefully tuned reward functions to innovative model architectures.

Lay Summary

Today’s AI models excel at understanding text and images—but they often stumble when asked to solve math problems shown in pictures, like interpreting a complex chart or performing step-by-step calculations from a diagram. This thesis examines Group Relative Policy Optimization (GRPO), a reinforcement-learning method that “rewards” the model for correct, well-formatted answers. By fine-tuning a vision-language model with GRPO, we boosted its accuracy on multi-step visual math tasks. Our experiments reveal that adapting both the vision and language components together delivers the greatest gains, although challenges remain in reading fine graph details and applying formulas exactly. These findings set the stage for future improvements to VLMs.

Preface

This work was conducted with the Deep Structured Learning Lab at UBC under the guidance of Renjie Liao. I was responsible for designing and running experiments, reviewing and presenting relevant literature, and compiling the results. I extensively consulted with Muchen Li and Tanzila Rahman, whose suggestions, insights, and expertise helped guide the project.

Table of Contents

Abstract	iii
Lay Summary	iv
Preface	v
Table of Contents	vi
List of Figures	viii
Acknowledgements	ix
1 Introduction	1
1.1 Problem Statement	1
1.2 Contribution	2
2 Literature Review	3
2.1 Introduction to VLMs	3
2.2 Foundational VLM Architectures and Advancements	3
2.3 VLM Datasets for Reasoning Tasks	4
2.4 Fine-tuning and Alignment Strategies for VLMs	5
2.4.1 Supervised Finetuning	5
2.4.2 Alignment	5
2.5 Reinforcement Learning for VLM Alignment And Tuning	5
2.5.1 RL for transformer based models	6
2.5.2 RL from Human Feedback	6
2.6 Group Relative Policy Optimization (GRPO)	7
3 Experimental Setup: GRPO Finetuning Trials for Math Reasoning	8
3.1 Base Model	8
3.2 Datasets	8

Table of Contents

3.3	Finetuning Procedures and Hardware	9
3.4	Experimental Design	9
3.4.1	Reward Functions	10
4	Results and Findings	11
4.1	Quantitative Results	11
4.2	Qualitative Results	13
4.2.1	Failure Cases and Trends in MathVista TestMini Responses	14
5	Conclusion	20
5.1	Future work	20
5.2	Limitations	20
5.3	Closing Remarks	20
	Bibliography	22
 Appendices		
A	Additional Test Results and Example Responses	25
A.1	Qwen2VL-2B Test	25
A.2	Geometry Response Example	25
A.3	Physics Response Example	26

List of Figures

4.1	GRPO Benchmark Accuracy Evaluation	11
4.2	GRPO Training Accuracy Reward	12
4.3	GRPO Training Format Reward	12
4.4	GRPO Training Total Reward	13
4.5	GRPO Training Completion Length	13
4.6	Example Irrelevant Question	14
4.7	Chart Question Example	15
4.8	Stem Leaf Plot Image Example	15
4.9	Unit Blocks Image Example	16
4.10	Measuring Tool Image Example	16
4.11	Physics Image Example	17
4.12	Geometry Image Example	17
4.13	Precision Error Example	18
4.14	Reasoning Disconnect Example	19
A.1	Qwen2VL-2B Accuracy Reward	25
A.2	Geometry Image Example Full	26
A.3	Physics Image Example Full	27

Acknowledgements

I must express my gratitude to Muchen Li, Tanzila Rahman, and Renjie Liao for their direct help with this project. They helped me tremendously and were paramount in my learning. Additionally, I would like to extend my thanks to the rest of the Deep Structured Learning lab. I attended various reading groups and talks in which I gained exposure to new research, ideas, and concepts.

Chapter 1

Introduction

Recent advances in transformer-based Large Language Models (LLMs) have been a significant development in the application of AI to many new problems. Many of the problems humans interact with are inherently multi-modal as they require vision such as geometric reasoning, counting, and physics. This signifies that language capabilities alone are not adequate to solve a variety of fundamental problems. Solving these problems is the goal of Vision Language Models (VLMs)

Modern Vision Language Models are typically large transformer-based models. They are able to complete a variety of tasks, including captioning, describing images, and Visual Question Answering (VQA). These models perform very well on simple problems that do not require complex or logical reasoning[2][3]. There is a significant gap in the capabilities for these more challenging problems that require vision. These visual reasoning tasks go beyond object detection and require nuanced image understanding in context to language and domain-specific contexts. For example, a geometry question that requires the Pythagorean theorem requires not only recognizing the need to use the theorem but also how to apply it to the specific image.

Visual mathematical reasoning presents a unique set of challenges, as it requires precise perception of fine-grained detail, understanding of language and logic, symbolic reasoning, and translating between visual and linguistic knowledge

1.1 Problem Statement

Modern vision language models are far from perfect for mathematical reasoning tasks [2]. Common failure modes of these models include poor multi-step reasoning, numerical precision, and incorrectly applying techniques or strategies. This thesis addresses the problem of enhancing the mathemati-

cal reasoning capabilities of VLMs by exploring the efficacy of Group Relative Policy Optimization (GRPO)[4], a reinforcement learning technique, to specifically improve numerical accuracy in visual mathematical problem-solving.

1.2 Contribution

This paper provides a qualitative evaluation of GRPO’s improvements to mathematical reasoning in VLMs and examines some of the architectural effects of GRPO through parameter freezing.

Chapter 2

Literature Review

2.1 Introduction to VLMs

VLMs have rapidly evolved over recent years. The advent of the transformer architecture has been a significant part of these changes[5]. Much work has been done exploring how visual information can be effectively integrated into the Large Language Models (LLMs) of today[6][7][8]. Many techniques have been employed to improve these models for human alignment and domain expertise. This chapter reviews the progression of VLMs, highlighting impactful literature pertinent to understanding their current state and providing context for the contributions made in this thesis. We will delve into the challenges of visual reasoning, with a specific focus on mathematical reasoning and the datasets used for evaluation. Following this, a brief overview of Supervised Fine-Tuning (SFT)[9] as a standard adaptation method will be provided. Finally, we will discuss the evolution of reinforcement learning techniques for model alignment, including Reinforcement Learning from Human Feedback (RLHF)[10] and Group Relative Policy Optimization (GRPO), which is central to this work.

2.2 Foundational VLM Architectures and Advancements

The introduction of the transformer and the related Vision Transformer (ViT) allowed for high performance models with visual and textual information being converted to the same intermediary representation as tokens[11]. Despite this, early attempts at vision language models struggled with aligning textual and visual information. This meant that models struggled to understand the relationship between text and images. Contrastive Language-Image Pretraining (CLIP) was a significant milestone in this problem[12]. In short, this allowed for a vision and text encoder with a shared understanding of images and their associated words. This meant the representation for the

2.3. VLM Datasets for Reasoning Tasks

image of something would look similar to the words describing that image via cosine similarity. Due to this, CLIP could solve classification problems it wasn't explicitly trained for by comparing its embeddings.

Papers like Bootstrap Language Image Pre-training (BLIP)[13] expanded on the ideas introduced in CLIP allowing for captioning and image-text matching tasks. Later models like LLaVA[6] and BLIP-2[14] used ViTs trained in a similar fashion to CLIP in conjunction with LLMs with a projection layer between. These models also incorporated multistage training strategies with parameter freezing. These performed extremely well and allowed for generalized visual question answering. Scaling this general architecture of integrating a ViT with a large language model with more data and parameters was done in subsequent papers[15][16] showing significant improvements. Additionally, there has been work done on image preprocessing[15][7], intermediary vision-language layers[14][6], and even new positional encoding schemes[8].

The current landscape of open source VLMs is dominated by the Vision Encoder + Projection Layer + LLM architecture due to its effectiveness[8]. This modular design allows leveraging independently developed, powerful unimodal models. Contrastive pre-training remains a cornerstone for learning robust visual representations. Furthermore, multi-stage training strategies with parameter freezing are common.

2.3 VLM Datasets for Reasoning Tasks

The BLINK[3] and MMMU[17] datasets both showcase a large gap in the capabilities of vision language models and humans in visual perception capabilities. In these datasets we see problems more complex than recognition, classification, captioning. These tasks require understanding of what is happening in the image and how it relates to spatial relationships, causal relationships, perspective, or domain-specific concepts like math or physics. These tasks encompass a diverse range of problems and apply to many domains.

Mathematical reasoning is a particularly challenging area for VLMs. Leading models struggle to understand complex figures, perform rigorous reasoning, and analyze spatial position [2][18]. Many questions that are straightforward for humans are difficult for VLMs to answer[3]. MathVision identified that 42% and 32% of GPT-4V's[1] errors in their dataset are from "Reasoning" and "Vision Recognition" respectively[2]. Performance on benchmarks

like this may reveal gaps between current VLMs and the robust, precise reasoning required for reliable mathematical problem-solving.

2.4 Fine-tuning and Alignment Strategies for VLMs

The significant gaps observed between current VLM capabilities and the requirements for robust mathematical reasoning, as highlighted by performance on challenging benchmarks necessitate exploring advanced methods beyond standard pre-training. Fine-tuning and alignment techniques play a crucial role in adapting pre-trained models to specific downstream tasks and desired behaviors, such as logical consistency, numerical accuracy, and domain expertise.

2.4.1 Supervised Finetuning

Supervised fine tuning (SFT) is one of the most common techniques used to improve task specific model performance. It involves adding an additional training step to the model in which we train using a dataset that is representative to the specific task we are aiming to improve. [9] Effective use of SFT involves a robust dataset of task specific input-output pairs which may not be available.

2.4.2 Alignment

Reinforcement Learning Human Feedback (RLHF) is a technique used for aligning models to human based objectives that are harder to define in a strict dataset[10]. Reinforcement learning techniques don't necessarily require a ground truth label but instead require a reward signal. RLHF generates this reward from human responses to answers. Further specifics of RLHF and reinforcement learning are discussed in Section 2.5

2.5 Reinforcement Learning for VLM Alignment And Tuning

Reinforcement Learning (RL) is a powerful technique that can fine-tune VLMs. It aims to align model behavior with complex objectives that may be difficult to specify or optimize effectively using SFT alone [4]. RL allows

models to learn through optimizing a reward signal and as a result learn a policy that maximizes expected cumulative reward.

2.5.1 RL for transformer based models

Applying RL to large transformer-based models required improvements to Policy Gradient Methods. These are reinforcement learning algorithms that allow the model policy to be updated by gradient ascent of the estimated expected reward. Some key policy gradient techniques include:

- **Trust Region Policy Optimization (TRPO):** [19] introduced constraints for more stable learning compared to earlier policy gradient techniques
- **Proximal Policy Optimization (PPO):** [20] simplified TRPO while maintaining strong performance, becoming the standard algorithm for RL-based fine-tuning of large language and vision-language models due to its balance of sample efficiency, ease of implementation, and stability

2.5.2 RL from Human Feedback

RLHF [10] is one of the most highly successful applications of RL in large transformer-based models. This technique specifically aims to align models with human preferences and instructions. The typical RLHF process involves:

1. **SFT Pre-training:** Training a base model using supervised fine-tuning.
2. **Reward Modeling:** Training a separate model to predict human preferences between pairs of model-generated outputs. Human labelers rank different responses, and the reward model learns to assign higher scores to preferred outputs.
3. **RL Fine-tuning:** Further tuning the model using an RL algorithm (commonly PPO) where the reward signal is provided by the trained reward model.

RLHF has proven effective in enhancing the helpfulness, instruction-following capabilities, and safety of large models [10]

2.6 Group Relative Policy Optimization (GRPO)

GRPO was introduced in DeepSeekMath in order to enhance mathematical reasoning capabilities while also optimizing the memory usage of PPO[4]. In short the primary difference between the two techniques is PPO uses a learned “value” function to estimate the advantage of a specific action or output while GRPO does not do this explicitly. Instead GRPO generates multiple outputs and estimates the normalized advantage of each action relative to group. The number of outputs is a hyperparameter called `num_generations`. The relative normalized advantage of an action \hat{A}_i is simply: $\frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$. Where \hat{A}_i is the advantage for an output o_i and r_i is the reward for the output o_i .

Chapter 3

Experimental Setup: GRPO Finetuning Trials for Math Reasoning

In this chapter we detail the experiments conducted to improve VLMs for Math Reasoning.

3.1 Base Model

The VLM we chose as the basis of our experiments is Qwen2VL-7B[8]. Qwen2VL is part of the Qwen large model series developed by Alibaba Cloud, specifically designed as a large vision-language model capable of handling diverse multimodal tasks. We chose the 7 billion parameter variant as initial tests showed the smaller variant Qwen2VL-2B, might lack the capacity for logical reasoning involved in the target benchmarks[REFER Appendix test]. In addition, larger models would require significantly more computational resources.

3.2 Datasets

The selection of appropriate datasets are critical for finetuning and evaluating VLMs, especially for specialized tasks like visual mathematical reasoning. We utilized the following datasets tailored for mathematical reasoning:

- **MathVista:** [18] This dataset is a diverse collection of visual mathematical reasoning problems. It encompasses areas such as geometry, statistical charts, general mathematics. It includes various image types (diagrams, plots, natural images) and provides image-question-answer triplets, often including reasoning steps. We utilized the testmini split for training purposes and also as one of the evaluation sets.

- **MathVision:** [2] This dataset, also known as MATH-V, provides a collection of mathematical problems with visual contexts sourced from math competitions. The questions span 16 disciplines and 5 difficulty levels. The dataset aims to provide a diverse and challenging evaluation of mathematical reasoning against human performance. We utilized the test and test-mini split for evaluation.

3.3 Finetuning Procedures and Hardware

The specific GRPO implementation we used was based on the R1-V[21] implementation on GitHub. All models were trained using the following parameters:

- num generations 4
- max prompt length 1024
- max completion length 1024
- per device train batch size 1
- gradient accumulation steps 2
- learning rate 1e-6
- warmup steps 50

The remaining parameters were kept to their default values and further details can be found in the appendix [CITE]. Multiple choice questions options were included in the prompt question[CITE APPENDIX]. Training checkpoints were saved at 700 and 1700 steps or 1.4 and 3.4 epochs respectively. Typical training runs were conducted on a single NVIDIA A100 80GB GPU taking up to 12 hours per run.

3.4 Experimental Design

In order to explore the effects of GRPO on the VLM architecture we evaluate four models. The first is the baseline model. This is the performance of Qwen2VL-7B without any modifications. After this, we evaluate the effects of applying GRPO to the entire model. Next we examine the changes when only the visual encoder portion of the model is left unfrozen. Finally, we examine the converse situation in which the vision encoder is frozen.

3.4.1 Reward Functions

We utilized two reward functions. The first was a format reward. This reward returned a value of 1 if the response followed the format described in DeepSeekMath[4]. In other words the response must use the tags "think" and "answer" to format its response. Otherwise the format reward was 0. This was done via Regex pattern matching.

Additionally, we incorporated an accuracy reward. When the model answer is exactly the same as the correct answer it received an accuracy reward of 1 and otherwise received a 0.

Chapter 4

Results and Findings

4.1 Quantitative Results

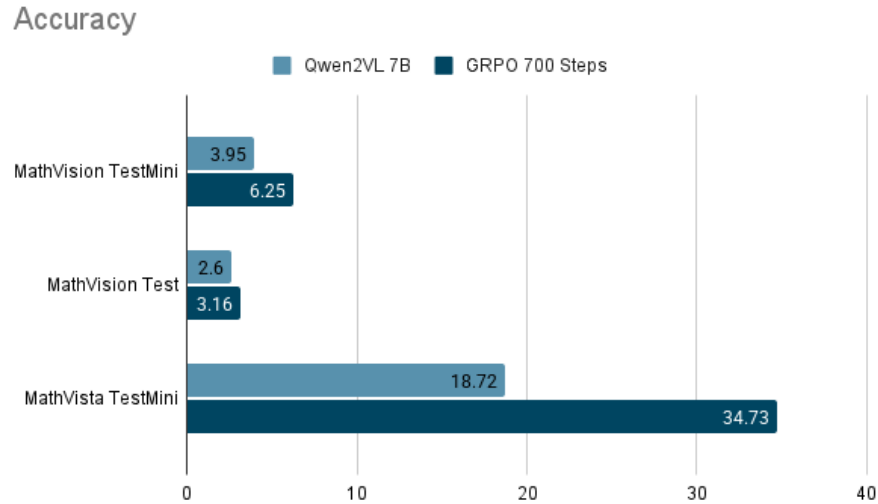


Figure 4.1: Here is an evaluation of GRPO’s performance for MathVista testmini (our finetuning set) and MathVision, our evaluation set

We see a notable improvement in test accuracy after incorporating GRPO for math reasoning. On the MathVision test split we observe a 21.5% improvement in accuracy relative to the baseline Qwen2VL-7B accuracy Figure 4.1. There is a notable drop in training accuracy after incorporating parameter freezing in both cases. The drop is more significant when the non-vision portion of the model is frozen.

4.1. Quantitative Results

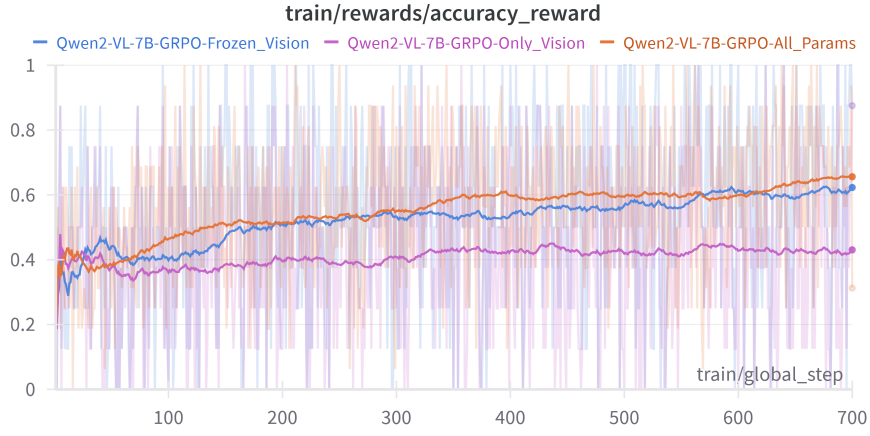


Figure 4.2: Here we have the training curves for the accuracy reward of each model per step

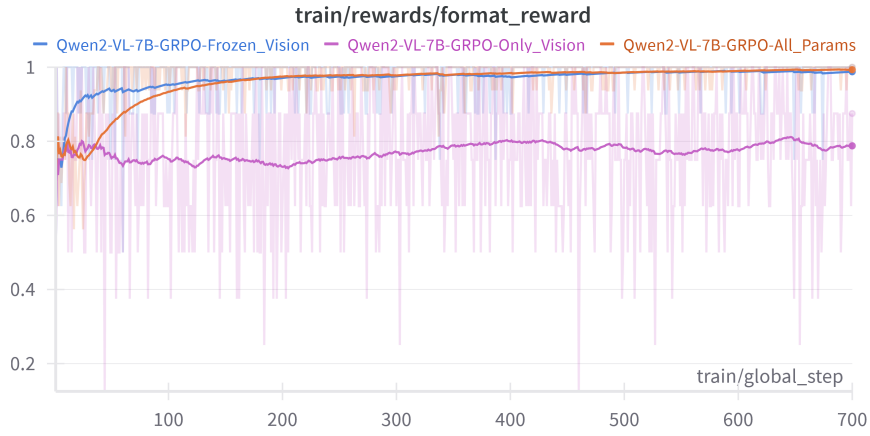


Figure 4.3: Here we have the training curves for the format reward of each model per step

4.2. Qualitative Results

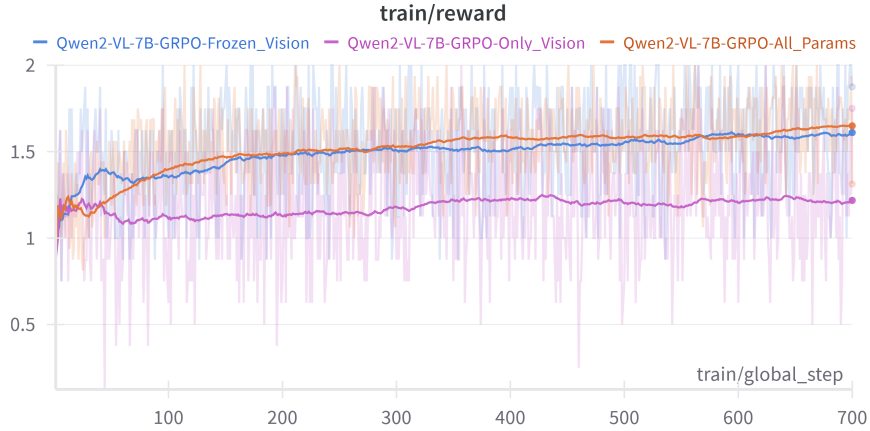


Figure 4.4: Here we have the training curves for the total reward of each model per step

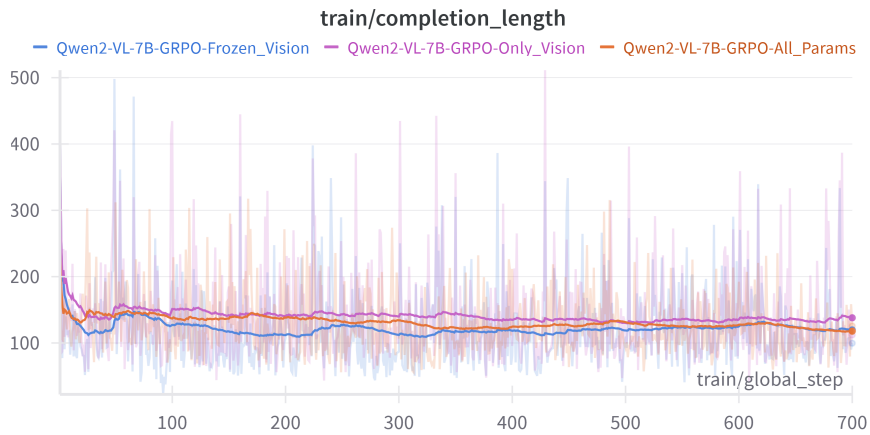


Figure 4.5: Here we have the completion length of each model's responses per step

4.2 Qualitative Results

Response quality and answer trends are difficult to capture with numerical analysis alone. Thus, we manually examined incorrect responses throughout

training to look for trends.

4.2.1 Failure Cases and Trends in MathVista TestMini Responses

Irrelevant Questions

A significant portion of the testmini split of the MathVista dataset contain irrelevant questions Figure 4.6. Specifically 51 instances or 5% of the dataset contain questions asking the age of individuals. Although these questions represent a challenging visual language problem to LLMs this question type does not pertain to math reasoning. Given the limited context of many of these questions, they involve identification and classification of people without complex reasoning. The VLM often resorted to responding with zero or mentioning not enough information was provided.



Figure 4.6: This image was provided with the question "What is the age gap between these two people in image?". This question answer pair in MathVista’s dataset does not reflect the target of problem of mathematical reasoning

Reading Fine Details in Graphs

Many incorrect responses involved reading fine-grained details in graphs. This was a commonly observed error type and showcases a possible area for further improvement in VLMs.

4.2. Qualitative Results

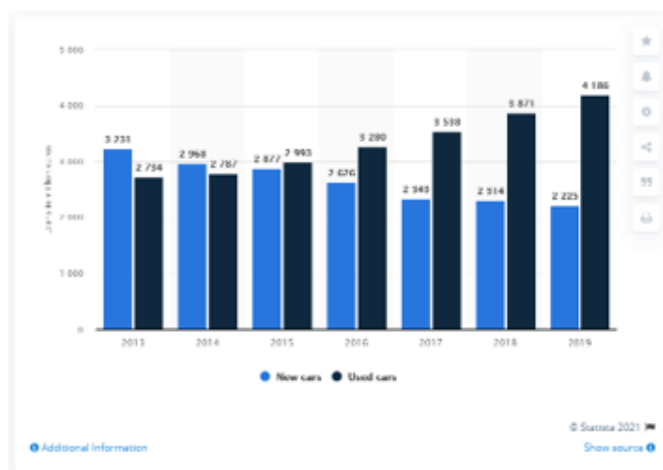


Figure 4.7: This image was provided with the question "Which year has the least difference between the used and new cars?". The answer is 2015 but the VLM answered 2016

Stem and Leaf Plots

Stem and leaf plot problems appear difficult for the VLM. This may be due to a lack of domain knowledge on how to read these specific charts. More detailed analysis is required to confirm this.

Billboards per city	
Stem	Leaf
4	1 1
5	0 2 7
6	0 2 3 6
7	1 2 4
8	1 2
9	0

Figure 4.8: This image was provided with the question "The advertising agency counted the number of billboards in each city in the state. How many cities have fewer than 70 billboards? (Unit: cities)". The answer is 9 but the VLM answers varied from 10-19

Counting Blocks

The VLM often failed to count unit blocks. Reading the thinking portion of the response showed that many errors were due to misunderstanding what each block represented, while other errors were from incorrectly counting the number of blocks.



Figure 4.9: This image was provided with the question "What number is shown?". The answer is 38 but the VLM answered 39

Reading Measuring Tools

Some questions involved reading measuring tools such as rulers, scales, or scientific equipment. The VLM incorrectly read these tools. This was not universally true for all such problems, but was nonetheless a common incorrect question type.

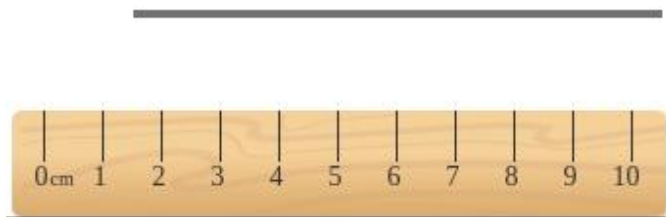


Figure 4.10: This image was provided with the question "Move the ruler to measure the length of the line to the nearest centimeter. The line is about (.) centimeters long." The answer is 9 but the VLM answered 7

Physics and Formulas

Incorrectly applying physics and math formulas is a common error the VLM made. It would often mention valid techniques and strategies, yet apply

them incorrectly Appendix A.3.

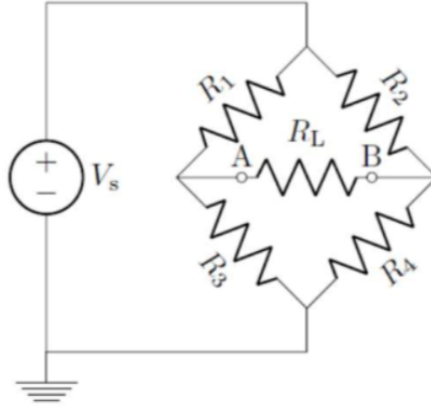
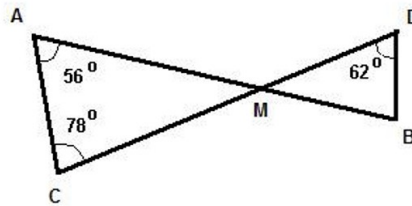


Figure 4.11: This image was provided with the question "Given $V_s = 5V$, $R_1 = 1k\Omega$, $R_2 = 2.2k\Omega$, $R_3 = 2.2k\Omega$, $R_4 = 1.5k\Omega$, and $R_L = 4.7k\Omega$. Determine the voltage and current across R_L . Answer in unit of V (3 sig.fig.)?". The answer is 1.06 but the VLM answered 3.4

Geometric Reasoning

Like in physics problems, the VLM failed to correctly apply relevant techniques to solve the problem despite mentioning them in the reasoning response Appendix A.2.



www.analyzemath.com

Figure 4.12: This image was provided with the question "Find the size of angle MBD in the figure below". The answer is 72 but the VLM answered 23

4.2. Qualitative Results

Output Formatting/Precision

One occasional source of errors is precision errors. Specifically, an answer may be correct, but the response provides more or fewer digits than requested. This type of error represents a case where visual reasoning is not the error but the instruction following is. Fortunately, these cases were rare.

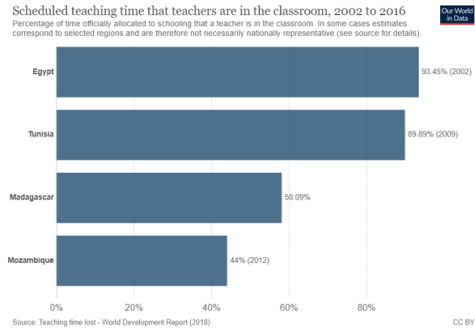


Figure 4.13: This image was provided with the question "Find out the average of the bottom two countries?". The answer is 51.04 but the VLM answered 51.045

Reasoning Disconnect

A rare instance of errors is reasoning disconnect. In this case, the VLM correctly identifies the answer while thinking but answers incorrectly. This was exceptionally rare, but it is not clear why these errors are observed.

4.2. Qualitative Results

Price	Quantity demanded	Quantity supplied
\$250	14,400	1,200
\$430	11,300	5,700
\$610	8,200	10,200
\$790	5,100	14,700
\$970	2,000	19,200

Figure 4.14: This image was provided with the question "Look at the table. Then answer the question. At a price of \$790, is there a shortage or a surplus?". The answer is surplus but this was VLM's full response: <think> At a price of \$790, the quantity demanded is 5,100 and the quantity supplied is 14,700. Since the quantity demanded is less than the quantity supplied, there is a surplus.</think> <answer>shortage</answer>

Chapter 5

Conclusion

5.1 Future work

There are several avenues for future research continuing this work. Exploring more sophisticated reward functions for GRPO that evaluate intermediate steps, rather than just final answer accuracy and format, could potentially guide the model towards more coherent and correct problem-solving processes. This could also help with reward scarcity. Investigating the application of GRPO to larger, more capable base VLMs could also yield further improvements. Additionally, curating more targeted finetuning datasets that specifically address the common failure modes identified could be beneficial. It could also be useful to examine other advanced RL techniques like Direct Preference Optimization (DPO) within this specific VLM context would also be valuable.

5.2 Limitations

This study has several limitations. The experiments were conducted using a single base VLM (Qwen2VL-7B) and primarily one dataset (MathVista testmini) for training, which may limit the generalizability of the findings. The reward functions used were relatively simple, focusing on final answer accuracy and format, potentially missing nuances in reasoning quality. The scale of the experiments was also constrained by available computational resources, limiting the number of training steps and hyperparameter exploration.

5.3 Closing Remarks

Improving the mathematical reasoning capabilities of Vision Language Models is a significant challenge at the intersection of computer vision and natural language processing. This thesis demonstrated that GRPO is a viable

5.3. *Closing Remarks*

pathway for improving VLM performance on visual mathematical tasks, yielding measurable gains over baseline models. However, our qualitative analysis highlights the complexity of these tasks. There are remaining gaps in areas such as precise perception, robust multi-step reasoning, and reliable application of domain knowledge. Techniques like GRPO push the boundaries, but it is clear that achieving human-level proficiency in visual mathematical reasoning will likely require continued advancements in model architectures, training methodologies, and reward engineering.

Bibliography

- [1] OpenAI. Gpt-4v(ision) system card, September 2023. [Online]. Available: <https://openai.com/index/gpt-4v-system-card/>. [Accessed: Apr. 23, 2025].
- [2] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.
- [3] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive, 2024.
- [4] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [6] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.
- [7] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and

- Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, 2024.
- [8] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution, 2024.
- [9] Xiaohu Jiang, Yixiao Ge, Yuying Ge, Dachuan Shi, Chun Yuan, and Ying Shan. Supervised fine-tuning in turn improves visual foundation models, 2024.
- [10] 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. Training language models to follow instructions with human feedback, 2022.
- [11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.
- [12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.
- [13] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.
- [14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.
- [15] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024.

- [16] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.
- [17] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024.
- [18] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024.
- [19] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017.
- [20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- [21] Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-v: Reinforcing super generalization ability in vision-language models with less than \$3. <https://github.com/Deep-Agent/R1-V>, 2025. Accessed: 2025-02-02.

Appendix A

Additional Test Results and Example Responses

A.1 Qwen2VL-2B Test

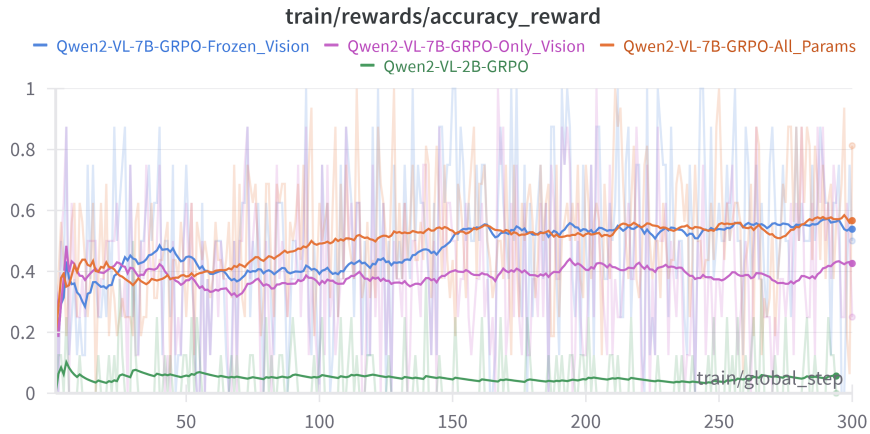


Figure A.1: This is a training curve for an early experiment exploring Qwen2VL-2B for math reasoning fine tuning via GRPO.

A.2 Geometry Response Example

Question: Find the size of angle MBD in the figure below

Solution: 72

Response: <think> The sum of the angles in a triangle is 90° .

- In triangle ACD, the angles are given as 56° , 78° , and angle ACD (which we need to find).

- The angle at A is 78° and the angle at D is 56° , so angle ACD = $180^\circ - 78^\circ$

A.3. Physics Response Example

- $56^\circ = 46^\circ$.
 - Therefore, the measure of angle MBD is half of angle ACD, since M is the midpoint of AD.
 - So, angle MBD = $46^\circ / 2 = 23^\circ$.
- <answer>23°</answer>

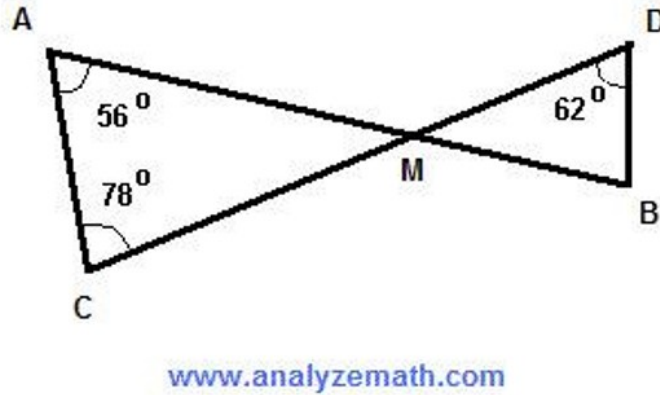


Figure A.2: Geometry problem in MathVista Dataset

A.3 Physics Response Example

Question: Given $V_s = 5\text{V}$, $R_1 = 1\text{k}\Omega$, $R_2 = 2.2\text{k}\Omega$, $R_3 = 2.2\text{k}\Omega$, $R_4 = 1.5\text{k}\Omega$, and $R_L = 4.7\text{k}\Omega$. Determine the voltage and current across R_L . Answer in unit of V (3 sig.fig.).

Solution: 1.06

Response: <think>First, we need to find the equivalent resistance of all the resistors in parallel. Since R_2 and R_3 are both $2.2\text{k}\Omega$, they have the same resistance and can be treated as a single resistor with a value of $1.1\text{k}\Omega$ in parallel with R_1 and R_4 . The equivalent resistance of these two parallel resistors is:

$$R_{eq} = \frac{R_1 \times R_2}{R_1 + R_2} = \frac{1.8\text{k}\Omega \times 1.8\text{k}\Omega}{1.8\text{k}\Omega + 1.8\text{k}\Omega} = 0.9\text{k}\Omega$$

Finally, we can combine R_{eq} with R_L in parallel to get the total equivalent resistance of the circuit. The total resistance across the battery is:

$$R_{total} = \frac{0.9\text{k}\Omega \times 4.7\text{k}\Omega}{0.9\text{k}\Omega + 4.7\text{k}\Omega} = 1.73\text{k}\Omega$$

A.3. Physics Response Example

Now, we can use Ohm's law to find the current and voltage across R_L . The voltage across R_L is:

$$V_L = \frac{(V_s * R_L)}{R_{total}} = \frac{(5V * 4.7k\Omega)}{1.73k\Omega} = 13.5V$$

And the current is:

$$I_L = V_L/R_L = 13.5V/4.7k\Omega = 2.86A$$

</think>

<answer>3.4V</answer>

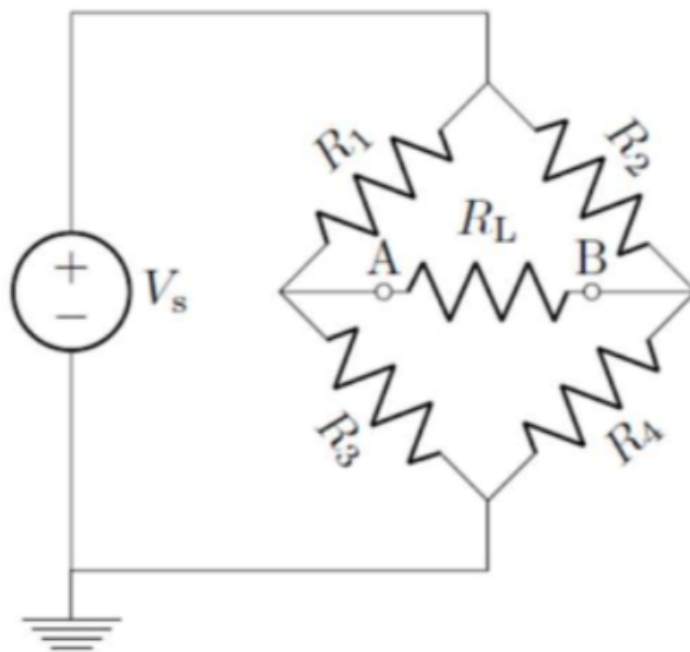


Figure A.3: Physics problem in MathVista Dataset

Appendix B

Full Parameter List and Prompt List

B.1 Question Prompt

All questions from the dataset were formatted in the following way:

- "{Question} Output the thinking process in <think> </think> and final answer (number) in <answer> </answer> tags."

Additionally, if the question is multiple choice the following text is also appended:

- " Choose from the following options " + " ".join(x["choices"])

Where x[choices] is the array containing the answer choices

B.2 Training Parameters and Meta Data

- `_attn_implementation_autoset`: true
- `_name_or_path`: "Qwen/Qwen2-VL-7B-Instruct"
- `accelerator_config`:
 - `dispatch_batches`: null
 - `even_batches`: true
 - `gradient_accumulation_kwargs`: null
 - `non_blocking`: false
 - `split_batches`: false

- use_seedable_sampler: true
- adafactor: false
- adam_beta1: 0.9
- adam_beta2: 0.999
- adam_epsilon: 1×10^{-8}
- add_cross_attention: false
- architectures (1 collapsed):
- attention_dropout: 0
- auto_find_batch_size: false
- average_tokens_across_devices: false
- bad_words_ids: null
- batch_eval_metrics: false
- begin_suppress_tokens: null
- beta: 0.04
- bf16: true
- bf16_full_eval: false
- bos_token_id: 151,643
- chunk_size_feed_forward: 0
- cross_attention_hidden_size: null
- data_seed: null
- dataloader_drop_last: false
- dataloader_num_workers: 0
- dataloader_persistent_workers: false
- dataloader_pin_memory: true
- dataloader_prefetch_factor: null
- ddp_backend: null
- ddp_broadcast_buffers: null

- `ddp_bucket_cap_mb`: null
- `ddp_find_unused_parameters`: null
- `ddp_timeout`: 1,800
- `debug` (0 collapsed):
- `decoder_start_token_id`: null
- `deepspeed`: `"./zero3.json"`
- `disable_tqdm`: false
- `dispatch_batches`: null
- `diversity_penalty`: 0
- `do_eval`: false
- `do_predict`: false
- `do_sample`: false
- `do_train`: false
- `early_stopping`: false
- `encoder_no_repeat_ngram_size`: 0
- `eos_token_id`: 151,645
- `eval_accumulation_steps`: null
- `eval_delay`: 0
- `eval_do_concat_batches`: true
- `eval_on_start`: false
- `eval_steps`: null
- `eval_strategy`: `"no"`
- `eval_use_gather_object`: false
- `evaluation_strategy`: null
- `exponential_decay_length_penalty`: null
- `finetuning_task`: null
- `forced_bos_token_id`: null

- forced_eos_token_id: null
- fp16: false
- fp16_backend: "auto"
- fp16_full_eval: false
- fp16_opt_level: "O1"
- fsdp (0 collapsed):
- fsdp_config:
 - min_num_params: 0
 - xla: false
 - xla_fsdp_grad_ckpt: false
 - xla_fsdp_v2: false
- fsdp_min_num_params: 0
- fsdp_transformer_layer_cls_to_wrap: null
- full_determinism: false
- gradient_accumulation_steps: 2
- gradient_checkpointing: false
- gradient_checkpointing_kwargs: null
- greater_is_better: null
- group_by_length: false
- half_precision_backend: "auto"
- hidden_act: "silu"
- hidden_size: 3,584
- hub_always_push: false
- hub_model_id: null
- hub_private_repo: false
- hub_strategy: "every_save"
- hub_token: "<HUB_TOKEN>"

- id2label:
 - 0: "LABEL_0"
 - 1: "LABEL_1"
- ignore_data_skip: false
- image_token_id: 151,655
- include_for_metrics (0 collapsed):
- include_inputs_for_metrics: false
- include_num_input_tokens_seen: false
- include_tokens_per_second: false
- initializer_range: 0.02
- intermediate_size: 18,944
- is_decoder: false
- is_encoder_decoder: false
- jit_mode_eval: false
- label_names: null
- label_smoothing_factor: 0
- label2id:
 - LABEL_0: 0
 - LABEL_1: 1
- learning_rate: 1×10^{-6}
- length_column_name: "length"
- length_penalty: 1
- load_best_model_at_end: false
- local_rank: 0
- log_level: "passive"
- log_level_replica: "warning"
- log_on_each_node: true

- logging_dir: "/data/tayyibc/r1-v_out/runs/Mar26_08-53-48_moss"
- logging_first_step: false
- logging_nan_inf_filter: true
- logging_steps: 1
- logging_strategy: "steps"
- lr_scheduler_type: "linear"
- max_completion_length: 1,024
- max_grad_norm: 1
- max_length: 20
- max_position_embeddings: 32,768
- max_prompt_length: 1,024
- max_steps: 4,000
- max_window_layers: 28
- metric_for_best_model: null
- min_length: 0
- model_init_kwargs: null
- model_type: "qwen2_v1"
- model_num_parameters: 0
- mp_parameters: ""
- neftune_noise_alpha: null
- no_cuda: false
- no_repeat_ngram_size: 0
- num_attention_heads: 28
- num_beam_groups: 1
- num_beams: 1
- num_generations: 4
- num_hidden_layers: 28

B.2. Training Paramters and Meta Data

- num_key_value_heads: 4
- num_return_sequences: 1
- num_train_epochs: 1
- optim: "adamw_torch"
- optim_args: null
- optim_target_modules: null
- output_attentions: false
- output_dir: "/data/tayyibc/r1-v_out"
- output_hidden_states: false
- output_scores: false
- overwrite_output_dir: false
- pad_token_id: null
- past_index: -1
- per_device_eval_batch_size: 8
- per_device_train_batch_size: 1
- per_gpu_eval_batch_size: null
- per_gpu_train_batch_size: null
- prediction_loss_only: false
- prefix: null
- problem_type: null
- push_to_hub: false
- push_to_hub_model_id: null
- push_to_hub_organization: null
- push_to_hub_token: "<PUSH_TO_HUB_TOKEN>"
- ray_scope: "last"
- remove_invalid_values: false
- remove_unused_columns: false

- repetition_penalty: 1
- report_to (1 collapsed):
- restore_callback_states_from_checkpoint: false
- resume_from_checkpoint: null
- return_dict: true
- return_dict_in_generate: false
- rms_norm_eps: 1×10^{-6}
- rope_scaling:
 - mrope_section (3 collapsed):
 - rope_type: "default"
 - type: "default"
- rope_theta: 1,000,000
- run_name: "Qwen2-VL-7B-GRP0-MathVista-5k"
- save_on_each_node: false
- save_only_model: true
- save_safetensors: true
- save_steps: 100
- save_strategy: "steps"
- save_total_limit: null
- seed: 42
- sep_token_id: null
- skip_memory_metrics: true
- sliding_window: 32,768
- split_batches: null
- suppress_tokens: null
- task_specific_params: null
- temperature: 0.9

- `tf_legacy_loss`: false
- `tf32`: null
- `tie_encoder_decoder`: false
- `tie_word_embeddings`: false
- `tokenizer_class`: null
- `top_k`: 50
- `top_p`: 1
- `torch_compile`: false
- `torch_compile_backend`: null
- `torch_compile_mode`: null
- `torch_dtype`: "float32"
- `torch_empty_cache_steps`: null
- `torchdynamo`: null
- `torchscript`: false
- `tpu_metrics_debug`: false
- `tpu_num_cores`: null
- `transformers_version`: "4.49.0.dev0"
- `typical_p`: 1
- `use_bfloat16`: false
- `use_cache`: null
- `use_cpu`: false
- `use_ipex`: false
- `use_legacy_prediction_loop`: false
- `use_liger_kernel`: false
- `use_mps_device`: false
- `use_sliding_window`: false
- `use_vllm`: false

- video_token_id: 151,656
- vision_config:
 - _attn_implementation_autoset: false
 - _name_or_path: ""
 - add_cross_attention: false
 - architectures: null
 - bad_words_ids: null
 - begin_suppress_tokens: null
 - bos_token_id: null
 - chunk_size_feed_forward: 0
 - cross_attention_hidden_size: null
 - decoder_start_token_id: null
 - depth: 32
 - diversity_penalty: 0
 - do_sample: false
 - early_stopping: false
 - embed_dim: 1,280
 - encoder_no_repeat_ngram_size: 0
 - eos_token_id: null
 - exponential_decay_length_penalty: null
 - finetuning_task: null
 - forced_bos_token_id: null
 - forced_eos_token_id: null
 - hidden_act: "quick_gelu"
 - hidden_size: 3,584
 - id2label:
 - * 0: "LABEL_0"

B.2. Training Parameters and Meta Data

```
    * 1: "LABEL_1"
  - in_channels: 3
  - in_chans: 3
  - is_decoder: false
  - is_encoder_decoder: false
  - label2id:
    * LABEL_0: 0
    * LABEL_1: 1
  - length_penalty: 1
  - max_length: 20
  - min_length: 0
  - mlp_ratio: 4
  - model_type: "qwen2_v1"
  - no_repeat_ngram_size: 0
  - num_beam_groups: 1
  - num_beams: 1
  - num_heads: 16
  - num_return_sequences: 1
  - output_attentions: false
  - output_hidden_states: false
  - output_scores: false
  - pad_token_id: null
  - patch_size: 14
  - prefix: null
  - problem_type: null
  - remove_invalid_values: false
  - repetition_penalty: 1
```

B.2. Training Parameters and Meta Data

- return_dict: true
- return_dict_in_generate: false
- sep_token_id: null
- spatial_merge_size: 2
- spatial_patch_size: 14
- suppress_tokens: null
- task_specific_params: null
- temperature: 1
- temporal_patch_size: 2
- tf_legacy_loss: false
- tie_encoder_decoder: false
- tie_word_embeddings: true
- tokenizer_class: null
- top_k: 50
- top_p: 1
- torch_dtype: "float32"
- torchscript: false
- typical_p: 1
- use_bfloat16: false
- vision_end_token_id: 151,653
- vision_start_token_id: 151,652
- vision_token_id: 151,654
- vllm_device: "auto"
- vllm_gpu_memory_utilization: 0.9
- vocab_size: 152,064
- warmup_ratio: 0
- warmup_steps: 50

B.2. Training Paramters and Meta Data

- `weight_decay`: 0