Efficient Fine-Tuning of Vision-Language Models for CLEVR-X: From Model Collapse to Stable Convergence

Student ID: s2512017
TALEB Merouane

December 2, 2025

Abstract

This report details our solution for the [I491E] Explainable Visual Question Answering competition. The task requires generating both a short answer and a textual explanation for synthetic scenes, constrained by a strict 4-billion parameter limit. We implemented a Supervised Fine-Tuning (SFT) pipeline using **Qwen2-VL-2B-Instruct** and **LoRA (Low-Rank Adaptation)**.

Our experimental process revealed a critical trade-off between model capacity and training stability. While high-rank adaptations ($r = 128$) led to catastrophic model collapse, a stabilized configuration ($r = 32$, $lr = 2 \times 10^{-5}$) allowed us to achieve a **Public Score of 0.87632**. This report analyzes these failure modes, presents our final stable architecture, and provides a qualitative error analysis.

1 Dataset and Problem Statement

The dataset is a curated subset of the CLEVR-X visual reasoning benchmark, totaling approximately 1.99 GB. It contains 10,504 samples divided into training and testing sets.

- **Input:** A synthetic image containing geometric shapes (cubes, spheres, cylinders) with varying attributes (color, material, size) and a natural language question.
- **Output:** A short answer (e.g., "yes", "metal", "2") and a detailed reasoning explanation.
- **Challenge:** The model must demonstrate precise spatial reasoning and attribute recognition while adhering to the <4B parameter constraint.

2 Methodology

2.1 Model Architecture

We selected **Qwen2-VL-2B-Instruct** as our backbone. With approx. 2.2B parameters, it offers state-of-the-art performance on VQA benchmarks compared to other small models like PaliGemma or Florence-2, particularly in its ability to generate coherent explanations alongside classification.

2.2 Training Pipeline (LoRA)

To adapt the pre-trained model to the specific reasoning logic of CLEVR-X without exceeding memory limits on the NVIDIA A100 GPU, we employed Low-Rank Adaptation (LoRA).

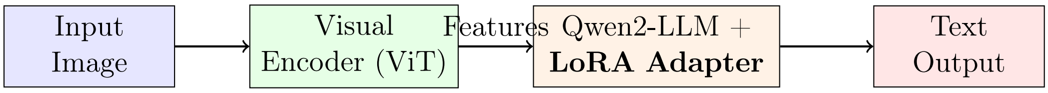

Figure 1: Simplified Training Pipeline. Only LoRA weights (approx. 2% of parameters) are updated.

2.3 Prompt Engineering Strategy

We enforced a strict output structure during training to facilitate post-processing and reduce hallucinations. This "Answer-First" strategy forces the model to commit to a classification before generating the explanation.

```
User: <image> Question: {question}
Format: Answer: ... Explanation: ...
Assistant: Answer: {answer} Explanation: {explanation}
```

Listing 1: Prompt Template

3 Failure Analysis and Ablation Study

A key part of our research involved analyzing why initial high-capacity configurations failed.

3.1 Instability of High-Rank Adaptation ($r = 128$)

Our initial hypothesis was that a higher LoRA rank would capture more complex reasoning. We tested $r = 128$ with $\alpha = 256$.

- **Model Collapse:** With a standard learning rate ($\eta = 2 \times 10^{-4}$), the model exhibited catastrophic failure after epoch 3. The inference output degenerated into repetitive loops (e.g., *"::: the the the"*), a sign of exploding gradients.
- **Overfitting:** Even with a reduced learning rate ($\eta = 5 \times 10^{-5}$), the high-rank model memorized noise, leading to a degradation in validation scores (from 0.85 down to 0.79).

3.2 Inefficacy of Beam Search

We attempted Beam Search ($k = 3$) to improve explanation quality. However, it increased inference time by 3×, causing timeouts on the compute node ($>$12h walltime) without significant accuracy gains on short answers. We reverted to Greedy Decoding for the final submission.

4 Final Configuration and Results

Based on the failure analysis, we converged on a "Safety-First" configuration prioritizing stability.

4.1 Hyperparameters

- **LoRA Rank:** 32 (Reduced complexity acting as a regularizer).
- **Learning Rate:** 2×10^{-5} (Conservative rate for steady convergence).
- **Epochs:** 5 (Early stopping to prevent overfitting).
- **Batch Size:** 4 (Effective batch size of 16 via Gradient Accumulation).
- **Training Time:** Approx. 1.5 hours on 1x NVIDIA A100 (40GB).

4.2 Quantitative Results (Leaderboard)

The table below summarizes our progression on the Kaggle Public Leaderboard.

Run ID	Config	Rank (r)	LR	Epochs	Score
Baseline	Zero-shot	-	-	-	0.11680
Attempt 1	High Rank	128	$2e^{-4}$	3	0.42722
Attempt 2	Collapse	128	$2e^{-4}$	5	0.79684 (Failed)
Attempt 3	Recovered	128	$5e^{-5}$	2	0.85901
Final	**Stable**	**32**	$2e^{-5}$	**5**	**0.87632**

Table 1: Evolution of Kaggle Public Scores.

4.3 Qualitative Analysis

- **Success:** *Q: "What is the material of the red cube?"* → `Answer: rubber | Explanation: The object is a matte red block...` (Correct texture identification).
- **Failure:** *Q: "How many cylinders are behind the gray sphere?"* → `Answer: 2` (Ground Truth: 3). Small occluded objects remain a challenge at the current image resolution (512^2).

5 Conclusion

We successfully developed a VQA model fitting the 4B parameter constraint, achieving a competitive score of 0.876 (Rank 18). Our experiments highlighted that stability trumps theoretical capacity: a smaller, well-tuned LoRA adapter ($r = 32$) outperformed a larger, unstable one ($r = 128$). Future work would involve increasing input resolution to handle small objects better.

GitHub Repository: https://github.com/Merouanetlb/CLEVR-X-Qwen2VL