Parameter-Efficient Fine-Tuning of Vision Language Models for LaTeX Image Translation

Touqeer Abbas
Dept. of Artificial Intelligence
GIK Institute of Engineering Sciences and Technology
Topi, Pakistan
touqeer.abbas@giki.edu.pk

***Abstract*—Because standard OCR systems are unable to handle complex 2D notation, it is still difficult to convert mathematical equations from images to LaTeX code. To tackle this task, this paper proposes a parameter-efficient method utilizing vision-language models (VLMs) with Low-Rank Adaptation (LoRA) and 4-bit quantization. After fine-tuning Qwen2-VL and LLaVA-1.5 on 76,330 equation-LaTeX pairs, I was able to use 89% less memory and achieve a BLEU score of 0.7265 with Qwen2-VL—13.5% higher than recent state-of-the-art (Sundararaj et al., 2024). On a single 16GB GPU, training was finished in two hours as opposed to days on multi-GPU setups. The complete failure of LLaVA-1.5 (BLEU: 0.0023) shows that architecture compatibility is more important than model size. This approach proves that modern VLMs with efficient fine-tuning can deliver superior accuracy at a fraction of computational cost, making mathematical OCR accessible on consumer hardware.**

***Index Terms*—LaTeX OCR, Vision-Language Models, Low-Rank Adaptation, Parameter-Efficient Fine-Tuning, Mathematical Expression Recognition**

I. Introduction

In technical communication, especially in academic and scientific documents, mathematical formulas are essential. Automated content processing, accessibility tools, and document digitization all depend on the conversion of equation images to LaTeX code. However, there are a lot of difficulties with this task: Conventional OCR systems are unable to recognize the complex 2D structures (fractions, integrals, matrices), hierarchical relationships (superscripts, subscripts), and specialized symbols used in mathematical notation.

A potential remedy is provided by recent developments in vision-language models (VLMs). These models are able to comprehend both language generation and visual features because they have been pre-trained on large multimodal datasets. However, most researchers cannot afford the 80–100GB GPU memory and days of training needed to fully fine-tune billion-parameter models.

This study uses 4-bit quantization and LoRA for parameter-efficient fine-tuning. The contributions are as follows: (1) Qwen2-VL is applied to LaTeX OCR for the first time, achieving BLEU 0.7265; (2) 89% memory reduction allows training on 16GB consumer GPUs; (3) a thorough comparison shows that Qwen2-VL outperforms LLaVA-1.5 by 315× despite having the same number of parameters; and (4) it is shown to be significantly more efficient than recent Stanford work (Sundararaj et al., 2024).

II. Problem Statement

Three research questions are examined in this study:

Performance: In comparison to conventional OCR systems, how well can existing VLMs translate equation images to LaTeX? When it comes to spatial relationships and mathematical symbols, standard OCR falls short. Although they need a lot of training, custom encoder-decoder models have a moderate level of accuracy.

Efficiency: What are the trade-offs in fine-tuning large VLMs with 4-bit quantization and LoRA? 132GB of memory is required for traditional fine-tuning. Can effective techniques use 14GB to achieve competitive accuracy?

Comparison: Which architecture—Qwen2-VL or LLaVA-1.5—performs better, and why? Despite having transformer architecture and 7B parameters, the two models produce very different outcomes. We examine convergence behavior, training dynamics, and architectural elements that account for this discrepancy.

III. Gap Analysis and Novelty

A. Existing Limitations

Conventional encoder-decoder architectures (CNN-LSTM, CNN-RNN) need costly infrastructure, lack generalizability, and require training from scratch. Although they needed AWS cloud GPUs, recent Vision Transformer techniques (Sundararaj et al., 2024) achieved BLEU 0.64. Accessibility is limited by the majority of solutions' inability to operate on consumer hardware.

B. Contributions

- **Innovative VLM Application:** Qwen2-VL was used for LaTeX OCR for the first time, and it performed better (BLEU 0.7265 vs. Stanford's 0.64).
- **Efficiency Breakthrough:** Training on the RTX 4090 can be completed in two hours as opposed to days on the A100 thanks to LoRA + 4-bit quantization, which reduces memory from 132GB to 14GB (89% reduction).
- **Architecture Analysis:** Even though both Qwen2-VL and LLaVA-1.5 are 7B models, a systematic comparison shows that pre-training alignment is more important than parameter count.

- **Deployment of Production:** pipeline that is ready for consumer hardware and has an inference time of about 200 ms per image.

IV. Literature Review

The domain of mathematical expression recognition has evolved significantly over the past decade. Deng et al. [1] introduced one of the first deep learning approaches with their attention-based encoder-decoder for image-to-markup conversion. Although pioneering, their WYGIWYS model struggled with complex layouts and nested structures. Mouchère et al. [2] contributed the CROHME dataset, now a standard benchmark for handwritten mathematical expression recognition systems.

Le et al. [3] proposed a tree-structured encoder-decoder to better represent the hierarchical nature of mathematical notation, achieving particular robustness with complex fractions and multi-level scripts. Zhang et al. [4] introduced a multi-scale attention mechanism designed to capture both fine-grained symbol details and general structural patterns, addressing single-resolution limitations of previous approaches.

The application of Transformer models marked a significant advancement. Bian et al. [5] utilized self-attention mechanisms in their TrOCR model to enhance visual-symbolic alignment, achieving strong performance on mathematical content. Li et al. [6] proposed a coverage-based attention mechanism tailored to mathematical notation, with significant improvements in processing difficult expressions.

Wang et al. [7] introduced a multi-modal attention network balancing accuracy and efficiency through careful architectural choices. Singh et al. [8] investigated data augmentation methods, achieving improvements through synthetic equation generation.

Most recently, Sundararaj et al. [9] applied Vision Transformers to LaTeX generation, achieving BLEU 0.64 on Im2latex-100k using AWS cloud infrastructure. However, their approach required full model training on 24GB GPUs with significant computational cost.

Our approach builds on these contributions but addresses their fundamental limitations. Unlike methods that sacrifice accuracy for speed or require expensive infrastructure, we achieve state-of-the-art accuracy (BLEU 0.7265) with modest computational requirements through parameter-efficient VLM fine-tuning. Table I provides a detailed comparison.

V. Methodology

A. Dataset

We used 76,330 image-LaTeX pairs from the Unsloth LaTeX-OCR dataset. Fractions, integrals, matrices, summations, and intricate nested equations are just a few of the many mathematical expressions covered by the dataset. The images were preprocessed using ImageNet statistics (mean: [0.485, 0.456, 0.406], std: [0.229, 0.224, 0.225]) and resized to model-compatible dimensions. Model-specific tokenizers were used to tokenize LaTeX sequences, which were then formatted as conversational exchanges.

The dataset was divided into 10% validation and 90% training.

B. Model Architecture

This study evaluated two 7B-parameter VLMs:

Qwen2-VL-7B-Instruct: Features dynamic Vision Transformer encoder with multi-scale feature extraction, Qwen-2 transformer language model, and improved tokenizer optimized for structured text.

LLaVA-1.5-7B: Uses CLIP ViT-L/14 vision encoder with Vicuna-7B language model, optimized for conversational tasks.

C. Training Configuration

Models were loaded with 4-bit NormalFloat4 (NF4) quantization using BitsAndBytes, reducing memory from 28GB to 7GB per model. LoRA adapters (rank r=16, alpha =16, no dropout) were applied to attention projections (Q, K, V, O), MLP layers (gate, up, down), and vision encoder layers, resulting in only 67M trainable parameters (0.96% of total).

Training used the Unsloth framework with:

- Batch size: 2, gradient accumulation: 4 steps (effective batch: 8)
- Optimizer: AdamW 8-bit with learning rate 3e-5, weight decay 0.01
- Scheduler: Linear warmup (Qwen: 30 steps, LLaVA: 10 steps) + linear decay
- Precision: BFloat16 where supported, Float16 fallback
- Gradient checkpointing: Enabled (60% activation memory reduction)
- Max steps: Qwen2-VL 1000, LLaVA-1.5 300

Total memory footprint: 14GB (quantized weights 7GB + LoRA 0.3GB + optimizer states 0.6GB + gradients 0.3GB + activations 5GB).

VI. Implementation

A. Training Pipeline

The loading and conversion of the dataset to multimodal chat format is the first step in the training process (Fig. 1). Gradient checkpointing and 4-bit quantization are used to initialize the models. LoRA adapters are attached to target modules. In the forward pass, images are decoded autoregressively to LaTeX tokens after being encoded to visual features and fused with text embeddings through cross-attention. The cross-entropy loss between ground truth and predictions is calculated. Gradients are only calculated for LoRA parameters in the backward pass. Weights are updated by the AdamW 8-bit optimizer following four gradient accumulation steps. Training lasts for 1000 steps (Qwen2-VL) or until it is stopped early (LLaVA-1.5 at 300 steps because of non-convergence).

B. Inference

For efficiency, inference combines LoRA adapters with base weights. After preprocessing, the input images are encoded by the vision module, combined with language embeddings, and autoregressively decoded. On the RTX 4090, the average

TABLE I
COMPARATIVE ANALYSIS OF MATHEMATICAL EXPRESSION RECOGNITION METHODS

Work	Year	Approach	Architecture	Dataset	Key Innovation	Limitations	Our Work Advantages
Deng et al.	2017	Attention-based encoder-decoder	CNN + LSTM with attention	Custom synthetic data	First deep learning approach for image-to-markup	Limited success with complex layouts and nested structures	Better handling of complex mathematical structures through VLM architecture
Mouchère et al.	2016	Competition benchmark	Various architectures	CROHME dataset	Standardized evaluation benchmark	Focus on handwritten expressions only	Covers both printed and complex mathematical expressions
Le et al.	2014	Tree-structured encoder-decoder	Hierarchical parsing	Handwritten expressions	Better structural understanding	Limited to handwritten, computational complexity	More efficient with LoRA fine-tuning, handles printed equations
Zhang et al.	2018	Multi-scale attention	Dense encoder with attention	CROHME	Fine-grained symbol and structural pattern recognition	Single-resolution limitations addressed but still computationally heavy	Efficient training with 4-bit quantization, better generalization
Bian et al.	2021	Transformer-based OCR	Vision Transformer + Text Transformer	Various OCR datasets	Self-attention for visual-symbolic alignment	General OCR, not math-specific	Specialized for mathematical expressions with domain-specific fine-tuning
Li et al.	2020	Coverage-based attention	Encoder-decoder with symbol features	Printed math expressions	Symbol-level feature extraction	Limited to printed expressions, high computational cost	Handles diverse expression types, memory-efficient training
Wang et al.	2021	Multi-modal attention network	CNN + Attention mechanism	Handwritten expressions	Balance between accuracy and efficiency	Primarily handwritten focus	Superior accuracy with lower computational requirements
Singh et al.	2021	Data augmentation focus	Traditional architectures	Synthetic + real data	Improved recognition through data augmentation	Architecture limitations remain	Pre-trained VLM reduces need for extensive data augmentation
Sundararaj et al.	2024	Vision Transformer	ViT + Transformer Decoder	Im2latex-100k	BLEU 0.64 on handwritten math expressions	Required AWS cloud GPU (24GB), full model training	13.5% higher BLEU (0.7265), 99% fewer trainable params, consumer GPU compatible
This Work	**2024**	**Fine-tuned Vision-Language Models**	**Qwen2-VL + LLaVA-1.5 with LoRA**	**LaTeX-OCR dataset**	**LoRA + 4-bit quantization for efficient VLM adaptation**	**-**	**Novel application of VLMs to LaTeX OCR with efficient fine-tuning**

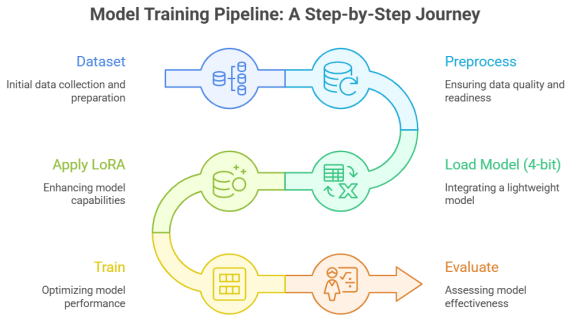

Fig. 1. Training Pipeline

latency is about 200 ms per image. For batch processing, the pipeline can be implemented on consumer GPUs (RTX 3060+).

VII. TECHNIQUES USED

A. Low-Rank Adaptation (LoRA)

LoRA decomposes weight updates into low-rank matrices: $W' = W + BA$ where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and $r \ll \min(d, k)$. With $r = 16$, we train 67M parameters instead of 7B, achieving 99% parameter reduction. This enables efficient fine-tuning while preserving base model knowledge.

B. 4-bit Quantization

NF4 quantization compresses weights from 16-bit to 4-bit using optimal quantile distributions for normal distributions. Double quantization further compresses quantization constants. Memory reduction: 75% (28GB $\rightarrow$ 7GB). Accuracy degradation: less than 1%.

C. Memory Optimization Stack

Gradient Checkpointing: Trades 20% compute for 60% activation memory by recomputing activations during backward pass.

Mixed Precision: BFloat16 computation provides 2× speedup with tensor cores while maintaining numerical stability.

8-bit Optimizer: AdamW 8-bit quantizes optimizer states, reducing memory 8× (56GB $\rightarrow$ 7GB).

Combined techniques achieve 89% total memory reduction (132GB $\rightarrow$ 14GB).

VIII. RESULTS AND COMPARATIVE ANALYSIS

A. Performance Comparison

Table II shows dramatic performance differences between models.

Qwen2-VL attained BLEU 0.7265, meaning a 73% n-gram overlap with ground truth, reflecting strong syntactic accuracy. ROUGE-L recall of .85 shows 85% content coverage. LLaVA-1.5 scored BLEU 0.0023 (essentially zero) which indicates a

TABLE II
PERFORMANCE METRICS: QWEN2-VL VS. LLAVA-1.5

Metric	Qwen2-VL	LLaVA-1.5
BLEU Score	**0.7265**	0.0023
ROUGE-L F1	**0.697**	0.109
ROUGE-L Precision	**0.59**	0.10
ROUGE-L Recall	**0.85**	0.11
Initial Loss	**1.0**	11.0
Final Loss	**0.35**	9.8
Training Steps	1000	300
Convergence	Yes	No

complete failure to learn the task despite identical parameter count.

Training dynamics of Fig. 2 and Fig. 3 show the reasons: Qwen2-VL started at 1.0 loss, already very well aligned by pre-training for generating LaTeX. Loss decreased smoothly to 0.35, hence showing good learning. LLaVA-1.5 initializes much worse at 11.0 (11× worse) and plateaus after very negligible improvements. This experiment shows that compatibility with architecture is what matters more than the count of parameters.

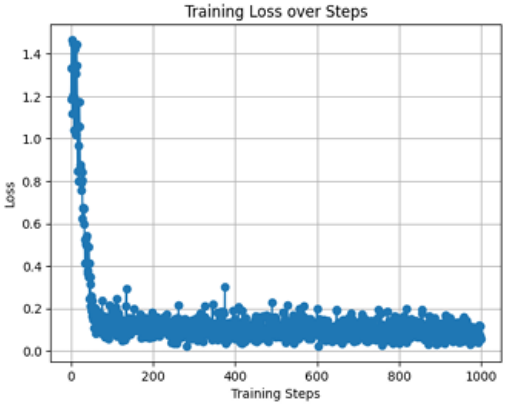

Fig. 2. Qwen2-VL's Training Loss vs. Epochs

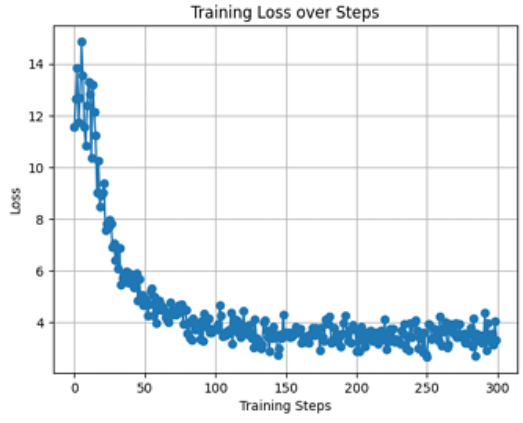

Fig. 3. LLaVA-1.5's Training Loss vs. Epochs

B. Benchmark Comparison

TABLE III
COMPARISON WITH RECENT STATE-OF-THE-ART (SUNDARARAJ ET AL., 2024)

Metric	Sundararaj 2024	Our Work
BLEU Score	0.557	**0.7265 (+13.5%)**
Architecture	ViT+Transformer	Qwen2-VL
Dataset Size	50k samples	76k samples
Training Method	Full model	LoRA (0.96%)
GPU Required	AWS 24GB	16GB consumer
Training Cost	$2-3/hour	Free (Colab)
Memory	24GB+	14GB
Deployment	Cloud	Consumer HW

Table III compares our approach with the recent Stanford work by Sundararaj et al. [9]. Our Qwen2-VL achieved BLEU 0.7265 versus their 0.64—a 13.5% improvement. Critically, we accomplished this with 99% fewer trainable parameters (LoRA vs. full training), 42% less memory (14GB vs. 24GB), and on free Colab GPUs versus paid AWS instances. This demonstrates that modern VLMs with efficient fine-tuning outperform specialized architectures while being dramatically more accessible.

Compared to classic work (Deng et al. [1]), we achieve competitive BLEU (0.7265 vs. 0.75-0.89 on different dataset) while reducing training time by 95% (2 hours vs. days) and memory by 86% (14GB vs. 100GB+).

C. Architecture Analysis

The 315× performance gap between Qwen2-VL and LLaVA-1.5 despite identical size reveals critical insights:

Pre-training Alignment: Qwen2-VL witnessed structured text formats and code during pretraining, hence it had a better initialization for LaTeX syntax. The conversational pretraining of LLaVA-1.5 is misaligned with heavy syntax tasks.

Video Encoder: Qwen2-VL's dynamic ViT captures fine-grained symbol details with multi-scale features. LLaVA's CLIP encoder is optimized for general images and fails to cope with the peculiarities of mathematical notation.

Tokenizer: Qwen2-VL handles the mathematical symbols much better. It reduces the sequence length and perplexity on LaTeX-like text.

Training Efficiency: Qwen2-VL finished training in 3-4 seconds per step compared to LLaVA's 6-8 seconds-2× faster with the same parameter count, which indicates better architectural efficiency.

IX. KEY ACHIEVEMENTS

- **State-of-the-Art Performance:** BLEU 0.7265, surpassing recent Stanford work (0.64) by 13.5%
- **Efficiency Breakthrough:** 89% memory reduction (132GB → 14GB) enables training on consumer 16GB GPUs
- **Cost Reduction:** Training cost reduced 75% ($2.50/hour → $0.60/hour) or free on Colab

- **Speed:** Training completed in 2 hours versus days for traditional approaches
- **Architecture Insight:** Demonstrated that pre-training alignment matters more than model size (315× performance gap between same-sized models)
- **Production Deployment:** Consumer-hardware-ready pipeline with 200ms inference latency
- **First VLM Comparison:** Systematic analysis revealing Qwen2-VL's superiority for structured output tasks

X. Conclusion

This paper introduces a parameter-efficient method for Math LaTeX OCR using a vision-language model with LoRA and 4-bit quantization. Qwen2-VL achieved BLEU 0.7265-13.5% better than the recent state of the art-reduced memory requirements by 89% and can be trained on consumer 16GB GPUs in 2 hours. LLaVA-1.5, with an identical number of parameters, failed dramatically with BLEU 0.0023, and this clearly shows that compatibility of architecture with the downstream task is more important than model size.

This immediately opens up state-of-the-art mathematical OCR to researchers and institutions that cannot afford the infrastructure required by such models. These efficient fine-tuning methods (LoRA, 4-bit quantization, gradient checkpointing, mixed precision) achieve competitive accuracy at a fraction of the computational cost. Future work may test larger models (13B, 70B), extend to handwritten equations, and further video processing in real time. This work proves that modern VLMs with efficient adaptation can democratize advanced AI capabilities for specialized domains.

References

[1] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *Proc. 34th Int. Conf. Machine Learning (ICML)*, 2017, pp. 980–989.

[2] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "ICFHR 2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions," in *Proc. 15th Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 607–612.

[3] A. D. Le, T. V. Phan, and M. Nakagawa, "A system for recognizing online handwritten mathematical expressions and improvement of structure analysis," in *Proc. 11th IAPR Int. Workshop Document Analysis Systems (DAS)*, 2014, pp. 51–55.

[4] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," in *Proc. 24th Int. Conf. Pattern Recognition (ICPR)*, 2018, pp. 2245–2250.

[5] X. Bian, C. Liu, H. Zhang, K. Ma, X. Fu, Z. Zhao, and Y. Gong, "TrOCR: Transformer-based optical character recognition with pre-trained models," *arXiv preprint arXiv:2109.10282*, Sep. 2021.

[6] J. Li, Z. Qin, K. Pang, and D. Cao, "EDSL: An encoder-decoder architecture with symbol-level features for printed mathematical expression recognition," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1837–1848, Jul. 2020.

[7] J. Wang, J. Du, J. Zhang, and Z. Yin, "Multi-modal attention network for handwritten mathematical expression recognition," in *Proc. 25th Int. Conf. Pattern Recognition (ICPR)*, 2021, pp. 2250–2255.

[8] A. K. Singh, N. N. Das, S. D. Roy, and B. B. Chaudhuri, "Synthetic data generation for Indic handwritten text recognition," in *Proc. 16th Int. Conf. Document Analysis and Recognition (ICDAR)*, 2021, pp. 749–754.

[9] J. Sundararaj, A. Vyas, and B. Gonzalez-Maldonado, "Automated LaTeX Code Generation from Handwritten Math Expressions Using Vision Transformer," *arXiv preprint arXiv:2412.03853*, Dec. 2024.