



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

《设计和理解深度神经网络》课程报告

基于 Qwen2.5-VL 的 LaTeX OCR 模型

学院 _____ 电子信息与电气工程学院

学号 _____ 521021910107, 521021910438, 124020990005

姓名 _____ 胡晨志, 王瑶瑶, MING JIE LIM

2025 年 8 月 16 日

目录

1 项目背景	3
1.1 研究动机与应用背景	3
1.2 挑战分析	4
1.3 模型选择	4
1.3.1 模型架构	5
1.3.2 模型优势	6
2 相关工作	6
2.1 模型微调	6
2.2 视觉语言模型	6
3 技术路线	7
3.1 数据集	7
3.2 模型微调	8
4 效果分析	9
4.1 评估指标	9
4.2 实验结果	10
4.2.1 定性结果分析	10
4.2.2 识别性能	11
4.2.3 推理效率与显存占用	11
5 端侧部署	12
5.1 工具介绍	12
5.2 部署细节	13
6 总结展望	13

1 项目背景

数学公式的数字化转换是科研、教育领域的重要需求，但现有光学公式识别（OCR）系统对复杂公式结构（如多级分数、矩阵、特殊符号）的识别精度有限，尤其在处理手写公式时性能显著下降。

本项目旨在通过微调视觉语言模型（VL Model），开发一个鲁棒的公式图像转 LaTeX 源码系统，解决现有方案对打印/手写公式的泛化性问题。

1.1 研究动机与应用背景

在科学文献和教育技术等场景中，数学公式作为表达变量关系的重要方式，其自动识别任务具有重要的实际应用价值。然而，相比于通用的文本识别任务，数学公式识别面临更高的挑战。这是因为公式不仅包含多种复杂的符号，还具有显著的二维空间结构，例如上下标、分数、根号和嵌套表达式等。因此，传统的光学字符识别（OCR）方法在面对数学公式时往往效果不佳。

传统的公式识别方法通常采用分阶段的处理流程，主要包括字符检测、符号识别和结构解析三个步骤。例如，INFTY Reader 系统通过水平与垂直投影进行字符定位，然后识别符号，最后构建符号之间的结构关系。然而，这种基于规则和启发式结构的方案在面对复杂结构如矩阵和嵌套表达式时容易出现误识，且流程之间的误差易于传递，严重影响最终识别效果。

近年来，随着深度学习技术的发展，研究者开始探索使用神经网络来替代传统的 OCR 流程。典型的深度学习方法采用编码-解码结构，例如 CNN 提取图像特征，RNN 解码为 LaTeX 字符序列，并通过注意力机制加强上下文建模能力。Im2LaTeX 等模型在公式识别任务中已取得明显提升，证明了数据驱动的端到端方法在该领域的有效性。

在此基础上，Transformer^[1] 架构因其卓越的全局建模能力和可扩展性，逐渐成为公式识别中的主流方法。视觉 Transformer（如 ViT^[2]、Swin Transformer^[3]）可对图像进行全局编码，而语言解码器（如 BART）则可将图像特征解码为 LaTeX 表达式。近年来，像 Pix2Tex¹、Donut^[4] 以及 Qwen2.5-VL^[5] 等模型纷纷被提出，这些模型不仅实现了从图像到 LaTeX 的直接转换，还具备了在无 OCR 条件下执行视觉文档理解的能力。

与 OCR 方法相比，基于 Transformer 的端到端方法具备诸多优势。首先，它规避了基于规则的 OCR 模型对复杂公式识别鲁棒性不足以及深度学习 OCR 计算开销大等问题。其次，注意力机制天然适合建模公式中符号之间复杂的空间关系和语法结构。第三，模型整体可以统一优化，训练更稳定，效果更具鲁棒性。此外，端到端方案还可避免 OCR 误识别对后续结构解析的影响，特别适用于手写体、多语种以及复杂排版的公式识别任务。Donut 等模型已在文档信息提取与视觉文档理解任务中验证了这一优势，其性能在准确率、速度和内存开销方面均优于传统 OCR 依赖方案^[6]。

¹<https://github.com/lukas-blecher/LaTeX-OCR>

1.2 挑战分析

尽管近年来端到端的视觉语言模型在公式识别任务中取得显著进展，但将其应用于实际场景仍面临很多挑战。

虽然通用视觉语言模型能识别简单的数学公式，但在面对一些复杂公式或者手写字体等多样化的实际情况时表现不佳。

- **复杂结构表达的建模困难：**数学公式中广泛存在上下标、嵌套分数、根式、多层次括号、矩阵等复杂的二维结构。这类结构在图像中具有显著的空间排布特性，与传统文本的线性序列结构不同，要求模型能够精准捕捉视觉布局并转换为对应的线性 *LaTeX* 表达形式，对模型的空间理解能力提出了很高要求。
- **手写公式的多样性与模糊性：**与印刷体公式相比，手写公式存在显著的个体差异、连笔、形变、重叠、笔画不完整等问题，极大地增加了视觉编码的难度。当前主流 VLM 模型多在自然图文或印刷体文档上训练，缺乏对手写风格的泛化能力，这对提升系统的实用性构成了重大挑战。
- ***LaTeX* 表达存在结构歧义：**相同的数学表达可以对应多种合法的 *LaTeX* 表达形式，例如 $\frac{a+b}{c}$ 可以写作 `frac{a+b}{c}` 或 `dfrac{a+b}{c}` 等；此外，括号使用、空格控制等也可能存在风格差异。这使得模型输出即使语义正确，也可能与参考答案存在“文本不匹配”，从而影响评估指标结果。因此，如何构造更鲁棒的评估方法，或引入语义等价性判断，是一个亟需解决的问题。

除了任务本身的复杂性，当前在数据资源、模型规模与硬件部署方面也存在实际限制，制约了大规模视觉语言模型在公式识别任务中的进一步落地应用。

- **训练数据稀缺：**公开的 *LaTeX* 图像对齐数据集数量有限，且覆盖的公式类型、长度和复杂性存在偏差。在大模型微调过程中，数据稀缺问题可能导致过拟合或结构泛化能力不足。此外，手写公式数据集（如 CROHME）体量更小，直接用于大型模型训练会面临训练不稳定和收敛困难等问题。
- **推理效率与部署成本：***LaTeX* 公式的识别通常应用于编辑软件或笔记软件。大规模 VLM 模型（如 Qwen2.5-VL）虽然性能强大，但在推理过程中占用显存较大，处理速度相对较慢，不利于资源受限环境下的部署，如移动端、教学平板或服务器并发任务环境。因此，如何在保证精度的前提下对模型进行压缩与加速，是实现端侧部署的重要挑战。

1.3 模型选择

本项目选择了由通义千问团队提出的多模态大模型 Qwen2.5-VL^[5] 系列作为视觉语言建模的基础。模型结构如图 1 所示。相比传统的两阶段公式识别模型（先使用视觉

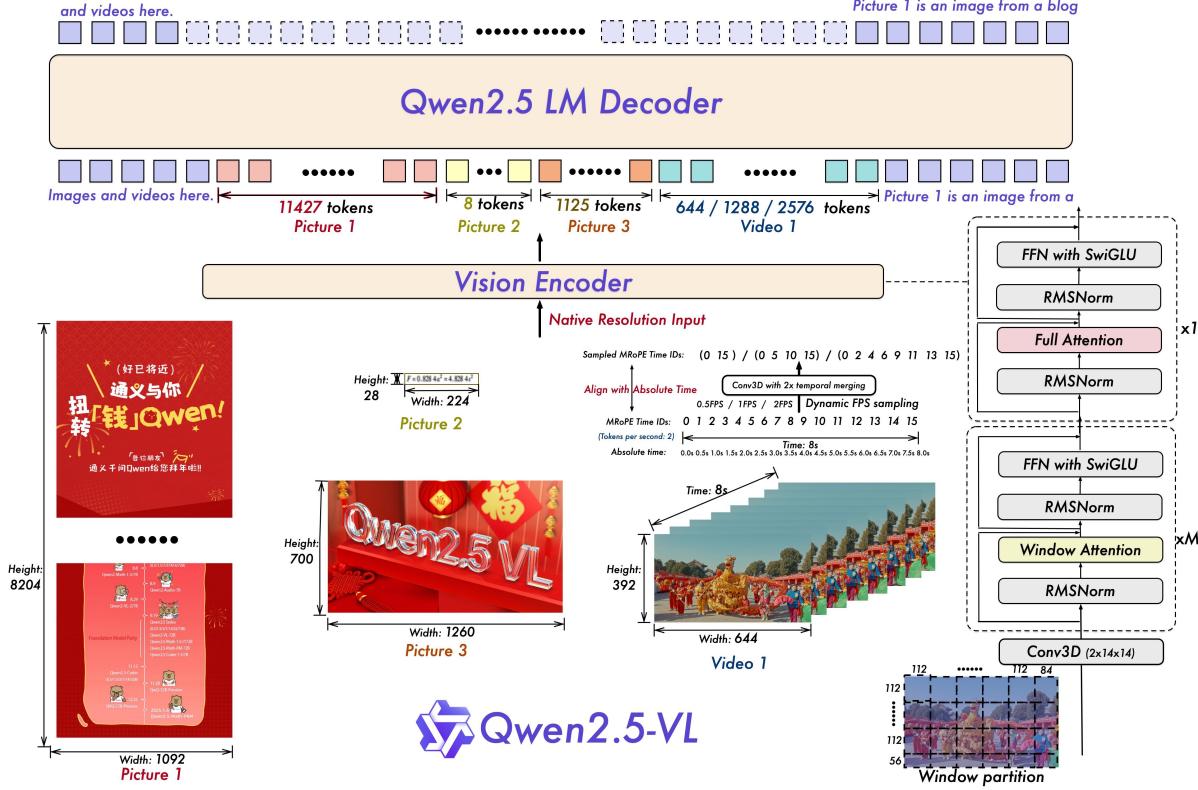


图 1: Qwen2.5 VL 模型结构。图片来源于<https://github.com/QwenLM/Qwen2.5-VL>

模块进行公式结构识别，再由语言模块生成 LaTeX 代码），Qwen2.5-VL 采用端到端的 Transformer 架构，将图像编码与文本解码统一建模，能够更好地捕捉图像与语言之间的对应关系，特别适用于公式图像转 LaTeX 的生成任务。

1.3.1 模型架构

Qwen2.5-VL 的整体模型架构由三个部分组成。

大型语言模型 Qwen2.5-VL 系列采用了大型语言模型作为其基础组件。该模型初始化时使用了 Qwen2.5 LLM 的预训练权重。为了更好地满足多模态理解的需求，其将 1D RoPE（旋转位置嵌入）修改为对齐绝对时间的多模态旋转位置嵌入。

视觉编码器 Qwen2.5-VL 的视觉编码器采用了重新设计的 Vision Transformer (ViT) 架构。结构上，其结合了 2D-RoPE 和窗口注意力，以支持原生输入分辨率并加速整个视觉编码器的计算。在训练和推理过程中，输入图像的高度和宽度会被调整为 28 的倍数，然后再送入 ViT。视觉编码器通过将图像分割成步幅为 14 的补丁来处理图像，生成一组图像特征。

基于 MLP 的视觉-语言融合 为了解决长序列图像特征带来的效率挑战，Qwen2.5-VL 采用了一种简单而有效的方法，在将特征序列送入大型语言模型 (LLM) 之前对其进行

压缩。具体来说，其不是直接使用 Vision Transformer (ViT) 提取的原始补丁特征，而是首先将空间相邻的四个补丁特征分组。然后将这些分组的特征连接起来并通过一个两层的多层感知机 (MLP) 传递，将其投影到与 LLM 中使用的文本嵌入对齐的维度。这种方法不仅减少了计算成本，还提供了一种灵活的方式来动态压缩不同长度的图像特征序列。

1.3.2 模型优势

快速高效的视觉编码器 重新设计了 Vision Transformer (ViT) 架构，在大多数层中引入了窗口注意力，确保计算成本随补丁数量线性增长而不是二次增长。此外，其采用 2D 旋转位置嵌入编码 (RoPE) 以有效地捕捉二维空间中的空间关系。

预训练 OCR 数据丰富 使用大规模预训练数据。其中 OCR 数据通过整理和收集不同来源的数据得到，包括合成数据、开源数据和内部收集的数据。数据集经过精心策划，以确保多样性和质量，利用高质量的合成图像和现实世界的自然场景图像。这种组合确保了在各种语言环境下的稳健性能，并提高了模型对不同文本外观和环境条件的适应性。

2 相关工作

2.1 模型微调

随着预训练技术的广泛应用，对于数据量有限的下游任务，可以利用经过预训练的基底模型的迁移能力，使用少量的下游任务数据集对模型进行微调，从而获得很好的效果。在 LLM 领域，主要有三种微调方法：(1) 全量微调，指在微调过程中更新所有的模型权重。该方法虽然能取得很好的效果，但需要极大的算力资源。(2) 在原模型的基础上增加适配器 (Adapter) 层，只训练适配器层的权重。该方法虽然减少了训练算力需求，但由于模型变深，会增加推理时延。(3) LoRA^[7]，该方法利用模型权重矩阵低秩的性质，将权重矩阵分解为小矩阵的乘积。该方法可以在保证高性能的同时极大的减少训练开销。LoRA 还有许多改进版本，如 QLoRA^[8] 在 LoRA 的基础上进行模型量化，进一步降低训练开销。由于本项目所用模型较小，为了获得较高性能，选择使用全量微调。

2.2 视觉语言模型

近年来，随着深度学习技术的发展，视觉与语言跨模态理解与生成能力成为人工智能领域的重要研究方向。视觉语言模型 (Vision-Language Models, VLMs) 作为连接图像与文本语义桥梁的关键技术，取得了显著进展。早期的视觉语言模型如 CLIP 通过对比学习的方式，在大规模图文对数据上进行训练，实现了高效的零样本迁移能力。这些模型将图像和文本分别编码为向量表示，并在共享语义空间中最大化正样本对的相似性，从而实现跨模态检索、分类等任务。随后，基于 Transformer 架构的多模态融合模

型逐渐兴起。例如，OFA 和 Flamingo 引入了更复杂的跨模态注意力机制，支持统一处理多种视觉-语言任务，包括图像描述生成、视觉问答（VQA）以及图文推理等。这类模型通常采用预训练-微调范式，在多个下游任务上展现出强大的泛化能力。近期，随着大语言模型（LLMs）的发展，诸如 LLaVA [9] 和 Qwen-VL[5] 等模型开始尝试将先进的语言生成能力引入视觉理解系统，让将视觉特征嵌入文本上下文，从而让大语言模型拥有视觉能力。

3 技术路线

3.1 数据集

Im2LaTeX-100k[10] 是一个广泛用于图像转 LaTeX 任务的公开数据集，由 Harvard NLP 团队在 2016 年首次整理发布。数据主要来源于公开的科研论文 LaTeX 源文件，总共从超过 60,000 篇文献中提取了 103,356 条数学公式。整个数据集共包含 103,556 条样本，训练集包含 83,883 条公式，测试集包含 9,319 条，验证集包含 10,354 条。每条数据包括一段 LaTeX 代码以及其对应的渲染后 PNG 图片，构成 LaTeX 与公式图像的一一对应标签对。每条 LaTeX 表达式的长度在 38 到 997 个字符之间。由于原始提取的 LaTeX 公式中各字符单元之间缺乏明确的分隔标记，因此在预处理阶段需要插入空格，以便后续模型能够更好地学习以及识别公式结构。此外，部分公式存在语法错误，无法被正常渲染，故在预处理阶段需对其进行筛选以保证数据质量。

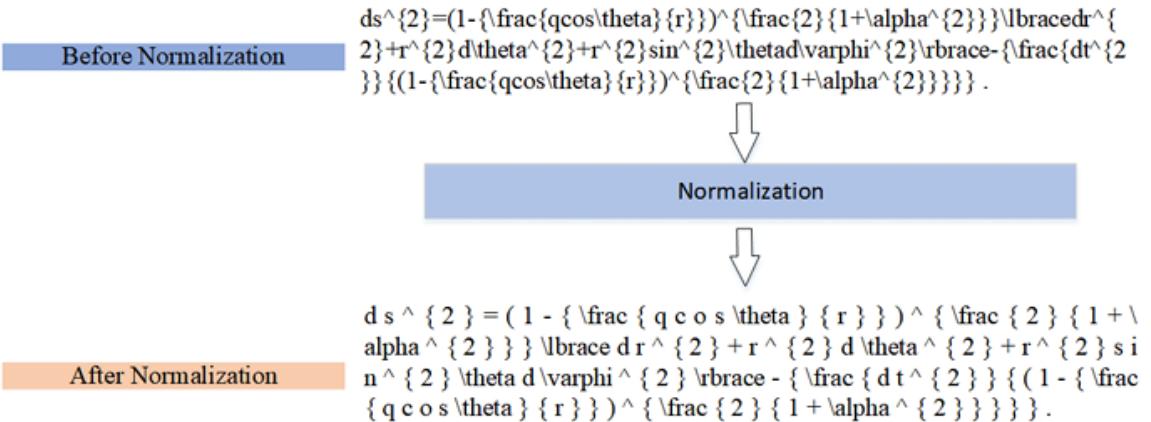


图 2: 基于抽象语法树的 LaTeX 规范化表达

数据预处理 为提高模型在公式识别任务中的学习效率，需对原始 LaTeX 表达式进行归一化预处理，以消除冗余结构并标准化字符表示。例如，表达式 $a_{\{b\}}$ 与 $(a)_{\{\{b\}\}}$ 在渲染结果上完全一致，但后者存在冗余的括号嵌套，适当减少括号层级有助于降低模型的计算开销。原始 LaTeX 表达式可借助 LaTeX 解析库转换为结构化的抽象语法

树 (Abstract Syntax Tree)，其中较为经典的解析工具是由 JavaScript 编写的 KaTeX。针对每条公式生成对应的 AST 之后，通过对该树进行遍历，可以提取出规范化之后的 LaTeX 表达序列。这一过程对 LaTeX 表达式进行字符单元间空格插入、冗余括号剔除、非法语法过滤、符号写法统一、函数格式规范化等，最大限度提升模型训练阶段的表达一致性与识别效率。

3.2 模型微调

在本研究中，我们基于预训练的 Qwen2.5-VL-3B 模型进行了全参数监督微调。考虑到视觉编码器在大规模图文数据上已具备强大的图像表征能力，我们在进行微调时冻结 ViT 的参数，这样不仅能保留其在预训练中学习到的视觉特征，同时还能显著降低训练过程中的显存与计算开销。此外，该策略可进一步减轻多模态训练中的优化难度，使语言模型部分更专注于图文联合建模与跨模态理解能力的提升。

训练在一台搭载 4 张 NVIDIA A800 GPU (单卡 80GB 显存) 的服务器上进行，系统环境为 Rocky Linux 8.8。我们基于 PyTorch 2.5.0 (CUDA 11.8 编译) 构建训练框架，采用 Hugging Face 的 Transformers 4.51.3 进行模型加载与训练调度，并借助 Accelerate 实现多卡并行训练能力。同时，我们采用 bfloat16 精度格式进行模型权重存储与训练推理，在保持计算精度的同时进一步降低内存开销。优化器采用 AdamW，初始学习率为 $1e-5$ ，相较预训练阶段适当降低，以适应全参数更新下更敏感的权重调整。我们采用 cosine 退火调度策略，并设置 warmup 比例为 0.1。训练共进行 2 个 epoch。同时，我们加入了 0.01 的正则化衰减项，防止过拟合。

由于多模态模型在处理高分辨率图像和大长度 token 序列时显存压力极高，我们将每张 GPU 的 batch size 设为 1，并引入 gradient accumulation steps = 16 的梯度累积策略，使训练过程在显存受限的前提下仍能模拟等效全局 batch size 为 16 的训练效果，从而提高参数更新的稳定性与梯度估计的可靠性。该配置尤其适合全参数微调任务，在有效降低梯度震荡的同时避免了 mini-batch size 太小时可能带来的收敛困难。

为防止前向传播阶段中间激活占用过多显存，我们启用了 gradient checkpointing 机制。该机制仅保存部分关键层的中间激活值，其余激活值在反向传播阶段动态重计算，从而将空间复杂度从 $O(n)$ 降低到 $O(\sqrt{n})$ ，有效缓解显存瓶颈问题。虽然此策略带来一定的前向重计算开销，但在大型多模态模型中，其节省显存所带来的训练可行性提升远大于其计算代价。

微调时模型的输入采用了 ChatML 格式 (OpenAI, 2024) 对多模态指令数据进行结构化建模，以提升模型对指令任务的理解能力。该格式有别于预训练阶段的纯文本或图文对齐数据格式，但在结构上保持了与 Qwen2-VL 架构的一致性。引入 ChatML 格式带来了三项关键优势：首先，通过明确的角色标记 (如 `<|user|>` 和 `<|assistant|>`)，实现了对多轮图文交互中说话角色的显式建模，有助于模型理解多模态对话中的角色轮换关系；其次，视觉特征被结构化地注入至文本指令中，使得视觉嵌入能够与语言输入共同构成统一的跨模态输入序列；最后，借助格式感知的 token 打包方式 (format-aware

packing)，模型能够保持图文之间的空间与顺序关系，从而提升视觉上下文的理解精度。通过在此格式下构造的图文指令-响应对，SFT 能够实现有效的知识迁移，同时保持预训练模型已学得的语义与表征能力。

在 2 个完整训练轮次结束后，模型共处理了约 1787 万个 token，总训练耗时为 48,982 秒。训练期间平均每秒处理样本数为 3.12，训练步数速度为 0.049 step/s，在较小 batch size 与梯度累积设置下模型训练进度稳定。最终模型在验证集上的损失为 0.271，在训练集上的平均 token 准确率（mean token accuracy）达到 98.7%。

4 效果分析

4.1 评估指标

为了全面评估公式图像转 *LaTeX* 系统的识别性能，本项目采用多种文本相似度指标对模型输出结果进行定量分析。主要包括 BLEU 分数（Bilingual Evaluation Understudy）、完全匹配率（Exact Match Accuracy）和字符错误率（Character Error Rate, CER）三类指标，分别从整体匹配质量、严格精确性与细粒度差异性三个维度对生成结果进行评估。

BLEU 分数 (Bilingual Evaluation Understudy) BLEU[11] 最初用于机器翻译任务，是一种衡量生成文本与参考文本之间 n-gram 重合度的指标。在本项目中，BLEU 分数用于评估模型生成的 *LaTeX* 序列与真实标注之间的整体相似程度。具体而言，设参考序列为 R ，模型生成序列为 H ，则 BLEU- n 分数定义为：

$$\text{BLEU-}n = \text{BP} \cdot \exp \left(\sum_{i=1}^n w_i \log p_i \right) \quad (1)$$

其中 p_i 表示第 i 阶 n-gram 的匹配精度， w_i 为加权系数（常设为 $1/n$ ）， BP 为惩罚过短生成序列的 brevity penalty 项。BLEU 分数反映了模型在生成语义结构相近表达式方面的能力，但在处理 *LaTeX* 表达式中存在多种等价写法的情况下，其敏感性也可能带来一定误判。

完全匹配率 (Exact Match Accuracy) 完全匹配率衡量模型生成的 *LaTeX* 序列是否与参考序列在字符级别完全一致，是一种严格的准确率评价指标。设总测试样本数为 N ，其中有 M 个样本模型生成序列与参考序列完全一致，则：

$$\text{Exact Match Accuracy} = \frac{M}{N} \quad (2)$$

该指标对公式的结构、空格、符号及命令写法均保持高度敏感，是当前 OCR 类任务中应用最广泛的主指标之一，适用于场景要求高度精确输出的情况。

字符错误率 (Character Error Rate, CER) CER 是评估生成文本与参考文本在字符级别差异的细粒度指标，通常用于计算两个字符串之间的编辑距离。CER 定义为模型输出序列到参考序列的最小编辑距离（包括插入、删除、替换操作）除以参考序列的总字符数：

$$\text{CER} = \frac{S + D + I}{N} \quad (3)$$

其中 S 为替换操作数， D 为删除操作数， I 为插入操作数， N 为参考序列的字符总数。CER 越低，表示模型输出与参考序列越接近。相比 BLEU 和 Exact Match，CER 更适合捕捉细节错误，常用于分析模型对小结构扰动的鲁棒性。

评估方法的局限性 需要指出的是，LaTeX 表达式在语法上具有一定的冗余性和表达歧义性，即多个语法不同但语义等价的表达可能对应同一数学含义（如 \backslashfrac 与 \backslashdfrac ）。因此，单纯依赖表面字符相似度进行评估可能存在一定偏差。考虑到此点，本项目综合了多种文本相似度的指标，综合评估模型的表现。

4.2 实验结果

4.2.1 定性结果分析

为进一步验证微调策略对模型预测结构准确性的提升，我们在测试集样例 Test_03 上进行可视化对比，结果如图 3 所示。

$$\begin{aligned} \frac{\partial b_{n,l}^*}{\partial t_1} &= (a_{n,l} - 1) + \frac{1}{t_1} \frac{b_{n,l}^*(t_{-1}b_{n,l} - l)}{1 - a_{n,l}}. & \text{(a) Ground Truth} \\ \frac{\partial b_{n,t}^*}{\partial t_1} &= (a_{n,t} - 1) + \frac{b_{n,t}^*(t_n - b_{n,t} - l)}{1 - a_{n,t}}. & \text{(b) Baseline Prediction} \\ \frac{\partial b_{n,l}^*}{\partial t_1} &= (a_{n,l} - 1) + \frac{1}{t_1} \frac{b_{n,l}^*(t_{-1}b_{n,l} - l)}{1 - a_{n,l}}. & \text{(c) Fine-tuned Prediction} \end{aligned}$$

图 3: 不同模型预测结果与真实标签的可视化对比

从图中可以观察到，baseline 模型在分式结构识别方面存在明显偏差，例如：

- 未能正确生成分式中分子与分母的完整表达；
- 对于变量下标如 t_1 、 n 、 l 等结构解析不准确；
- 中间出现了将指数与乘法混淆的情况，符号优先级处理也不稳定。

相比之下，微调后的模型能够准确识别公式结构，不仅正确输出了嵌套分式，还精确捕捉了上下标、括号层级、变量下标等细节，最终输出与 ground truth 完全一致。该

结果表明，在图像转 LaTeX 的场景中，针对结构表达能力的微调能显著提升模型的结构对齐能力和公式语法保真度，尤其对复杂表达式的结构还原具有重要意义。

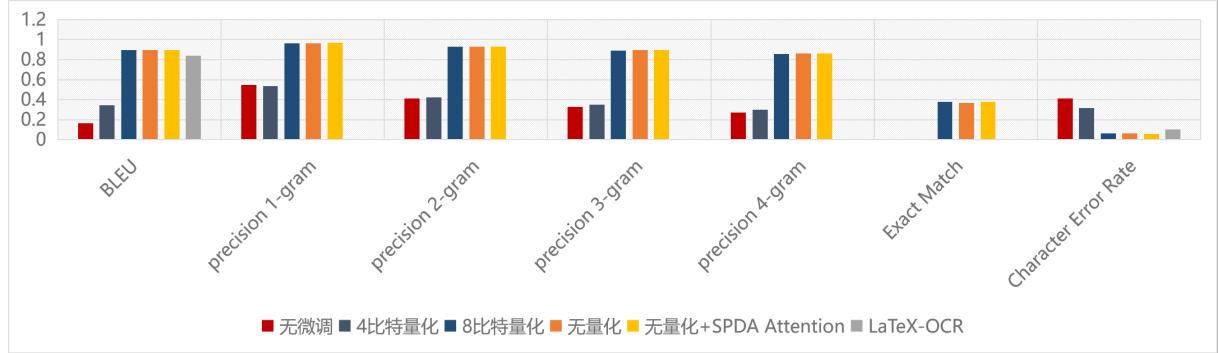


图 4: 实验结果汇总

4.2.2 识别性能

所有实验结果的汇总如图 4 所示。可以看到，只有在 4 比特量化下模型的性能表现较差，但 BLEU 分数相对于无微调的原始模型仍有约一倍的提升；而无量化、无量化且使用了 SPDA attention 及 8 比特量化的模型表现均有非常大的提升，三者的 BLEU 分数均在 0.9 左右，超过了原始论文中（图中 LaTeX-OCR）的 BLEU 分数。

4.2.3 推理效率与显存占用

由于本项目希望解决学生使用笔记软件等端侧设备时对 LaTeX 公式识别的需求，因此在提升模型的公式识别性能的同时也需要对模型的推理效率和显存占用进行分析。

表 1: 不同设置下的推理性能与文本生成质量

方法	BLEU 分数 ↑	长度比 ↑	推理时间 (秒) ↓	显存占用 (MB) ↓
无微调	0.1658	0.5509	0.8212	7754.88
4 比特双精度量化	0.3438	0.8821	2.3477	3878.00
8 比特量化	0.8954	0.9842	7.9428	4025.55
无量化	0.8959	0.9836	2.3151	7754.88
无量化 +SPDA	0.8964	0.9829	2.0987	7754.88

如表 1 所示。可以看到，经过微调和量化后的模型在推理时间上表现均不如原始模型，这与模型的输出长度提升有关，模型输出的长度比（输出 token 序列的长度与真实标注的 token 序列长度比值）从初始的 0.55 提升为 0.88-0.98。

未经微调的模型虽然推理速度最快 (0.82s)，但 BLEU 分数仅为 0.1658，length ratio 也显著偏低，表明其生成的 LaTeX 表达式存在严重缺失，无法有效完成任务。引入 4

比特双精度量化后，显存占用降低至 3878 MB，但 BLEU 分数仍仅为 0.3438，显示其在压缩同时损失了较多生成质量。

相比之下，未量化的完整模型在保持合理推理时间（2.31s）的同时，实现了 0.8959 的 BLEU 分数和接近 1 的 length ratio (0.9836)，显示出良好的生成准确性与表达完整性。在此基础上进一步引入 SPDA 注意力机制（即“无量化 +SPDA”方案）后，BLEU 分数略有提升至 0.8964，length ratio 达到 0.9829，推理时间反而有所减少（2.10s），表明该优化在不增加显存的情况下提升了效率和准确性。

此外，尽管 8 比特量化后的模型在显存占用上达到与 4 比特双精度量化接近的表现，在 BLEU 和 length ratio 上表现接近未量化模型，但其推理时间却大幅上升至 7.94s，说明该方案在当前设置下存在效率瓶颈。

因此，从整体平衡角度来看，“无量化 +SPDA”方案在精度与效率之间实现了最优折中，适合部署于对性能和资源均有要求的实际场景中，但其显存占用仍相对较高，如何在保持较高推理性能的情况下显著降低显存占用可以是后续工作的一个主要方向。

5 端侧部署

本项目还探索了微调后模型在端侧的部署，整体方案为：通过 llama.cpp 将模型权重保存为 GGUF 格式，然后利用 Ollama 框架及 Open WebUI，实现微调后的 Qwen2.5-VL 在个人电脑上的部署及可视化界面。

5.1 工具介绍

llama.cpp llama.cpp 是一个开源的高效推理框架，专为在资源受限环境下部署大规模语言模型而设计。该项目以 Meta 发布的 LLaMA 系列模型 [12, 13] 为基础，采用纯 C/C++ 实现，具备跨平台兼容性，支持在 CPU、GPU 甚至移动端设备上运行。llama.cpp 通过集成多种量化方案（如 4-bit、5-bit、8-bit）以及高效的推理优化（如 KV-cache、multi-threading 和 SIMD 指令加速），可以显著降低推理时的显存占用与运行延迟，在不牺牲太多模型精度的前提下，实现轻量化部署。

llama.cpp 支持 GGUF(GPT-Generated Unified Format) 格式，可兼容多种预训练或微调后的模型权重，便于用户快速加载和评估模型性能，已成为端侧部署 LLM（大语言模型）最主流的方案之一，尤其适用于教育设备、边缘计算节点和嵌入式系统等对算力敏感的场景。

GGUF 采用紧凑的二进制格式，相较于文本格式的文件，减少了读取和解析时所需的 I/O 操作和处理时间，可以更快地被读取和解析。同时支持将多种模型元信息（如 tokenizer、训练超参、量化元数据）集成到同一个文件中，简化了模型加载流程，也方便后续的版本控制和格式兼容。此外，GGUF 支持多种主流量化算法，在保持推理精度的同时，大幅压缩模型文件体积。

5.2 部署细节

为实现微调后模型在端侧的本地部署与交互测试，我们采用了轻量级推理框架 `llama.cpp`。首先将训练得到的 `safetensors` 格式模型权重文件转换为 `.gguf` 格式，后者为 `llama.cpp` 支持的统一二进制模型格式，具备结构紧凑、加载高效等优势。转换过程中采用了 Q4 或 Q8 等主流量化策略，在压缩模型大小的同时尽量保持推理精度。此外，GGUF 文件中嵌入了 tokenizer 配置、超参数、量化元信息等元数据，实现了模型结构与上下游组件的高度集成，极大简化了加载流程。

转换后的 `.gguf` 模型文件被导入至本地部署的 Ollama 推理引擎中，该引擎基于 `llama.cpp` 封装并提供服务接口。为了提升可用性与交互体验，我们进一步集成了 Open-WebUI 前端界面，使用户可以通过网页方式向本地模型发送图文指令并获取响应，便于验证模型在图像转 LaTeX 任务中的实际推理表现。该部署方案具备部署简便、运行高效、界面友好等特点，尤其适用于端侧验证、嵌入式集成或教学演示等场景，为后续多模态模型在实际系统中的落地提供了可行路径。

6 总结展望

在本项目中，我们使用 LaTeX OCR 数据集端到端微调了 Qwen2.5-VL 7B 模型，在多个指标上获得了巨大的提升。在未来，我们会继续探索更加轻量化的部署方案，使模型支持在边缘设备（如平板电脑、笔记本电脑等）上高效部署，以及针对用户字迹的个性化高效定制方案，如使用 QLoRA 技术在端侧高效微调。

参考文献

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [3] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

-
- [4] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, JinYeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Donut: Document understanding transformer without ocr. *arXiv preprint arXiv:2111.15664*, 7(15):2, 2021.
 - [5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.
 - [6] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, JinYeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer, 2022.
 - [7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
 - [8] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.
 - [9] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.
 - [10] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M Rush. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, pages 980–989. PMLR, 2017.
 - [11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
 - [12] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

-
- [13] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xi-ang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.