Qwen3-VL技术报告解读

Qwen3VL

论文链接：https://arxiv.org/pdf/2511.21631

代码链接：https://github.com/QwenLM/Qwen3-VL

⚽ 论文总结

Qwen3-VL实现了三方面能力的突破：

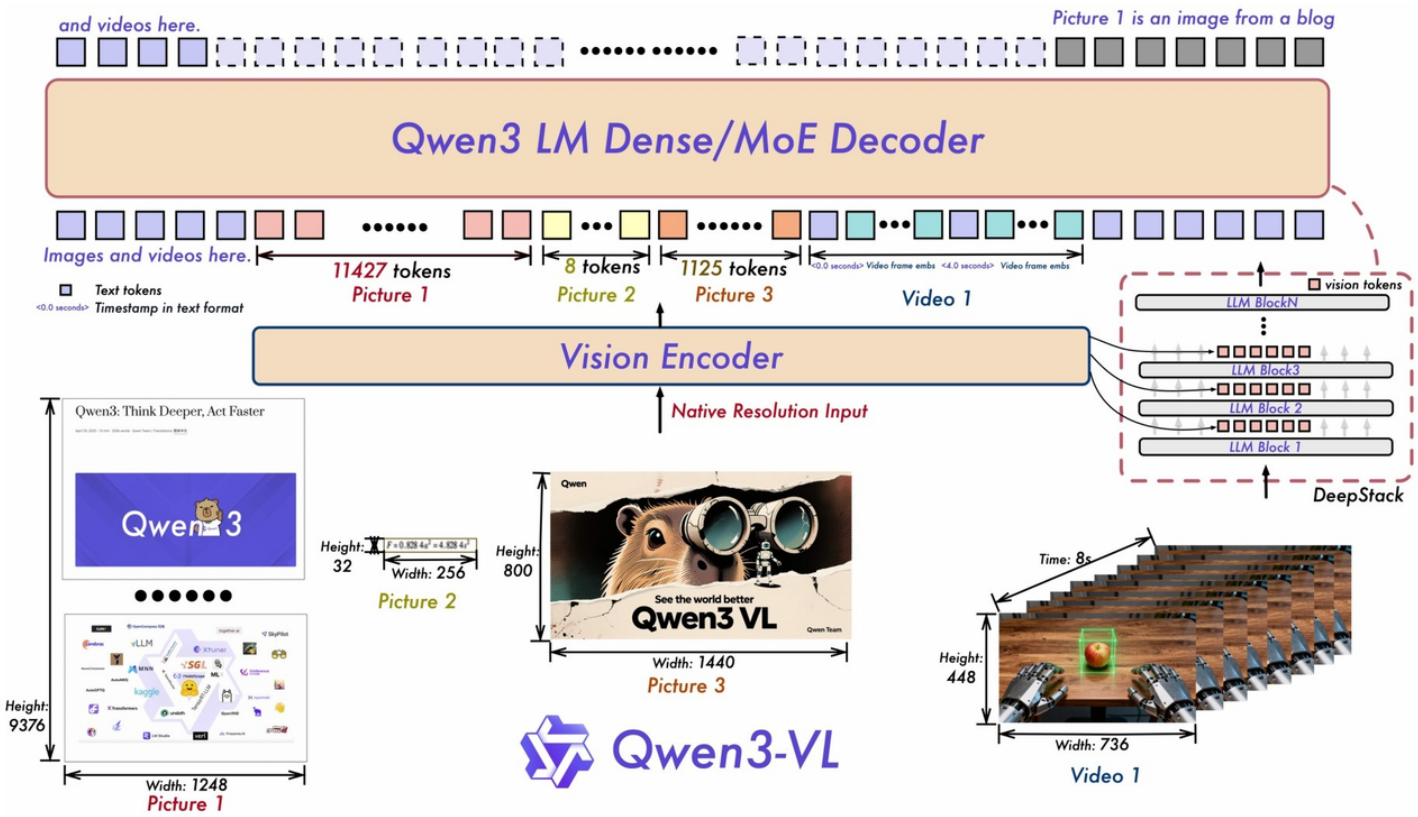
1. 更强的**纯文本理解能力**，在多个场景下超越同类纯文本模型
2. 更强的**长上下文理解能力**，支持文本及交错多模态输入的 256K token窗口。
3. 更强的对单图、多图和视频的**多模态推理能力**。

在架构方面，做了以下改进

1. **交错-MRoPE**，原始MRoPE将特征维度按照时间（T）、高度（H)和宽度（W)的顺序分块划分，使得时间信息全部分布在高频维度上。Qwen3-VL将时间、高度、宽度三个维度均匀分布在低频和高频带中，显著提升图像与视频中的时空建模能力；
2. **DeepStack**，ViT不同层的视觉token通过残差连接路由至对应的 LLM 层，能够有效保留从底层（low-level）到高层（high-level）的丰富视觉信息，在不增加额外上下文长度的情况下增强多层级融合，强化视觉-语言对齐；
3. 采用**基于文本的时间对齐机制**，通过显式的文本时间戳对齐替代 Qwen2.5-VL 中通过位置编码实现的绝对时间对齐，采用“时间戳-视频帧”交错的输入形式，实现更精确的时空定位。为平衡纯文本与多模态学习目标，采用平方根重加权策略，在不损害文本能力的前提下显著提升多模态性能。

Qwen3-VL的训练流程：

- **预训练**：四阶段（**视觉语言对齐→全参数多模态训练→长上下文适应→超长上下文优化**）
- **后训练**：三阶段（**监督微调→知识蒸馏→强化学习**）



模型架构

Qwen3-VL仍然采用**ViT+Merger+LLM**的架构，重点关注下面架构中红色的部分：

- **LLM**：包含3种dense模型和2种MOE模型，旗舰模型为 Qwen3-VL-235B-A22B。在自注意力中使用了**QK-Norm**，并且在前3层进行Deepstack特征融合。
- **ViT**：复用**SigLIP-2架构，**在其基础上进行继续预训练（Qwen2.5VL重新训练ViT）。
  - patch_embed中开启了bias，**patch-size变成16**（Qwen2.5VL关闭 bias，patch-size14）。
  - MLP中激活函数变成 `PytorchGELUTanh` （Qwen2.5VL `SiLU`）。
  - 位置编码仍然采用**2D-RoPE**，支持动态分辨率，并根据输入尺寸插值绝对位置嵌入。
  - 采用**LayerNorm**（Qwen2.5VL 采用RMSNorm）。
  - 定位从绝对坐标又改回了相对坐标。
- **Merger**：与Qwen2.5VL一样采用两层的MLP，将 2×2 视觉特征压缩为1个token。区别是采用LayerNorm，并使用了DeepStack机制(后面介绍)。

代码块

```
Qwen3-VLMoeForConditionalGeneration(
  (model): Qwen3-VLMoeModel(
    (visual): Qwen3-VLMoeVisionModel(
      (patch_embed): Qwen3-VLMoeVisionPatchEmbed(
        (proj): Conv3d(3, 1152, kernel_size=(2, 16, 16), stride=(2, 16, 16))
      )
      (pos_embed): Embedding(2304, 1152)
```

```
        (rotary_pos_emb): Qwen3-VLMoeVisionRotaryEmbedding()
        (blocks): ModuleList(
          (0-26): 27 x Qwen3-VLMoeVisionBlock(
            (norm1): LayerNorm((1152,), eps=1e-06, elementwise_affine=True)
            (norm2): LayerNorm((1152,), eps=1e-06, elementwise_affine=True)
            (attn): Qwen3-VLMoeVisionAttention(
              (qkv): Linear(in_features=1152, out_features=3456, bias=True)
              (proj): Linear(in_features=1152, out_features=1152, bias=True)
            )
            (mlp): Qwen3-VLMoeVisionMLP(
              (linear_fc1): Linear(in_features=1152, out_features=4304,
bias=True)
              (linear_fc2): Linear(in_features=4304, out_features=1152,
bias=True)
              (act_fn): PytorchGELUTanh()
            )
          )
        )
        (merger): Qwen3-VLMoeVisionPatchMerger(
          (norm): LayerNorm((1152,), eps=1e-06, elementwise_affine=True)
          (linear_fc1): Linear(in_features=4608, out_features=4608, bias=True)
          (act_fn): GELU(approximate='none')
          (linear_fc2): Linear(in_features=4608, out_features=4096, bias=True)
        )
        (deepstack_merger_list): ModuleList(
          (0-2): 3 x Qwen3-VLMoeVisionPatchMerger(
            (norm): LayerNorm((4608,), eps=1e-06, elementwise_affine=True)
            (linear_fc1): Linear(in_features=4608, out_features=4608, bias=True)
            (act_fn): GELU(approximate='none')
            (linear_fc2): Linear(in_features=4608, out_features=4096, bias=True)
          )
        )
      )
      (language_model): Qwen3-VLMoeTextModel(
        (embed_tokens): Embedding(151936, 4096)
        (layers): ModuleList(
          (0-93): 94 x Qwen3-VLMoeTextDecoderLayer(
            (self_attn): Qwen3-VLMoeTextAttention(
              (q_proj): Linear(in_features=4096, out_features=8192, bias=False)
              (k_proj): Linear(in_features=4096, out_features=512, bias=False)
              (v_proj): Linear(in_features=4096, out_features=512, bias=False)
              (o_proj): Linear(in_features=8192, out_features=4096, bias=False)
              (q_norm): Qwen3-VLMoeTextRMSNorm((128,), eps=1e-06)
              (k_norm): Qwen3-VLMoeTextRMSNorm((128,), eps=1e-06)
            )
            (mlp): Qwen3-VLMoeTextSparseMoeBlock(
```

```
            (gate): Qwen3-VLMoeTextRouter(in_features=4096, out_features=128, bias=False)
            (experts): Qwen3-VLMoeTextExperts(
              (act_fn): SiLU()
            )
          )
          (input_layernorm): Qwen3-VLMoeTextRMSNorm((4096,), eps=1e-06)
          (post_attention_layernorm): Qwen3-VLMoeTextRMSNorm((4096,), eps=1e-06)
        )
      )
      (norm): Qwen3-VLMoeTextRMSNorm((4096,), eps=1e-06)
      (rotary_emb): Qwen3-VLMoeTextRotaryEmbedding()
    )
  )
  (lm_head): Linear(in_features=4096, out_features=151936, bias=False)
)
```

Qwen3-VL 模型架构

QK Norm介绍

论文链接：https://arxiv.org/pdf/2010.04245

应用：在Qwen3、LLaMA4等模型都应用

面临问题：**注意力计算中的 $\frac{QK^T}{\sqrt{d}}$ 是无界的，导致 $SoftMax(\frac{QK^T}{\sqrt{d}})$ 容易饱和**

原理：对 Q 和 K 分别沿头维度进行 L2 归一化，这样就**将点积注意力转换为余弦相似度注意力**，将 $\hat{Q}\hat{K}^T$ 限制在 $[-1,1]$ 区间，避免softmax饱和导致的梯度消失爆炸

$$\hat{Q}=\frac{Q}{\|Q\|_2},\quad \hat{K}=\frac{K}{\|K\|_2}, \hat{Q}\hat{K}^T=cosine\_similarity(\hat{Q},\hat{K})$$

工程实现：在这篇论文中，用可学习参数 g 代替标准注意力中的固定缩放因子 $\frac{1}{\sqrt{d}}$

$$Attention=softmax(g*\hat{Q}\hat{K}^T)V$$

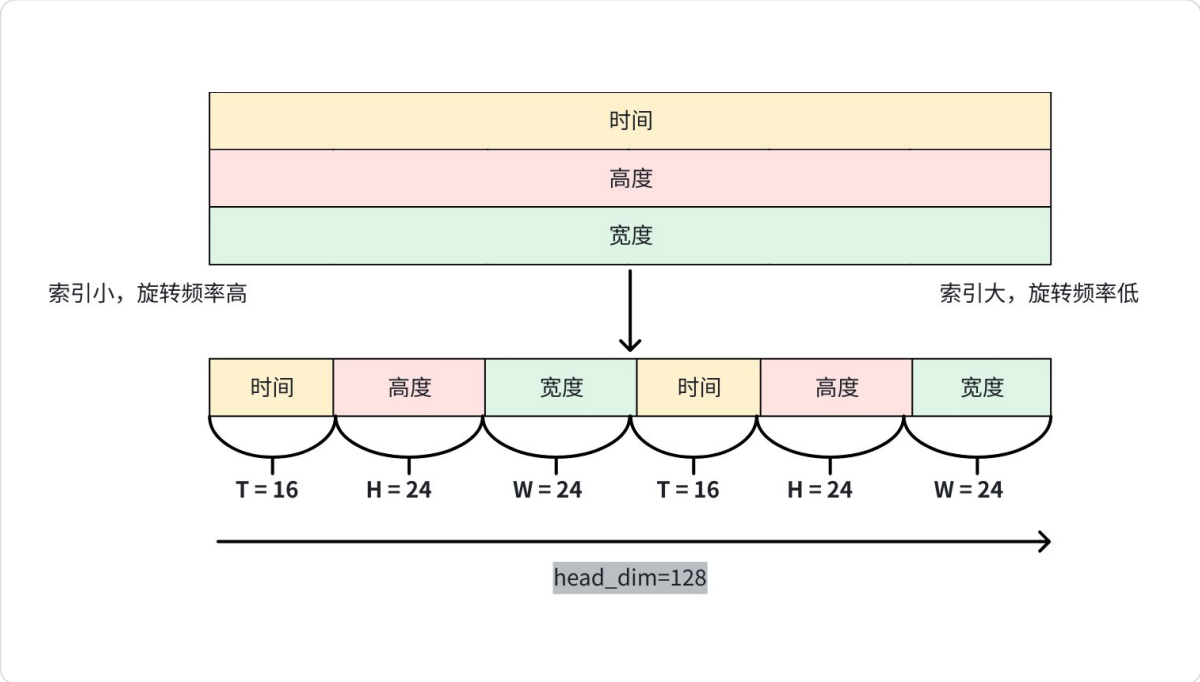
但在大模型中，选择用**RMSNorm**实现QKNorm

$$\hat{Q}=Q/\sqrt{(E[Q^2]+\varepsilon)}*\gamma$$
$$\hat{K}=K/\sqrt{(E[K^2]+\varepsilon)}*\gamma$$

其中 γ 是可学习的缩放参数，ε 是防止除零的小常数，γ 与 g 的本质是一样的，都是为了将 $\hat{Q}\hat{K}^T$ 的取值范围从 $[-1,1]$ 拓展到更大的区间（但不是 $\frac{QK^T}{\sqrt{d}}$ 那种无界），避免softmax输出差异过小，难以区分 token 间的关联性。

交错-MRoPE

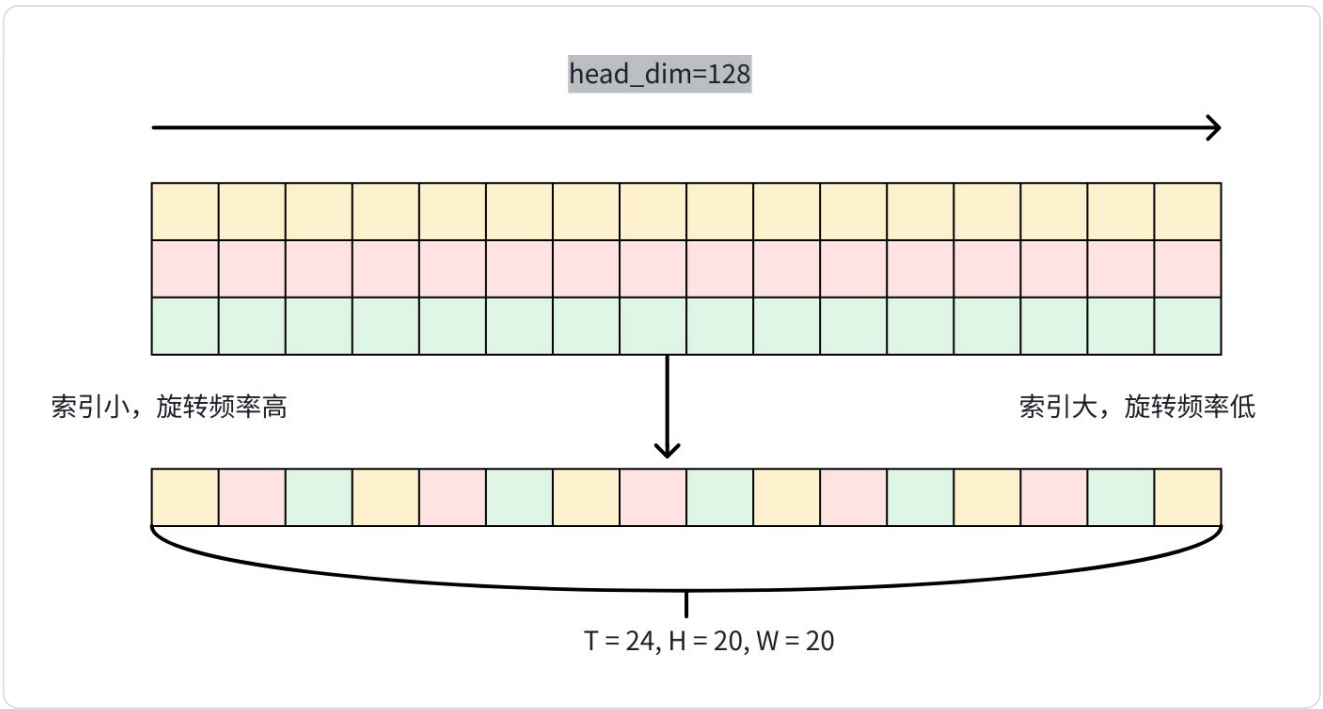
回忆一下Qwen2.5VL 中的MRoPE，使用3D位置信息（**时间，高度，宽度**）。其位置向量的组成方式为：



> 一个token的sin/cos向量

但这种方式存在问题，即RoPE中 $\theta_i = 10000^{\frac{-2i}{d}}$ ，i 表示索引，由于旋转频率随着索引增加而降低，MRoPE会导致时间维度的信息全部在高频维度上，不利于长序列的理解，会导致注意力随着时间快速衰减。

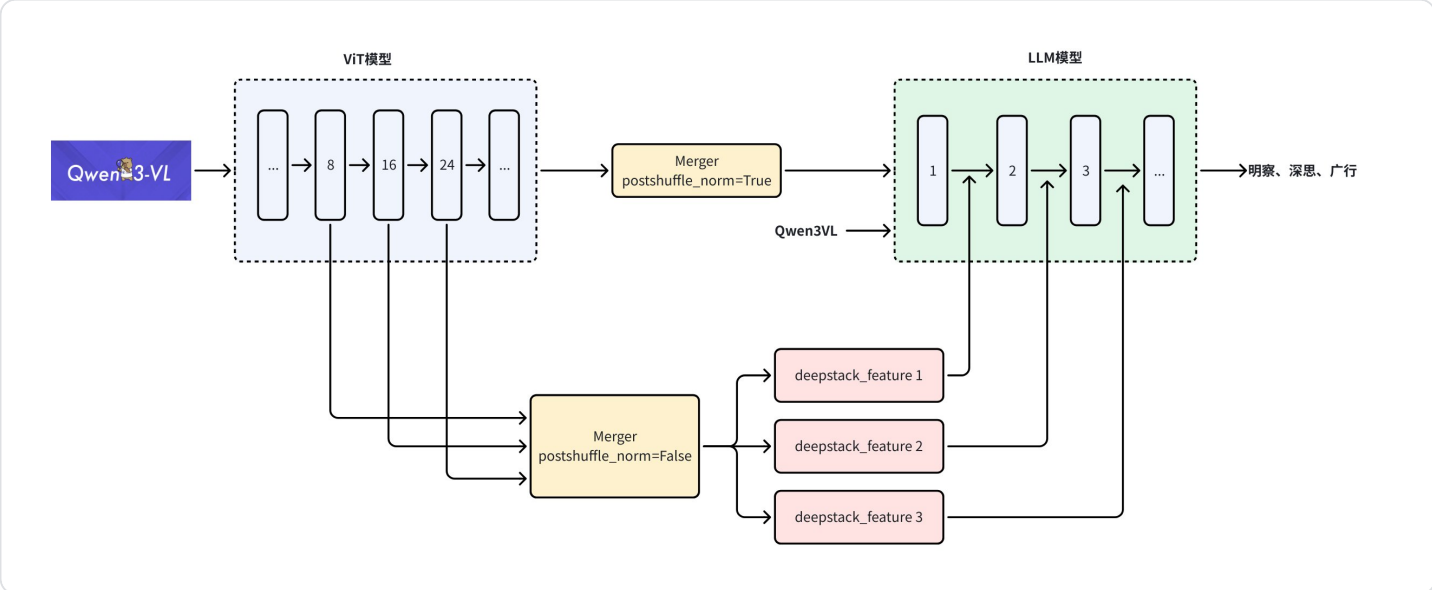
为此，Qwen3-VL在LLM中采用Interleaved MRoPE，以细粒度的轮询方式将特征通道分配到**时间，高度，宽度**轴上，确保每个位置轴都使用从高到低的完整频谱进行编码。



> 上图中黄、粉、绿分别表示T、H、W维度，T=24，H和W=20，1:4缩小，所以最后会有一个单独的时间块。

DeepStack

从 ViT的中间层提取视觉标记，注入到LLM的多个层中，保留了从低级到高级表示的丰富视觉信息。从视觉编码器的三个 $[8, 16, 24]$ 不同层级选择特征，使用Merger将这些特征投影为视觉token，然后添加到前三个LLM层的对应hidden states中。



基于文本的时间对齐机制

Qwen2.5VL将时间位置 ID 直接关联到绝对时间（即3DRoPE，时间维度的值对应帧数），该方法在处理长视频时会产生过大且稀疏的时间位置 ID，削弱模型对长时序上下文的理解能力。 并且为了有效学习，需要在不同帧率（fps）下进行广泛且均匀的采样，显著增加了训练数据构建的成本。

Qwen3-VL采用**基于文本的时间对齐机制**，为每个视频时序patch都添加时间戳前缀，在训练过程中添加了“秒”和“时:分:秒”两种格式的时间戳以确保模型能够学习理解多种时间码表示。这种方法会带来适度的上下文长度增加。

在数据预处理时就已经在文本中添加了时间戳，输入是 `聪明的<t1> <vision_start> <video_token> [视觉特征token序列] <vision_end> 小羊` 。其中 `<t1>` 表示时间戳， `[视觉特征token序列]` 包含1个帧，每一帧是2×2 网格 `（llm_grid_h=2， llm_grid_w=2）` 。

Token	类型	位置ID （T， H， W）	解释
聪明	文本	(0, 0, 0)	文本token，THW三维相同
的	文本	(1, 1, 1)	文本token，THW三维相同
<t1>	文本	(2, 2, 2)	时间戳被视为文本token，TH
<vision_start>	文本	(3, 3, 3)	视觉开始标记token，THW三
<video_token>	文本	(4, 4, 4)	视频token标记，THW三维相
(f1_0,0)	视觉	(0, 0, 0) + 5 = (5, 5, 5)	视觉特征token，t=0，h=0, w
(f1_0,1)	视觉	(0, 0, 1) + 5 = (5, 5, 6)	视觉特征token，t=0，h=0, w
(f1_1,0)	视觉	(0, 1, 0) + 5 = (5, 6, 5)	视觉特征token，t=0，h=1, w
(f1_1,1)	视觉	(0, 1, 1) + 5 = (5, 6, 6)	视觉特征token，t=0，h=1, w
<vision_end>	文本	(9, 9, 9)	视觉结束标记token，THW三
小	文本	(10, 10, 10)	文本token，THW三维相同
羊	文本	(11, 11, 11)	文本token，THW三维相同

预训练

Stage	Objective	Training	Token Budget	Sequence Length
S0	Vision-Language Alignment	Merger	67B	8,192
S1	Multimodal Pre-Training	All	~1T	8,192
S2	Long-Context Pre-Training	All	~1T	32,768
S3	Ultra-Long-Context Adaptation	All	100B	262,144

四阶段预训练

Qwen3-VL的预训练分为四个阶段

1. **视觉-语言对齐：**弥合视觉编码器与LLM的模态差距

- **训练策略：****仅训练MLP merger参数**，冻结ViT和LLM主
- **数据：**约67B tokens的高质量图像-文本对、视觉知识库和OCR数据
- **序列长度：**8,192 tokens

2. **多模态预训练：**全参数端到端联合训练

- **训练策略：****解冻视觉编码器、合并器和LLM所有参数**
- **数据：**约1T tokens混合b数据（视觉语言数据+纯文本数据），前者包含交错图文文档、视觉定位、VQA、STEM领域数据及少量视频数据
- **序列长度：**保持8,192 tokens

3. **长上下文预训练：**扩展上下文处理能力

- **训练策略：****训练所有模型参数，序列长度增至32,768 tokens**
- **数据：**约1T tokens数据，增加纯文本数据比例，强化长文本理解；增加视频和agent指令遵循数据

4. **超长上下文适应**：将上下文窗口扩展至极限

- **关键改进**：**训练所有模型参数，序列长度增至262,144 tokens**
- **数据**：100B tokens数据集 ，重点是长视频理解和长文档分析。

训练数据处理

1. **图像-标题对与图文交错数据**

- **图像-标题对**：对于网页多语言图文对，**用Qwen2.5-VL-32B重写描述**，强化视觉元素与语义表达；基于**语义相似性进行语义去重**，通过**聚类**识别稀疏数据，并进行针对性增强。
- **图文交错**：采集中英文文档，基于**微调后的轻量级 Qwen 的评分器进行领域分类**，过滤广告等低价值内容；对书籍类数据**用微调后的 Qwen2.5-VL-7B 模型进行解析**，精确提取并对齐文本与嵌入的图表、示意图和照片。**合并页面生成最长256K tokens的序列**，实现超长上下文建模。

2. **知识类数据**

- 覆盖10+语义类别。采用**重要性采样平衡长尾分布**：高频实体多采样，低频实体少量保留；替换稀疏标注为LLM生成的包含属性、场景等的详细描述。

3. **OCR、文档解析与长文档理解**

- **OCR**：构建粗到精的流水线，**利用OCR模型和Qwen2.5VL优化OCR标注**。包含3000万内部样本+3000万多语言合成样本。
- **文档解析**：包含300万Common Crawl PDF+400万内部文档，先用**模型标注文本区域和非文本区域的顺序和边界**，**用Qwen2.5-VL-72B进行区域识别**，最后将输出结果重新组合为具有位置感知、版面对齐的解析数据。
- **长文档理解**：将单页文档合成长文档解析序列，生成长文档VQA数据，并平衡问题类型分布。

4. **Grounding 与计数**

- **边界框 Grounding**：整合COCO等开源数据集，开发自动化合成标注（**Qwen2.5-VL提取物体候选+Grounding DINO标注+过滤低置信样本**）。
- **点 Grounding**：融合PixMo等公开数据并合成聚焦细粒度图像细节等标注数据。
- **计数**：包含直接计数、框计数、点计数三类任务，**采用[0,1000]归一化坐标**，提升分辨率适应性。

5. **空间理解与3D识别**

- **空间理解**：为了让模型能够推理二维场景中的空间关系、物体可操作性以及可行操作，**构建了一个包含提升含关系标注（如“杯子在电脑左侧”）、可操作性标签（如“可抓取”）、动作规划查询（如“为了拿到显示器后面的书，我应该先移动什么？”）的数据集**，采用**相对坐标**鼓励关系推理。
- **3D 定位**：构建3D视觉定位数据集（**图像+自然语言指代+边界框**），将所有数据统一到一个相机坐标系。

6. **代码数据**

- **纯文本代码**：复用**Qwen3 和 Qwen3-Coder系列数据集**，覆盖软件开发、算法、数学推理等场景。
- **多模态代码**：包含**截图转HTML/CSS、图像转SVG代码、视觉编程题、流程图转代码**等任务。

7. **视频数据**

- **时序感知视频理解**：**长视频采用从短到长字幕生成策略**，利用字幕生成模型生成细粒度的标注；为增强模型的时空定位能力，构建时空定位数据，在物体、动作和人物层面进行了标注。
- **数据平衡**：按数据来源平衡分布，**根据不同的序列长度约束，动态调整采样参数，如每秒帧数（fps）和最大帧数，进行长度自适应采样**。

8. **STEM类数据**

- **视觉感知**：通过程序生成几何图表，包含100万点定位样本、200万面向感知的视觉问答对；经过两阶段标注+模型验证，生成600万图表描述数据集。
- **多模态推理**：6000万K12至本科习题，清洗低质量数据、统一答案格式；采用推理模型合成1200万带图像的长CoT样本，**基于规则和模型验证推理轨迹**，筛选高难度问题。
- **语言推理**：复用**Qwen3的推理数据，因为多模态推理能力在很大程度上源于语言推理能力。**

9. **智能体数据**

- **GUI**：GUI界面感知包含元素描述、密集标注等任务；智能体能力方面，构建多步骤任务轨迹+人工审核；补充CoT推理，强化规划与自我修正能力。
- **函数调用**：多模态函数调用轨迹合成流水线（生成查询、函数定义、调用逻辑、响应，此过程重复进行，直到用户查询被认为已解决）。
- **搜索**：结合图像与文本搜索工具收集多模态事实查询轨迹，**鼓励模型对陌生实体主动搜索。**

后训练

3阶段后训练

1. **SFT：**激活指令遵循能力和潜在推理技能

- 分两阶段实施：**32k上下文长度训练 + 扩展到256k上下文窗口**，专注长文档/长视频数据
- 训练数据分两类：**用于非思考型模型的标准格式，以及用于思考型模型的CoT格式。**

2. **强弱知识蒸馏：**将教师模型能力迁移到学生模型

- **使用纯文本数据进行LLM微调**，显著提升文本/多模态任务的推理能力

3. **强化学习：**分两个阶段：

- **推理RL**：覆盖数学、编码、逻辑推理、视觉基础等任务
- **通用RL**：增强指令跟随和人类偏好对齐

SFT阶段

SFT数据

在Qwen2.5VL的基础能力上（包含分成 8 个核心领域，30 个细粒度领域），新增了以下能力：

- **具身智能的空间推理**
- **视频时空定位的鲁棒目标追踪**
- **细粒度视觉理解的图像推理**
- **数百页的长技术文档的理解**

- **数据集构成**：**约 120 万样本 ，1/3 为纯文本 ，2/3 为图像-文本和视频-文本对** 。（对比Qwen2.5用了200万数据，文本多模态1:1）。引入单轮和多轮对话，支持单图、多图序列的对话动态模拟。包含交错图像-文本示例，用于工具增强的图像搜索和视觉推理 。
- **训练策略**
  a. 第一阶段：**32K token** 序列长度，训练 1 epoch
  b. 第二阶段：**256K token 序列长度**，32k 和 256k 数据混合的训练，训练 1 epoch
- **数据质量控制**
  - **查询过滤**：
    - **使用 Qwen2.5-VL 筛选不可验证的查询**
    - **修正模糊指令**，去除无实质内容的网络来源查询
    - 所有剩余的查询经过**复杂度和上下文相关性**的最终评估，仅保留适当难度且相关的样本进入下一阶段。
  - **响应过滤**：
    - **规则过滤**：**去除重复、不完整或格式错误的响应，过滤偏离主题或有害内容**
    - **模型过滤**：**基于 Qwen2.5-VL 的奖励模型**，评估答案正确性、完整性、语言一致性等维度；视觉任务验证视觉信息的准确应用；以及过滤掉规则方法难以识别的问题，如不恰当的语言混用或突兀的风格转换

冷启动数据

- **数据构成与领域覆盖：**（**视觉语言:纯文本 ≈ 1:1**），多模态部分覆盖 VQA、OCR、2D/3D 定位、视频分析等传统领域，特别强化 **STEM 和Agent**相关任务；文本部分跟Qwen3数据一致。
- **数据过滤**：
  - 先做难度过滤：只保留**base模型做不对和回复更长更详细的数据**。
  - 多模态必要性过滤：过滤掉Qwen3-30B-nothink 能**不依赖图片就能做对的题**
  - 与Qwen3一样对相应进行处理：过滤到错误、重复、语言混乱、**猜答案**（Qwen3-VL 中新提到的）、缺乏推理步骤的数据。

强到弱蒸馏

使用纯文本数据进行LLM微调，分为两个阶段

- **off-policy蒸馏**：直接把教师模型回复给学生模型做微调。
- **on-policy蒸馏**：**最小化教师和学生模型的logits之间的KL散度。**

强化学习

推理强化学习

目的：提升模型推理能力

1. **数据准备**

- **数据来源**：包含**文本和多模态数据**，覆盖数学、编程、逻辑推理、视觉定位和视觉谜题领域。
- **数据预处理**：使用Qwen3-VL-235B-A22B对每个查询生成16个响应，若全部做错则丢弃该查询（**删掉太难的**）。
- **数据筛选**：**每个数据源单独做实验，如果RL实验之后没提升就剔除**（看起来工作量巨大）。最终得到30K数据。
- **训练阶段过滤**：训练时rollout16次，通过率>90%的简单查询进行过滤掉（**删掉太简单的**）。
- **批次构建**：**一个batch混合不同任务数据，每个batch的比例固定，通过预实验确定各任务样本比例**。

2. **奖励系统设计**

- 构建统一的奖励框架，不同任务的奖励需要分别实现，共享数据预处理、工具函数、奖励管理器等。
- **删除格式奖励**：**通过prompt引导模型输出规范格式，无需显式格式奖励**
- **语言惩罚**：**对输出语言与prompt中要求语言不一致的情况添加惩罚。**

3. **RL算法**：采用**SAPO**，对比GRPO/GSPO，能更长时间的稳定学习，达到更高的Pass@1准确率。主要创新为：

   a. 用**受温度控制的软门控机制替代了硬裁剪**

   b. **为负token设置更高的温度**，使得负token上的梯度衰减得更快，从而提升训练的稳定性和性能

通用强化学习

目的：提升模型的泛化能力和鲁棒性，进行多任务RL训练。

1. **多任务奖励机制**：基于SFT阶段的多个任务（VQA、图像描述、OCR、文档解析、grounding、时钟识别等）构建综合奖励函数，优化以下两个维度：
   - **指令遵循**：评估模型对显式用户指令的遵守能力，包括**内容、格式、长度和结构化输出的约束**。
   - **偏好对齐**：针对开放式或主观性查询，优化**输出的帮助性、事实准确性和风格适宜性**，以符合人类偏好。
2. **错误先验纠正**：通过**设计可验证任务（如反直觉对象计数、复杂时钟时间识别）触发SFT阶段形成的错误知识先验，用事实知识替代错误先验**。

3. **低频问题抑制**：针对**不恰当语言混合、过度重复、格式错误等低频问题**，跟着其他数据一起做RL训练样本效率太低，因此**构建会诱发此类不良行为的prompt的数据集专门训练**，通过高频有针对性的惩罚策略抑制这些错误。
4. **混合奖励设计：**
    - **规则奖励**：可验证问题基于明确规则（如格式遵循）提供高精度反馈，缓解**奖励劫持**。
    - **模型奖励**：开放性问题利用**Qwen2.5-VL-72B-Instruct或Qwen3作为评估模型**，对比模型生成回复与真实答案。

Think with Image

目标：**增强多模态模型的工具调用能力**

1. 创建一个冷启动agent数据集，包含10k个视觉问答任务，对Qwen2.5-VL-32B微调，模拟视觉agent的行为：`think → act → analyze feedback → answer`，最后进行**多轮、工具集成的RL**。
2. 再用训练好的Qwen2.5-VL-32B蒸馏出120k的多轮agent交互数据。用这些数据对Qwen3-VL进行相同的SFT+RL流程。

强化学习采用三种奖励信号：

- **准确性奖励**：用 Qwen3-32B 来衡量**最终答案是否正确**
- **多轮推理奖励：**利用 Qwen2.5-VL-72B 评估agent的**推理过程奖励**
- **工具调用奖励**：对比实际工具调用次数与 Qwen2.5-VL-72B估算的调用次数，**鼓励适当的工具调用，防止hack到不调用工具或者只调用一次工具的情况。**

Infrastructure

训练使用PAI-Lingjun，基于**Megatron**进行分布式训练（整合了张量并行（TP）、流水线并行（PP）、上下文并行（CP）、专家并行（EP）以及 ZeRO-1 数据并行（DP）），在万卡规模仍能保持高吞吐量和低通信延迟。

本地部署采用**vLLM**或**sglang**，前者通过PageAttention实现高吞吐量，后者能更好的结构化生成和处理复杂提示。

实验

技术报告的实验部分，首先证明在各种任务上赢麻了，本文只关注最后的消融实验。

Vision Encoder

在进行预训练之前，Qwen3-VL中的ViT是在SigLIP-2的基础上，使用动态分辨率进行CPT得到的。因此为了证明CPT训练的有效性，本文对比了Qwen3-ViT和原始的SigLIP-2模型。

指标如下图，在clip预训练的指标上与SigLIP-2基本相同，当拼接上1.7B的Qwen3 语言模型并训练1.5T token后，在VLM Bench上取得更高的性能。

ViT	Clip Bench							VLM Bench				
	ImageNet-1K	ImageNet-V2	ImageNet-A	ImageNet-R	ImageNet-S	ObjectNet	Omni	OCRB	AI2D	RLWDQA	InfoVQA	Omni
SigLIP-2	84.2	78.6	87.0	96.1	76.2	79.9	36.9	77.2	74.1	58.7	65.3	50.1
Qwen3-ViT	84.6	78.8	87.1	95.7	74.5	81.0	45.5	78.7	76.2	66.1	67.0	53.0

DeepStack

内部15B-A2B LLM上对DeepStack进行消融实验，消耗200B token进行预训练，证明了**deepstack的有效性，能够整合丰富的视觉信息，从而有效增强细粒度视觉理解能力。**

ViT	Clip Bench							VLM Bench				
	ImageNet-1K	ImageNet-V2	ImageNet-A	ImageNet-R	ImageNet-S	ObjectNet	Omni	OCRB	AI2D	RLWDQA	InfoVQA	Omni
SigLIP-2	84.2	78.6	87.0	96.1	76.2	79.9	36.9	77.2	74.1	58.7	65.3	50.1
Qwen3-ViT	84.6	78.8	87.1	95.7	74.5	81.0	45.5	78.7	76.2	66.1	67.0	53.0

Needle-in-a-Haystack

评估模型处理长上下文的能力，在 Qwen3-VL-235B-A22B-Instruct 上构建了一个视频版的“大海捞针”评估任务。**在视频中插入一帧含有答案的内容，让模型长视频中准确定位目标帧并回答相关问题**。视频以 1 FPS 的频率统一采样，帧分辨率动态调整以保持恒定的视觉标记预算。

- 在256k token上下文，对应30分钟的视频上，准确率100%。
- 在使用YaRN将序列长度外推到1M token上下文，对应2小时的视频上，准确率99.5%。证明了该模型强大的长序列建模能力。