面经—模型篇

QWen25VL

分为几个部分：模型整体设计、关键技术点、相较 Qwen2 的改进、应用场景，以及可能的追问点。

---

1. 模型整体设计

- **架构**：
  Qwen2.5-VL 基于 **Qwen2.5 LLM 解码器**，前面加多模态输入编码器（图像/视频 → patch embedding → 投影到 LLM 词向量空间）。采用统一的 **decoder-only Transformer** 来进行跨模态建模。
- **输入模态**：
  - 文本：常规分词 + RoPE。
  - 图像：切分成 patch（类似 ViT），通过线性投影映射到 token space。
  - 视频：在图像 patch 的基础上增加时间维度（frame patch embedding + temporal merging）。

---

2. 关键技术点

(1) 图像编码

- 使用 **Vision Transformer (ViT-Large/ViT-Huge)** 预训练 backbone，将图像分成 patch，每个 patch 转换成 embedding。
- **Patch Token 化**：比如一张 224×224 输入，切 14×14 patch，共 256 个 token。
- 输入 Qwen2.5 的是投影后的 token，保证与文本 embedding 维度一致。

(2) 视频处理

- **帧采样**：动态 FPS（比如从视频中采 8–32 帧），保证高效和覆盖性。
- **Temporal Merging**：Conv3D 或时序池化，将帧序列信息聚合。
- **时间 ID（Time IDs）**：在 RoPE 的基础上增加时间位置编码，确保帧间顺序信息。

(3) 位置编码 — M-RoPE

- **文本**：普通 RoPE（旋转位置编码）。
- **图像**：2D-RoPE（对 height/width 各自编码，保留空间结构）。
- **视频**：在 2D-RoPE 基础上再加时间 ID，实现三维位置编码。

面试时举例：

> 假设输入一段 80×80 的图像，如果 patch size=14，能得到 40×40≈1600 个 patch，每个 patch 投影成一个 token，最后就是约 1.6k token；加上文本 token，一起进入 LLM。

---

3. 相较 Qwen2 的改进

- **更强的多模态能力**：Qwen2.5-VL 在图像 + 视频上同时支持，并优化了 **空间 + 时间位置编码**。
- **输入粒度更细**：支持高分辨率图像，能处理小目标与细节。
- **跨模态对齐**：采用 M-RoPE 解决长序列对齐问题，在长视频场景中效果更稳定。
- **推理增强**：在 VL 任务中支持 Chain-of-Thought 推理（图文问答、视频理解）。
- **训练数据**：更大规模的图文对齐数据 + 合成数据增强。

---

4. 应用场景

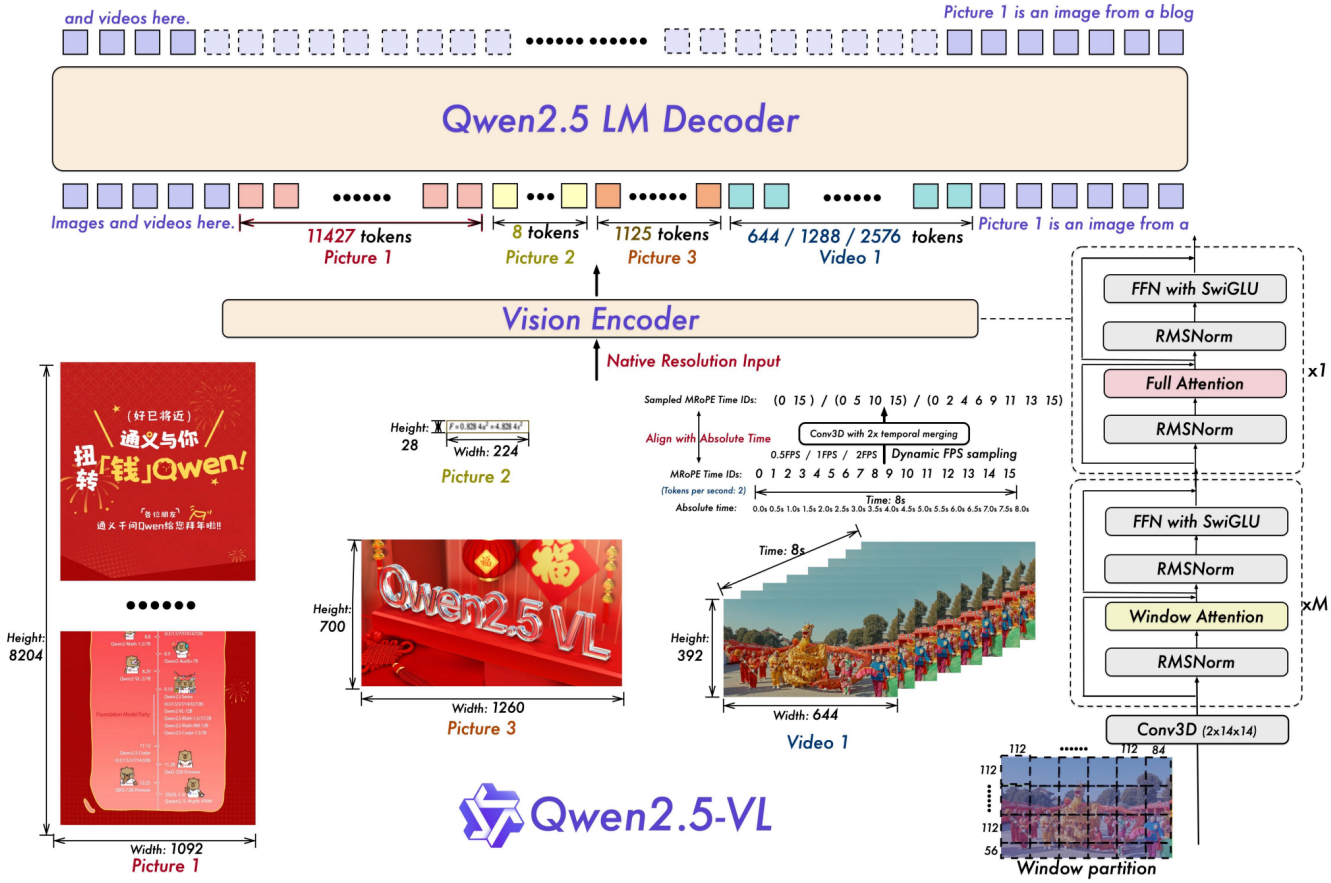
- **图文问答**：OCR 场景，复杂表格/图像理解。
- **视频理解**：多帧输入，问答/摘要/时序推理。
- **多模态推理**：比如“根据视频判断人物动作是否连续”、“从图表里读出趋势”。
- **Agent 场景**：可以结合视觉输入执行复杂任务（如操作界面）。

---

5. 面试可能追问点

1. **Q：Qwen2.5-VL 如何解决长视频输入时 token 爆炸的问题?**
    - A：采用 **动态帧采样 + temporal merging**，减少冗余帧，同时用 M-RoPE 保持时间顺序。

2. **Q：为什么要用 M-RoPE，而不是单纯的 RoPE?**
    - A：RoPE 是一维的，无法同时表达图像的二维结构和视频的时间维度。M-RoPE 扩展到 (H, W, T)，确保不同模态在同一解码器中能对齐。
3. **Q：Qwen2.5-VL 相比 Qwen2-VL 的主要提升点?**
    - 更细粒度 patch，支持高分辨率；
    - M-RoPE 时空编码；
    - 增强视频能力；
    - 更大规模多模态训练集。
4. **Q：如果输入一张 80×80 的图像，会产生多少 token?**
    - 举例：假设 patch size=16 → (80/16)²=25 个 patch → 25 token；
    - patch size=8 → 100 token；
    - patch size=4 → 400 token；
    - 最终 token 数取决于 patch size 和分辨率。



Qwen2.5-VL 通过增强的视觉识别、精确的物体定位、强大的文档解析和长视频理解，在理解和与世界互动方面实现了重大飞跃。Qwen2.5-VL 的一大亮点是能够使用边界框或点精确定位物体。它能够从发票、表单和表格中提取强大的结构化数据，并对图表、示意图和布局进行详细分析。为了处理复杂的输入，Qwen2.5VL 引入了动态分辨率处理和绝对时间编码，使其能够处理不同大小的图像和长达数小时的长视频，并实现秒级事件定位。这使得该模型能够原生地感知空间尺度和时间动态，而无需依赖传统的归一化技术。通过从头开始训练原生动态分辨率 Vision

Transformer (ViT) 并结合窗口注意力机制，我们在保持原生分辨率的同时显著降低了计算开销。因此，Qwen2.5-VL 不仅在静态图像和文档理解方面表现出色，而且作为交互式视觉代理，能够在实际场景（例如操作计算机和移动设备）中进行推理、工具使用和任务执行。该模型实现了跨领域的强大泛化能力，无需针对特定任务进行微调

我们的贡献主要有四方面：（1）在视觉编码器中引入窗口注意力机制，优化推理效率；（2）引入动态 FPS 采样，将动态分辨率扩展到时间维度，实现跨不同采样率的全面视频理解；（3）在时间域中，我们通过与绝对时间对齐的方式升级了 MRoPE，从而促进更复杂的时间序列学习；（4）我们大力投入高质量数据用于预训练和监督微调，将预训练语料库从 1.2 万亿个 token 进一步扩展至 4.1 万亿个 token。

Qwen2.5-VL 的闪光特性如下：

- 强大的文档解析能力：Qwen2.5-VL 将文本识别升级为全文档解析，能够出色地处理多场景、多语言以及各种内置（手写、表格、图表、化学式、乐谱）文档。
- 跨格式的精确物体接地：Qwen2.5-VL 提高了检测、指向和计数物体的准确性，适应绝对坐标和 JSON 格式，实现高级空间推理。
- 超长视频理解和细粒度视频接地：我们的模型将原生动态分辨率扩展到时间维度，增强了理解数小时视频的能力，同时在数秒内提取事件片段。 -
- 增强计算机和移动设备的代理功能：利用先进的接地技术，推理和决策能力，通过智能手机和计算机上的卓越代理功能增强模型。

太好了，这张 Figure 1 其实是在“把整条多模态流水线展开给你看”。我按“从输入到输出”的顺序把每个标注的意思、它解决的问题，以及跟 Qwen2-VL 相比的新点讲清楚（配上论文原文要点作为旁证）。

① 输入侧：原生分辨率 + 动态帧率

- 左下角展示了不同尺寸的图片（海报、长图、表单等）和一个视频片段。模型**不把它们强行缩成同一分辨率**，而是按**原生分辨率**切成 patch 后再编码，因而**得到的 token 序列长度因图/视频大小而异**（图上方“Picture 1/2/3/Video 1 … → 不同 token 数”就是这个含义）。这解决了“高分辨率细节被压缩丢失”的问题。
- 对视频，支持**动态 FPS 采样**（图里标出 0.5/1/2/…/15 FPS 之类），帧数多少、采样速率不同，映射出来的 token 长度也跟着变。这样既能覆盖快慢节奏的视频，又能控制计算量。

② Vision Encoder（右侧模块）

- 这是一个**重构过的 ViT**：多数层使用**窗口注意力（Window Attention）**来把二次方复杂度降到**近似线性**，只在少数层用全局注意力（图示 “Window Attention ×M + 少量 Full Attention”）。论文里明确：最大窗口对应 **112×112 像素（即 8×8 个 14×14 的 patch）**，仅**4 层**用全注意力，其余是窗口注意力。这样既保留原生分辨率细节，又把大图的计算压住。
- **规范化/激活**：用 **RMSNorm + SwiGLU**，风格上向 LLM 靠拢，提升效率与稳定性（图里每个 Block 右边的 RMSNorm/FFN with SwiGLU）。
- **视频的时序聚合**：图里最底部有 **Conv3D(2×14×14)**，表示把**相邻两帧**在 patch 级做 3D 卷积/合并，减少送进 LLM 的时序 token 数量，同时保留局部时间信息。

- **输入预处理**：高宽会被对齐到 **28 的倍数**，随后以**步长 14**切 patch（也就是“patch size=14”）；这些细节对应图里 Vision Encoder 输入前的“Native Resolution Input”标注。

③ 时空位置编码：MRoPE 与“绝对时间”

- 文本/图像/视频统一用 **MRoPE**：图像用**二维 RoPE（高/宽各自旋转编码）**；视频在此基础上加**时间维**。这让同一解码器里，空间与时间的位置都能对齐。
- **关键升级点**：Qwen2.5-VL 把**时间 ID 与“绝对时间”对齐**——图里时间轴上用“Absolute Time”标注。意思是：不同 FPS 的视频，时间 ID 的间隔**按真实时间**来刻度，而不是仅按“第几帧”。这样模型能更稳地感知“节奏/速度”，在**时间定位（找事件发生的秒级位置）**上更准。

④ 视觉→语言的“合并器”（Merger）

- 大图会产生很多 patch 特征。为**降低送入 LLM 的序列长度**，图像特征会先做**空间邻近的 4 个 patch 分组**，拼接后通过**两层 MLP**投到与文本**同维度**的嵌入，再送入解码器（这一步在图里没有单独画出来的盒子，但对应“Vision-Language Merger”的设计）。这一步就是“既压缩计算、又保留细节”的关键。

⑤ 顶部：Qwen2.5 LLM Decoder（统一解码器）

- 所有多模态 token（文本 + 图像/视频压缩后的视觉 token）**串接**到同一个 **Qwen2.5 解码器**里进行跨模态建模与生成。图里最上条长条就是这个“统一的 decoder-only Transformer”。

这张图想表达的三件事（面试话术）

1. **原生分辨率与可变序列**：不同尺寸/时长/采样率→不同长度的视觉 token；模型不牺牲细节而是让序列长度自然变化。好处：小目标、票据表格、长图都更准。
2. **高效的视觉编码器**：窗口注意力 + 少量全局注意力 + RMSNorm/SwiGLU + 3D 时序合并 → 在保持分辨率/时序信息的同时把计算打下来。
3. **时间对齐的新位置编码**：MRoPE 的时间轴与**绝对时间对齐**，跨 FPS 仍能稳定理解“快/慢”和“具体发生在第几秒”。这直接提升**时间定位、密集描述**等视频任务。

你可能会被追问的细节（速答要点）

- **patch 与窗口的典型尺寸?**
  patch size=14；窗口最大覆盖 112×112（约 8×8 个 patch）；仅 4 层用全局注意力。
- **为什么还要一个 Merger?**
  大图/长视频会产生超长序列。先按 4 邻域合并+MLP 投影到文本维度，可**动态压缩**视觉序列再送入 LLM，算得起还尽量不丢局部结构。
- **相比 Qwen2-VL 的显著升级?**
  动态 FPS + **绝对时间对齐的 MRoPE**（跨 FPS 更稳），**窗口注意力**的高效 ViT，从训练/推理两端降低成本并强化长视频与文档细粒度理解。

如果你愿意，我可以基于开源仓库再把**Vision Encoder（窗口注意力 + 3D 时序合并）**、**MRoPE-Absolute Time**、**Merger**这三处的核心代码路径/调用关系画一张“小蓝图”，并配上“80×80/224×224/长图/视频”的**token 粗估公式**，方便你在面试里又快又准地讲清楚。

Q&A

1.VIT内容

2.M-ROPE

3.RMSNorm

4. SwiGLU