DeepSeek-OCR 2: 视觉因果流

Haoran Wei, Yaofeng Sun, Yukun Li

DeepSeek-AI

Abstract

我们提出 DeepSeek-OCR 2，旨在探索一种新型编码器——DeepEncoder V2 的可行性，该编码器能够依据图像语义动态重排视觉 token。传统的视觉语言模型（VLMs）在将视觉 token 输入大语言模型（LLMs）时，一贯采用固定的光栅扫描顺序（从左上至右下）配合固定的位置编码进行处理。然而，这与人类视觉感知相悖，人类视觉遵循由内在逻辑结构驱动的、灵活且语义连贯的扫描模式。尤其在处理布局复杂的图像时，人类视觉展现出受因果信息引导的顺序处理特性。受此认知机制启发，DeepEncoder V2 旨在为编码器赋予因果推理能力，使其能够在基于 LLM 的内容理解之前智能地重排视觉 token。本研究探索了一种全新范式：是否可通过两个级联的 1D 因果推理结构有效实现 2D 图像理解，从而提供一种有望实现真正 2D 推理的新架构思路。代码与模型权重已公开于 http://github.com/deepseek-ai/DeepSeek-OCR-2。

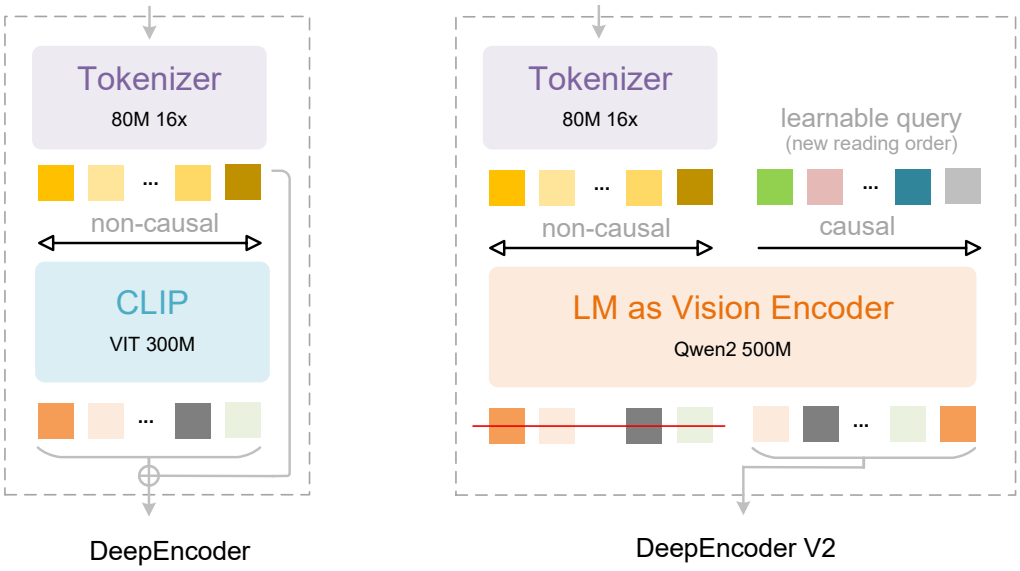

图 1 | 我们将 DeepEncoder 中的 CLIP 组件替换为类 LLM 架构。通过自定义注意力掩码，视觉 token 采用双向注意力，而可学习查询（learnable queries）采用因果注意力。因此，每个查询 token 能够关注所有视觉 token 及前置查询，从而实现对视觉信息的渐进式因果重排。

目录

1. 引言

人类视觉系统与基于 Transformer 的视觉编码器 [? ?] 高度相似: 中央凹注视点充当视觉 token，局部清晰且具备全局感知能力。然而，与现有编码器严格从左上方到右下方扫描 token 不同，人类视觉遵循由语义理解引导的因果驱动流。以追踪螺旋线为例——我们的眼动遵循内在逻辑，即每一次后续的注视都因果性地依赖于前一次注视。同理，模型中的视觉 token 应被选择性处理，其排序应高度依赖于视觉语义而非空间坐标。

这一洞察促使我们从根本上重新审视视觉语言模型（VLMs）的架构设计，尤其是编码器组件。LLMs 本质上是在 1D 序列数据上训练的，而图像是 2D 结构。按照预定义的光栅扫描顺序直接展平图像块会引入不必要的归纳偏置，从而忽略语义关系。为解决这一问题，我们提出 DeepSeek-OCR 2，并采用新型编码器设计——DeepEncoder V2，以迈向更拟人化的视觉编码。沿袭 DeepSeek-OCR [?] 的工作，我们将文档阅读作为主要实验测试平台。文档带来了丰富的挑战，包括复杂的布局顺序、复杂的公式和表格。这些结构化元素内在蕴含着因果视觉逻辑，需要复杂的推理能力，这使得文档 OCR 成为验证我们方法的理想平台。

我们的主要贡献有三方面:

首先，我们提出 DeepEncoder V2，具备几项关键创新: (1) 如图 1 所示，我们将 DeepEncoder [?] 中的 CLIP [?] 组件替换为紧凑的 LLM [?] 架构，以实现视觉因果流; (2) 为实现并行处理，我们引入了可学习查询 [?]，称为因果流 token，并将视觉 token 作为前缀前置——通过自定义注意力掩码，视觉 token 保持全局感受野，而因果流 token 可获得视觉 token 重排能力; (3) 我们保持因果 token 与视觉 token 的数量相等（包含填充和边框等冗余），以提供充足的重新注视（re-fixation）容量; (4) 仅将因果流 token（即编码器输出的后半部分）输入 LLM [?] 解码器，从而实现级联的因果感知视觉理解。

其次，基于 DeepEncoder V2，我们提出 DeepSeek-OCR 2，在保留 DeepSeek-OCR 图像压缩率与解码效率的同时，实现了显著的性能提升。我们将输入 LLM 的视觉 token 数量限制在 256 至 1120 之间。下限（256）对应 DeepSeek-OCR 对 1024×1024 图像的 token 化结果，而上限（1120）则与 Gemini-3 pro [?] 的最大视觉 token 预算相匹配。该设计使 DeepSeek-OCR 2 既成为用于研究探索的新型 VLM 架构，也成为为 LLM 预训练生成高质量训练数据的实用工具。

最后，我们为采用语言模型架构作为 VLM 编码器提供了初步验证——这是一条通往统一全模态编码的有前景的路径。该框架仅需配置模态特定的可学习查询，即可实现跨多种模态（图像、音频、文本 [?]）的特征提取与 token 压缩。关键在于，它能够自然继承 LLM 社区的高级基础设施优化成果，包括混合专家（MoE）架构、高效注意力机制 [?] 等。

综上所述，我们为 DeepSeek-OCR 2 提出了 DeepEncoder V2，采用专用注意力机制有效建模文档阅读的因果视觉流。与 DeepSeek-OCR 基线相比，DeepSeek-OCR 2 在 OmniDocBench v1.5 [?] 上实现了 3.73% 的性能提升，并在视觉阅读逻辑方面取得了显著进展。

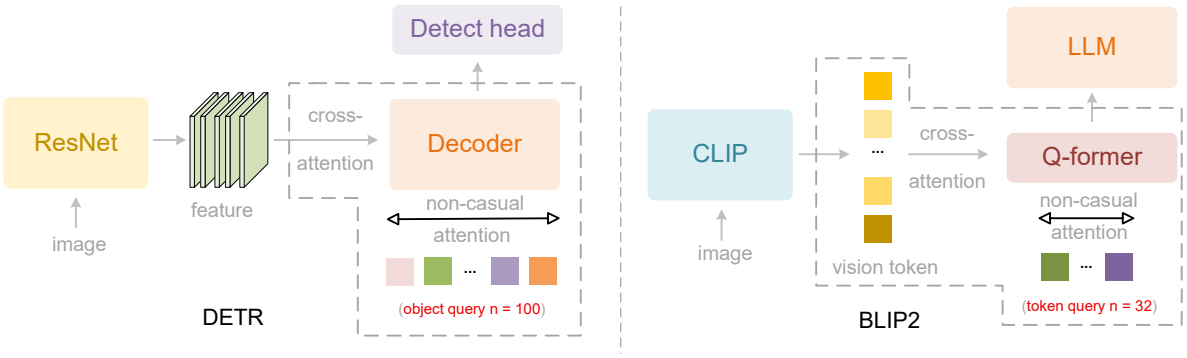

图 2 | 本图展示了两个采用并行化查询的计算机视觉模型：用于目标检测的 DETR 解码器 [?] 和用于视觉 token 压缩的 BLIP2 Q-former [?]。两者均在查询之间采用双向自注意力机制。

2. 相关工作

2.1. 解码器中的并行化查询

DETR [?] 开创了将 Transformer 架构引入目标检测的先河，从根本上打破了传统检测范式 [? ?]。为克服 Transformer 块中串行解码的效率瓶颈，DETR 引入了预设的并行化可学习查询——一组 100 个对象查询，通过训练编码形状和位置等对象先验。这些查询通过交叉注意力机制与特征图 [?] 交互，同时通过自注意力机制在查询之间进行双向信息交换。DETR 确立了一种基础范式，使 Transformer 能够处理并行化 token。自此，对象查询设计已成为后续基于 Transformer 的检测方法 [? ?] 中事实上的标准架构组件。

2.2. 投影器中的并行化查询

近年来，视觉语言模型 [? ? ? ?] 发展迅速，其架构逐渐向编码器-投影器-LLM 范式收敛。投影器将视觉 token 与 LLM 的嵌入空间对齐，作为使 LLM 理解视觉内容的关键桥梁。BLIP-2 [?] 中引入的 Q-former 是投影器设计的有效典范，它采用可学习查询进行视觉 token 压缩。Q-former 采用类 BERT [?] 架构，并借鉴了 DETR 的对象查询 [?] 思想，利用 32 个可学习查询通过交叉注意力与数百个 CLIP [?] 视觉 token 交互。这些压缩后的查询表示随后被输入 LLM，实现了从视觉空间到语言空间的有效映射。Q-former 的成功表明，并行化可学习查询不仅在检测任务的特征解码中有效，在多模态对齐的 token 压缩中同样行之有效。

2.3. 基于 LLM 的多模态初始化

在大规模互联网数据上训练的大语言模型（LLM）已被证明可作为多模态模型的有效初始化方法。Pang 等人 [?] 表明，冻结的 LLM Transformer 层能够提升视觉判别任务的性能。此外，视觉领域的无编码器或轻量级编码器模型（如 Fuyu [?] 和 Chameleon [?]），以及语音领域的 VALL-E [?]，进一步验证了 LLM 预训练权重在多模态初始化方面的潜力。

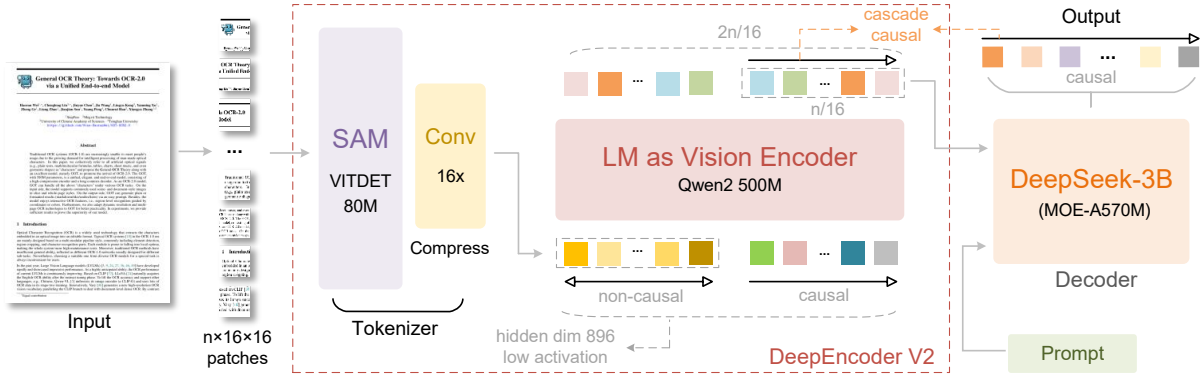

图 3 | DeepSeek-OCR 2 采用了源自 DeepEncoder 的视觉 token 压缩机制，使用一个 80M 参数的图像压缩器，将视觉 token 数量减少 16 倍。DeepEncoder V2 的不同之处在于，用紧凑的语言模型架构替换了 DeepEncoder 中的 CLIP 模块。通过自定义的注意力掩码，这种类语言模型的视觉编码器在获得 CLIP 的知识压缩能力的同时，实现了对视觉序列的因果建模。

3. 方法

3.1. 架构

如图 3 所示，DeepSeek-OCR 2 继承了 DeepSeek-OCR 的整体架构，由编码器和解码器组成。编码器将图像离散化为视觉 token，解码器则基于这些视觉 token 和文本提示生成输出。关键区别在于编码器：我们将 DeepEncoder 升级为 DeepEncoder V2，它在保留前代所有能力的同时，通过新颖的架构设计引入了因果推理机制。我们将在下文详细阐述 DeepSeek-OCR 2 的具体细节。

3.2. DeepEncoder V2

传统编码器是一个重要组件，它通过注意力机制提取和压缩图像特征，其中每个 token 都会关注所有其他 token，从而实现类似于人类中央凹视觉和周边视觉的全图像感受野。然而，将二维图像块展平为一维序列会通过面向文本的位置编码（如 RoPE [?]）引入严格的顺序偏差。这与自然的视觉阅读模式相悖，尤其是在光学文本、表单和表格中的非线性布局。

3.2.1. 视觉分词器

DeepEncoder V2 的第一个组件是视觉分词器。遵循 DeepEncoder 的设计，我们采用了一种结合 80M 参数 SAM-base [?] 与两个卷积层 [?] 的架构。最终卷积层的输出维度从 DeepEncoder 中的 1024 降低至 896，以与后续流水线对齐。需要注意的是，这种基于压缩的分词器并非必需，可以替换为简单的 patch 嵌入。我们保留它是因为它通过窗口注意力机制以极少的参数量实现了 16 倍的 token 压缩 [? ? ? ?]，显著降低了后续全局注意力模块的计算成本和激活内存。此外，其参数量（约 80M）与 LLM 中文本输入嵌入通常使用的约 100M 参数量相当。

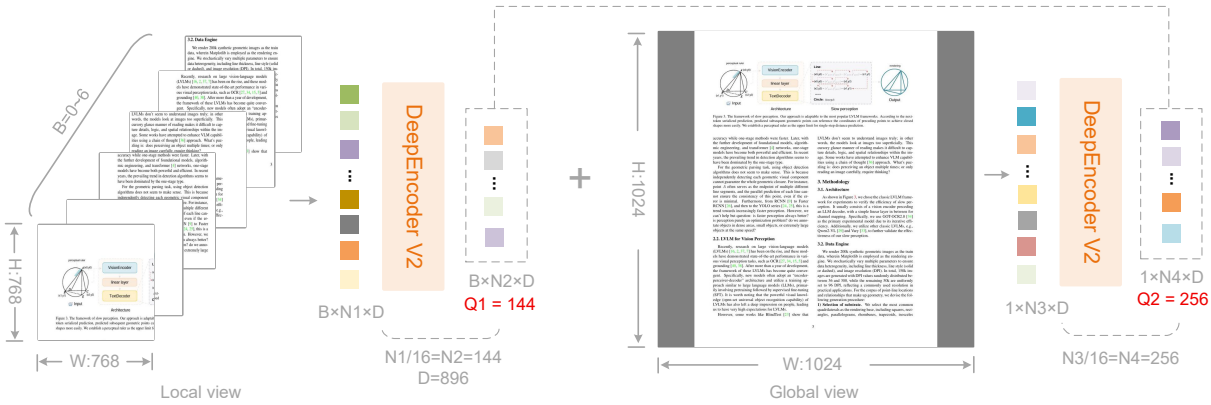

图 4 | DeepEncoder V2 中的 token 数量计算。DeepEncoder V2 采用包含 0 到 6 个局部视图的多裁剪策略，每张图像输出 256 至 1120 个 token。当局部视图为 0 时，仅全局视图产生 256 个 token；当局部视图为 6 时，数量达到 1120（6×144+256）。

3.2.2. 作为视觉编码器的语言模型

在 DeepEncoder 中，视觉分词器后接一个 CLIP ViT 用于压缩视觉知识。DeepEncoder V2 将该组件重新设计为具有双流注意力机制的类 LLM 架构。视觉 token 使用双向注意力以保留 CLIP 的全局建模能力，而新引入的因果流查询则采用因果注意力。这些可学习查询作为后缀附加在视觉 token 之后，其中每个查询都会关注所有视觉 token 和前面的查询。通过保持查询与视觉 token 数量相等，该设计在不改变 token 数量的前提下，对视觉特征施加了语义排序和蒸馏。最后，仅将因果查询的输出输入到 LLM 解码器中。

我们使用 Qwen2-0.5B [**?**] 实例化该架构，其 500M 参数与 CLIP ViT（300M）相当，且不会引入过多的计算开销。带有视觉 token 前缀拼接的仅解码器架构被证明至关重要：在 mBART 风格 [**?**] 的编码器-解码器结构中使用交叉注意力的额外实验均未能收敛。我们假设这种失败源于视觉 token 被隔离在独立编码器中时交互不足。相比之下，前缀设计使视觉 token 在所有层中保持活跃，促进了与因果查询之间有效的视觉信息交换。

该架构实际上建立了两阶段级联因果推理：编码器通过可学习查询对视觉 token 进行语义重排序，而 LLM 解码器则对排序后的序列执行自回归推理。与通过位置编码施加严格空间顺序的传统编码器不同，我们的因果排序查询能够适应平滑的视觉语义，同时自然契合 LLM 的单向注意力模式。该设计有望弥合二维空间结构与一维因果语言建模之间的差距。

3.2.3. 因果流查询

如前所述，因果查询 token 的数量等于视觉 token 的数量，计算公式为 $\frac{W\times H}{16^2\times 16}$，其中 W 和 H 表示输入到编码器的图像宽度和高度。为避免为不同分辨率维护多组查询集，我们采用在预定义分辨率下固定查询配置的多裁剪策略。

具体而言，全局视图使用 1024×1024 的分辨率，对应 256 个查询嵌入，记为 $\text{query}_{\text{global}}$。局部裁剪采用 768×768 的分辨率，裁剪数量 k 范围为 0 到 6（当图像两个维度均小于 768 时

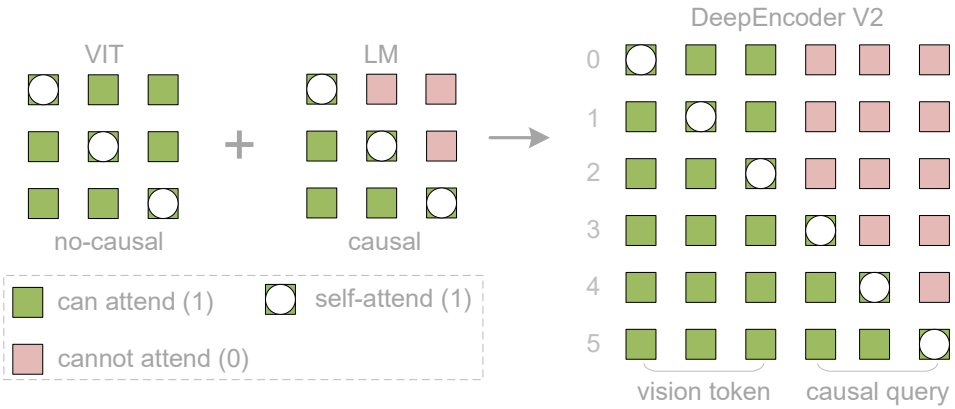

图 5 | DeepEncoder V2 的注意力掩码架构。双向掩码（视觉 token，类 ViT）与因果三角掩码（流 token，类 LLM 解码器）的拼接。

不进行裁剪）。所有局部视图共享一组统一的 144 个查询嵌入，记为 $\text{query}_{\text{local}}$。因此，输入到 LLM 的重排序视觉 token 总数为 $k \times 144 + 256$，范围在 $[256, 1120]$ 之间。该最大 token 数量（1120）低于 DeepSeek-OCR 的 1156（Gundam 模式），并与 Gemini-3-Pro 的最大视觉 token 预算相匹配。

3.2.4. 注意力掩码

为了更好地说明 DeepEncoder V2 的注意力机制，我们在图 5 中可视化了注意力掩码。该注意力掩码由两个不同的区域组成。左侧区域对原始视觉 token 应用双向注意力（类似 ViT），允许完全的 token 间可见性。右侧区域对因果流 token 采用因果注意力（三角掩码，与仅解码器 LLM 相同），其中每个 token 仅关注前面的 token。这两个组件沿序列维度拼接，以构建 DeepEncoder V2 的注意力掩码（M），如下所示：

$$M = \begin{bmatrix} \mathbf{1}_{m \times m} & \mathbf{0}_{m \times n} \\ \mathbf{1}_{n \times m} & \text{LowerTri}(n) \end{bmatrix}, \quad \text{where } n = m \tag{1}$$

其中 n 为因果查询 token 的数量，m 表示传统视觉 token 的数量，LowerTri 表示下三角矩阵（对角线及其下方为 1，上方为 0）。

3.3. DeepSeek-MoE 解码器

由于 DeepSeek-OCR 2 主要侧重于编码器改进，我们未对解码器组件进行升级。遵循这一设计原则，我们保留了 DeepSeek-OCR 的解码器 – 一个拥有 3B 参数、约 500M 激活参数的 MoE 结构。DeepSeek-OCR 2 的核心前向传播过程可表述为：

$$\mathbf{O} = \mathcal{D}\left(\pi_Q\left(\mathcal{T}^L\left(\mathcal{E}(\mathbf{I}) \oplus \mathbf{Q}_0; \mathbf{M}\right)\right)\right) \tag{2}$$

其中 $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ 为输入图像，$\mathcal{E}$ 为将图像映射为 m 个视觉 token $\mathbf{V} \in \mathbb{R}^{m\times d}$ 的视觉分词器，$\mathbf{Q}_0 \in \mathbb{R}^{n\times d}$ 为可学习的因果查询嵌入，$\oplus$ 表示序列拼接，$\mathcal{T}^L$ 表示带有掩码注意力的 L 层 Transformer，$\mathbf{M} \in \{0,1\}^{2n\times 2n}$ 为公式 1 中定义的块因果注意力掩码，π_Q 为提取最后 n 个 token 的投影算子（即 $\mathbf{Z} = \mathbf{X}_{m+1:m+n}$），$\mathcal{D}$ 为语言解码器，$\mathbf{O} \in \mathbb{R}^{n\times|\mathcal{V}|}$ 为基于 LLM 词表的输出 logits。

4. 实验设置

4.1. 数据引擎

DeepSeek-OCR 2 采用与 DeepSeek-OCR 相同的数据源，包括 OCR 1.0、OCR 2.0 [? ? ?] 以及通用视觉数据 [?]，其中 OCR 数据占训练混合数据的 80%。我们还引入了两项修改：(1) 对 OCR 1.0 数据采用更均衡的采样策略，按内容类型（文本、公式、表格）以 3:1:1 的比例划分页面；(2) 通过合并语义相似的类别（例如统一 “图注” 和 “图题”）对布局检测标签进行细化。鉴于这些差异极小，我们将 DeepSeek-OCR 视为有效的对比基线。

4.2. 训练流程

我们分三个阶段训练 DeepSeek-OCR 2：(1) 编码器预训练，(2) 查询增强，以及 (3) 解码器专项训练。第一阶段使视觉分词器和类 LLM 编码器获得特征提取、token 压缩和 token 重排序的基础能力。第二阶段在增强视觉知识压缩的同时，进一步强化编码器的 token 重排序能力。第三阶段冻结编码器参数并仅优化解码器，从而在相同 FLOPs 下实现更高的数据吞吐量。

4.2.1. 训练 *DeepEncoder V2*

遵循 DeepSeek-OCR 和 Vary [?] 的做法，我们使用语言建模目标训练 DeepEncoder V2，将编码器与轻量级解码器 [?] 耦合，通过下一个 token 预测进行联合优化。我们采用两个数据加载器，分辨率分别为 768×768 和 1024×1024。视觉分词器从 DeepEncoder 初始化，类 LLM 编码器从 Qwen2-0.5B-base [?] 初始化。预训练完成后，仅保留编码器参数用于后续阶段。我们使用 AdamW [?] 优化器，学习率按余弦策略从 1e-4 衰减至 1e-6，在 160 块 A100 GPU（20 个节点 × 8 块 GPU）上进行训练，批量大小为 640，迭代 4 万次（序列打包长度为 8K，约 1 亿个图像-文本对样本）。

4.2.2. 查询增强

在 DeepEncoder V2 预训练完成后，我们将其与 DeepSeek-3B-A500M [? ?] 集成，作为最终的流水线。我们冻结视觉分词器（SAM-conv 结构），同时联合优化 LLM 编码器和 LLM 解码器，以增强查询表示。在此阶段，我们通过多裁剪策略将两种分辨率统一到一个数据加载器中。我们采用 4 阶段流水线并行：视觉分词器（PP0）、类 LLM 编码器（PP1）以及 DeepSeek-LLM 层（PP2-3 每个阶段 6 层）。使用 160 块 GPU（每块 40GB），我们配置了 40 个数据并行副本（每个副本 4 块 GPU），并使用相同的优化器，学习率从 5e-5 衰减至 1e-6，全局批量大小为 1280，训练 1.5 万次迭代。

4.2.3. LLM 持续训练

为了快速消耗训练数据，在此阶段我们冻结所有 DeepEncoder V2 参数，仅更新 DeepSeek-LLM 参数。该阶段加速了训练（在相同全局批量大小下训练速度提升一倍以上），同时帮助 LLM 更好地理解 DeepEncoder V2 重排序后的视觉 token。延续第二阶段，我们在本阶段执行另一次学习率衰减，从 1e-6 降至 5e-8，训练 2 万次迭代。

表 1 | 在 OmniDocBench v1.5 上的文档阅读综合评估。V-token^{max} 表示该基准中每页使用的最大视觉 token 数量。R-order 表示阅读顺序。除 DeepSeek OCR 和 DeepSeek OCR 2 外，本表中所有其他模型的结果均来源于 OmniDocBench 仓库。

模型	V-token^{max} ↓	整体 ↑	文本 ^{Edit} ↓	公式 ^{CDM} ↑	表格 ^{TEDs} ↑	表格 ^{TEDS_s} ↑	阅读顺序 ^{Edit} ↑
流水线							
Marker-1.8.2 [?]	-	71.30	0.206	76.66	57.88	71.17	0.250
MinerU2-pp [?]	-	71.51	0.209	76.55	70.90	79.11	0.225
Dolphin [?]	-	74.67	0.125	67.85	68.70	77.77	0.124
Dolphin-1.5 [?]	-	83.21	0.092	80.78	78.06	84.10	0.080
PP-StructureV3 [?]	-	86.73	0.073	85.79	81.68	89.48	0.073
MonkeyOCR-pro-1.2B [?]	-	86.96	0.084	85.02	84.24	89.02	0.130
MonkeyOCR-3B [?]	-	87.13	0.075	87.45	81.39	85.92	0.129
MonkeyOCR-pro-3B [?]	-	88.85	0.075	87.25	86.78	90.63	0.128
MinerU2.5 [?]	-	90.67	0.047	88.46	88.22	92.38	0.044
PaddleOCR-VL [?]	-	92.86	0.035	91.22	90.89	94.76	0.043
端到端模型							
OCRFlux [?]	>6000	74.82	0.193	68.03	75.75	80.23	0.202
GPT-4o [?]	-	75.02	0.217	79.70	67.07	76.09	0.148
InternVL3 [?]	>7000	80.33	0.131	83.42	70.64	77.74	0.113
POINTS-Reader [?]	>6000	80.98	0.134	79.20	77.13	81.66	0.145
olmOCR [?]	>6000	81.79	0.096	86.04	68.92	74.77	0.121
InternVL3.5-241B [?]	>7000	82.67	0.142	87.23	75.00	81.28	0.125
MinerU2-VLM [?]	>7000	85.56	0.078	80.95	83.54	87.66	0.086
Nanonets-OCR-s [?]	>7000	85.59	0.093	85.90	80.14	85.57	0.108
Qwen2.5-VL-72B [?]	>6000	87.02	0.094	88.27	82.15	86.22	0.102
Gemini-2.5 Pro[?]	-	88.03	0.075	85.82	85.71	90.29	0.097
dots.ocr [?]	>6000	88.41	0.048	83.22	86.78	90.62	0.053
OCRVerse [?]	>6000	88.56	0.058	86.91	84.55	88.45	0.071
Qwen3-VL-235B [?]	>6000	89.15	0.069	88.14	86.21	90.55	0.068
DeepSeek-OCR (9-crops)	1156	87.36	0.073	84.14	85.25	89.01	0.085
DeepSeek-OCR 2	1120	91.09	0.048	90.31	87.75	92.06	0.057
	↓ 36	↑ 3.73	↓ 0.025	↑ 6.17	↑ 2.5	↑ 3.05	↓ 0.028

5. 评估

我们选择 OmniDocBench v1.5 [?] 作为主要评估基准。该基准包含 1,355 页文档，涵盖中英文 9 大主要类别（包括杂志、学术论文、研究报告等）。凭借其多样化的测试样本和稳健的评估标准，OmniDocBench 为验证 DeepSeek-OCR 2 的性能（尤其是 DeepEncoder V2 的有效性）提供了有效的框架。

表 2 | OmniDocBench v1.5 中不同类别文档元素的编辑距离。V-token^{max} 表示最低的最大视觉 token 数量。

模型	V-token^{max}↓	文本 ^{Edit}↓	公式 ^{Edit}↓	表格 ^{Edit}↓	阅读顺序 ^{Edit}↓	整体 ^{Edit}↓
Gemini-3 pro [?]	1120	-	-	-	-	0.115
Seed-1.8 [?]	5120	-	-	-	-	0.106
DeepSeek-OCR	1156	0.073	0.236	0.123	0.085	0.129
DeepSeek-OCR 2	1120	0.048	0.198	0.096	0.057	0.100

5.1. 主要结果

如表 1 所示，DeepSeek-OCR 2 在使用最小视觉 token 上限（V-token^{max}）的情况下，取得了 91.09% 的先进性能。与 DeepSeek-OCR 基线相比，在相似的训练数据源下，其性能提升了 3.73%，验证了我们新设计架构的有效性。除了整体性能的提升，阅读顺序（R-order）的编辑距离（ED）也显著降低（从 0.085 降至 0.057），表明新的 DeepEncoder V2 能够基于图像信息有效地选择和排列初始视觉 token。如表 2 所示，在相似的视觉 token 预算（1120）下，DeepSeek-OCR 2（0.100）在文档解析方面比 Gemini-3 Pro（0.115）实现了更低的 ED，进一步证明我们的新模型在确保卓越性能的同时，保持了极高的视觉 token 压缩率，展现出巨大的潜力。

5.2. 提升空间

我们在 9 种文档类型上对 DeepSeek-OCR 和 DeepSeek-OCR 2 进行了详细的性能对比，发现 DeepSeek-OCR 2 仍有相当大的提升空间，如表 3 所示。在文本识别编辑距离（ED）方面，DeepSeek-OCR 2 在大多数情况下优于 DeepSeek-OCR，但也存在明显的短板，例如在报纸类别上，其 ED 大于 > 0.13。我们认为主要有两个原因：(1) 较低的视觉 token 上限可能影响对文本极其密集的报纸的识别，未来可以通过增加局部裁剪数量来简单解决；(2) 报纸数据不足——我们的训练数据仅包含 25 万相关样本，不足以训练 DeepEncoder V2 处理此类文档。当然，在阅读顺序（R-order）指标上，DeepSeek-OCR 2 全面优于 DeepSeek-OCR，进一步验证了我们视觉因果流编码器设计的有效性。

表 3 | DeepSeek-OCR 2 与 DeepSeek-OCR 在 9 种文档类型上的详细对比。R-order 表示阅读顺序。所有指标均为编辑距离，数值越低越好。

模型	编辑距离↓	PPT	学术论文	书籍	彩色教材	考试试卷	杂志	报纸	笔记	研究报告
DS-OCR	文本	0.052	0.028	0.022	0.130	0.074	0.049	0.131	0.145	0.015
	阅读顺序	0.052	0.021	0.040	0.125	0.083	0.101	0.217	0.089	0.016
DS-OCR 2	文本	0.031	0.013	0.033	0.053	0.047	0.026	0.139	0.068	0.008
	阅读顺序	0.025	0.013	0.027	0.066	0.048	0.100	0.176	0.035	0.011

表 4 | DeepSeek-OCR 与 DeepSeek-OCR 2 的生产环境性能对比。对于服务于 LLM 流水线的 OCR 模型，在生产环境中无法获取真实标签。因此，重复率构成了主要的可观测质量指标。

模型	指标	在线用户日志（图像）	预训练数据（PDF）
DeepSeek-OCR	重复率 ↓	6.25%	3.69%
DeepSeek-OCR 2		4.17% ↓ 2.08%	2.88% ↓ 0.81%

5.3. 生产环境就绪度

DeepSeek-OCR 主要服务于两个生产用例：为 DeepSeek-LLM 读取图像/文档的在线 OCR 服务，以及执行批量 PDF 处理的预训练数据流水线。我们对比了 DeepSeek-OCR 2 与 DeepSeek-OCR 的生产环境性能。由于在生产环境中无法获取真实标签，我们主要关注重复率作为关键指标。如表 4 所示，与前代模型（DeepSeek-OCR）相比，DeepSeek-OCR 2 展现出显著提升的生产环境就绪度，将在线用户日志图像的重复率从 6.25% 降至 4.17%，将 PDF 数据生产的重复率从 3.69% 降至 2.88%。这些结果进一步验证了 DeepSeek-OCR 2 架构的有效性，尤其是其逻辑视觉理解能力。

6. 讨论与未来工作

6.1. 迈向真正的二维推理

DeepSeek-OCR 2 提出了一种新颖的架构范式，将类 LLM 编码器与 LLM 解码器进行级联。这种两个一维因果推理器的级联结构为实现真正的二维推理带来了希望：编码器执行阅读逻辑推理（通过查询令牌对视觉信息进行因果重排序），而解码器则在这些因果有序表征上执行视觉任务推理。将二维理解分解为两个互补/正交的一维因果推理子任务，可能代表着迈向真正二维推理的突破。当然，实现这一目标仍任重道远。例如，为了实现视觉内容的多次复查与多跳重排序，我们可能需要比原始视觉令牌序列长得多的因果流令牌。在未来的工作中，我们将继续完善该架构，并探索其在通用视觉推理任务上的有效性。

6.2. 迈向原生多模态

DeepEncoder V2 初步验证了类 LLM 编码器在视觉任务中的可行性。更重要的是，该架构具备演变为统一全模态编码器的潜力：一个共享 Wk, Wv 投影、注意力机制与前馈网络（FFN）的单一编码器，可以通过模态特定的可学习查询嵌入来处理多种模态。此类编码器可以在同一参数空间内压缩文本、提取语音特征并重组视觉内容，其区别仅在于查询嵌入的学习权重。DeepSeek-OCR 的光学压缩代表了迈向原生多模态的初步探索，而我们相信 DeepSeek-OCR 2 的类 LLM 编码器架构标志着我们在此方向上的进一步迈进。未来，我们也将继续探索通过该共享编码器框架集成更多模态的可能性。

7. 结论

在本技术报告中，我们介绍了 DeepSeek-OCR 2，这是 DeepSeek-OCR 的一次重大升级，它在保持高视觉令牌压缩率的同时，实现了显著的性能提升。这一进步得益于新提出的 DeepEncoder V2，它通过融合双向与因果注意力机制，隐式地提炼了对视觉世界的因果理解，从而赋予视觉编码器因果推理能力，并显著提升了视觉阅读逻辑。

尽管光学文本阅读（尤其是文档解析）代表了 LLM 时代最具实用价值的视觉任务之一，但它仅构成了更广阔的视觉理解图景中的一小部分。展望未来，我们将完善并适配该架构以应用于更多样化的场景，深入探索，迈向更全面的多模态智能愿景。