Chain-of-Visual-Thought: Teaching VLMs to See and Think Better with Continuous Visual Tokens

周宇露 | 今天修改

Chain-of-Visual-Thought: Teaching VLMs to See and Think Better with Continuous Visual Tokens

Yiming Qin[1] Bomin Wei[2] Jiaxin Ge[1] Konstantinos Kallidromitis[3]
Stephanie Fu[1] Trevor Darrell[1] XuDong Wang[1†]
[1]UC Berkeley [2]UCLA [3]Panasonic AI Research

Project Page: https://wakalsprojectpage.github.io/covt-website/

目前的视觉语言模型擅长于在语言空间中进行推理，但对于需要密集的视觉感知的理解的任务，例如空间推理和几何感知，表现得不好。

这些局限性源于以下两点：

- **语言空间的局限性**：当连续的视觉信息被投影到离散的文本空间时，丰富的视觉表征，例如边界、布局、深度和几何图形，会丢失或不能很好地表示。
- **错误累积与训练偏差**：纯文本推理链过长易导致早期错误累积。同时，现有模型的监督信号主要由文本主导，缺乏提取底层知觉特征（如边缘、深度）的动力。

纯文本的cot是使用语言表述连续的空间和几何关系，这会带来信息损失，可能会误导甚至降低视觉推理性能。比如说Qwen3-VL-Think在空间理解的bench上比instruct 模型低5%。

那么一个自然的解决方法就是调用外部工具来增强vlm,补充细粒度的视觉信息。--->更多的gpu计算消耗，以及受外部工具的限制。

另一种方式是在思考过程中裁剪或者生成图像。 --->本质还是将图像投影到文本空间，丢失了密集的视觉信息。

这篇文章想做的是通过视觉思维而不是将一切转化为文字来推理，更具体地说，是将细粒度的视觉信号直接注入到VLM的推理过程中，不仅能够在语言空间中推理，而且还可以通过连续的编码了紧凑的视觉表征进行推理。并且是通过一个自包含的，可解释性强的方式。（latent比较难的就是设计监督方式）

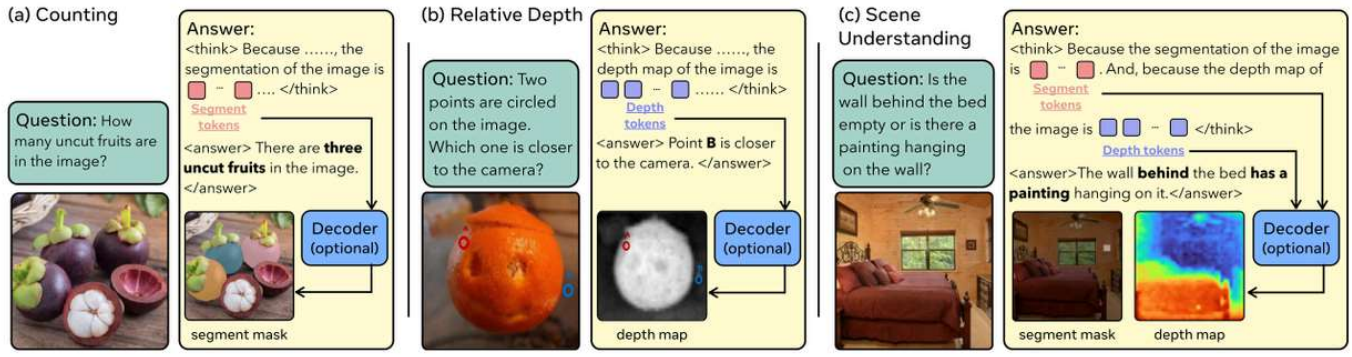


Figure 2. **Continuous visual thinking with CoVT.** CoVT introduces compact, continuous visual tokens that encode fine-grained perceptual cues, such as object localization, spatial structure, and scene semantics, directly into VLM reasoning. These tokens ground multimodal reasoning in visual space, enabling the model to capture fine-grained relationships across vision-centric tasks (*e.g.*, counting, depth ordering, and scene understanding) without relying on external tools. They can also be decoded into dense predictions, offering human-interpretable visualizations of the model's reasoning process.

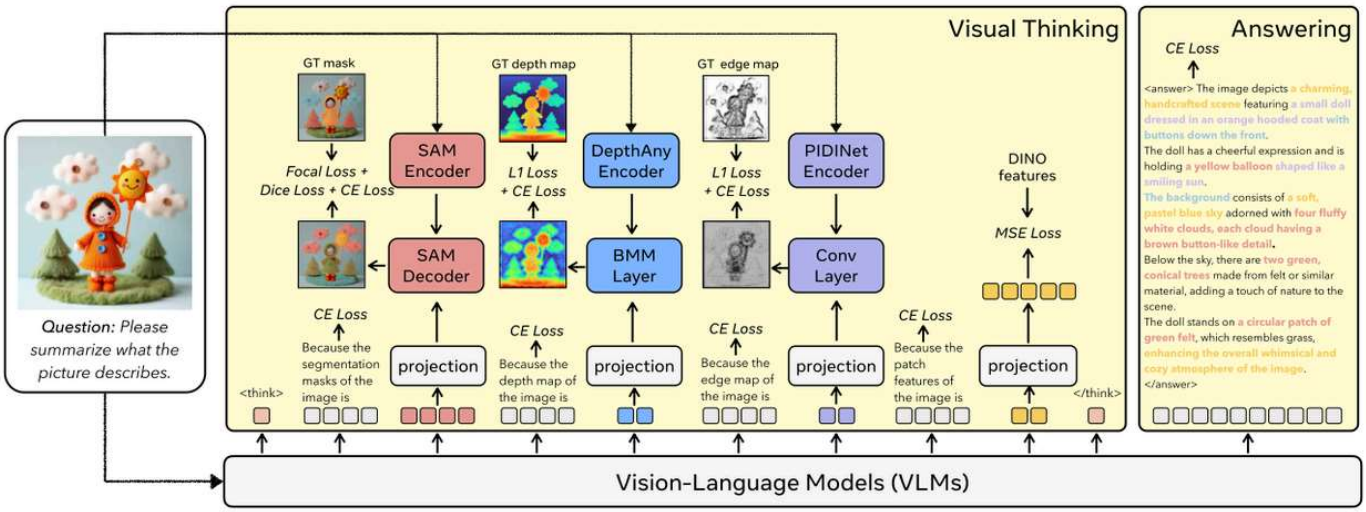
core perception ability:

1. instance recognition（sam)
2. 2D and 3D spatial relationship（DepthAnything)

3. structure detection (PIDINet)

4. deep mining of semantic information（dino)

在训练过程中，要求 VLM 在其推理链中预测一些连续的视觉token，从而将丰富的感知信息压缩进一个紧凑的latent space。每一组视觉token都与一个轻量级的视觉专家对齐（例如：SAM, DepthAnything, PIDINet, DINO），用于编码特定的视觉特征。



这些token由特定任务的轻量级解码器进行解码，以重建对应的目标（例如：分割掩码、深度图、边缘图或 DINO 特征）。其实模型就像一个dense的视觉编码器，能够主动提取并生成代表分割、深度等关键信息的token。

Loss function:

COVT 并没有对所有视觉专家模型采用统一的对齐方式，而是根据输出的精细程度将其分为两类：

- 任务导向型（如分割、深度）：输出很精细（Pixel-level），采用提示级对齐（Prompt-level Alignment）。
- 表征型（如 DINOv2）：输出相对宏观（Patch-level），采用特征级对齐（Feature-level Alignment）。

SAM:

模型生成的 8 个分割token被视为Prompts，直接喂给 SAM 的解码器。这些token会与 SAM 编码器提取的稠密图像特征f结合，还原出掩码图。

$$\hat{M}_i = \text{Decoder}(T_i^{\text{sam}}, f), \quad \hat{M}_i \in [0,1]^{H \times W},$$

匹配与损失计算:

- 匹配机制：使用匈牙利匹配算法。由于模型生成的是多个token，该算法能自动将预测的掩码与gt中最合适的掩码一一对应。（gt: 基于稳定性得分 (Stability Score) 和面积，筛选出质量最高的 Top-8 掩码作为 GT）
- 优化目标：结合 Dice Loss（关注区域重合度）和 Focal Loss（关注难分类像素），确保分割结果的极高精度。

DepthAnything v2

- 模型预测 4 个深度token，充当驱动深度重建的 Prompt，与depthanything的 4 层中间特征图进行bmm。

$$\hat{D}_i = \text{softmax}\left(T_i^{\text{depth}} \cdot F_i^{\text{depth}\top}\right),$$

- 多层融合：生成 4 张预测深度图并取平均值，得到最终的深度预测。
- loss: L1 loss

PIDINet

- 模型预测 4 个边缘token，被直接当作1*1卷积核使用，作用于对应的pidinet中间层特征图上，最后求平均值
- Loss: L1 loss

DINOv2

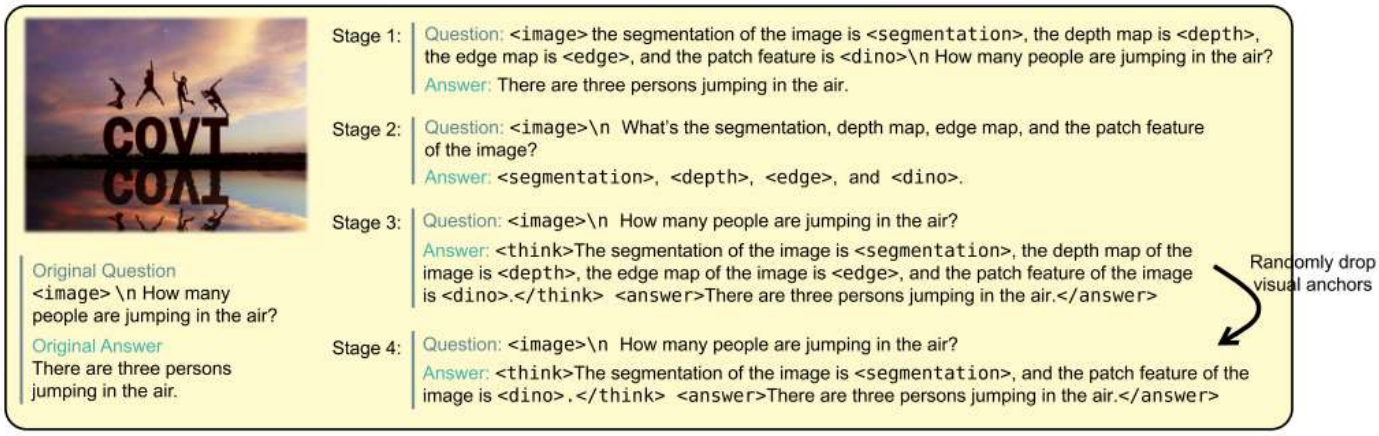
- 模型预测 4 个 DINO 标记，经过Projection Layer处理后，被直接映射到与 DINOv2 编码器空间中。
- Loss：特征层面的mse

token类型	监督专家	对齐层级	实现手段	损失函数	核心感知能力
分割 (Seg)	SAM	提示级	生成 Mask Prompts	Dice + Focal	物体实例与形状
深度 (Depth)	DepthAnything	提示级	BMM 矩阵乘法	L1 Loss	3D 空间关系
边缘 (Edge)	PIDINet	提示级	1x1 卷积核	L1 Loss	几何轮廓与结构
语义 (DINO)	DINOv2	特征级	投影后直接对齐	MSE Loss	patch级语义信息

最终的loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \gamma\big(\lambda_{\text{seg}} \cdot \mathcal{L}^{\text{seg}}_{\text{visual}} + \lambda_{\text{depth}} \cdot \mathcal{L}^{\text{depth}}_{\text{visual}} + \lambda_{\text{edge}} \cdot \mathcal{L}^{\text{edge}}_{\text{visual}} + \lambda_{\text{dino}} \cdot \mathcal{L}^{\text{dino}}_{\text{visual}}\big),$$

训练数据:



四个阶段：逐步教会模型如何生成和利用这些视觉token来辅助回答。

第一阶段帮助模型理解视觉token;

第二阶段引导模型生成视觉token;

第三阶段使VLM能够将视觉表征整合到其推理过程中;

第四阶段使模型能够在视觉思维链中有效地选择和利用视觉思维表征。（会随机丢弃（drop）某些视觉 token 类型，目的是让模型学会"并不是每次都能拿到所有视觉思维工具"，也要能稳住回答、并能在可用工具之间灵活切换。）

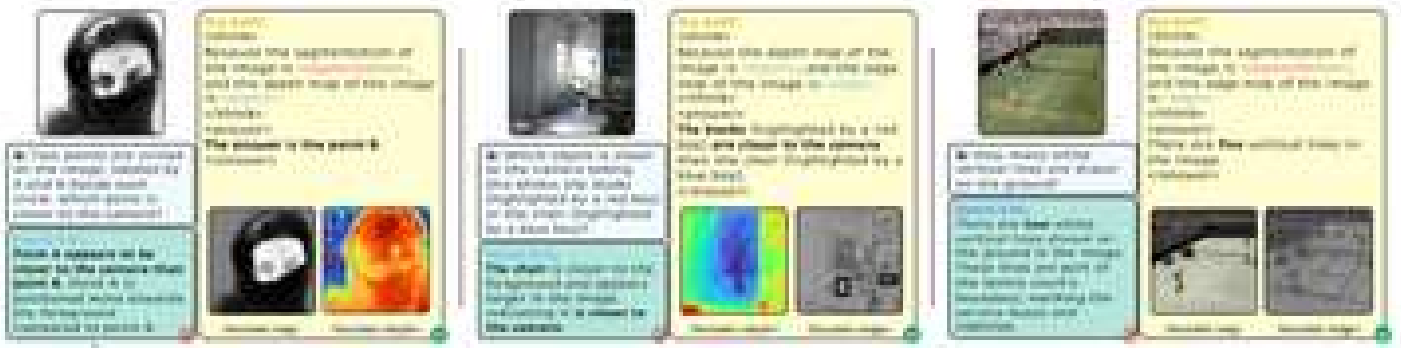
推理阶段:

如果需要，这些token可以被解码成人类可读的图像辅助说明（具有可解释性）；否则，模型仅在latent space高效推理。

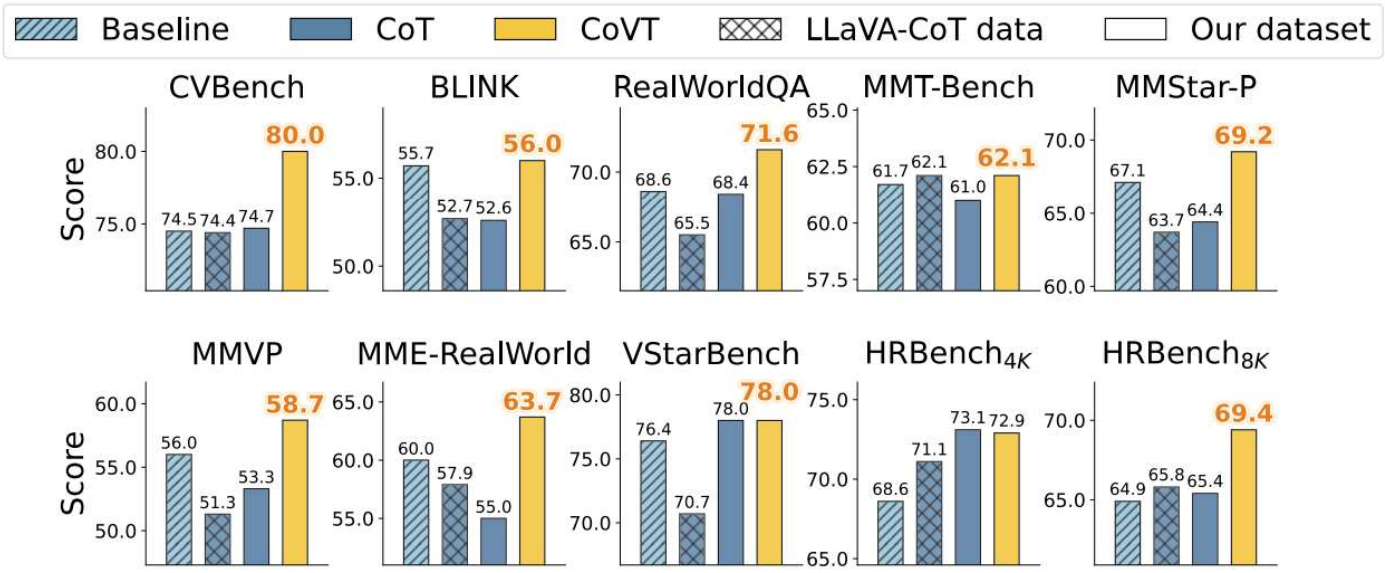
实验:

Visual tokens				CV-Bench				Other vision-centric benchmarks							
Seg	Depth	DINO	Edge	CVBench	Count	Depth	Dist.	BLINK	RW-QA	MMT	MMStar-P	MMVP	MME-RW	V*	HR_{4K}	HR_{8K}
Closed-source Models																
Claude-4-Sonnet				76.3	62.2	77.7	80.5	39.6	63.7	-	58.8	48.7	-	15.2	32.3	22.7
GPT-4o				79.2	65.6	86.7	81.0	63.0	69.7	-	65.2	72.0	-	42.9	50.6	46.7
Qwen2.5-VL-7B				74.5	65.0	72.8	75.5	55.7	68.6	61.7	67.1	56.0	60.0	76.4	68.6	64.9
CoVT (1 Visual Token)																
✓				77.9	66.0	80.8	80.5	**57.4**	71.1	62.1	68.5	58.7	62.1	**79.1**	71.9	69.0
	✓			78.7	65.4	83.2	78.2	56.4	71.5	**62.7**	**69.9**	58.7	62.0	**79.1**	71.9	69.4
		✓		71.3	64.7	72.3	66.7	55.8	71.5	62.5	67.9	57.3	61.1	77.5	71.0	68.6
CoVT (3 Visual Tokens)																
✓	✓	✓		**80.0**	**66.2**	86.8	**82.5**	56.0	71.6	62.1	69.2	**58.7**	**63.7**	78.0	**72.9**	69.4
Δ *(vs Baseline)*				**+5.5**	**+1.2**	**+14.0**	**+7.0**	**+0.3**	**+3.0**	**+0.4**	**+2.1**	**+2.7**	**+3.7**	**+1.6**	**+4.3**	**+4.5**
CoVT (4 Visual Tokens)																
✓	✓	✓	✓	79.8	66.1	**89.2**	80.5	56.2	**71.8**	61.9	68.4	56.7	63.3	78.5	72.5	**69.9**
Δ *(vs Baseline)*				**+5.3**	**+1.1**	**+16.4**	**+5.0**	**+0.5**	**+3.2**	**+0.2**	**+1.3**	**+0.7**	**+3.3**	**+2.1**	**+3.9**	**+5.0**

covt和和闭源模型以及baseline的比较结果。



一些推理的case，直观地展示这些token是否为推理提供了有用的信息，以及作者认为不同的视觉token提供的线索具有很强的互补性（depth子任务加上edge的效果更好）。



Text setting: cvot的数据去掉视觉token，全部转换为 LLaVA-CoT 风格的纯文本格式; 以及llava cot数据。

文本 CoT 不等于视觉感知。在处理深度、形状、计数等感知密集型任务时，文本无法表征高维的视觉信息，会导致推理性能退化。COVT 避开了“视觉 -> 文本 -> 推理”的有损路径，实现了“视觉 -> 视觉latent space -> 推理”的高保真路径 。

Type	Align	CVBench	BLINK	RW-QA	MM*-P	MMVP	V*	HR_{4K}
Seg	Feature	76.8	55.2	70.6	67.7	56.0	78.0	69.8
	Ours	**77.9**	**57.4**	**71.1**	**68.5**	**58.7**	**79.1**	**71.9**
Depth	Feature	77.0	54.2	70.5	67.6	55.3	**78.0**	71.3
	Ours	**78.7**	**56.4**	**71.5**	**69.9**	**58.7**	77.5	**71.9**

Table 5. Our tailored **alignment strategy** plays a crucial role in further enhancing the performance of CoVT.

这个说明对齐方式的有效性。

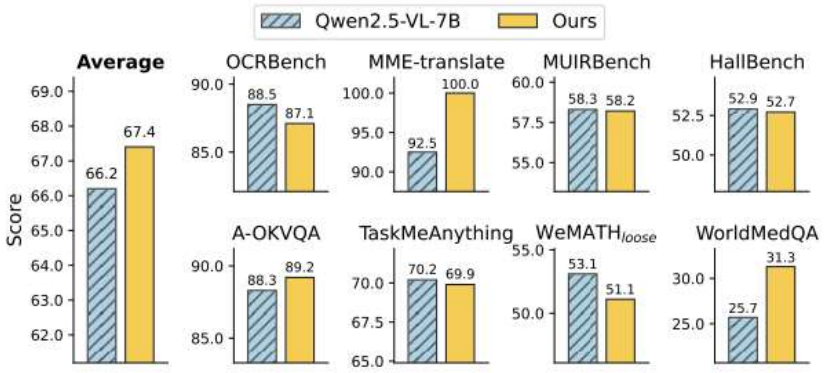


Figure 7. Beyond the gains on vision-centric benchmarks, CoVT also achieves slight improvements on **non–vision-centric tasks**

在不是visual centric的bench上的测试，也有总体上的改善。

这篇文章存在的一些不足：

1. 论文后面用"随机 drop 某些 token 类型"来提升鲁棒性，但这更像是"避免过拟合某个工具"，并不等于"动态决定需要多少 token、需要什么工具"。
2. SAM 的 GT mask 不是人工标注，而是用 SAM 自己生成并筛选（area + stability score 过滤后保留 8 个）。这等于"把 SAM 蒸馏进 token"。优点是便宜，但缺点是：token 学到的可能是"SAM 的偏好"，不一定是 VLM 任务最需要的视觉中间表示，并且上限受到sam的限制。