香港中文大學
The Chinese University of Hong Kong

VisionReasoner: Unified Reasoning-Integrated Visual Perception via Reinforcement Learning

Yuqi Liu[1], Tianyuan Qu[1], Zhisheng Zhong[1], Bohao Peng[1]
Shu Liu[2], Bei Yu[1], Jiaya Jia[2,3]
The Chinese University of Hong Kong[1]
SmartMore[2]
The Hong Kong University of Science and Technology[3]
March 31, 2026

Outline

Section 1

Introduction

Introduction

- Through an analysis of diverse visual perception tasks, we observe that many can be categorized into three fundamental types: detection (e.g., object detection[1]), segmentation (e.g., referring expression segmentation[2], reasoning segmentation[3]), and counting (e.g., object counting[4]).
- Notably, our analysis reveals that these three task types share a common structure as multi-object cognition problems, suggesting that they can be addressed through a unified framework.

[1] **Tsung-Yi Lin et al. (2014). "Microsoft coco: Common objects in context".** In: *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, pp. 740–755.

[2] **Licheng Yu et al. (2016). "Modeling context in referring expressions".** In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, pp. 69–85.

[3] **Xin Lai et al. (2024). "Lisa: Reasoning segmentation via large language model".** In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9579–9589.

[4] **Roni Paiss et al. (2023). "Teaching clip to count to ten".** In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3170–3180.

Introduction

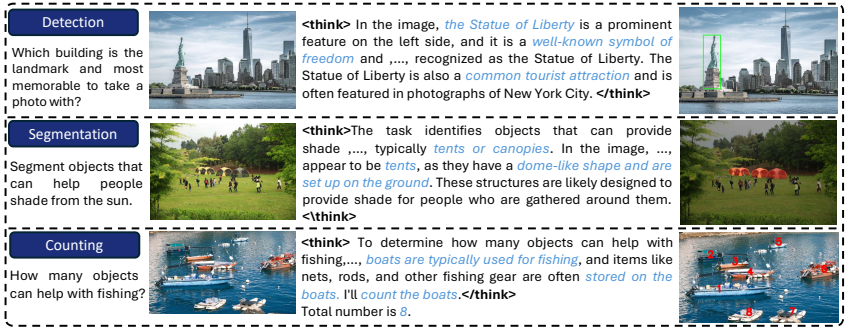

Figure: (a) VisionReasoner addresses diverse tasks within a unified framework. It generates a reasoning process and outputs the expected result corresponding to each query. (b) VisionReasoner significantly outperforms Qwen2.5VL. (c) VisionReasoner retains strong VQA capabilities.

Section 2

Method

Method - Why LVLM is needed?

Although traditional vision models[5,6] achieve strong performance on standard visual perception benchmarks, they are struggle with complex instructions.

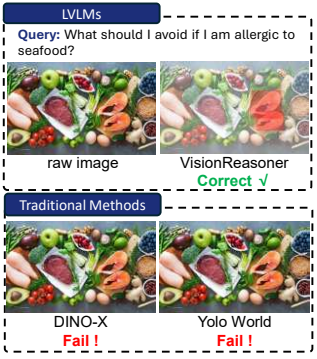

Figure: VisionReasoner correctly localizes objects from a complex instruction, whereas both commercial DINO-X and open-source YOLO-World fail.

[5] **Tianheng Cheng et al. (2024). "Yolo-world: Real-time open-vocabulary object detection".** In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16901–16911.

[6] **Tianhe Ren et al. (2024). "Dino-x: A unified vision model for open-world object detection and understanding".** In: *arXiv preprint arXiv:2411.14347*.

Method - Task Categorization

Our analysis of vision perception tasks reveals that many of them can be categorized into three fundamental task types.

- **Detection**. Given an image $\mathbf{I}$ and a text query $\mathbf{T}$, the detection task type aims to generate a set of bounding boxes $\{\mathbf{B}_i\}_{i=1}^{N}$ that localize objects of interest.
- **Segmentation** Given an image $\mathbf{I}$ and a text query $\mathbf{T}$, the segmentation task type aims to generate a set of binary segmentation masks $\{\mathbf{M}_i\}_{i=1}^{N}$ that identify the regions of interest.
- **Counting** Given an image $\mathbf{I}$ and a text query $\mathbf{T}$, the counting task type aims to estimate the number of target objects specified by the query.

Method - Architecture

Our VisionReasoner $\mathcal{F}$ model incorporates a reasoning module, which processing image and locates targeted objects, and a segmentation module that produces segmentation masks if needed.

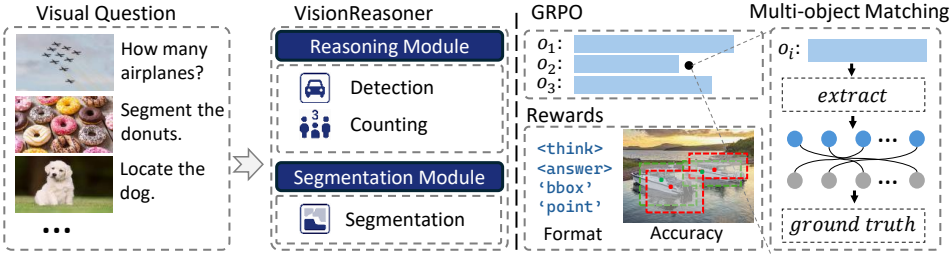

Figure: Illustration of VisionReasoner. (a) For a given image $\mathbf{I}$ and text instruction $\mathbf{T}$, our model generates the expected output corresponding to the instruction. (b) For each observation o_i, we calculate the rewards and attain the optimal match of multi-objects.

Given an image $\mathbf{I}$ and a text query $\mathbf{T}$, the VisionReasoner $\mathcal{F}$ generates an interpretable reasoning process, and then produces the bounding boxes $\{\mathbf{B}_i\}_{i=1}^{N}$ and binary masks $\{\mathbf{M}_i\}_{i=1}^{N}$ if needed. The inference process can be formulated as:

$$(\{\mathbf{B}_i, \mathbf{M}_i\})_{i=1}^{N} = \mathcal{F}(\mathbf{I}, \mathbf{T}). \tag{1}$$

During inference, we define a specified task type $\mathbf{C} \in \{\text{detection, segmentation, counting}\}$. The system then produces the expected outputs as follows:

$$\text{Output} = \begin{cases} \{\mathbf{B}_i\}_{i=1}^{N}, & \text{if } \mathbf{C} \text{ is detection,} \\ \{\mathbf{M}_i\}_{i=1}^{N}, & \text{if } \mathbf{C} \text{ is segmentation,} \\ N, & \text{if } \mathbf{C} \text{ is counting.} \end{cases} \tag{2}$$

Method - Reward Functions

We design a unified reward mechanism for visual perception tasks, including format rewards and accuracy rewards.

- **Thinking Format Reward** This reward is 1.0 if the model output a thinking process between <think> and </think>tags, and output the final answer between the <answer> and </answer>tags.
- **Answer Format Reward** We use bounding boxes $\{\mathbf{B}_i\}_{i=1}^{N}$ and points $\{\mathbf{P}_i\}_{i=1}^{N}$ as the answer as it has better training efficiency. So this reward restrict the model output answer in $[\{'\text{bbox\_2d}' : [x_1, y_1, x_2, y_2], '\text{point\_2d}' : [x_1, y_1]\}, ...]$.
- **Non Repeat Format Reward** We split the reasoning process into sentences to detect repeated pattern. A reward of 1.0 is assigned for those with unique or non-repetitive thinking processes.

Method - Reward Functions

We design a unified reward mechanism for visual perception tasks, including format rewards and accuracy rewards.

- **Bboxes IoU Reward.** Given a set of N ground-truth bounding boxes and K predicted bounding boxes, this reward computes their optimal one-to-one matched Intersection-over-Union (IoU) scores. For each IoU exceeding 0.5, we increment the reward by $\frac{1}{\max\{N, K\}}$.
- **Bboxes L1 Reward** Given a set of N ground-truth bounding boxes and K predicted bounding boxes, this reward computes their one-to-one matched L1 distance. For each L1 distance below the threshold of 10 pixel, we increment the reward by $\frac{1}{\max\{N, K\}}$.
- **Points L1 Reward** Given a set of N ground-truth points and K predicted points, this reward computes their one-to-one matched L1 distance. For each L1 distance below the threshold of 30 pixel, we increment the reward by $\frac{1}{\max\{N, K\}}$.

Algorithm 1: Multi-object Matching

Input: pred bboxes $\boldsymbol{b}_{\text{pred}} \in \mathbb{R}^{K\times 4}$; pred points $\boldsymbol{p}_{\text{pred}} \in \mathbb{R}^{K\times 2}$; GT bboxes $\boldsymbol{b}_{\text{gt}} \in \mathbb{R}^{N\times 4}$; GT points $\boldsymbol{p}_{\text{gt}} \in \mathbb{R}^{N\times 2}$

Function *AccuracyReward*$(\boldsymbol{b}_{pred}, \boldsymbol{p}_{pred}, \boldsymbol{b}_{gt}, \boldsymbol{p}_{gt})$**:**

$r \leftarrow 0$; $L_{\max} \leftarrow \max(K, N)$;

$IoU \leftarrow \text{BatchIoU}(\boldsymbol{b}_{\text{pred}}, \boldsymbol{b}_{\text{gt}}) \in \mathbb{R}^{K\times N}$

$BL1 \leftarrow \text{BatchBoxL1Distance}(\boldsymbol{b}_{\text{pred}}, \boldsymbol{b}_{\text{gt}}) \in \mathbb{R}^{K\times N}$

$PL1 \leftarrow \text{BatchPointL1Distance}(\boldsymbol{p}_{\text{pred}}, \boldsymbol{p}_{\text{gt}}) \in \mathbb{R}^{K\times N}$

$R_{\text{IoU}} \leftarrow [IoU > \text{IoU threshold}]$

$R_{\text{BL1}} \leftarrow [BL1 < \text{Box L1 threshold}]$

$R_{\text{PL1}} \leftarrow [PL1 < \text{Point L1 threshold}]$

$C \leftarrow (3 - (R_{\text{IoU}} + R_{\text{BL1}} + R_{\text{PL1}})) \in \mathbb{R}^{K\times N}$

$(\boldsymbol{r}, \boldsymbol{c}) \leftarrow \text{Hungarian}(C)$

$\text{total} \leftarrow 3|\boldsymbol{r}| - \sum_t C_{\boldsymbol{r}_t, \boldsymbol{c}_t}$

$r \leftarrow \text{total}/L_{\max}$; **return** r

Output: Accuracy reward r

Section 3

Experiment

Experiment - Datasets

We use ten benchmarks to evaluate model performance across general vision perception tasks, including three fundamental task types: detection, segmentation and counting.

Type	Data	# of samples
Det	COCO	36,781
	RefCOCO	5,786
	RefCOCO+	5,060
	RefCOCOg	7,596
Seg	RefCOCO	1,975
	RefCOCO+	1,975
	RefCOCOg	5,023
	ReasonSeg	979
Count	Pixmo-Count	1,064
	CountBench	504
SUM		**66,023**

Table: Statistics of evaluation benchmarks. We report the number of instances for detection and segmentation tasks. The reported numbers combine validation and test splits where applicable.

Experiment - Main Results

Method	Detection							Avg.
	COCO	RefCOCO		RefCOCO+		RefCOCOg		
	val	val	testA	val	testA	val	test	
Task-specific Models								
VGTR	-	79.0	82.3	63.9	70.1	65.7	67.2	-
TransVG	-	81.0	82.7	64.8	70.7	68.7	67.7	-
RefTR	-	85.7	88.7	77.6	82.3	79.3	80.0	-
MDETR	-	86.8	89.6	79.5	84.1	81.6	80.9	-
GLIP-T	46.6	50.4	54.3	49.5	52.8	66.1	66.9	55.2
G-DINO-T	48.4	74.0	74.9	66.8	69.9	71.1	72.1	68.2
DQ-DETR	50.2	88.6	91.0	81.7	86.2	82.8	83.4	**80.6**
Large Vision-language Models								
Shikra-7B	-	87.0	90.6	81.6	87.4	82.3	82.2	-
InternVL2-8B	-	87.1	91.1	79.8	87.9	82.7	82.7	-
Qwen2-VL-7B	28.3	80.8	83.9	72.5	76.5	77.3	78.2	71.1
Qwen2.5-VL-7B	29.2	88.8	91.7	82.3	88.2	84.7	85.7	78.6
VisionReasoner-7B	37.7	88.6	90.6	83.6	87.9	86.1	87.5	**80.3**

Table: Performance comparison on detection tasks.

Experiment - Main Results

Method	Segmentation					Avg.
	ReasonSeg		RCO	RCO+	RCOg	
	val	test	testA	testA	test	
Task-specific Models						
LAVT	-	-	75.8	68.4	62.1	-
ReLA	22.4	21.3	76.5	71.0	66.0	51.4
Large Vision-language Models						
LISA-7B	44.4	36.8	76.5	67.4	68.5	58.7
GLaMM-7B	-	-	58.1	47.1	55.6	-
PixelLM-7B	-	-	76.5	71.7	70.5	-
Seg-Zero-7B	62.6	57.5	80.3	76.2	72.6	69.8
Qwen2-VL-7B	44.5	38.7	68.7	65.7	63.5	56.2
Qwen2.5-VL-7B	56.9	52.1	79.9	76.8	72.8	67.7
VisionReasoner-7B	66.3	63.6	78.9	74.9	71.3	**71.0**

Table: Performance comparison on segmentation tasks. We use SAM2 for vision-language models if necessary.

Experiment - Main Results

Method	Counting			Avg.
	Pixmo		Count	
	val	test	test	
Large Vision-language Models				
LLaVA-OV-7B	55.8	53.7	78.8	62.8
GLaMM-7B	-	-	-	-
PixelLM-7B	-	-	-	-
Seg-Zero-7B	-	-	-	-
Qwen2-VL-7B	61.6	56.3	80.4	66.1
Qwen2.5-VL-7B	58.1	53.1	78.8	63.6
VisionReasoner-7B	70.1	70.7	89.2	**76.7**

Table: Performance comparison on counting tasks.

Experiment - Main Results

We compare the results with LVLMs and task-specific models on each of the three fundamental task types. It is worthy note that our VisionReasoner is capable of handling different tasks within the same model.

- **Detection**. VisionReasoner achieves superior performance among LVLMs. While our model shows a performance gap compared to some task-specific baselines on COCO datasets, it maintains competitive advantages due to its superior generalization capability.
- **Segmentation**. VisionReasoner achieves state-of-the-art performance, outperforming both general-purpose LVLMs and task-specific approaches.
- **Counting**. VisionReasoner achieves state-of-the-art performance on counting tasks.

VisionReasoner achieves a slight performance gain even though we do not train on VQA data.

Method	OCRBench	RealworldQA	MMMUPro$_{vision}$	MMMUPro$_{std}$
Qwen2VL-7B	809	66.1	28.0	33.8
Qwen2.5VL-7B	822	69.2	32.4	36.4
VisionReasoner-7B	**825**	**69.5**	**32.6**	**37.4**

Table: Performance comparison on VQA tasks.

Experiment - Visualization

Figure: Qualitative results on different tasks.

Section 4

Conclusion

Conclusion

- We present VisionReasoner, a unified vision-language framework for reasoning visual perception tasks.
- VisionReasoner demonstrates strong capabilities in analyzing visual inputs, generating structured reasoning processes and delivering task-specific outputs.
- Experiments across ten diverse tasks, spanning detection, segmentation and counting, validates the robustness and versatility of our approach.

THANK YOU!