Lumen: Unleashing Versatile Vision-Centric Capabilities of Large Multimodal Models

Yang Jiao[*1,2,3], **Shaoxiang Chen**[3], **Zequn Jie**[†3], **Jingjing Chen**[†1,2]
Lin Ma[3], **Yu-Gang Jiang**[1,2]
[1]Shanghai Key Lab of Intell. Info. Processing, School of CS, Fudan University
[2]Shanghai Collaborative Innovation Center on Intelligent Visual Computing
[3]Meituan

Abstract

Large Multimodal Model (LMM) is a hot research topic in the computer vision area and has also demonstrated remarkable potential across multiple disciplinary fields. A recent trend is to further extend and enhance the perception capabilities of LMMs. The current methods follow the paradigm of adapting the visual task outputs to language-oriented formats. This adaptation leads to the convenient development of such LMMs with minimal modifications, however, it overlooks the inductive biases within diverse visual tasks and hinders the learning of perception capabilities. To address this issue, we propose a novel LMM architecture named **Lumen**, which decouples the learning of perception capabilities into task-agnostic and task-specific stages. Firstly, Lumen promotes fine-grained vision-language concept alignment, which is the fundamental capability for various visual tasks. Thus the output of the task-agnostic stage is a shared representation for all vision-centric tasks we address in this paper. Afterward, the task-specific decoding is carried out by flexibly routing the shared representation to lightweight task decoders with negligible training efforts. Comprehensive experimental results on a series of vision-centric and VQA benchmarks indicate that our Lumen model not only achieves or surpasses the performance of existing LMM-based approaches in a range of vision-centric tasks while maintaining general visual understanding and instruction following capabilities.

1 Introduction

As the Large Language Models (LLMs) [1, 2] currently ignite the spark of Artificial General Intelligence (AGI), Large Multimodal Models (LMMs) [3, 4, 5, 6, 7, 8, 9] take a step forward by integrating visual modalities with the linguistic prowess of LLMs. With the instruction-following and content-reasoning capabilities inherited from LLMs, the LMM has successfully functioned as a versatile assistant across a wide range of tasks, including visual question answering [10, 11, 12], image captioning [13, 14], text-to-image generation [15], etc.

In pursuit of more convenient and efficient human-AI interaction, it is crucial to further explore fundamental vision-centric capabilities encapsulated in the LMMs, which aid in detailed object referencing and dialogue responses. Early works, e.g., MiniGPT-v2 [16], Kosmos-2 [17] and Qwen-VL [18], equip the LMM with the grounding ability by reformulating bounding boxes as a sequence of coordinate tokens and adapting them to the language model's output space. Griffon [19] extends this design to object detection by meticulously curating a language-prompted detection dataset. Owing to such language model-oriented reformulation, these methods can be implemented with minimal modifications to existing LMMs.

[*]This work is done when Yang Jiao is an intern at Meituan. [†] Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

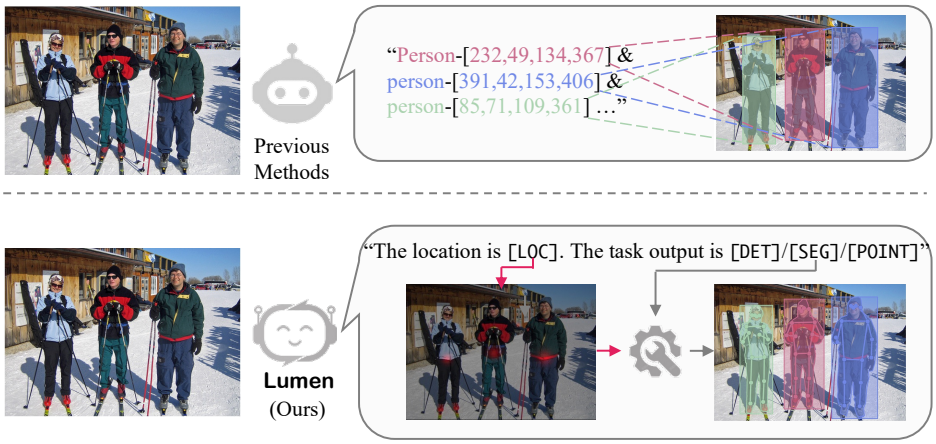


Figure 1: **Comparison of our proposed Lumen with previous methods.** Previous methods (e.g., Griffon [19]) serialize bounding box coordinates into discrete token sequences to conform to the language-oriented outputs format of LMMs, disregarding the unordered nature inherent in bounding boxes. Our Lumen first predicts unified heatmaps for various tasks. These heatmaps are further used for guiding simple decoding tools with the parsed task-type indicators to support versatile visual tasks. We omit the user instruction of referring to all persons in the image for conciseness.

Although convenient, the aforementioned approaches encounter challenges when scaling up to more intricate scenarios and vision-centric tasks. First, those LLMs highly relies on language models's auto-regressive sequence generation method, which leads to high uncertainty when multiple objects are concurrently addressed. As shown in the first row of Fig 1, the three persons highlighted with bounding boxes of different colors lack an inherent order. And imposing a sequential order on them would exacerbate the confusion in the decoding process, as the model would be compelled to produce drastically different outputs after the same word "person". Although sequence augmentation techniques introduced in Pix2Seq [20] can alleviate this issue, they are tailored for the object detection task and can not be seamlessly transferred to other vision-centric tasks such as instance segmentation and pose estimation. On the other hand, in contrast to tasks within the NLP field, which often exhibit stronger inter-task correlations [21], vision-centric tasks are inherently discrete due to the inductive bias introduced by their distinct task definitions [22]. Therefore, reformulating heterogeneous vision-centric task outputs into language-oriented formats tends to overemphasize the format consistency with language models, while the learning of the underlying visual perception capabilities and intrinsic characteristics of diverse vision tasks is overlooked.

When delving into fundamental vision-centric tasks, namely object detection, instance segmentation, and pose estimation[1], we observe that they can be decoupled into task-agnostic and task-specific learning processes. Essentially, these tasks share a common task-agnostic objective of identifying individual instances with an instruction like *"finding the area addressed in instructions"*, while their task-specific definitions introduce different decoding rules for generating diverse formats of outputs (e.g., boxes, masks, or points). Compared with the general task-agnostic objective, task-specific outputs relies less on semantics but is more difficult for the LMMs to learn to output. Based on the above analysis, in this paper, we propose **Lumen**, a **L**arge m**u**ltimodal **m**odel with vision-centric capabilities **en**hancement, which decouples the task-agnostic and task-specific learning in two consecutive stages as shown in the Fig 1. Concretely, in the first stage, the aforementioned visual tasks are unified by reformulating them into the same matching problem. We start by feeding the user's instruction and image into a LMM for content comprehension. The obtained responses contain a designed special token (i.e, `[LOC]` in Fig 1) that encapsulates the visual concepts conveyed in the provided instruction, regardless of the specific task. Subsequently, this special token interacts with image patches via a transformer-based aligner to generate a heatmap, wherein the response at each location indicates the matching probability between the instruction and the corresponding image region. In the second stage, utilizing this heatmap as indicative guidance, task-specific decoding

[1]We only investigate these three visual tasks in this paper, as LMMs inherently lack image generation capabilities.

processes are further managed by flexibly assembling predefined decoding rules and lightweight decoders to generate the final outputs with different formats. Due to such decoupled learning behaviors, on the one hand, Lumen can concentrate on promoting fine-grained multimodal content comprehension, rather than being trapped in learning diverse specialized decoding rules lacking in semantics. On the other hand, Lumen can be affected less by the various inductive biases associated with vision-centric tasks within the LLM token space, thereby seamlessly maintaining general visual understanding and instruction following capabilities, as demonstrated in Table 2.

In summary, our contributions are three folds: (1) We introduce **Lumen**, a **L**arge m**u**ltimodal **m**odel with vision-centric capabilities **en**hancement, which unleashes the vision-centric potential of the LMM by decoupling the task-agnostic and task-specific learning processes; (2) Our Lumen can seamlessly adapt to tasks such as object detection, instance segmentation, and pose estimation without requiring meticulously curated specialized dialogue datasets as done in the previous work [19]. (3) Our Lumen not only matches or exceeds the performances of existing LMM-based approaches on a series of vision-centric tasks, but also maintains general visual understanding and instruction following capabilities.

2 Related Work

2.1 Large Multimodal Models (LMMs)

Benefiting from the remarkable reasoning capabilities of Large Language Models (LLMs), LMMs [23, 9, 24, 25] transfer these abilities to the vision domain by aligning visual tokens with LLMs' input space. To achieve this, pioneering work, Flamingo [26] resamples the visual features and feeds them into attention-based adapter layers inserted in the LLM. Aiming at more comprehensive vision-language alignment, BLIP-2 [27] designs a Q-Former and jointly performs cross-modal representation learning and generative learning. Inspired by the instruction tuning technique [21, 28] in NLP field, Instruct-BLIP [29], LLaVA [4] and Mini-GPT4 [3] curate high-quality multi-modal instruction data for enhanced instruction-following abilities. Meanwhile, to systematically evaluate their general-purpose conversational capabilities, more advanced benchmarks [30, 31, 32, 33, 34, 35] have been explored. However, these methods focus on high-level visual content comprehension and reasoning, ignoring the fundamental visual perception functions, such as object detection, instance segmentation, pose estimation, etc.

2.2 Vision Generalist Models

Generalist models in the vision domain aim at unifying a wide range of vision tasks using one model. Motivated by the success of sequence-to-sequence models in NLP field [13], OFA [36] and GIT [37] unify various tasks in the sequence generation format. Following this trend, Unified-IO [38], Pix2Seq v2 [39] and UniTab [40] add discrete coordinate tokens into the vocabulary, so as to accommodating more tasks. Moreover, Gato [41] successfully unifies reinforcement learning tasks as the sequence generation format. Nevertheless, the sequence generation modeling can lead to low inference speeds and degraded performances. Toward this end, Uni-Perceivers [42, 43] unify different tasks as the maximum likelihood matching problem by calculating representation similarities of all targets and each input. With such an objective, generative and non-generative tasks can be unified by selecting corresponding input and target candidates. However, these generalist models are restricted to pre-defined tasks, failing to be flexibly instructed by natural languages like LMMs.

2.3 LMMs with Vision-Centric Capabilities

To endow LMMs with vision-centric capabilities, two research directions are investigated. On the one hand, a line of work regards LLMs/LMMs as intelligent planners, and allows them to trigger a wide range of task-specific Application Program Interfaces (APIs) according to user's instructions. HuggingGPT [44] connect GPT with a suite of visual experts. AutoGPT [45] can further execute post-processing programs after detection. Moreover, BuboGPT [46] further combines visual grounding specialists with LMMs. On the other hand, an alternative approach explores the intrinsic localization potential of LMMs with task-specific modifications. Kosmos-2 [17], MiniGPT-v2 [16], Qwen-VL [18] and Shikra [47] enlarge the vocabulary size of LMMs with discrete coordinate tokens to deal with the visual grounding task [48]. LISA [49] merges the LMM with SAM [50] for enhanced

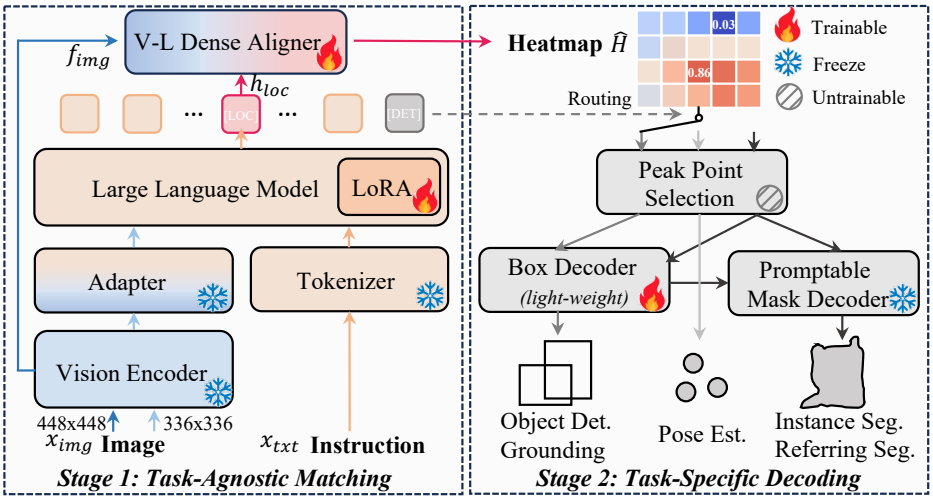


Figure 2: **Overall framework of the proposed Lumen**. Our Lumen consists of two stages. In the first stage, the input image and the instruction with designed special tokens are embedded and fed into a large language model to interact and comprehend visual and textual contents. Then, the `[LOC]` token output and high-resolution image features are further aligned to produce a heatmap denoting cross-modal matching probabilities. In the second stage, the heatmap can serve as a strong indication for various vision tasks, and the task outputs can be obtained with lightweight task-specific decoders. The routing of the decoding pathway is determined by the task token (e.g., `[DET]` in image) output generated in the first stage.

reasoning capability in the referring image segmentation scenario [51]. However, these methods do not address fundamental vision tasks like object detection and instance segmentation, where multiple objects should be detected or segmented simultaneously. To address this issue, VisionLLM [52] regards an LLM as a DETR-like task decoder and customizes structural prompts for the detection task with Hungarian matching [53] for label assignment. Griffon [19] takes a different approach by leveraging the inherent detection capabilities of the LMM, introducing a language-prompted detection dataset for the instruction tuning. However, these methods leverage discrete coordinate tokens as outputs for different vision tasks, while ignoring their inherent disparities. In this paper, we disentangle the task-agnostic and task-specific learning of various vision-centric tasks to unlock the visual potential of LMMs.

3 Method

As shown in Fig 2, our proposed Lumen comprises two consecutive stages. In the first stage, we concentrate on promoting fine-grained multimodal content comprehension via densely aligning visual regions and language instructions, disregarding the discrepancies between various vision tasks. In the second stage, with the resulting alignment heatmap from the first stage, task-specific decoding is performed with specialized operations or decoders. In the following parts, we will elaborate on the detailed designs within each stage.

3.1 Stage 1: Task-Agnostic Matching

3.1.1 Conversation Reformulation

A preliminary step for adapting vision-centric tasks to LMMs is to reformulate the visual data into conversation formats. For different tasks, we employ a unified instruction-response template: "`USER`: `[IMG]`. *Please find the location of* `{description}`. *Respond with* `{format}`. `ASSISTANT`: *Sure, the location is* `[LOC]`. *The task output is* `[DET]`/`[SEG]`/`...`" Here, `{description}` and `{format}` can be customized according to specific tasks. For vision-centric tasks like object detection, instance segmentation and pose estimation, `{description}` is a certain class name, and `{format}` can be boxes, masks, or points. For vision-language tasks like visual grounding and referring segmentation,

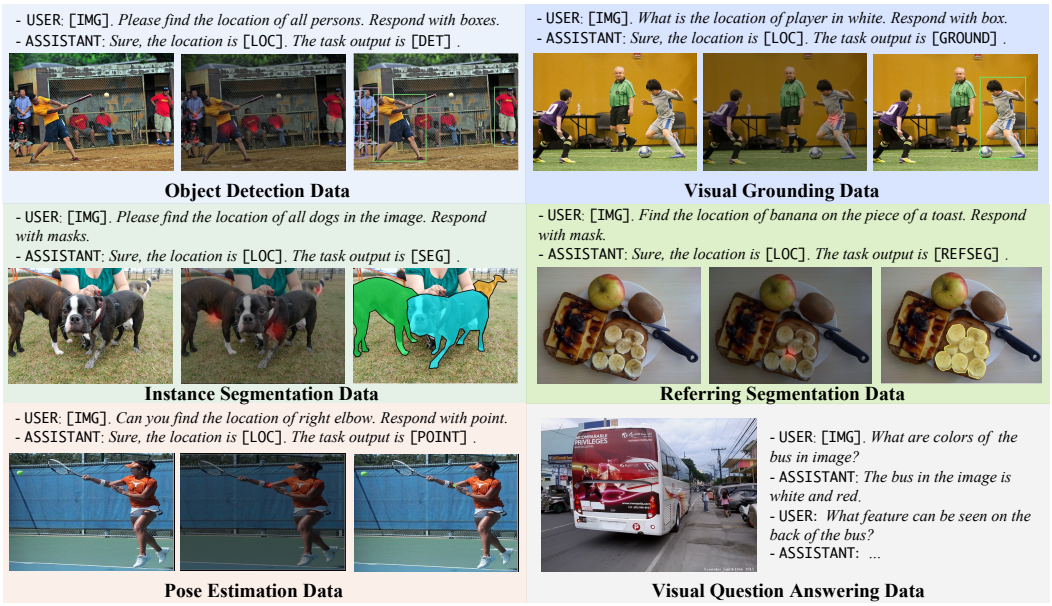


Figure 3: **The illustration of reformulated conversation data examples of different tasks.** For each reformulated data example, we sequentially present the original image (left), the heatmap generated from annotations (middle), and the task-specific ground truth (right) in the figure.

`{description}` is the referring sentence and `{format}` can be boxes or masks. Apart from `[IMG]` that denotes image features, we also introduce a task-agnostic special token `[LOC]` and a set of task-specific special tokens, including `[DET]`, `[SEG]`, `[POINT]`, `[GROUND]` and `[REFSEG]`. As shown in Fig 2, regardless of task types, `[LOC]` is tasked to summarize the visual concepts in instructions and further densely align with image regions. Task-specific tokens merely serve as routers for guiding the decoding pathway in the second stage **without encoding task-specific inductive bias**. Examples of reformulated conversation data of various tasks are illustrated in Fig 3.

3.1.2 Model Architecture

With the given image x_{img} and reformulated instruction x_{txt} as inputs, we first feed them into a conventional LMM (LLaVA-1.5 [54] in our implementation) to generate language responses. We also extract high resolution image features $f_{img} \in \mathbb{R}^{32\times32\times256}$ following prior works [18, 19]:

$$f_{img} = \mathcal{F}_{\mathcal{V}}(x_{img}), \tag{1}$$

where $\mathcal{F}_{\mathcal{V}}$ denotes the vision encoder. Since the adopted vision encoder, CLIP-ViT [55], is pretrained to accept image inputs of 336×336 pixels, we resize its positional embeddings to adapt to higher resolution, e.g., 448×448 pixels. Afterward, the feature of `[LOC]` token $h_{loc} \in \mathbb{R}^{1\times256}$ from the generated response, together with high-resolution image features, are fed into the proposed V-L dense aligner to calculate their matching probabilities, resulting in a heatmap $\hat{H} \in \mathbb{R}^{32\times32}$:

$$\hat{H} = \mathcal{F}_{\mathcal{A}}(f_{img}, h_{loc}), \tag{2}$$

where $\mathcal{F}_{\mathcal{A}}$ represents the V-L dense aligner. In our implementation, we employ a lightweight transformer-based architecture as $\mathcal{F}_{\mathcal{A}}$ to interact high resolution image feature tokens and the `[LOC]` token. More discussions on the architecture design are included in Sec.4.3 and Appendix A.1.

3.1.3 Model Training

To enrich semantics during the dense alignment, we merge data from various tasks by reformulating them into uniform conversation formats as shown in Fig 3. For vision-centric tasks, we hide diverse task-specific output format details and their learning targets are reformulated and unified as heatmaps, where each element denotes the matching probability between the instruction and the corresponding

region. To construct the ground-truth heatmap H, we use a Gaussian kernel to fill the probabilities into a blank map following prior works [56, 57]:

$$H_{xy} = \exp(-\frac{(x-l_x)^2 + (y-l_y)^2}{2\sigma^2}), \tag{3}$$

where the (l_x, l_y) is the location of the one referred by the instruction, and σ is the standard deviation. For object detection and visual grounding, (l_x, l_y) is the center coordinate of each object, and σ is object size-adaptive following [56]. For instance segmentation and referring segmentation, we first convert their masks into bounding boxes by calculating enclosing rectangles of masks, and then calculate (l_x, l_y) and σ with the same rules as detection and visual grounding tasks. For pose estimation, (l_x, l_y) is the coordinate of the annotated keypoint, σ is a hyper-parameter designated as 2.

With the ground-truth heatmap H and predicted heatmap $\hat{H}$, we apply Gaussian focal loss following [56] as:

$$\mathcal{L}_h = -\frac{1}{N}\sum_{xy}\begin{cases} (1-\hat{H}_{xy})^{\alpha}\log(\hat{H}_{xy}) & \text{if } H_{xy}=1, \\ (1-H_{xy})^{\beta}(\hat{H}_{xy})^{\alpha}\log(1-\hat{H}_{xy}) & \text{otherwise.} \end{cases} \tag{4}$$

α and β are hyper-parameters of the focal loss, and N is the number of the location where the ground-truth matching probability equals to 1. We set $\alpha = 2$ and $\beta = 4$ in our implementation. For supervising the language outputs, we use the standard cross-entropy loss $\mathcal{L}_t$ following previous LMM-based methods [19, 18]. The overall loss function $\mathcal{L}$ can be formulated as:

$$\mathcal{L} = \lambda_h \mathcal{L}_h + \lambda_t \mathcal{L}_t, \tag{5}$$

where λ_h and λ_t are hyper-parameters for balancing two losses. During training, since the vision encoder, adapter, tokenizer, and large language model parts in Fig 2 inherit the weights of LLaVA, we only finetune the large language model using the LoRA [58] technique and train the V-L dense aligner under the supervision of the loss function $\mathcal{L}$.

3.2 Stage 2: Task-Specific Decoding

Decoding Modules. We first introduce the operations of the three key modules illustrated in Fig 2. (1) **Peak point selection** parses the heatmap into a set of points, where each point indicates the center location of an identified object or keypoint. Specifically, we filter the locations with heatmap response greater than their 8 surrounding neighbors as candidate peak points, and then retain the top K candidates as the selected results. The value of K can vary across different tasks, which will be elaborated on in Sec.4.1. (2) **Box decoder** is used for further regressing the extent of the objects designated by the selected peak points. For efficiency, instead of introducing an additional network as the box decoder, we reuse the V-L dense aligner by appending two new learnable special tokens after the `[LOC]` token as additional text-side inputs. Accordingly, the V-L dense aligner will generate two additional 1-channel map predictions, which are used for regressing the height and width under the supervision of the L1 loss functions similar to [56]. (3) **Promptable mask decoder** can accept both points and boxes as visual instructions to generate the mask for the referred object. We directly use the SAM model [50] for executing the above process without finetuning. In general, these decoding modules introduce negligible training costs and can be easily implemented.

Decoding Pathways. On the basis of these three decoding modules, we customize their different cooperation patterns to perform diverse tasks, which will be introduced as follows: (1) **Pose estimation** requires to return the keypoint coordinates. We can easily obtain these coordinates by parsing the predicted heatmap with the peak point selection operation. (2) **Object detection** and **visual grounding** share the same output format of bounding box, therefore, we employ the identical decoding pathway for them. Specifically, as shown in Fig 2, we feed the heatmap into the cascaded peak point selection and box decoder modules to generate bounding boxes. (3) **Instance segmentation** and **referring segmentation** share a relationship analogous to the one between object detection and visual grounding, therefore, we also adopt the same decoding pathway for them. Concretely, we first parse both the center point and bounding box of the object, and then, we use them as visual instructions to guide the promptable mask decoder to generate the mask prediction.

4 Experiments

4.1 Experimental Setup

Datasets. Our training data consists of datasets from the following different tasks. (1) For object detection, we use MSCOCO [59], Objects365 [60] and OpenImages [61]. (2) For visual grounding, we use RefCOCO, RefCCO+ and RefCOCOg [62]. (3) For pose estimation, we use MSCOCO keypoints [59] and AIC [63]. (4) For visual question answering, we employ a subset of ShareGPT4V dataset [33] with 665K samples. It is worth noting that for instance segmentation and referring segmentation, since we first transform their masks into bounding boxes as mentioned in Sec 3.1.2, the constructed heatmaps are the same as the object detection. Additionally, given that we do not finetune the mask decoder in the second decoding stage, we actually do not use segmentation annotations throughout the entire training process.

Model Configurations. For the first task-agnostic matching stage, we utilize pre-trained CLIP ViT-L/14-336 [55] and LLaVA-7B-v1.5 [4] as our vision encoder and large multimodal model, respectively. For the second decoding stage, since the different decoding modules and pathways have been introduced in Sec 3.2, we primarily specify the K value choices and corresponding post-processing details of different tasks here. (1) For object detection and instance segmentation, which generally includes multiple objects in a given image, we set $K = 100$ to generate 100 box candidates. Then we apply regular NMS [64] to filter redundant boxes, and the remaining boxes are used for further prompt mask decoder to generate instance segmentation results. (2) For visual grounding and referring segmentation, given that only one object is referred by a referring sentence, we set $K = 1$ to only generate one predicted box/mask, and no post-processing is required. (3) For pose estimation, we follow previous works [65, 20] to first crop the single object from the image using bounding boxes. Then, for each keypoint category, we set $K = 1$ to extract the point with the highest matching probability as the prediction result.

Training Details. For the task-agnostic stage, our training comprises two phases. (1) In Phase 1, we mix the object detection, visual grounding and pose estimation data with sampling rates of 0.69, 0.23, 0.08, respectively, for balanced data sampling. (2) In Phase 2, we mix the visual question-answering, object detection, visual grounding and pose estimation data with sample rates of 0.67, 0.23, 0.07, 0.03, respectively. We set the batch size to 160 and train the first step for 50,000 steps and the second step for 10,000 steps on 8 NVIDIA 80G A100 GPUs. The loss function balance terms λ_h and λ_t are both set to 1. For each phase, we use AdamW as the optimizer with an initial learning rate of 3×10^{-4} and weight decay of 0. During training, we do not calculate heatmap loss $\mathcal{L}_h$ for visual question-answering data. We do not use any data augmentation techniques for all tasks.

Evaluation Metrics. We adopt evaluation metrics commonly used within each field of task. For object detection and instance segmentation, we use mAP based on box IoU and mask IoU, respectively. For pose estimation, we use mAP based on OKS (object keypoint similarity). For visual question-answering, we comply with the evaluation protocol of each individual benchmark.

4.2 Results on Versatile Tasks

We evaluate our method on vision-centric and vision-language tasks as shown in Table 1 and 2. We categorize existing approaches into three groups, namely *"task-specific specialists"*, *"vision generalists"* and *"LMM generalists"*, according to their functions and architectures. (1) **Task-specific specialists** are customized models in different fields. They have diverse architectural designs and are limited to a single task. (2) **Vision generalists** pursue handling multiple tasks with a unified architecture. To excel in fundamental visual perception tasks, they typically utilize a powerful vision encoder or additional designs (e.g, data augmentations or decoding strategies) adaptive to vision-centric tasks. (3) **LMM generalists** aim to resolve every task in a conversational format. Focusing on improving the conversational quality, they adopt vision encoders proficient in multimodal content comprehension. In line with these methods, our Lumen pursues ***unleashing the inherent vision-centric capabilities of LMMs***.

Object Detection & Instance Segmentation. Object detection and instance segmentation require the model to make dense predictions across the image, therefore they pose great challenges in capturing fine-grained object cues. COCO val set is used for evaluation. (1) Compared with other LMM generalists, our Lumen surpasses them by clear margins, achieving a 15.6 AP_{50} boost over

Table 1: **Results on fundamental vision-centric tasks and vision-language tasks.** We use "-" and gray cell to indicate the result is not reported and the corresponding task is not supported by the method, respectively.

Method	Object Det.			Instance Seg.			Pose Est.			Grounding	Refer Seg.
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP_{50}	cIoU
Task-specific Specialists											
Faster R-CNN [66]	40.3	61.0	44.0								
DETR [67]	43.3	63.1	45.9								
Mask R-CNN [68]	41.0	61.7	44.9	37.1	58.4	40.1					
PolarMask [69]				30.5	52.0	31.1					
CPM [69]							62.7	86.2	70.9		
RTMPose [70]							68.2	88.3	75.9		
MDETR [71]										83.4	
SeqTR [72]										82.7	64.7
LAVT [73]											61.2
ReLA [74]											65.0
Vision Generalists											
Pix2Seq-v2 [20]	46.5	-	-	38.2	-	-	64.8	-	-		
Uni-Perceiver-v2 [43]	58.6	-	-	50.6	-	-					
mPLUG-2 [75]	46.9	-	-	40.6	-	-				84.7	
VisionLLM [52]	44.6	64.0	48.1	25.1	50.0	22.4				-	
LMM Generalists											
Shikra-7B [47]										82.3	
MiniGPT-v2-7B [16]										**84.4**	
Griffon-13B [19]	23.2	37.6	23.4							81.5	
InstructCV [76]	-	48.5	-								
Lumen-7B (Ours)	**35.3**	**53.2**	**35.8**	**30.4**	**49.8**	**31.0**	**67.2**	**90.4**	**75.6**	83.6	**65.1**

Table 2: **Results on prevalent VQA benchmarks.** Here, we employ English MMBench dev, SEEDBench image, MME test, MMMU val and MathVista mini sets for evaluation.

Method	Param	MMBench	SEED	MME	MMMU	MathVista
InstructBLIP [29]	7B	36.0	58.8	1213/292	32.9	**25.3**
MiniGPT-4 [3]	7B	24.3	47.4	582/144	-	23.1
Shikra [47]	7B	58.8	-	-	-	-
Qwen-VL-Chat [18]	7B	60.6	58.2	1488/**361**	**35.9**	-
LLaVA-v1.5 [54]	7B	64.3	**66.1**	**1511**/296	35.6	23.5
Lumen (Ours)	7B	**64.9**	65.8	1426/332	35.2	24.6

Griffon [19] and a 4.7 AP_{50} boost over InstructCV [76]. This indicates the effectiveness of our method in discovering dense object cues. (2) Compared with vision generalists and task-specific specialists, our method further approaches their performances. We analyze that the performance gap might originate from two major aspects. First, we use the input size of 448×448, which is much smaller than these competitors, for example, 1024×1024 in [20, 75]. Second, we do not introduce DETR-like object decoding techniques as done in [52, 75].

Pose Estimation. We employ COCO human 2D body keypoint val set for evaluation. Following the top-down paradigm employed by previous works [65, 20], we first crop a single object from the image using the bounding box. Therefore, the pose estimation is simplified to discovering keypoints of a single object. Since the LMM generalists do not perform this task, we only compare our method with vision generalists and task-specific specialists. As illustrated in Table 1, our method outperforms the vision generalist model Pix2Seq-v2 [20] with 2.4 AP. Meanwhile, our Lumen also achieves comparable performances with task-specific specialists.

Visual Grounding & Referring Segmentation. Compared to object detection and instance segmentation, visual grounding and referring segmentation underscore the language comprehension ability. Here, we report the results on RefCOCOg val set for comparison because the referring expressions in it are more diverse and abundant than those in RefCOCO and RefCOCO+. We also provide complete results on these three benchmarks in the Appendix B. As illustrated in Table 1, our method achieves better performances than Shikra [47] and Griffon [19] on visual grounding task, with AP_{50} 83.6 vs 82.3 and 83.6 vs 81.5, respectively.

Table 3: **Ablation studies on model designs.**

(a) Effect of different "V-L dense aligner" design choices.

Arch.	AP	AP_{50}	AP_{75}
Conv.	18.4	30.2	15.7
Trans.	**28.4**	**45.0**	**28.1**

(b) Effect of the pretrained mask decoder. cIoUs are reported.

Mask Dec.	RC	RC+	RCg
TransDec.	65.8	58.6	**62.0**
SAM	**66.6**	**59.2**	61.5

(c) Effect of LMM baseline.

Baseline	AP	AP_{50}	AP_{75}
LLaVA-v1.0	24.0	39.0	23.0
LLaVA-v1.0*	26.8	43.2	26.0
LLaVA-v1.5	**28.4**	**45.0**	**28.1**

Table 4: **Effect of multi-task training.** AP_{50} on RefCOCOg val set and COCO val set are reported.

#	Object Det.	Grounding	Pose Est.	VQA	RefCOCOg	COCO
1		✓	✓	✓	72.3	26.3
2	✓		✓	✓	24.6	41.2
3	✓	✓		✓	**75.8**	**45.7**
4	✓	✓	✓		75.0	44.8
5	✓	✓	✓	✓	74.8	45.0

Table 5: **Effect of the training recipe on VQA performances.** MMBench is used for evaluation[2].

#	Phase 1	Phase 2	AR	CP	FP-C	FP-S	LR	RR	Overall
1			**69.3**	74.3	54.5	66.6	28.0	55.7	62.5
2		✓	67.3	**80.0**	49.0	**67.9**	28.8	**58.3**	63.7
3	✓	✓	67.3	77.0	**58.7**	67.2	**33.0**	56.5	**64.2**

Visual Question Answering. To examine general visual understanding and instruction following of our model, we follow common practices and employ MMBench [77], MME [78], SEEDBench [79], MMMU [30] and MathVista [80] for evaluation. As shown in Table 2, our Lumen achieves VQA results comparable with state-of-the-art LMMs while extending versatile vision-centric capabilities.

4.3 Ablation Studies

For efficiency, we default to training the model for 10,000 iterations in the first phase of task-agnostic learning. This protocol is standard for our ablation studies

Multi-task Training. We use different task combinations for training the model, and the results are shown in Table 4. Based on the results, we have the following observations: (1) Compared with adopting datasets from all tasks (#5), discarding object detection data (#1) will reduce visual grounding performances. Similarly, removing visual grounding data (#2) also results in performance degradation on object detection. This demonstrates that object detection and visual grounding can benefit from each other. (2) Excluding pose estimation data leads to performance enhancements on both visual grounding and object detection (#3 vs #5). This might be caused by the data conflict problem addressed in many generalist models [52, 23]. (3) Excluding VQA data (i.e., discarding Phase 2) does not significantly affect model's performances on both detection and visual grounding, which is reasonable as VQA data do not provide explicit object-level vision-language alignment cues.

Training Recipe. We ablate the effects of 2-phase training strategy to inspect the contribution of vision-centric data on VQA tasks. In Table 5, we list detailed scores on diverse aspects covered by MMBench. #1 is initialized from LLaVA's weights, without any tuning. By comparing #2 and #3, we observe that FP-C (i.e, Cross-instance Perception) improves remarkably, which indicates that incorporating vision-centric data in Phase 1 can promote multi-instance perception in VQA (as well as the overall performance). Scores of Phase 1-only are not reported as the model trained without rich instruction data tends to respond template answers like *"Sure, the location is* `[LOC]` *."*

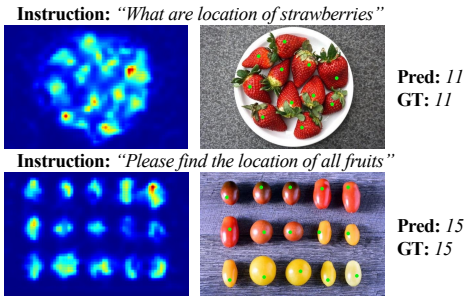
V-L Dense Aligner Architectures. We ablate different architecture design choices as shown in Table 3a, where "Conv." indicates the operation of concatenating feature of `[LOC]` token with every image patch features and fusing them with two convolutional layers, and "Trans." represents a light-weight transformer in our main method. Substituting our transformer-based aligner with simple

[2]Please refer to the official benchmark for the specific meaning of each term.

Table 6: Generalization ability evaluation on object detection. † indicates that VOC is not included in our training data, and thereby used for cross-dataset generalization evaluation.

Method	COCO	VOC†
RetinaNet [81]	-	77.3
Faster R-CNN [66]	-	80.4
Pix2Seq-v2 [20]	**57.4**	38.5
InstructCV [76]	48.5	61.7
Lumen (Ours)	53.2	**77.9**

Figure 4: Qualitiave results of our Lumen when generalizing to the object counting task.



convolutional layers incurs significant performance degradation. This result indicates that complete vision-language interaction is crucial in promoting explicit dense vision-language alignment.

Pretrained Mask Decoder. We adopt the mask annotations from RefCOCO (RC), RefCOCO+ (RC+) and RefCOCOg (RCg) to train a lightweight transformer-like mask decoder from scratch (TransDec.) and report cIoU on their val set to ablate the effect of leveraging pretrained SAM decoder in Table 3b. Although SAM outperforms TransDec., the narrow performance gap suggests that the primary challenge in advancing segmentation performances within our framework is the need to improve the dense vision-language alignment in the learned heatmap.

LMM Baseline. We ablate the effect of different LMM baselines on vision-centric task as shown in Table 3c. Since LLaVA-v1.5 employs CLIP ViT-L/14-336 as a stronger vision backbone, we use it to substitute the original CLIP ViT-L/14-224 in LLaVA-v1.0, denoted as "LLaVA-v1.0*" in Table 3c, to respectively ablate the effects of stronger vision backbone and multimodal language model. It can be proved that both stronger vision backbone and multimodal language model can benefit dense vision-language alignment.

4.4 Generalization Evaluation

Generalization to Unseen Datasets. To evaluate the generalization ability of our method, we perform zero-shot evaluation on PASCAL VOC2007 val set [82]. As illustrated in Table 6, our method demonstrates superior generalization ability than other generalist models. It is worth noting that compared with InstructCV which also inherits the enormous word vocabulary of LLM, our method outperforms it on VOC zero-shot evaluation by 16.2 AP. Besides, even compared with specialist models trained on VOC dataset (i.e, RetinaNet and Faster R-CNN in Table 6), our method still achieves comparable performances.

Generalization to Unseen Tasks. To prove that the heatmap produced by our first-stage task-agnostic training is a powerful intermediate representation for scaling up to more vision-centric tasks, we apply simple decoding rules on the heatmap to make our Lumen support the object counting task [83]. As shown in Fig 4 our Lumen can make the correct prediction even without specifically training on the object counting task.

5 Conclusions

In this paper, we present **Lumen**, a **L**arge m**u**ltimodal **m**odel with versatile vision-centric capabilities **en**hancement. Within the Lumen, we employ a decoupled design to initially promote learning task-agnostic vision-language dense alignment, and subsequently collaborate with task-specific decoding operations to address diverse tasks. With the task-agnostic and task-specific design, Lumen significantly broadens the range of visual tasks that existing LMM generalists can tackle, and also maintains general visual understanding and instruction following capabilities.

6 Acknowledgements

This work was supported in part by National Natural Science Foundation of China Project (No. 623B2027) and National Natural Science Foundation of China Project (No. 62072116).

References

[1] Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, et al. Map-neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv:2405.19327*, 2024.

[2] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

[3] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[4] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[5] Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. Mova: Adapting mixture of vision experts to multimodal context. *arXiv preprint arXiv:2404.13046*, 2024.

[6] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Peng Gao, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024.

[7] Zekun Wang, King Zhu, Chunpu Xu, Wangchunshu Zhou, Jiaheng Liu, Yibo Zhang, Jiashuo Wang, Ning Shi, Siyu Li, Yizhi Li, et al. Mio: A foundation model on multimodal tokens. *arXiv preprint arXiv:2409.17692*, 2024.

[8] Pengkun Jiao, Xinlan Wu, Bin Zhu, Jingjing Chen, Chong-Wah Ngo, and Yugang Jiang. Rode: Linear rectified mixture of diverse experts for food large multi-modal models. *arXiv preprint arXiv:2407.12730*, 2024.

[9] Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Vidcompress: Memory-enhanced temporal compression for video understanding in large language models. *arXiv preprint arXiv:2410.11417*, 2024.

[10] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[11] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. *arXiv preprint arXiv:2305.14836*, 2023.

[12] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Unleashing chain-of-thought reasoning in multi-modal language models. *arXiv preprint arXiv:2403.16999*, 2024.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[14] Yang Jiao, Shaoxiang Chen, Zequn Jie, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. More: Multi-order relation mining for dense captioning in 3d scenes. In *European Conference on Computer Vision*, pages 528–545. Springer, 2022.

[15] Bingqi Ma, Zhuofan Zong, Guanglu Song, Hongsheng Li, and Yu Liu. Exploring the role of large language models in prompt encoding for diffusion models, 2024.

[16] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[17] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023.

[18] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[19] Yufei Zhan, Yousong Zhu, Zhiyang Chen, Fan Yang, Ming Tang, and Jinqiao Wang. Griffon: Spelling out all object locations at any granularity with large language models. *arXiv preprint arXiv:2311.14552*, 2023.

[20] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

[21] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[22] Lingxi Xie, Longhui Wei, Xiaopeng Zhang, Kaifeng Bi, Xiaotao Gu, Jianlong Chang, and Qi Tian. Towards agi in computer vision: Lessons learned from gpt and large language models. *arXiv preprint arXiv:2306.08641*, 2023.

[23] Shaoxiang Chen, Zequn Jie, and Lin Ma. Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms. *arXiv preprint arXiv:2401.16160*, 2024.

[24] Zhijian Huang, Tao Tang, Shaoxiang Chen, Sihao Lin, Zequn Jie, Lin Ma, Guangrun Wang, and Xiaodan Liang. Making large language models better planners with reasoning-decision alignment. In *European Conference on Computer Vision*, pages 73–90. Springer, 2025.

[25] Yunxin Li, Baotian Hu, Xinyu Chen, Lin Ma, Yong Xu, and Min Zhang. Lmeye: An interactive perception network for large language models. *IEEE Transactions on Multimedia*, 2024.

[26] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[27] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[28] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[29] W Dai, J Li, D Li, AMH Tiong, J Zhao, W Wang, B Li, P Fung, and S Hoi. Instructblip: towards general-purpose vision-language models with instruction tuning. arxiv. *Preprint posted online on June*, 15:2023, 2023.

[30] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

[31] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[32] Ge Zhang, Xinrun Du, Bei Chen, Yiming Liang, Tongxu Luo, Tianyu Zheng, Kang Zhu, Yuyang Cheng, Chunpu Xu, Shuyue Guo, et al. Cmmmu: A chinese massive multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2401.11944*, 2024.

[33] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023.

[34] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*, 2024.

[35] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[36] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

[37] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[38] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022.

[39] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, 35:31333–31346, 2022.

[40] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer, 2022.

[41] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[42] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16804–16815, 2022.

[43] Hao Li, Jinguo Zhu, Xiaohu Jiang, Xizhou Zhu, Hongsheng Li, Chun Yuan, Xiaohua Wang, Yu Qiao, Xiaogang Wang, Wenhai Wang, et al. Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2691–2700, 2023.

[44] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*, 2023.

[45] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions. *arXiv preprint arXiv:2306.02224*, 2023.

[46] Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*, 2023.

[47] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

[48] Yang Jiao, Zequn Jie, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Suspected objects matter: Rethinking model's prediction for one-stage visual grounding. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 17–26, 2023.

[49] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[50] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[51] Yang Jiao, Zequn Jie, Weixin Luo, Jingjing Chen, Yu-Gang Jiang, Xiaolin Wei, and Lin Ma. Two-stage visual cues enhancement network for referring image segmentation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1331–1340, 2021.

[52] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023.

[53] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[54] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[56] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[57] Dongkai Wang and Shiliang Zhang. Contextual instance decoupling for robust multi-person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11060–11068, 2022.

[58] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[59] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[60] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[61] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[62] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.

[63] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017.

[64] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[65] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv preprint arXiv:2309.03895*, 2023.

[66] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[67] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[68] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[69] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12193–12202, 2020.

[70] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023.

[71] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[72] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 598–615. Springer, 2022.

[73] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165, 2022.

[74] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23592–23601, 2023.

[75] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. *arXiv preprint arXiv:2302.00402*, 2023.

[76] Yulu Gan, Sungwoo Park, Alexander Schubert, Anthony Philippakis, and Ahmed M Alaa. Instructcv: Instruction-tuned text-to-image diffusion models as vision generalists. *arXiv preprint arXiv:2310.00390*, 2023.

[77] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[78] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024.

[79] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[80] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

[81] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[82] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[83] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021.

[84] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15325–15336, 2023.

[85] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[86] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 10034–10043, 2020.

[87] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127, 2023.

Appendix

A Implementation Details

A.1 Detailed Design of V-L Dense Aligner

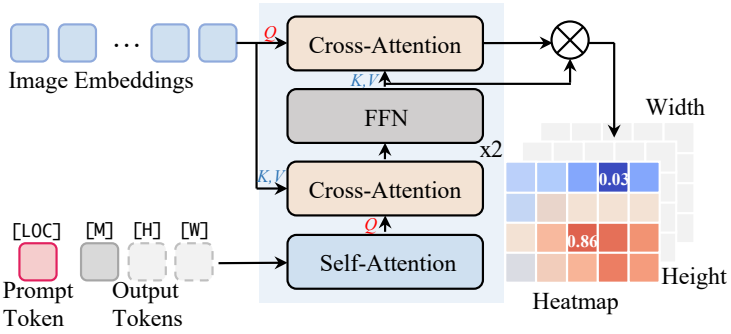


Figure 5: Detailed designs of "V-L Dense Aligner". [LOC] is the special token output from the LMM. [M] is the special token used for predicting the matching probabilities (i.e., heatmaps) in the first class-agnostic matching stage in the main paper. [H] and [W] are special tokens used as a simple box decoder for predicting additional height and width in the second class-specific decoding stage in the main paper. We use dotted lines to illustrate that these tokens are added in the second stage.

The detailed design of the transformer-based V-L dense aligner is shown in Fig 5. Motivated by SAM [50], we regard [LOC] token as a prompt token, which carries semantics conveyed in the instruction and is used to guide different output tokens, namely [M], [H] and [W], to generate corresponding predictions (i.e., heatmap, height and width in Fig 5). Specifically, the prompt token and output tokens are first interacted with a self-attention layer. Then, the resulting outputs are interacted with image embeddings through a sequence of cross-attention layers and a feed-forward network as shown in Fig 5 to perform dual attention. The above process is repeated twice. The final outputs are produced by performing an element-wise multiplication of the features of output tokens with the features of the image embeddings. For the sake of clarity and conciseness, we have omitted channel compression steps from Fig 5.

As for the output tokens, in the first stage, we only use [M] to generate the predicted heatmap. In the second stage, we add [H] and [W] to predict the height and width maps under the supervision of L1 loss as done in [56].

Table 7: Performance comparison with state-of-the-art specialists and generalists on the visual grounding task.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	test-A	test-B	val	test-A	test-B	val	test
Specialists								
UNINEXT [84]	92.64	94.33	91.46	85.24	89.63	79.79	88.73	89.73
G-DINO-L [85]	90.56	93.19	88.24	82.75	88.95	75.92	86.13	87.02
Generalists								
VisionLLM-H [52]	86.70	-	-	-	-	-	-	-
OFA-L [36]	79.96	83.67	76.39	68.29	76.00	61.75	67.57	67.58
Shikra-7B [47]	87.01	90.61	80.24	**81.60**	87.36	72.12	82.27	82.19
Lumen-7B (Ours)	**88.59**	**92.06**	**83.55**	80.35	**87.59**	**72.20**	**83.62**	**84.44**

B Complete Results on Visual Grounding and Referring Segmentation

We provide the complete performance comparison with existing approaches on visual grounding and referring segmentation tasks as shown in Table 7 and Table 8. We will analyze the results in the following parts.

B.1 Visual Grounding.

We report the visual grounding performances of our Lumen on RefCOCO, RefCOCO+ and RefCOCOg datasets in Table 7. First, compared with VisionLLM-H, a generalist that can also perform object detection and instance segmentation like us, our method outperforms it on the RefCOCO val set even without using a strong vision encoder, e.g., InternImage-H used in VisionLLM-H. Besides, compared with OFA-L and Shikra-7B, which reformulate the box prediction as a sequence generation manner, our method can achieve better or comparable performances with them. It indicates that our predicted heatmap can capture the correspondence between visual regions and complex language concepts.

B.2 Referring Segmentation.

We report the referring segmentation performances of our Lumen on RefCOCO, RefCOCO+ and RefCOCOg datasets in Table 8. First, compared with other generalists, our Lumen surpasses them across most benchmarks by clear margins. Besides, it is worth noting that we **do not use any pixel-level supervision** (i.e., segmentation masks) throughout our training process, but our method still achieves comparable performances with specialist models on the challenging RefCOCOg benchmark.

Table 8: Performance comparison with state-of-the-art specialists and generalists on the referring image segmentation task.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	test-A	test-B	val	test-A	test-B	val	test
Specialists								
MCN [86]	62.4	64.2	59.7	50.6	55.0	44.7	49.2	49.4
LAVT [73]	72.7	75.8	68.8	62.1	68.4	55.1	61.2	62.1
ReLA [74]	73.8	76.5	70.2	66.0	71.0	57.7	65.0	66.0
LISA [49]	74.1	76.5	71.1	62.4	67.4	56.5	66.4	68.5
Generalists								
X-Decoder [87]	-	-	-	-	-	-	64.6	-
Unified-IO [38]	46.4	46.1	48.1	40.5	42.2	40.2	17.3	-
InstructDiff [65]	61.7	65.2	60.2	46.6	52.3	39.0	**67.4**	-
Lumen (Ours)	**70.7**	**75.5**	**66.2**	**62.8**	**69.6**	**55.4**	65.1	**67.0**

C More Ablation Studies

C.1 Input Sizes

To examine the impact of various V-L dense aligner input resolutions, we resize the input image into three distinct sizes, concurrently adjusting the positional embeddings of the original CLIP ViT. As shown in Table 9a, enlarging the input size from 336×336 (default input size of CLIP ViT-L/14) to 448×448 can achieve nontrivial performance boosts. However, further increasing the size to 896×896 will harm the performances. This phenomenon indicates that excessively increasing the input size (to a scale dramatically different from its pretraining) will break the semantics of visual features, and thereby damaging the vision-language alignment.

Table 9: **Ablation studies on model designs.**

(a) Effect of different input sizes for the dense alignment.

Input Size	AP	AP_{50}	AP_{75}
336×336	26.1	40.2	25.6
448×448	**28.4**	**45.0**	**28.1**
896×896	24.5	36.4	22.5

(b) Effect of different pre-trained vision encoders.

Vis Enc.	AP	AP_{50}	AP_{75}
SAM ViT-H/16-1024	14.5	24.0	14.1
CLIP ViT-L/14-336	**28.4**	**45.0**	**28.1**

(c) Effect of different K value on dense prediction task.

$\#K$	20	50	100	200
AP	26.3	27.6	28.4	**28.5**
AP_{50}	39.5	43.2	45.0	**45.3**
AP_{75}	25.3	27.2	**28.1**	28.0

(d) Effect of different numbers of training iterations.

#Iteration	10k	20k	30k	40k	50k
AP	28.4	31.6	34.5	35.2	**35.3**
AP_{50}	45.0	49.2	52.4	53.2	**53.2**
AP_{75}	28.1	31.5	34.7	35.5	**35.8**

C.2 Vision Encoders

Since the input size can not be seamlessly scaled up as discussed above, it is worth further exploring to leverage vision encoders naturally adaptive to high-resolution image inputs. Toward this end, we employ SAM ViT-H/16 [50], which takes 1024×1024 images as inputs, as an additional vision backbone to provide high-resolution visual features for V-L dense aligner. However, as shown in Table 9b, the utilization of SAM ViT-H/16 results in a significant performance deterioration. We posit that this phenomenon can be attributed to the inadequate alignment between the SAM visual encoder and the language modality. Building upon the aforementioned experimental findings and analysis, we think that embracing vision encoders capable of processing high-resolution image inputs without compromising semantic coherence can pave the way for further enhancing the vision-centric capabilities of LMMs.

C.3 Number of *K* for Dense Prediction Tasks

We also investigate the impact of selecting different values for K on the dense prediction tasks, e.g., object detection. As shown in Table 9c, when K increases from 20 to 100, the model's performance consistently improves, which can be attributed to the increased recall of true positives. As K further increases from 100 to 200, the model's performance does not improve evidently, therefore, we set K as 100 in our implementation.

C.4 Training Epochs

We record the variations in detection performance as the number of training iterations increases in Table 9d. It can be observed that the model's performance slowly increases as the training proceeds. Due to the limitation of computational resources, we did not train our model for a longer time. The convergence speed might be limited by the optimization difficulty of utilizing a single `[LOC]` token to query the entire image regions. A feasible solution to mitigate this problem is to employ more special tokens like `[LOC]` serving as object queries in DETR [67] with the Hungarian matching for label assignment, which can accelerate training and promote performances as proved in VisionLLM [52]. We do not adopt this design as it is unclear whether these tokens customized for object detection can generalize well to other tasks.

D Limitations

Due to the vision-language dense alignment manner of the introduced V-L connector, one heatmap generated by it can only reflect the areas densely activated by a single instruction. To generate individual heatmaps for each category in standard vision-centric tasks like object detection and pose estimation, the model must be run separately for each category. This limitation can be addressed in the future by reformulating vision-centric data into instructions that simultaneously indicate multiple

categories. In our implementation, we use parallel programming techniques to expedite evaluation. As a result, our Lumen requires ~5 hours to evaluate COCO val set on a single A100, which is faster than Griffon [19] (~10 hours).

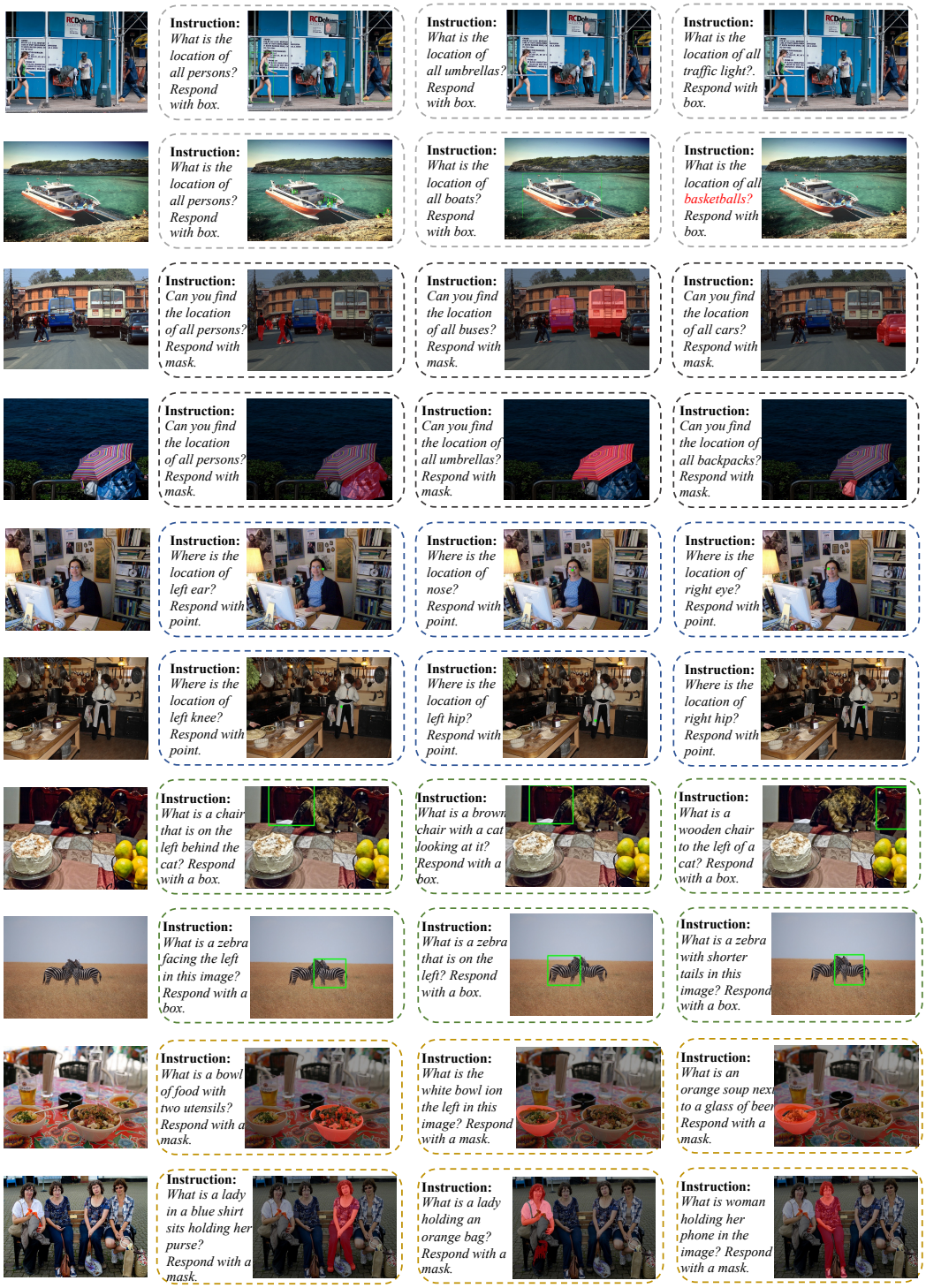


Figure 6: Qualitative results of versatile vision-centric capabilities of our proposed Lumen. We use different colors of dotted rectangles to denote performing different tasks.

E More Qualitative Results

We provide qualitative results to demonstrate the versatile capabilities of our Lumen as shown in Fig 6. Overall, the abundant qualitative results prove that our method possesses versatile vision-centric capabilities. It is worth noting that in the case within the second row, we instruct the model to detect the non-existing object (i.e., *"basketball"* in Fig 6), the maximum activation across the heatmap is relatively low, and filtered with a pre-defined threshold. Therefore, the model will not generate any output for this wrong instruction as illustrated in Fig 6.

F Broader Impacts

This research is expected to enhance the visual perception capabilities inherent in LMMs. Additionally, since our approach leverages open-source pre-trained LMMs and vision encoders, it requires low training resources, thus reducing the carbon footprint. Currently, we do not foresee any substantial negative ethical or social impacts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Main claims and contributions are clearly stated in the abstract and introduction, and are supported by experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix D for discussion of limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Theoretical assumptions and proofs do not apply to this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to Appendix 4.1 and A.1 for detailed training process and architectural designs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to the privacy policy of the institution the authors collaborated with, the code should be published only after obtaining authorization.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Sec.4.1 and Appendix 4.1, A.1 for experimental details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the standard evaluation protocol of each benchmark, and their evaluation results are stable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Appendix 4.1 for compute resource details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Appendix F for details.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper aims at equip the LMM with vision-centric capabilities, without introducing high-risk language data for training the model.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and pretrained models used in this study are open-sourced and licensed for academic research purposes.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: Due to the privacy policy of the institution the authors collaborated with, new assets like code should be published only after obtaining authorization.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Crowdsourcing and research with human subjects are not involved in this paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Human subjects are not involved in this paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.