MSc Artificial Intelligence
Master Thesis

Object-Guided Visual Tokens: Eliciting Compositional Reasoning in Multimodal Language Models

by
Matteo Nulli
15118290

July 11, 2025

36 Credits
06/01/2025 - 11/07/2025

Supervisor:
Dr. Ivona Najdenkoska
Vladimir Orshulevich
Examiner:
Prof. Cees G. M. Snoek

Second reader:
Prof. Yuki M. Asano

Universiteit van Amsterdam

Contents

Abstract

Standard Multimodal Large Language Models (MLLMs) employ contrastive pre-trained vision encoders whose performance, while undoubtedly good in a good range of tasks, falls short in Compositional Understanding and Reasoning on the visual input. This is mostly due their pre-training objective aimed at retrieval between similar image/captions rather than in-depth understanding of all components of an image. Moreover, while state-of-the-art image encoding methods yield strong performance, they inflate the number of visual input tokens by roughly two to three times, thereby significantly lengthening both training and inference times.
To alleviate these issues, we present **OG-LLaVA (Object-Guided LLaVA)**, a novel multimodal architecture which, through a novel connector design (`OG-Fusion`), enhances the model's ability to understand and reason about visual content without substantially increasing the number of tokens or unfreezing the Vision Encoder. A core element of `OG-Fusion` is the combination of CLIP output representations with segmentation masks. By leveraging the descriptive power of advanced segmentation models, **OG-LLaVA** attains superior performance at tasks which require a deeper understanding of object relationships and spatial arrangements and, more broadly, within the domains of compositional reasoning and visual grounding.[1]

[1]Work done during internship at eBay.

Chapter 1

Introduction

Recent advancements of Multi-modal Large Language Models (MLLMs) [43, 60, 66, 12] have consistently shown impressive capabilities in a wide variety of downstream applications, from simple image captioning [71, 10, 65] to broader visual-question answering [42, 21, 47] and complex reasoning [69]. Despite these innovations, MLLMs still struggle to obtain consistent results when faced with in-depth image understanding tasks. A central factor underlying this limitation is the absence of robust Compositional Reasoning (CR) [73, 29, 2]—the capacity to decompose a visual scene into constituent entities, model their spatial relations, and re-compose this information to draw coherent, context-sensitive inferences. Unlike surface-level pattern matching, compositional reasoning requires MLLM to build an internal spatial representation of the environment, track object identities, and reason over hierarchical relations such as containment, adjacency, or occlusion [29]. This ability enables a model not only to recognize objects but also to predict how changes in one region of an image might constrain possibilities in another, a prerequisite for tasks that demand fine-grained situational awareness. Closing this gap will require training regimes that explicitly reward spatial relational reasoning and architectures capable of dynamically binding visual and linguistic features into compositional representations.

Building on this observation, an emerging line of inquiry shifts the spotlight from language-driven training toward visual-centric strategies that explicitly structure the image input itself. Instead of treating the image as a monolithic token sequence, the authors of [42] presented the idea of Dynamic High Resolution, which consists in splitting the image into patches and encoding each image patch separately, allowing detailed Vision Encoder representations. Similarly, [24] propose an inference-time algorithm which segments images in nonoverlapping patches to avoid loss of information due to disappearing images or wrong warping when reshaping high-resolution inputs to custom input dimensions. Recent research [60] also argues that genuine progress in this field depends on advancing *visual representation learning*, rather than relying solely on improved *language understanding*, pushing for visual-centric data. Others like [9] improved existing image tiling methods to avoid image distortion by working on aspect ratio and area preservation. On a different approach, [76] and [72] have both leveraged vision encoder representations from Segmentation models [56, 38] to augmenting visual information and tokens and semantically partitioning the scene. While such pipelines all encourage the construction of disentangled, spatially grounded representations, these approaches all share a significant increase in the number of visual tokens given as input.

To turn this ever–growing set of patch- or segment-level features into signals that a language backbone can actually digest, multimodal systems interpose lightweight vision–language connectors (sometimes called adapters or projectors) between the vision encoder and the LLM. These modules transition these high-dimensional visual embeddings, aligning them with the linguistic token space, all while adding only a few million parameters. With the foundational BLIP-2 framework [37] the authors introduced the Q-Former, establishing an early template

for aligning high-dimensional visual embeddings with language decoders. Subsequently [43] demonstrated that a parameter-efficient Multi-Layer Perceptron (MLP) can substitute the heavier cross-attention blocks with negligible performance loss, making it the prevailing choice in many training pipelines. Nevertheless, passing generic vision-encoder features through such connectors imposes an information bottleneck—an issue compounded by the well-documented spatial myopia of CLIP representations [52, 73].

To confront the twin limitations of (a) an ever-expanding pool of visual tokens and (b) the weak spatial inductive biases of CLIP features, we introduce **OG-LLaVA**—an **O**bject-**G**uided extension of LLaVA that raises spatial acuity while keeping the visual token budget almost unchanged. At the heart of the architecture lies `OG-Fusion`, a lightweight connector that injects *object-centric* priors directly into the vision stream. Concretely, we obtain high-quality segmentation masks (via SAM2 [56]), (i) downsample each mask to the vision encoder output representation, (ii) apply each mask onto the vision encoder representations and (iii) create Object-Guided Visual Tokens through concatenation. This way each mask is represented by a segment of visual tokens which are directly proportional to the segmented region of the image. Because these segmentation are rarely overlapping—`OG-Fusion` keeps the visual sequence length on par with the vanilla LLaVA pipeline[1], thereby avoiding the quadratic cost explosion observed in aforementioned methods.

1.1 Main contributions

- *Object-Guided Visual Tokens.* We introduce **OG-LLaVA** a novel Multimodal Large Language Model architecture, with an innovative vision-language adapter `OG-Fusion`, a simple yet effective mechanism that fuses segmentation cues with CLIP features, reinstating spatial locality within visual features, providing the Large Language Model with Object-Guided Visual Tokens.

- *Unlocking Compositional Reasoning.* Across both the LLaVA-1.5 [41] and Cambrian1 [60] learning curricula, **OG-LLaVA** not only posts consistent quantitative gains over its vanilla counterparts[1], but also reveals a noticeably deeper grasp of how multiple visual elements combine and interact within an image. This translates into substantial performance jumps on dedicated benchmarks such as ConMe [29], ARO [73], and MMVP [61] and qualitative examples in Figure 1.1 further illustrate these model's improved ability to reason about object configurations, relative positions, layered semantics and the heightened robustness in modeling spatial relationships.

- *Token-efficient visual stream.* By representing each object with a compact block of tokens, `OG-Fusion` keeps the visual sequence length essentially unchanged—unlike tiling or cropping schemes—and thus retains computational efficiency of its vanilla pipelines[1].

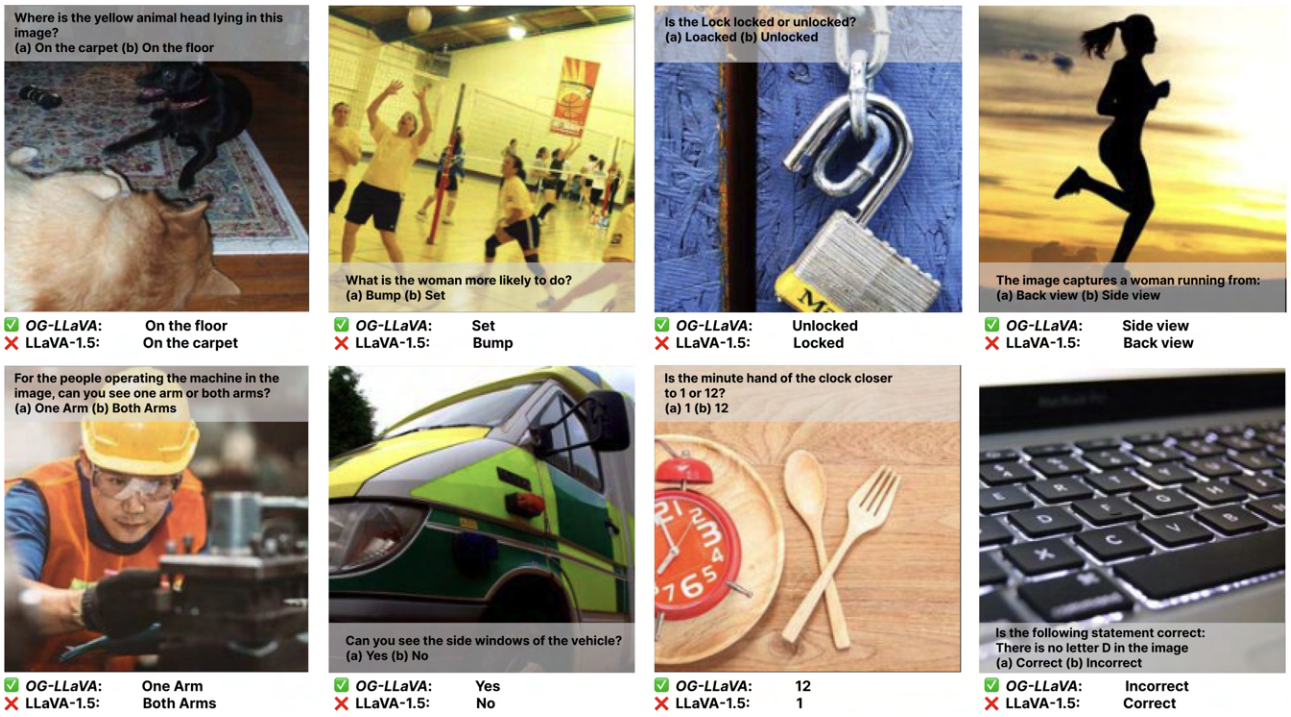
- *Superiority over alternative methods.* Compared with Subobject-level Image Tokenization [8], our **OG-LLaVA** attains considerably higher scores on both vision-centric and compositional tasks.

[1]With *"vanilla LLaVA pipeline" / "vanilla counterpart"* we indicate models trained with the standard training regime introduced in [43]. In particular when comparing to **OG-LLaVA**, we refer to models trained with identical backbones, data, and hyper-parameters, but with different connector design.

1.2 Thesis Outline

The remaining of the thesis is structured as follows: Chapter 2 surveys the relevant literature and foundational concepts: multimodal models, segmentation methods, the evaluation of MLLMs on compositional-reasoning and vision-centric tasks, and a concluding discussion of positional encodings in vision. Chapter 3 defines the research problem and introduces our solution, **OG-LLaVA**. We detail its architecture—highlighting the novel connector `OG-Fusion`—and provide a formal mathematical description of the overall pipeline, followed by the training and inference procedures. Chapter 4 presents the experimental work. We outline the training setup, datasets, and evaluation benchmarks, then compare **OG-LLaVA** across a broad set of models, backbones, and training strategies. Comprehensive ablation studies pinpoint the most effective components, and further experiments benchmark the system against state-of-the-art models. We also report a detailed qualitative analysis. Chapter 5 concludes the thesis with key takeaways, noted limitations, avenues for future research, and reflections on the broader applicability of our approach.

(a) ConMe [29] *replace-attribute* examples



(b) MMVP [61] examples

Figure 1.1: **OG-LLaVA vs LLaVA-1.5 on Compositional Reasoning**. Subfigures (a) and (b) visualize **OG-LLaVA** strengths across *replace-attribute* sub-task of the ConMe [29] benchmark and MMVP [61] benchmarks respectively.

Chapter 2

Background & Related work

2.1 Multi-modal Large Language Models

Multi Purpose MLLMs Since the advent of Visual Instruction Tuning [43], many have realized the impact of combining CLIP Vision Encoders [52] with Large Language Models (LLMs) [53, 16, 62, 20] to enable cross modality understanding with LLMs. Most notably LLaVA [43] and GPT4V [49], have paved the way for more diverse and varied MLLMs. Recent investigations have advanced along several complementary fronts. A first line of work systematically decomposing the training pipeline and characterizing model behavior across a variety of pre-trained backbones [46, 75, 35]. Parallel efforts address the efficient processing of images spanning multiple resolutions [42, 66, 50], the development of fully open multimodal foundation models like Molmo&Pixmo [22]. Multimodal Large Language Models have consistently achieved state-of-the-art results across a broad spectrum of downstream applications, encompassing image captioning [71, 10, 65], visual question answering [42], image understanding [43, 60], and complex reasoning tasks [69], as well as a range of additional settings [21, 47]. Yet, despite these impressive broad-spectrum achievements, equipping MLLMs with genuinely deep image comprehension remains a more demanding objective.

Vision-centric MLLMs Attaining truly deep image comprehension in Multimodal Large Language Models remains an exceptionally challenging endeavor. Recent work has proposed specialized architectures for vision-centric tasks that demand advanced computer-vision reasoning—an ability still difficult to realize in practice [42, 60, 50, 13]. Building on these efforts, the past few years have witnessed a surge of research aimed at strengthening the vision-grounding capabilities of MLLMs. Approaches range from redesigning the vision encoder to support arbitrary input resolutions [50, 4] to exploring refined spatial cropping and sampling strategies. A concrete example of these architectural refinements is the Dynamic High Resolution (DHR) paradigm proposed by [42], which operationalizes the vision-encoder redesign by extracting and separately encoding image patches, thereby enabling more detailed representations.

In a similar fashion the paper [24] presents an inference-time procedure that divides images into non-overlapping segments to prevent information loss resulting from vanishing regions or incorrect warping when adapting high-resolution inputs to fixed dimensions. The research team behind [9] further refined existing image-tiling approaches by enforcing both aspect-ratio and area preservation, effectively mitigating image distortion. Different approaches include augmenting training sets with spatially aware data [60], leveraging traditional computer vision techniques such as 3D scene graphs [13].

2.2 Segmentation Models

With the rise of Large Language Models (LLMs) [6], the field of computer vision has seen a shift towards NLP inspired approaches. One of the most notable example comes from Segment Anything Model (SAM) [33]. SAM defines a new promptable visual segmentation task allowing any user to define a segmentation mask for any object in an image, through a series of cue-like natural language, bounding boxes, and or points. More crucially, the authors of SAM released a new large scale open source dataset comprised of 1B masks and 11M images (SA-1B). Later works like OMG-Seg [38] focus on improving upon Segment Anything by delving into more complex segmentation tasks, supporting a wider range as well as significantly reduce the computational costs. The authors employ a unified transformers-based [63] encoder-decoder architecture, following [14] using a CLIP [52] based vision encoder. Concurrently to OMG-Seg, Segment Anything Model 2 (SAM2) [56] was presented. SAM2 expands the original SAM idea to support video segmentation while improving on existing image segmentation performance. In Figure 2.1 we report the overall process of SAM2. Each frame of a video (image) goes through a Hiera [57] Vision Encoder and through a memory attention mechanisms. This block is adjusting the output representation of the vision encoder through temporal information from previous frames. Later the mask encoder embedds the masks, points and/or boxes from the prompt, which leads to the final segmentation prediction. Lastly the information is stored into memory and added to the past memory of temporal sequence of the frames.

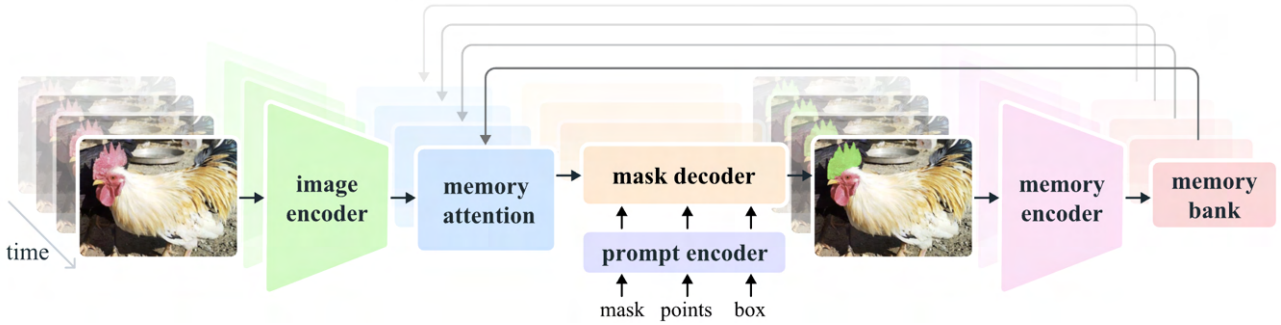


Figure 2.1: **SAM2 Architecture**: Segment Anything Model 2 [56] archiecture. We show the overal pipeline happening for any given video frame.

Segmentation infused VLMs Segmentation models have recently become very accurate in defining segmentation masks in both prompted and un-prompted scenarios. Due to these impressive capabilities, integrating segmentation capabilities into MLLMs has been a key focus in recent vision-centric research efforts. Notably, [72] and [76] have demonstrated how segmentation can be leveraged to create pixel-level representation as well as allow the model to perform both question-answering and segmentation at the same time. While other research [8] has shown how to leverage mask information to modify the visual tokens provided to the Language Model.

2.3 Compositional Reasoning & Visual Grounding

In spite of the recent advancements in MLLMs, Visual Grounding yet remains a relatively open track in research. Unlocking full image understanding capabilities in MLLMs remains challenging, and among its most exciting sub-fields is Compositional Reasoning (CR). This is the ability of a model to understand the relations and attributions between objects, while reasoning about their placement within the image. Even though this is easy to humans, MLLMs have a very hard time understanding the complexity and reasoning within the visual domain.

Evaluating Compositional Reasoning Notably, [59, 73] both address these issues in Vision-Language Models like CLIP [52], which are the visual encoding backbones to most Multi-Modal models. [73] argue that true compositional reasoning is directly in conflict with the standard contrastive pre-training of Vision Encoders [52] due to their retrieval oriented objective. They argue that these models tend to well in most benchmarks because they lack order and compositional cues. To solve the issue they propose the ARO benchmark, showing how poorly VLMs act in scenarios where the CR is crucial, an example being *"the horse is eating the grass"* vs *"the grass is eating the horse"*. In parallel [59] introduced Winoground[1] benchmark, tasked with a similar objective. More recently [48] have looked into how this issue translates into MLLMs, how In-Context-Learning (ICL) [6] can be leveraged to partially alleviate it and on the comparison between generative [43] and contrastive models [52]. Others [77, 26, 29] have looked into pitfalls of CR evaluation strategies, and how to improve upon them proposing new benchmarks like VL-Checklist [77], SugarCrepe [27] and ConMe [29]. Similar works [2], discuss the impacts of fine-grained understanding of spatial relation, count and object-attribute relations when including compositional reasoning data within their training mixture in both contrastive and generative VLMs. The authors also present a benchmark (Vis-Min), which stems from existing CR benchmarks, and is created by focusing on four kinds of minimal-changes within object, attribute, count and spatial relations. More information and pictures of the benchmarks can be found in Figure A.3, A.4.

Evaluating Visual Grounding In addition to Compositional Reasoning, other works have looked into the visual understanding capabilities of MLLMs. Famously, [61] explored the difference in feature understanding between DINO-v2 [51] and CLIP [52] embedding spaces. To this end they create MMVP (Multimodal Visual Patterns), a benchmark with CLIP-blind pairs of visual-question answers. Together with Cambrian [60], the authors presented CVBench, tasked with determining the level of spatial awareness in both 2D and 3D. Others have presented similar benchmarks, like RealworldQA [68] and V* [67]. More information and pictures of the benchmarks can be found in Figure 4.1, A.5.

2.4 Positional Encodings

One of the most crucial aspects in Language has always been the ability of encoding the relative and sequential position of words with respect to each other. Traditional Recurrent Neural Network Systems [25], due to their inherently periodic structure never had to face this challenge. It was only when the self-attention mechanism was introduced [3] and later applied on transformers [64] when the sequential nature of the model was not something which researchers could rely on.

Standard Positional Encodings In classical positional-encoding (PE) methods, the projections are typically applied to the sum of the token embedding and a positional-embedding vector. [63] were the first to introduce a non-learnable sinusoidal PE. Later work [18, 53] adopted various learnable absolute position encodings, replacing fixed vectors with trainable ones.

Rotary Positional Encoding More recent research led to Rotary Positional Encoding (RoPE) [58]. RoPE encodes position within the attention mechanism itself by rotating the query and key vectors so that their relative positions are reflected in the rotation angle. The embedding space is divided into two-dimensional subspaces, each associated with a particular

[1] the Winoground Hugging Face benchmark

angular frequency that dictates how much rotation is applied. A graphical illustration is provided in Figure A.1.

Positional Encodings in Vision In the visual domain, Convolutional Neural Networks [36], with their inherent spatial awareness, were long the standard for computer-vision tasks. That changed with Vision Transformers (ViTs) [19], where the internal representation of image patches lacks spatial information. [19] tried many configurations but ultimately adopted a one-dimensional learnable positional embedding, giving the transformer no explicit two-dimensional structure. Recently, Pixtral [1] has leveraged RoPE [58] instead of the standard one-dimensional positional encoding used in ViTs. Pixtral applies RoPE directly to the patch embeddings, encoding height and width positions in alternating dimensions.

Chapter 3

Methodology

3.1 Problem Formulation

Let $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ be an image and $t \in \Sigma$ be a language instruction input, where Σ is the input space of character sequences. Let $s_{\theta,\gamma,\phi}$ be a Multimodal Large Language Model, parametrized by θ, γ, ϕ, and $f_{v\theta}$ be a contrastive pre-trained Vision Encoder model, defined as:

$$f_{v\theta} : \mathbb{R}^{C \times H \times W} \to \mathbb{R}^{V \times F},$$

where V is the number of visual tokens and F their hidden size and $f_{t\theta'}$ is the corresponding Text Encoder. And let $m_\gamma : \mathbb{R}^{V \times F} \to \mathbb{R}^{V \times D}$ be a Multi-Layer Perceptron with two hidden layers. The token vocabulary is defined as:

$$\mathcal{V} \;=\; \mathcal{V}_{\text{vision}} \;\cup\; \mathcal{V}_{\text{text}}.$$

The Large Language Model is defined as follows:

$$g_\phi \;:=\; \mathcal{D}_d \;\circ\; \text{softmax} \;\circ\; F_{\phi'} \;\; : \; \mathbb{R}^{J \times D} \;\longrightarrow\; \mathcal{V}^J, \qquad \phi = \big(\phi', d\big),$$

where $F_{\phi'}$ is the transformer that produces logits and $\mathcal{D}$ a decoding operator (greedy, top-k, nucleus, ...) with hyper-parameters d. Thus g_ϕ maps an embedded input token sequence to an output token sequence.

Vision-Language Modeling For clarity, we describe the standard pipeline of MLLMs during both training and inference, assuming a batch size of 1. Vision Encoders $f_{v\theta}$, such as CLIP [52], are used in MLLMs to encode an image $\mathbf{X}$ into a representation:

$$\mathbf{X}' = f_{v\theta}(\mathbf{X}) \in \mathbb{R}^{V \times F}$$

where F is the feature dimension and V is the vision encoder hidden dimension $V = (\frac{\textit{image resolution}}{\textit{patch size}})^2$. Here *image resolution* corresponds to the $f_{v\theta}$ inner resizing to a specific resolution during pre-processing, and *patch size* is a pre-defined hyperparameter specifying the size of each patch when splitting the image at the beginning of the process. Refer to Figure A.2 and the paper [19] for an in-depth explanation of hyperparameters in CLIP-like encoders. Subsequently $\mathbf{X}'$ is transformed through m_γ into Visual Tokens ($\mathbf{VT}$)

$$\mathbf{VT} = m_\gamma(\mathbf{X}') \in \mathbb{R}^{V \times D},$$

which exist in the input space of the Large Language Model. In parallel a Tokenizer $\mathcal{T} : \Sigma \to \mathcal{V}^J$ and a learned embedding $E : \mathcal{V}^J \;\longrightarrow\; \mathbb{R}^D$, turn t is turned into textual tokens defined as:

$$TT = E^{\otimes}(\mathcal{T}(t)) \in \mathbb{R}^{J \times D},$$

where $E^{\otimes}$ is the sequence-wise lifting of operator E. Lastly $\mathbf{VT}$ together with TT are given as input to the g_ϕ obtaining the output tokens $\mathbf{T}_a$

$$\mathbf{T}_a = g_\phi(\mathbf{VT} \oplus TT) \in \mathcal{V}^J. \tag{3.1}$$

Vision Embeddings lack Spatial Awareness Vision Encoder outputs $\mathbf{X}'$ often suffer from poor spatial understanding and produce embedding representations, lacking in-depth understanding of the relation between objects in an image. This is mostly due to their contrastive pre-training objective, whose main goal is to match the best pairs of image and caption. Mathematically, during training each encoder extracts the feature representations $t' = f_{t\theta'}(t),\ \mathbf{X}' = f_{v\theta}(\mathbf{X})$. These are then normalized as follows:

$$\mathbf{X}'_e = t'_e \ := \ \frac{\mathbf{X}'\, t'}{\|\mathbf{X}'\, t'\|_2} \ ,$$

and used to compute the pairwise cosine similarities, $\mathit{logits} = (\mathbf{X}'_e \cdot t'^T_e) \cdot e^t$, where t'^T_e is the transpose of t'_e. These logits are finally used to compute the joint loss function using cross-entropy (CE) as defined:

$$\begin{aligned} \mathcal{L}_{\mathbf{X}} &= \mathrm{CE}(\mathit{logits}, \mathit{labels}, \mathrm{axis} = 0), \\ \mathcal{L}_t &= \mathrm{CE}(\mathit{logits}, \mathit{labels}, \mathrm{axis} = 1), \\ \mathcal{L} &= \tfrac{1}{2}\left(\mathcal{L}_{\mathbf{X}} + \mathcal{L}_t\right). \end{aligned} \tag{3.2}$$

where *labels* are the ground-truths for that sample, and axis $= i$, with $i \in 0, 1$ represents the dimension where the loss is computed through. Averaging the image- and text-based losses urges the model to downplay fine-grained visual details and instead prioritize broader, high-level concepts, thereby discarding subtle nuances, as is also evident in other research [73, 61]. Inevitably, this lack of spatial understanding translates into poor visual tokens representations $\mathbf{VT}$, and subsequently in the ability of g_ϕ to answer compositional and visually grounded questions.

Our objective is to alleviate the issue of contrastive pre-training without substituting nor re-training the $f_{v\theta}$ backbone. To this end, we propose a novel MLLM architecture adjusting the representations $\mathbf{X}'$ and Visual Tokens $\mathbf{VT}$ with the help of Segmentation models.

3.2 Object-Guided Visual Tokens

Given $\mathbf{X} \in R^{C \times H \times W}$ denotes a single input image, let:

$$\mathbf{M} = \{\mathbf{m}_i \mid i = 1, \ldots, N\} \subset \mathbb{R}^{H \times W},\ \ \mathbf{m}_i \in \{0, 1\}^{H \times W},$$

be its corresponding list of binary masks of length N. Our intention is to produce a set of segmentation-aware Visual Tokens, where each length-varying token segment can be mapped to one of the masks. To this end, we encode the image information through a vision encoder and combine this with a down-sampled representation of the segmentation maps. In the next sections, we describe our internal process and, for simplicity, assume a batch size of one.

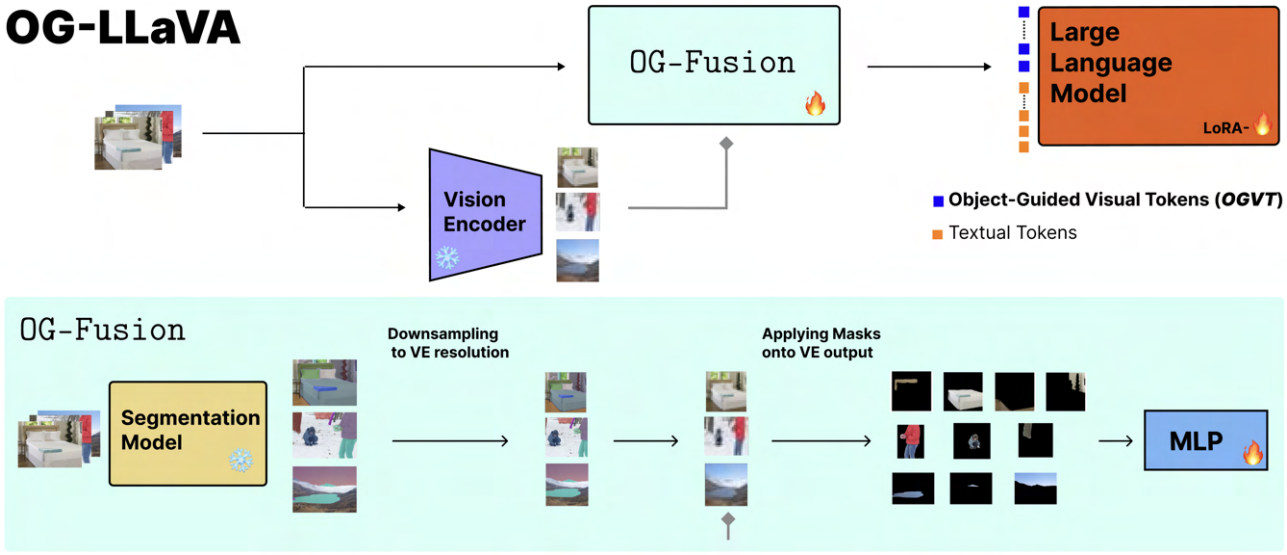


Figure 3.1: **OG-LLaVA architecture with `OG-Fusion` internal process**: We extract visual features from the input image through a Vision Encoder. Concurrently, we pass the input image through `OG-Fusion`. Here we (i) use a Segmentation model to retrieve the masks then (ii) downsample the segmentations and (iii) apply these masks onto the visual features. These Visual Embeddings are then (iv) concatenated together and passed through a Multi-Layer Perceptron to produce Object-Guided Visual Tokens (***OGVT***). The ***OGVT*** are then given as input to a Large Language Model together with Textual Tokens to produce an output. The snowflake and fire represent modules whose parameters are kept frozen or turned on. LoRA emphasizes that not all parameters of the LLM are unfrozen, only the LoRA layers.

Masks & Features Extraction Through a segmentation model, we first obtain a set of binary masks $\mathbf{M}$ for each image. During training, we extract the visual features from a vision encoder ($f_{v\theta}$):

$$\mathbf{X}' = f_{v\theta}(\mathbf{X}) \in \mathbb{R}^{V,F}, \tag{3.3}$$

Once the visual features $\mathbf{X}'$ are extracted, we retrieve the masks $\mathbf{M}$ and apply an ad-hoc downsampling operator.

Downsampling Operator We define our downsampling operator Φ_α as the functional composition of four elementary operators described below. First, we apply a flattening operation:

$$\mathcal{F} : \{0,1\}^{H\times W} \longrightarrow \{0,1\}^{HW}, \quad \mathcal{F}(\mathbf{M}) = \mathrm{vec}(\mathbf{M})$$

yielding $\mathbf{m}_{\text{flat}} = \mathcal{F}(\mathbf{m}_i) \in \{0,1\}^{HW}$. Then we proceeded by performing an average pooling into V bins. We divide the index set $\{1,\dots,H\cdot W\}$ into equally-sized, contiguous blocks $B_k,\ k = 1,\dots,V$ of length $\text{size}_b = H\cdot W/V$, corresponding to the average number of pixels per output bin.
Subsequently, we define:

$$\mathcal{P}_V : \mathbb{R}^{HW} \longrightarrow \mathbb{R}^V, \quad \big[\mathcal{P}_V(\mathbf{x})\big]_k = \frac{1}{|B_k|}\sum_{n\in B_k} x_n$$

so that, $\mathbf{m}_{\text{pool}} = \mathcal{P}_V(\mathbf{m}_{\text{flat}}) \in [0,1]^V$ stores the *fraction* of entries corresponding to "1" in each bin.
We later scale to pixel counts by defining:

$$\mathcal{S}_{\text{size}_b} : \mathbb{R}^V \longrightarrow \mathbb{R}^V, \quad \mathcal{S}_{\text{size}_b}(\mathbf{x}) = \text{size}_b \cdot \mathbf{x}$$

which produces: $\mathbf{m}_{\text{count}} = \mathcal{S}_{\text{size}_b}(\mathbf{m}_{\text{pool}}) \in \mathbb{R}^V$, i.e. the estimated *number* of mask pixels per bin. Finally we threshold $\mathbf{m}_{\text{count}}$ with an Indicator function

$$\mathcal{T}_\alpha : \mathbb{R}^V \longrightarrow \{0,1\}^V, \quad \big[\mathcal{T}_\alpha(\mathbf{x})\big]_k = \mathbf{1}\big\{x_k \geq \alpha\big\},$$

returning $\mathbf{m}'_i = \mathcal{T}_\alpha(\mathbf{m}_{\text{count}}) \in \{0,1\}^V$, a down-sampled binary mask whose entries indicate whether bin k contains at least α foreground pixels.
Collecting the four steps:

$$\Phi_\alpha = \mathcal{T}_\alpha \circ \mathcal{S}_{\text{size}_b} \circ \mathcal{P}_V \circ \mathcal{F} \quad \Longrightarrow \quad \mathbf{m}'_i = \Phi_\alpha(\mathbf{m}_i).$$

Applied to all N masks, we obtain the set of down-sampled masks:

$$\mathbf{M}' = \Big\{ \Phi_\alpha\big(\mathbf{m}_i\big) \;\Big|\; i = 1, \ldots, N \Big\} \subset \{0,1\}^V. \tag{3.4}$$

A Python implementation of the operator Φ_α is available in the Appendix A.2.2

Applying Segmentation After these pre-processing steps, we can apply the down-sampled segmentations $\mathbf{M}'$ onto the representation of the vision encoder $\mathbf{X}'$. Practically, for every sample i we turn the mask $\mathbf{m}'_i$ into an *index set*[1]

$$\mathcal{J}_i = \big\{\, j \in \{1, \ldots, V\} \mid (\mathbf{m}'_i)_j = 1 \big\}, \qquad t_i = |\mathcal{J}_i| = \|\mathbf{m}'_i\|_0.$$

Arrange the elements of $\mathcal{J}_i$ in ascending order $j_1 < \cdots < j_{t_i}$ and define the *row-selection matrix*

$$P_i = \begin{bmatrix} e_{j_1}^\top \\ \vdots \\ e_{j_{t_i}}^\top \end{bmatrix} \in \{0,1\}^{t_i \times V},$$

where e_j is the j-th canonical basis vector in $\mathbb{R}^V$.
Multiplying by P_i simply *keeps* the rows whose indices are in $\mathcal{J}_i$ and discards the rest, yielding

$$\boxed{\mathbf{Y}_i = P_i \mathbf{X}' \in \mathbb{R}^{t_i \times F}} \qquad (i = 1, \ldots, N). \tag{3.5}$$

The matrices P_i contain no learnable parameters; they merely *select and reorder* rows of $\mathbf{X}'$ in a deterministic, object-guided manner.

Object-Guided Visual Tokens With the down-sampled visual fragments $\mathbf{Y}_i \in \mathbb{R}^{t_i \times F}$ $(i = 1, \ldots, N)$ derived in previous section, we can denote their *row-wise* concatenation ($\big\|$) by

$$\mathbf{Y} = \big\|_{i=1}^N \mathbf{Y}_i = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix} \in \mathbb{R}^{T \times F}, \qquad T = \sum_{i=1}^N t_i. \tag{3.6}$$

We then project every F-dimensional visual embedding onto the g_ϕ embedding space $\mathbb{R}^D$ by the learned linear map m_γ applied row-wise to $\mathbf{Y}$

$$\boxed{\boldsymbol{OGVT} := m_\gamma(\mathbf{Y}) \in \mathbb{R}^{T \times D}} \tag{3.7}$$

Because T equals the total number of object-bearing bins retained by the masks, its value varies *per image*—yet in expectation $T \approx V$.
We lastly feed the concatenation of visual and textual tokens to the Large Language Model g_ϕ:

$$\mathbf{T}'_a = g_\phi\big([\boldsymbol{OGVT}; \mathbf{TT}]\big) \tag{3.8}$$

where $[\,\cdot\,;\,\cdot\,]$ denotes sequence concatenation along the token (row) dimension.

[1]Here $\|\mathbf{m}'_i\|_0 = \sum_{j=1}^V (\mathbf{m}'_i)_j$ denotes the ℓ_0-pseudo-norm (number of non-zeros).

3.3 Model Architecture

Vision Encoder To keep the same architectural structure as LLaVA-1.5 we choose as $f_{v\theta}$ a CLIP ViT-L/14@336, which has patch size of 14 and image resolution corresponding to 336 [19, 52]. Regardless of our particular choice of $f_{v\theta}$ our approach is applicable to any kind of Vision Encoder backbone.

`OG-Fusion` In Figure 3.1 we show an overview of our `OG-Fusion` process. Architecturally, it is comprised of a Segment Anything Model 2 (SAM2) [56] with a frozen backbone along with a series of detailed procedures from Sections 3.2 and a Multi-Layer Perceptron m_γ with 2 hidden layers with input size of 1024 and GeLU activations. SAM2 is used to extract the segmentation masks from the input image before training, then the we apply the steps from Eq. 3.4 to 3.7 employing m_γ to project the masked features into the same dimension of the Large Language Model input tokens, to obtain the Object-Guided Visual Tokens (***OGVT***).

Large Language Model We experiment mostly using Llama3.1-8B-Instruct [62] as our Large Language Model g_ϕ. We also employ Llama3.2-3B-Instruct [47] for some experiments in light of its lower parameter count and faster adaptability. The final architecture is visible in Figure 3.1.

Given these considerations, in this work we present two versions of **OG-LLaVA**, both with CLIP ViT-L/14@336 as backbone and `OG-Fusion` as Adapter:

- **OG-LLaVA-3B**: trained with Llama3.2-3B-Instruct as LLM,
- **OG-LLaVA-8B**: trained with Llama3.1-8B-Instruct as LLM.

3.4 Model Training & Inference

Following [43] and our definitions in Section 3.1, we train our model $s_{\theta,\gamma,\phi}$ using an auto-regressive objective. Specifically, we optimize the likelihood of:

$$p(\mathbf{T}'_a|\mathbf{X}, t) = \prod_{i=1}^{L} p_{\theta,\gamma,\phi}(x_i|\mathbf{X}, t_{<i}, \mathbf{T}'_{a,<i}), \tag{3.9}$$

where L is the length of the sequence, θ, γ, ϕ are the parameters of s, p is the probability of the target answers $\mathbf{T}'_a$ and $t_{<i}, \mathbf{T}'_{a,<i}$ correspond to the language instruction and target value until the prediction of x_i. With this loss, we train $s_{\theta,\gamma,\phi}$ in two different stages.

Vision-Language Alignment (VLA) We first perform vision-language alignment stage where we only unfreeze m_γ and do not train $f_{v\theta}, g_\phi$. The main objective of this phase is to teach m_γ to align the visual features $\mathbf{Y}$ into the input space of g_ϕ.

Supervised Fine-Tuning (SFT) The second stage is Supervised Fine-Tuning. Here we unfreeze m_γ and unfreeze only the linear weights of g_ϕ through Low-Rank Adaptation (LoRA) [28]. The target of SFT is to instruction tune the model to the tasks at hand and to steer the weights of g_ϕ just enough to incorporate information from the image $\mathbf{X}$.

Flexible Inference One peculiar aspect of our approach is its flexibility during inference time. During training we follow Eq. 3.3 through 3.7, which means that we train our model with a total number of $T(\approx V)$ visual token. However, at test time, our approach is robust enough to be evaluated with Object-Guided Visual Tokens ($T(\approx V)$), as well as *without* (V). Where the *without* corresponds to the standard multimodal information passing described in Section

3.1. This advantage derives from selectively re-utilizing components of the original $f_{v\theta}$ feature representation and subjecting them to minimal, targeted modifications. These adjustments preserve the model's semantic understanding of $\mathbf{X}'$ while strengthening its spatial awareness.

Chapter 4

Experiments

4.1 Experimental Setup

4.1.1 Training Implementation

Training Data As anticipated in Section 3.4 we train our model in two stages, Vision-Language Alignment (VLA) and Supervised Fine-Tuning (SFT). In the first, we train using BLIP-Laion dataset [43], comprised of 558k image-caption pairs, generated through BLIP2 [37] with images gathered from LAION dataset.
In SFT, we verify our approach on two different training datasets: *LLaVA-1.5* [41], which consists of 665k unique images each with multi-turn chat-like instructions [1], and *Cambrian-7M* [60], comprised of 7M image-text pairs, with 3.5M unique images. Specifically, we leverage the Cambrian7M system prompt version.

Setup Following the training setup of LLaVA1.5 [41], we employ the deepspeed library [55] with ZeRO-3 [54] to enable training of large models. We use a cosine learning rate scheduler with a warm-up ratio of 0.06 and perform gradient checkpointing to save memory and a batch size of 8 per device during captioning and 4 during SFT. We use the AdamW optimizer [32, 45] with a weight decay of 0 and a maximum gradient norm of 0.3. In total, we train for 1 epoch each stage, train in bfloat 16 and use a learning rate of 1e-4 for the `OG-Fusion` during VLA and 1e-4 for both the connector and the LoRA [28] layers in SFT, which we unfreeze only during SFT with a LoRA rank of 128 and a LoRA alpha of 256. Table 4.1 summarizes the training setup used in our experiments.

4.1.2 Evaluation Datasets & Metrics

Before we begin our analysis, we look into which are the best ways to evaluate and understand the performance of our proposed technique. As our aim is to elicit compositional reasoning and visual grounding, we will employ task-specific benchmarks. Furthermore, we evaluate the model on a suite of general-purpose image-understanding tasks to ascertain whether the gains achieved on the first two objectives can be realized without compromising overall performance.

Compositional Reasoning Benchmarks One of the most foundational works in compositional understanding and reasoning research is [73], whose visionary work introduced **ARO** benchmark[2]. This set, made up of 23937 samples, was uniquely constructed to evaluate the model's ability to understand the visual input by providing it with 5 examples of captions,

[1] LLaVA-1.5 Mixture-665k
[2] ARO Hugging Face benchmark

Parameter	Value
Training Stages	2 (Vision-Language Alignment, Visual SFT)
Optimizer	AdamW
Learning Rate	1e-4 (`OG-Fusion`), 1e-4 (LoRA layers)
Learning Rate Schedule	Cosine
Weight Decay	0.
Warmup Ratio	0.06
Batch Size	8 (VLA), 4 (SFT)
Gradient Accumulation Steps	4
Gradient Checkpointing	True
Mixed Precision	bfloat16
ZeRO Stage	3
LoRA Rank	128
LoRA Alpha	256

Table 4.1: **Training setup for our experiments**: Summary of the training parameters and configurations used in our experiments.

asking it to choose the correct one. The benchmark is constructed merging four different data sources, Visual Genome [34], GQA-annotations [30], COCO [40], and Flickr30k [70], to create four sub-sets:

- *Coco-Order* (CO) and *Flickr-Order* (FO), which consist of word-level perturbations, swapping adjectives, nouns, and other elements within a caption.
- *Visual Genome Relation* (VR), where given an image and a relation "X relation Y," they test if the model can choose the correct order (e.g., "dog behind tree" vs. "tree behind dog," or "horse eating grass" vs. "grass eating horse").
- *Visual Genome Attribution* (VA). Here, they see if the model assigns attributes to objects correctly by selecting the proper phrase (e.g., "crouched cat and open door" vs. "open cat and crouched door"). More details are presented in Figure A.3.

ConMe is a more recent evaluation dataset presented by [29] [3]. The authors set out to create a more complex and CR question-answer benchmark, based on an established one like SugarCrepe [26]. Through a negative text generation pipeline, they start from SugarCrepe images and concretely create more challenging questions through the help of then state-of-the-art VLMs like GPT-4V [49], LLaVA-1.5 [41], and InstructBLIP [17]. ConMe consists of 24347 samples divided into three subcategories:

- *replace-attribute*, which consists of 8863 samples where the difference between the ground truth and negative caption is a swapped attribute.
- *replace-object*, where the only difference is a noun (object) replaced instead of another; in total there are 8691 examples.
- *replace-relation* comprised of 6793 question/answer pairs, the relation between two objects is modified. Each question is a multiple-choice question with a letter option. Additional details in Figure A.4.

[3] ConMe huggingface benchmark

Vision-Centric Benchmarks Additionally, we set out to analyze the *visual grounded-ness* of our approach. We do so by choosing two well-established benchmarks.
Presented by [61] the MultiModal Visual Patterns benchmark (**MMVP**) is tailored around scenarios that made CLIP-like models completely fail, asking the model a multiple choice question. Given our premises of Section 1 and 2, this is a relevant benchmark we have in our analysis. This is mostly due to the CLIP-blind pairs, which are manually constructed, will give us a precise indication of the pitfalls of CLIP models in scenarios described by [73], which we are specifically addressing [4].

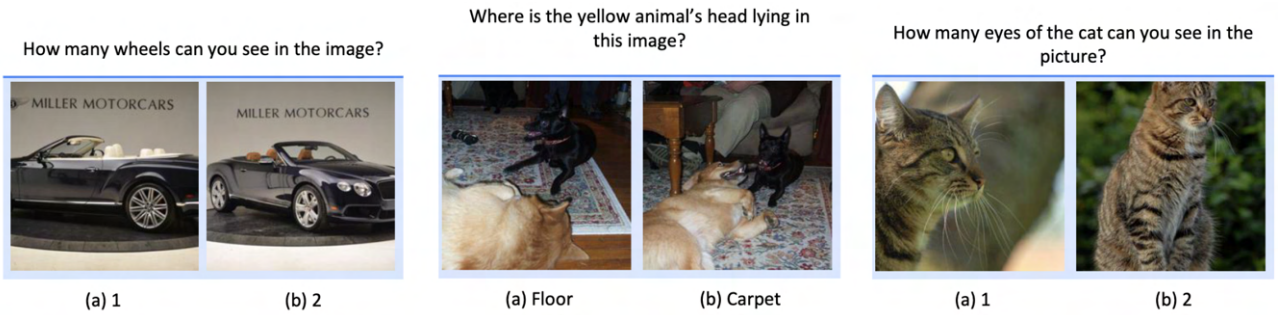


Figure 4.1: **MMVP benchmark examples**: Showing three qualitative examples from the MMVP [61] benchmark.

CVBench [60] [5] was recently presented along with Cambrian-1, and provides us with a unique perspective on multimodal capabilities in 2D and 3D scenarios. Within the multimodal evaluation realm, the authors noticed a lack of tasks that mostly revolved around traditional computer-vision tasks. With this in mind, they create a comprehensive set of 2638 manually-inspected samples, with four main tasks *Object Counting* and *Spatial Relation*, using ADE20K [78] and COCO [40] as data source, and *Depth Order* and *Relative Distance* estimation, leveraging [5]. For more details and examples, see Figure A.5.

General Image Understanding Even though our method is oriented toward Compositional Reasoning and Vision-Centric tasks, we believe it is of vital importance to also report performance on general image understanding evaluation sets. We report our model performance on:
AI2D [31] a ChartQA/Diagram understanding benchmark, with 3009 examples.
MME [23] a comprehensive Multimodal image-understanding benchmark. The benchmark consists of 2370 image Yes/No questions regarding two sub-tasks, general Perception and Cognition (Reasoning).
MMStar [11]. Stemming from knowledge-based multimodal benchmarks, the authors present MMStar to ensure the evaluation of MLLMs is entirely based on the model's ability to understand the image content. The benchmark includes 6 fine-grained categories (Fine-grained Perception, Coarse Perception, Mathematics, Science & Technology, Logical Reasoning, Instance Reasoning) over 1.5k samples.
MMBench [44] is a well-established benchmark focusing on a breadth of vision-language capabilities, ensuring the model's consistency over different tasks on 2040 visual question answers.

[4] MMVP Huggingface
[5] CVBench Huggingface

4.2 Comparison to Baselines

In this section we compare our **OG-LLaVA** with LLaVA-1.5 range of models and Subobject-level Image Tokenization method [8] (which we call SIT for brevity).
To ensure a fair comparison, we train in-house both LLaVA-1.5 and SIT with the same architectural backbones (Vision Encoder and LLM) as our models. Furthermore, we use the same training data, exploring both LLaVA-1.5 data mixture as well as Cambrian 7M introduced in Section 4.1.1, and hyperparameters of Table 4.1.
Specifically, we have:
LLaVA-1.5-3B with Llama3.2-3B as LLM, and trained with both LLaVA and Cambrian data.
LLaVA-1.5-7B with Vicuna-7B as LLM, and trained with LLaVA data.
LLaVA-1.5-8B with Llama3.1-8B as LLM, and trained with LLaVA data.
SIT-8B with Llama3.1-8B as LLM, and trained with LLaVA data.
All these models are trained with CLIP ViT-L/14@336 as Vision Encoder and the standard Multi-Layer Perceptron as Connector, except **SIT-8B** which is trained using a three-layer feed-forward network with ReLU activation function as adapter. This is due to the model-specific implementation and support for different visual input sizes. We will cover the technical details later.

4.2.1 Main Results

First we cover the comparison between **OG-LLaVA** and LLaVA-1.5 models on Compositional Reasoning, Vision Centric, and General Purpose Image understanding tasks.

Compositional Reasoning and Vision Centric In Table 4.2 we report our **OG-LLaVA** performance on Compositional Reasoning and Vision Centric benchmarks. Under both LLaVA-1.5 and Cambrian-1 training regimes, **OG-LLaVA** is consistently better than its baselines. We systematically see these advantages in compositional understanding benchmarks with ARO showing consistent gains across training datasets and backbones of +21% on *Coco-Order* with the biggest gap from 38.2 to 82.6, +16%, from 49.1 → 84.0 on *Flickr-Order*. The trend continues on *Visual Genome Attribution* with an average increase of +10% and on *Visual Genome Relation* with a whopping +20% across training data and model sizes. ConME likewise records a consistent improvement of at least 2% across model scales, reaching 65.2 in the 8B setting (+3.6 over the strongest baseline).
These compositional gains translate to the vision-centric side: MMVP rises by roughly three percentage points on average (e.g. 32.0 → 37.0 in the 8B comparison and 61.6 → 66.0 with Cambrian-1 data), while performance on CVBench is essentially retained (,±1-point fluctuations across configurations).
Finally, it is noteworthy that the light-weight Llama3.2-3B backbone (first and last two rows of Table 4.2) preserves most of the accuracy budget—indeed, it widens the margin on the two Visual Genome sub-tasks—despite its markedly lower parameter count, thereby offering the community an attractive latency–accuracy trade-off at both training and inference time.

General Purpose In Table 4.3 we extend our evaluation to *general-purpose* benchmarks and observe that **OG-LLaVA** maintains, and in some cases enhances, the strong trend outlined for compositional and vision-centric tasks. Under the LLaVA-1.5 supervision regime, our 8B model exhibits notable gains on the demanding MME diagnostic: the perceptual (Perc.) score rises by +36.2pp to 1551.5, while the cognitive (Cogn.) component improves by +25.0pp to 317.1 relative to the LLaVA-1.5-8B baseline. These advances highlight a better alignment between textual and visual modalities when traversing both low-level perceptual cues and higher-order

reasoning, despite a marginal drop on AI2D (–0.5) and MMStar (–1.9). At the smaller 3B scale, the performance gap narrows, yet OG-LLaVA preserves competitive accuracy across all tasks, matching or slightly exceeding the baseline on MMBench.

When trained with the richer Cambrian-1 corpus, **OG-LLaVA-3B** secures further improvements: AI2D climbs to 66.5 (,+1.7%,), and the MME perceptual score reaches 1511.7, while MMBench advances to 70.91. These results are achieved with a *token-balanced* visual representation during training ($T \approx V$). Taken together, the evidence suggests that **OG-LLaVA**'s object-guided token selection is well suited to broad, open-ended Vision-Language evaluation suites, delivering systematic benefits on perception-oriented and mixed-reasoning metrics with minimal sacrifices on outlier tasks.

Model Details		**Compositional Reasoning**					**Vision Centric**	
		ARO				ConMe	MMVP	CVBench
Method	**#Vis. Tok.**	CO	FO	VA	VR	Acc.	Acc.	2D+3D Acc.
Training data: LLaVA-1.5								
LLaVA-1.5-3B	V	63.8	70.9	60.2	52.4	59.6	**59.7**	63.3
OG-LLaVA-3B (Ours)	$T(\approx V)$	**79.1**	**82.2**	**75.2**	**75.5**	**61.2**	57.3	**63.5**
LLaVA-1.5-7B	V	36.2	44.1	28.1	28.2	57.7	33.7	60.1
LLaVA-1.5-8B	V	38.2	49.1	28.3	29.7	61.6	32.0	**65.2**
OG-LLaVA-8B (Ours)	$T(\approx V)$	**82.6**	**84.0**	**38.6**	**45.3**	**65.2**	**37.0**	62.5
Training Data: Cambrian-1								
LLaVA-1.5-3B	V	71.0	76.8	72.6	27.5	**68.6**	61.6	**66.2**
OG-LLaVA-3B (Ours)	$T(\approx V)$	**73.7**	**79.5**	**79.2**	**50.8**	66.7	**66.0**	64.5

Table 4.2: **OG-LLaVA performance on CR and VC tasks compared with LLaVA baselines.** Results are reported without segmentation masks at inference time. Here $T(\approx V)$ symbolizes the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for **OG-LLaVA**. For ARO, we present scores for its four sub-tasks (CO, FO, VA, VR). The highest performance is in **bold**, within the same training data and model size section.

Model Details		**General Purpose**				
		AI2D	MME		MMStar	MMB
Method	**#Vis. Tok.**	Acc.	Perc.	Cogn.	Acc.	Dev. Acc.
Training data: LLaVA-1.5						
LLaVA-1.5-3B	V	**59.2**	**1407.1**	**330.0**	**37.7**	**67.4**
OG-LLaVA-3B (Ours)	$T(\approx V)$	56.8	1394.2	325.3	37.45	67.60
LLaVA-1.5-7B	V	53.5	1479.7	**323.6**	34.4	62.5
LLaVA-1.5-8B	V	**60.6**	1515.3	292.1	**40.7**	**71.6**
OG-LLaVA-8B (Ours)	$T(\approx V)$	60.1	**1551.5**	317.1	38.8	67.3
Training Data: Cambrian-1						
LLaVA-1.5-3B	V	65.38	1480.9	**328.2**	**41.37**	**70.07**
OG-LLaVA-3B (Ours)	$T(\approx V)$	**66.5**	**1511.7**	300.6	38.35	70.91

Table 4.3: **OG-LLaVA performance on General purpose tasks compared with LLaVA baselines.** Results are reported without segmentation masks at inference time. Here $T(\approx V)$ symbolizes the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for **OG-LLaVA**. The highest performance is in **bold**, within the same training data and model size section.

4.2.2 Comparison to Subobject-level Image Tokenization

In a recent study, the authors of [8] propose a method for Subobject-level Image Tokenization. While their main contributions revolve around different areas like Segmentation and Image Tokenization, among the many experiments, they show how to leverage segmentation masks to create a more efficient visual tokenization process within MLLMs. Below, we will go over this method and analyze the differences in approach and performance with our **OG-LLaVA**.

Subobject-level Image Tokenization Let m'_γ be a Multi-Layer Perceptron comprised of 3 Linear layers with a ReLU activation in between and $f_{v\theta}$, g_ϕ be the previously defined Vision Encoder and Large Language Model from Section 3. Given an image $\mathbf{X} \in R^{C\times H\times W}$ and a set of segmentation masks $\mathbf{M} = \{\mathbf{m}_i \mid i = 1, \ldots, N\} \subset \mathbb{R}^{H\times W}$, the authors propose to extract visual features from the image using a vision encoder. They first encode the image to obtain the visual features

$$\mathbf{X}' = f_{v\theta}(\mathbf{X}) \in \mathbb{R}^{V,F}$$

where $f_{v\theta}$ is CLIP ViT-L/14@336. Now, instead of down-sampling the segmentation masks to match the visual token dimension, they up-sample the visual features out of $f_{v\theta}$ back to the original image resolution, i.e. $\mathbf{X}'' = \texttt{upsample}(\mathbf{X}') \in \mathbb{R}^{C\times H\times W}$.

Subsequently, the authors consolidate the pixel-level features within each segment to form its content embedding. They do so by defining an average pooling function pool : $\mathbb{R}^{C\times H\times W} \to \mathbb{R}^C$

$$\mathbf{x}_i^c := \text{pool}(\mathbf{X}''[h, w]|M[h, w] = i),$$

with $\mathbf{X}''[h, w] \in \mathbb{R}^C$ being the feature vector at coordinates $[h, w]$.
Later, $\mathbf{x}_i^c$ are concatenated with a positional encoding $\mathbf{x}_i^p$ to obtain a single visual token representations ($\mathbf{x}$) with

$$\mathbf{x}_i = m'_\gamma(\mathbf{x}_i^c \oplus \mathbf{x}_i^p)$$

collectively the final visual tokens for image $\mathbf{X}$ are $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$. Finally the output tokens ($\mathbf{T}_a$) are obtained through g_ϕ as:

$$\mathbf{T}_a = g_\phi\big([\mathbf{x}; \mathbf{TT}]\big)$$

To ensure a fair comparison, we implement SIT by adapting the source code to our training setup and train both our and the SIT model with the same data and model backbones.

Comparison with our OG-LLaVA In Figure 4.2 we report the comparison between our **OG-LLaVA-8B**, SIT-8B, and a standard LLaVA-1.5-8B, all models with the same backbones. Results clearly show how our approach consistently outperforms SIT in both Compositional Reasoning and Visual Grounding domains, with a decrease of more than 25% for the former and 10% for the latter.

Furthermore, in contrast to **OG-LLaVA**, which supports inference both with and without mask information, SIT mandates the availability of pre-computed segmentations at test time. Although the token count is reduced with $N < V$, this benefit is offset by the non-trivial overhead of running an additional segmentation model during inference. The requirement arises from the modified architecture of SIT's adapter: without masking metadata—or a significant redesign of the image-processing pipeline—images cannot traverse the standard LLaVA-1.5 flow.

4.3 Ablation Studies

In this section we present a series of ablation studies aimed at disentangling the contribution of the main design choices behind our framework. Specifically, we first analyze the influence of the

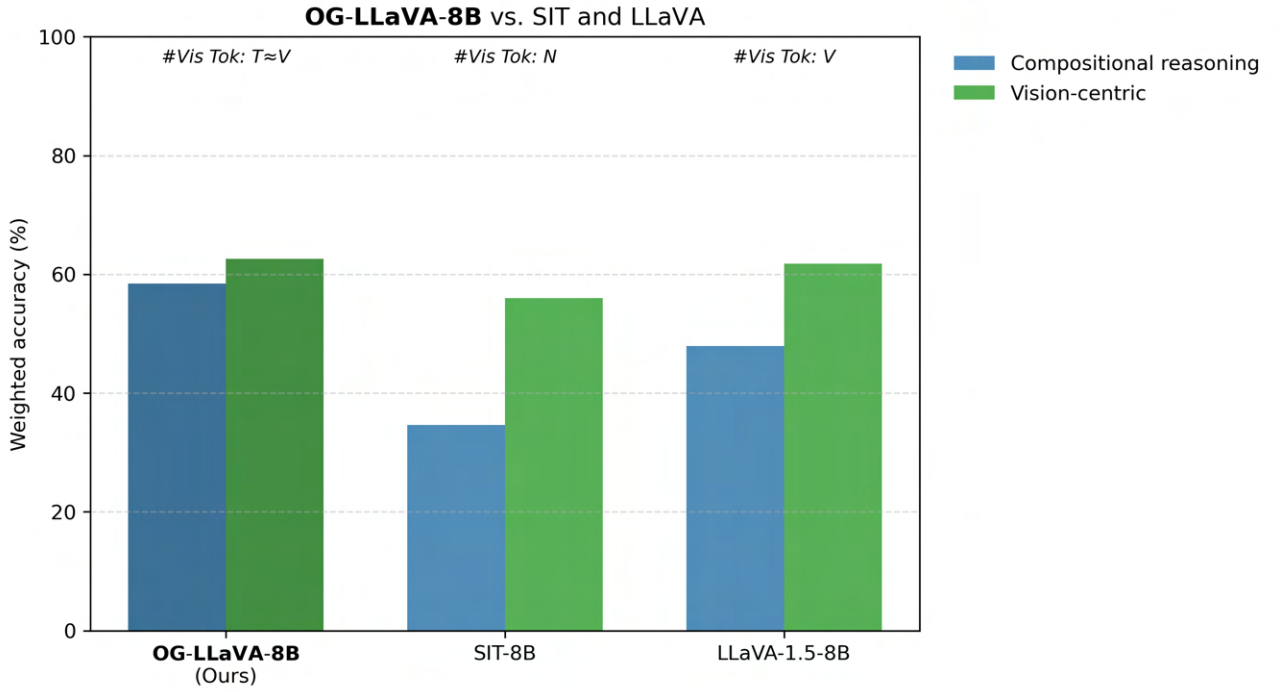


Figure 4.2: **Comparison between our OG-LLaVA, Subobject level Image Tokenization and LLaVA-1.5**: We report the performance of our **OG-LLaVA-8B** with `OG-Fusion` (darker bars) against the Subobject Level Image Tokenization (SIT-8B) [8] approach and LLaVA-1.5-8B. The weighted accuracies (higher is better) are reported for three macro-benchmarks—Compositional Reasoning (ARO [73] and ConMe [29]), Vision-Centric (MMVP [61] and CVBench [60]), with the test-set item counts as weights. Here $T(\approx V)$ symbolizes the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for **OG-LLaVA**.

visual masking scheme, asking whether the default configuration adopted by `OG-Fusion` already supplies sufficient contextual cues or if exposing additional image content leads to further gains. Later we assess the need for more explicit structure in the visual prompt, testing the effect of introducing specialized tokens and/or positional encodings that could convey finer-grained spatial information to the LLM.

4.3.1 Masking Approach

We start our analysis by looking into the impact of the masking approach, trying to understand which type of information passing is the most effective.
In the `OG-Fusion` setup, covered in Section 3.2, we apply the down-sampled masks ($\mathbf{M}'$) one by one onto the vision encoder output ($\mathbf{X}'$). We then concatenate the masked features along its latest dimension, and project them into the same dimension as the LLM input tokens (Eq. 3.7), obtaining $\boldsymbol{OGVT}$.
To assess whether Object–Guided Visual Tokens already capture all the visual information required by the LLM, we design an ablation variant in which we explicitly re-inject a global representation of the image.

Global View We first investigate whether the output Vision Encoder ($\mathbf{X}'$) carries meaningful *global* contextual signal, which should be incorporated along with the single masks. Our masking pipeline, by design, generates tokens that focus on isolated regions; if the model is exposed solely to these local snippets, it risks losing the holistic scene context that links separate masks to

one another. Without such a global embedding, **OG-LLaVA** may find it difficult to associate objects appearing in different masks and to reason about their mutual relationships. Injecting the full-image representation $\mathbf{X}'$ therefore tests whether furnishing this broader contextual view facilitates cross-mask integration and preserves relational understanding.

Mathematically, this accounts to substituting Eq. 3.7 with

$$\boldsymbol{OGVT}_{gv} \;:=\; m_\gamma\Big(\mathbf{X}' \oplus \big\|_{i=1}^{N} \mathbf{Y}_i\Big) \;\in\; \mathbb{R}^{(V+T)\times D} \tag{4.1}$$

where $\oplus$ denotes concatenation. We add $\mathbf{X}'$ at the beginning of the stack, therefore we increase the number of visual tokens from $T \to (V+T)$. We call this variant the "*Global View*" as we retain information from the entirety of the CLIP output providing global representation of the image.

Results Figure 4.4 reveals that augmenting the object-guided visual token set with a global feature is *counter-productive*. The global-view variant suffers a substantial 6.4% drop on the compositional-reasoning axis. On the vision-centric and general-purpose suites the declines are smaller (-0.8% and -1.7% respectively) but remain consistently negative, indicating no compensatory benefit on tasks that should profit the most from holistic context.
Crucially, these modest (or negative) accuracy changes come at a steep computational cost: adding the encoder grid doubles the number of visual tokens (*from* $T \approx V$ *to* $V+T \approx 2V$), thereby *doubling* the quadratic self-attention workload in the vision branch. In light of the clear performance degradation and the two-fold increase in sequence length, we deem the global-view extension "not worth the trouble" and employ the lean ***OGVT*** design for all subsequent experiments.

Given these poor results and the theoretical increase in information provided to the model, which should help contextualize information, *Why does the global view back-fire?* To address this, we conjecture two, not mutually exclusive, failure modes:
Dilution in an over-long context: The global-view variant doubles the visual sequence length ($T \approx V \;\to\; 2V$). For quadratic self-attention this not only inflates the FLOP/memory footprint but also enlarges the key/query space in which each token must compete for relevance. Empirically, transformers with a fixed number of heads often show degraded utility per token once the sequence becomes too long for the head dimension to "cover" all positions. Our observed accuracy drop is consistent with the model *incomplete coverage* of the now much broader context and thereby mixing or discarding object-specific evidence.
Lack of disambiguating priors. In the present ablation the global feature is appended as an *ordinary* visual token, indistinguishable—save its position—from the object-guided tokens. Without an explicit positional encoding or a dedicated type-embedding, the decoder has no cue that this particular vector summarises the *entire* image as opposed to one local patch. Introducing a special "GLOBAL" token type or a learned positional offset, akin to the CLS embedding in BERT-like models [18], may help the network to gate or re-weight the global context more effectively.

4.3.2 End-of-Mask Token

Building on these findings, we next explore a complementary strategy that makes the terminus of every mask explicit to the decoder. Concretely, we append a dedicated End-of-Mask (EoM) token after the last slot of each object mask. In theory, this lightweight delimiter serves the purpose of *Structural disambiguation*. Because the EoM vector appears only at mask boundaries, it provides an unambiguous cue about where one object description ends and another begins—even when all preceding tokens share the same type-embedding. The remainder of this section details the formulation of the End-of-Mask token, the minimal architectural changes needed to support

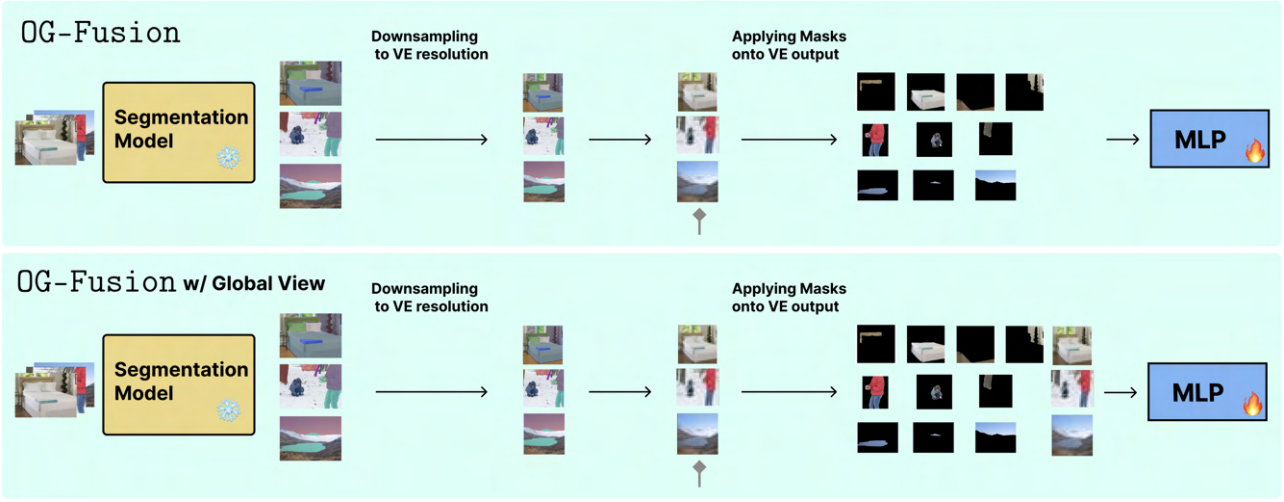


Figure 4.3: `OG-Fusion` **without and with Global View**: Reporting a visual example of our `OG-Fusion` with the standard pipeline (on the top) versus appending the *Global View* (on the bottom).

it, and its impact on performance.

Explicitly, we perform this ablation by comparing our `OG-Fusion` without (standard setup) and with the EoM token. In Section 3.2 we introduced the $\boldsymbol{OGVT}$ by concatenating the masked features across the last dimension. Here we modify Eq. 3.7 through EoM

$$\boldsymbol{OGVT}_{EoM} := m_\gamma\Big(\Big\|_{i=1}^{N} [\mathbf{Y}_i; \mathbf{e}_i]\Big) \in \mathbb{R}^{(T+N)\times D} \tag{4.2}$$

where $\mathbf{e}_i \in \mathbb{R}^{1\times D}$ is the End of Mask token, ";" denotes concatenation and $T = \left(\sum_{i=1}^{N} t_i\right)$ is the sum of all *True* values across $\mathbf{M}'$.

Results In Figure 4.5 we report the side-by-side comparison of our **OG-LLaVA** with and without the EoM token. The picture clearly shows how, in spite of the theoretical structural disambiguation provided by the EoM, the model seems to lose considerable amounts of performance. Specifically in Compositional Reasoning, with performance dropping (-10%) ($61.0 \rightarrow 52.1$), the extra EoM tokens elongate every sequence and appear to either interrupt the cross-mask attention patterns required to fuse multiple objects into a single chain of reasoning or to simply confuse the model on each segment. This gap is not present in Vision-Centric tasks with a small gain ($+ \leq 3\%$), and in General purpose, which remain statistically unchanged. Because the delimiter reduces performance precisely where multi-object reasoning is critical —and inflates the visual-token budget from $T \rightarrow T + N$, thereby slowing training and inference—we adopt the leaner, delimiter-free `OG-Fusion` setup described in Section 3.2 for all subsequent experiments.

4.3.3 Positional Encoding

As outlined in Section 2.4 positional encodings (PE) are a cornerstone of Transformer-based models [63], especially in the vision domain [19, 1]. Furthermore, positional embeddings may also address the concerns raised in Section 4.3.1, providing disambiguation in the spatial domain. Given this—and the particular design of our framework—we therefore evaluate how different positional-encoding schemes affect our model's performance. We run this ablation study on **OG-LLaVA-3B**.

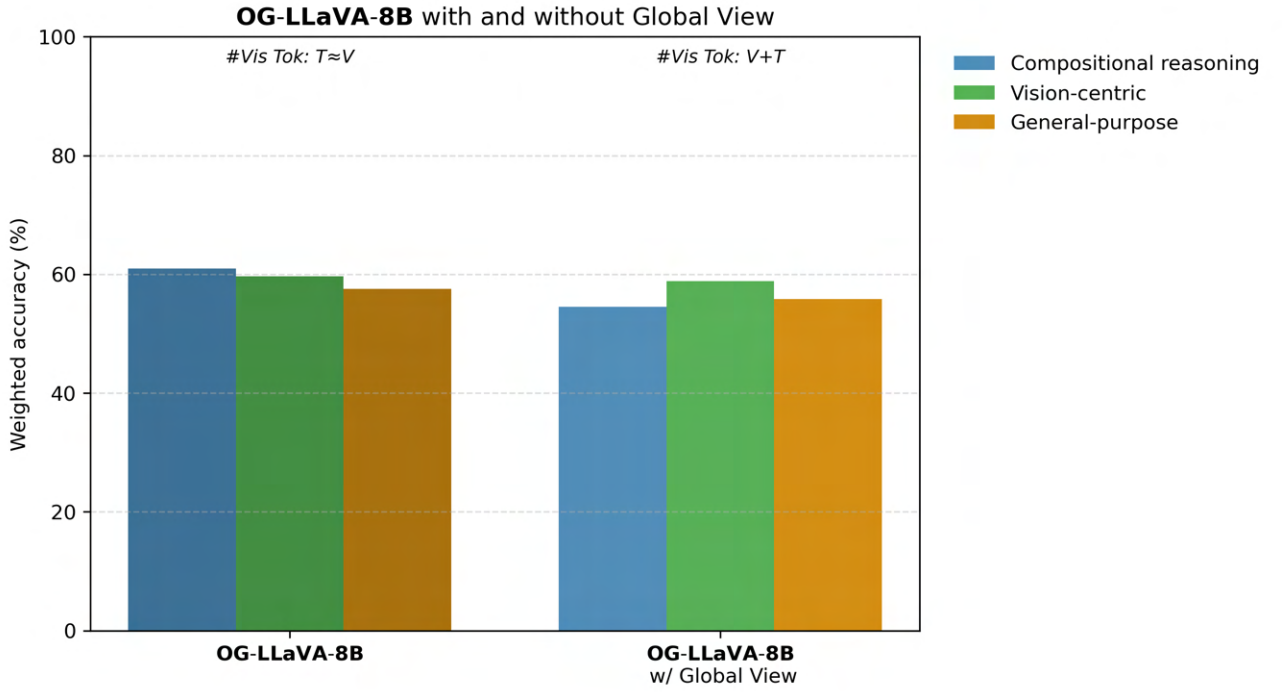


Figure 4.4: **OG-LLaVA-8B without and with Global View**: Performance difference with and without the Global View approach. The weighted accuracies (higher is better) are reported for three macro-benchmarks—Compositional Reasoning, Vision-Centric, and General-Purpose—using the test-set item counts as weights. Here $T(\approx V)$ and $T+V$ symbolize the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for both **OG-LLaVA** versions.

1D Absolute Sinusoidal Encodings We first examine 1D positional encodings, focusing on fixed sinusoidal. The guiding intuition for this experiment is to provide for every token associated with a given mask an identical positional vector.

To achieve this, we modify the current process of Section 3.2 by assigning every mask patch i a discrete segment label $s_i \in \{0, \dots, K\}$. We then compute a deterministic vector $\mathrm{PE}(s_i) \in \mathbb{R}^D$ whose even and odd coordinates follow the classical scheme of [63]:

$$\mathrm{PE}_{2k}(s) = \sin\Big(\tfrac{s}{10000^{2k/D}}\Big), \qquad \mathrm{PE}_{2k+1}(s) = \cos\Big(\tfrac{s}{10000^{2k/D}}\Big), \quad k = 0, \dots, \tfrac{D}{2} - 1,$$

with $\mathrm{PE}(0) = \mathbf{0}$ reserved for the special "no-segment" label. We finally apply all together, obtaining

$$\boldsymbol{OGVT}_{1D-s} \;=\; m_\gamma\Big(\big\|_{i=1}^{N} \mathbf{Y}_i\Big) \;+\; \big\|_{i=1}^{N} \mathrm{PE}(s_i) \;\in\; \mathbb{R}^{T\times D},$$

so each column now captures the content processed by m_γ together with the fixed sinusoidal signature of its segment.

1D Learnable Encodings We also introduce learnable encodings instead of absolute sinusoidal. We define $\mathrm{PE}(s)$ as:

$$\mathrm{PE}(s) \;:=\; E_s, \quad E = \begin{bmatrix} E_0^\top & E_1^\top & \dots & E_K^\top \end{bmatrix}^\top \in \mathbb{R}^{(K+1)\times D},$$

where $E_0 = \mathbf{0}$ corresponds to the padding index and $K+1$ is a fixed maximum number of segments. The rationale behind this is similar to that of fixed sinusoidal encodings, where the learned embedding provides a segment-specific shift, allowing the network to adaptively position

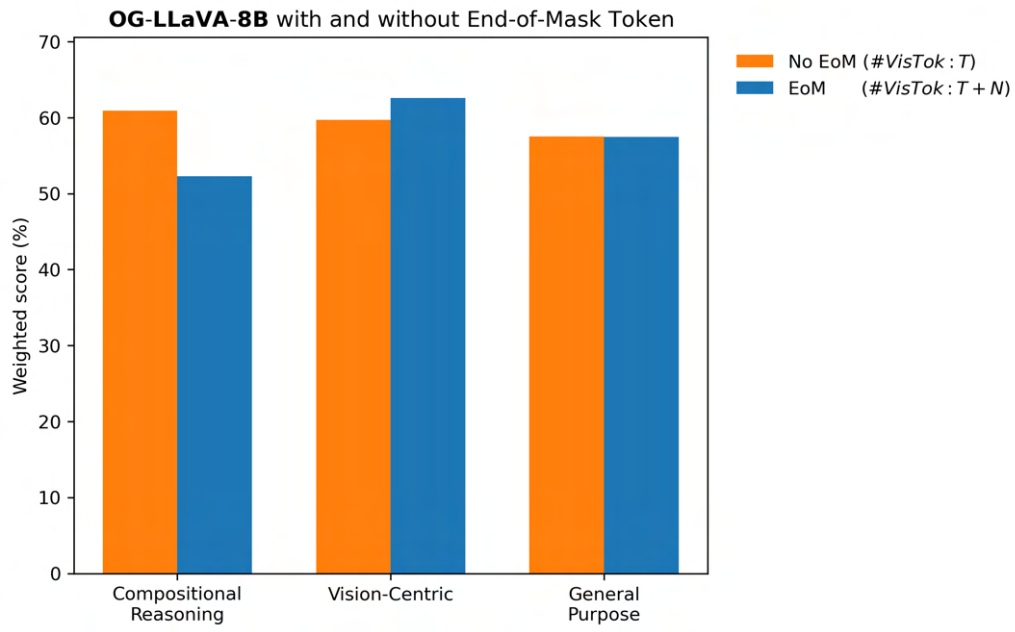


Figure 4.5: **OG-LLaVA-8B with and without End of Mask Token**: We show the performance of our `OG-Fusion` with and without End of Mask token. The weighted accuracies (higher is better) are reported for three macro-benchmarks—Compositional Reasoning, Vision-Centric, and General-Purpose—using the test-set item counts as weights. The orange bar represents the standard pipeline without EoM tokens, whereas the blue bar represents the ablated version with the EoM tokens. Here T and $T+V$ symbolize the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for both **OG-LLaVA** versions.

each segment in representation space. Stacking these position-aware vectors yields the final descriptor

$$\boldsymbol{OGVT}_{1D-l} \;=\; m_\gamma\Big(\big\|_{i=1}^{N}\mathbf{Y}_i\Big) + \big\|_{i=1}^{N}\mathrm{PE}(s_i) \in \mathbb{R}^{T\times D}$$

2D Sinusoidal Positional Encodings Building on the 1D segment encoding above, we further enrich every mask patch with an absolute 2D spatial code that depends on its row–column location inside the V feature grid of our vision encoder. Half of the D channels encode the vertical index $y\in\{0,\dots,H-1\}$, the other half encode the horizontal index $x\in\{0,\dots,W-1\}$, following two independent sinusoidal spectra:

$$\begin{aligned}
\mathrm{PE}^{\mathrm{row}}_{2k}(y) &= \sin\!\Big(\tfrac{y}{10000^{2k/(D/2)}}\Big), \quad \mathrm{PE}^{\mathrm{row}}_{2k+1}(y) = \cos\!\Big(\tfrac{y}{10000^{2k/(D/2)}}\Big),\\
\mathrm{PE}^{\mathrm{col}}_{2k}(x) &= \sin\!\Big(\tfrac{x}{10000^{2k/(D/2)}}\Big), \quad \mathrm{PE}^{\mathrm{col}}_{2k+1}(x) = \cos\!\Big(\tfrac{x}{10000^{2k/(D/2)}}\Big),
\end{aligned} \qquad k = 0,\dots,\tfrac{D}{4}-1,$$

Then we concatenate the two halves to obtain $\mathrm{PE}^{\mathrm{2D}}(x,y)\in\mathbb{R}^{V\times D}$, where (x_i,y_i) is the grid coordinate of patch i.
We only apply the positional encoding after passing the image features to m_γ, changing Eq. 3.7 to:

$$\boldsymbol{OGVT}_{2D} \;=\; m_\gamma\Big(\big\|_{i=1}^{N}\mathbf{Y}_i\Big) \;+\; \big\|_{i=1}^{N}\mathrm{PE}^{\mathrm{2D}}(x_i,y_i),\; \in\; \mathbb{R}^{T\times D},$$

This operation augments each mask patch with a fine-grained spatial signature, complementing the segment-level (1-D) encodings and allowing the network to reason jointly about where a patch lies inside the image grid and to which mask it belongs.

Segmentation aware 1D RoPE In our **OG-LLaVA**, the Large Language Model backbone is either Llama3.2-3B, Llama3.1-8B. Both of these encapsulate 1D Rotary Positional Embedding (RoPE) [58] within its attention mechanism, see [20]. In their one-dimensional version of RoPE, each token at position index $p_{b,j}$ (batch b, time step j) is assigned a rotation phase

$$\theta_{b,j,k} = p_{b,j}\,\omega_k, \qquad \omega_k = \frac{1}{\alpha^{\,2k/d}}, \quad k = 0, \ldots, \tfrac{d}{2} - 1,$$

so the complex embedding vector is $\exp\big(i\theta_{b,j}\big)$, whose real and imaginary parts are returned as $\cos\theta$ and $\sin\theta$.

We modify this setting by introducing a group mask $\mathcal{G}_b = \{(s_\ell, e_\ell)\}_{\ell=1}^{L_b}$ for every batch row. Positions that fall in the same interval are collapsed to the left-boundary index:

$$\tilde{p}_{b,j} = \begin{cases} s_\ell & \text{if } (s_\ell, e_\ell) \in \mathcal{G}_b \text{ and } s_\ell \le p_{b,j} \le e_\ell, \\ p_{b,j} & \text{otherwise.} \end{cases}$$

Subsequent rotary phases are computed with $\tilde{p}_{b,j}$ instead of $p_{b,j}$:

$$\tilde{\theta}_{b,j,k} = \tilde{p}_{b,j}\,\omega_k, \qquad \cos\tilde{\theta}_{b,j,k},\ \sin\tilde{\theta}_{b,j,k}.$$

Hence every token whose index lies inside the same interval (s_ℓ, e_ℓ) receives an identical phase $\tilde{\theta}_{b,j,k} = s_\ell\,\omega_k$. Operationally, this collapses fine-grained time steps into coarse "segments," making self-attention permutation-invariant within each segment while preserving standard RoPE behavior across segments. We call this variant *Segmentation-Aware 1D RoPE*. In this study, because we directly act on the internal mechanism of the LLM, we leave the ***OGVT*** invariant.

Results Figure 4.6 confirms that the plain, "no-PE" baseline is still the strongest configuration (first column). With nothing more than the implicit patch order, **OG-LLaVA-3B** reaches $\sim$ 69% weighted accuracy on compositional-reasoning, 63% on vision-centric, and 56% on general-purpose queries—topping every competitor across the board.

Adding *fixed 1D sinusoidal* shaves almost 7% off compositional reasoning accuracy and gives back only 3% on vision-related questions; the trigonometric basis appears to clash with the object-guided token order the network has already internalized. A *1D learnable* PE closes that gap by $\sim 0.3\%$ but still lags the baseline, hinting that the model cannot reliably discover a more helpful geometry from scratch.

The *2D sinusoidal PE* variant is especially puzzling. Apriori, enriching every token with its absolute (x, y) phase should let the decoder pinpoint where a patch sits on the image lattice, preserving fine-grained spatial cues that matter for both vision-centric recognition and the multi-object reasoning required in compositional tasks. Yet the curve tells the opposite story: once the mask-wise reordering scrambles the grid, those fixed waves seem to supply a noisy—and often conflicting—reference frame, driving *all* macro-benchmarks below 60%. This suggests that a hard-wired sinusoid is simply too rigid for our object-guided token stream. A more flexible alternative could be to adopt *2D RoPE*, as in PixTral-12B, or—even simpler—a lightweight *learnable 2-D embedding grid* that can bend to the permutation induced by masking while still encoding relative positions. We leave these richer formulations to future work and, for now, drop explicit positional signals altogether.

Finally, the *segmentation-aware 1-D RoPE* variant recovers some vision-centric accuracy, yet it, too, stays several points behind on multi-object reasoning and mixed workloads. We suspect this shortfall occurs because the LLM has never encountered mixed groups of either visual and textual tokens sharing the same positional encoding during neither its multimodal

nor language-only pre-training. Moreover, our fine-tuning keeps most of the network frozen, changing only a small fraction of its weights—likely too little for the model to adapt its internal representations to this novel signal.

Because every explicit positional scheme adds parameters and/or latency while lowering or, at best, matching accuracy, we adopt the leaner position-free setup in the remainder of the study.

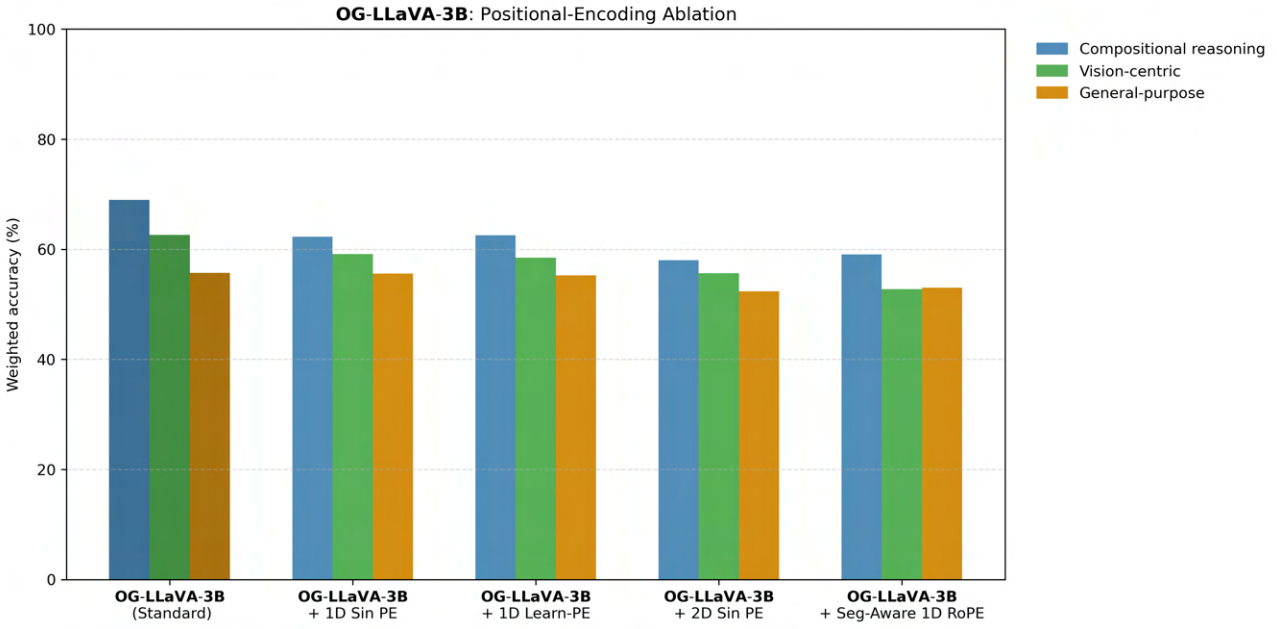


Figure 4.6: **Positional encoding ablation on OG-LLaVA-3B**: The weighted accuracies (higher is better) are reported for three macro-benchmarks—Compositional Reasoning, Vision-Centric, and General-Purpose—using the test-set item counts as weights. The darker bars correspond to the baseline **OG-LLaVA** model (no explicit positional encoding beyond the patch order), while lighter bars show four alternatives: 1D sinusoidal, 1D learnable, 2D sinusoidal, and segmentation-aware 1D RoPE.

4.4 Additional Experiments

4.4.1 *Sliding Windows* approach

Given new visual feature encodings like Dynamic High Resolution (DHR) [42], in this section, we choose to study the importance of our Object-Guided Visual Tokens against a much simpler, yet reasonably similar technique. We call this the *Sliding Windows* approach. While similar to DHR, this approach is done only after the image is already encoded by the ViT, and provides a good comparison between segmentation masks and patch regions.

Let $\mathbf{X} \in R^{C\times H\times W}$ denote a single input image and $\mathbf{X}' = f_{v\theta}(\mathbf{X}) \in \mathbb{R}^{V,F}$ be the output of the vision encoder. Instead of using the standard $\mathbf{M} = \{\mathbf{m}_i \mid i = 1, \ldots, N\} \subset \mathbb{R}^{H\times W}$, we create a collection of *sliding-window patches* $\mathcal{J}^{(1)}, \ldots, \mathcal{J}^{(k)} \subset \{1, \ldots, V\}$ that partition the image into k disjoint regions, i.e. $\bigcup_{q=1}^{k} \mathcal{J}^{(q)} = \{1, \ldots, V\}$.

For each patch set $t_q := |\mathcal{J}^{(q)}|$ (the number of pixels in patch q), arrange the indices in ascending

order, $\mathcal{J}^{(q)} = \{ j_1^{(q)} < \cdots < j_{t_q}^{(q)} \}$, and define the corresponding *row-selection matrix*

$$P^{(q)} := \begin{bmatrix} e_{j_1^{(q)}}^\top \\ \vdots \\ e_{j_{t_q}^{(q)}}^\top \end{bmatrix} \in \{0,1\}^{t_q \times V}, \qquad q = 1, \ldots, k,$$

where e_j denotes the j-th canonical basis vector in $\mathbb{R}^V$.
Multiplying by $P^{(q)}$ *keeps* the rows that belong to patch q and discards the rest, giving

$$\mathbf{Y}^{(q)} \;=\; P^{(q)}\,\mathbf{X}' \;\in\; \mathbb{R}^{\,t_q \times F},\; q = 1, \ldots, k.$$

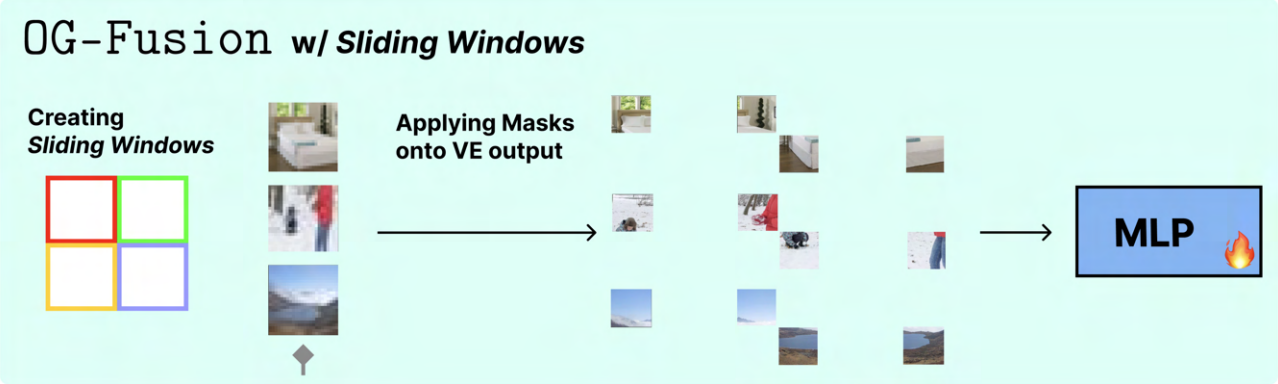
The matrices $P^{(q)}$ contain no learnable parameters; they simply *select and reorder* rows of $\mathbf{X}'$ in a deterministic fashion dictated by the sliding-window partition of the image. Therefore Equation 3.7 becomes:

$$\boldsymbol{OGVT}_{sw} \;:=\; m_\gamma\big(\mathbf{Y}^{(q)}\big) \;\in\; \mathbb{R}^{T_q \times D} \tag{4.3}$$

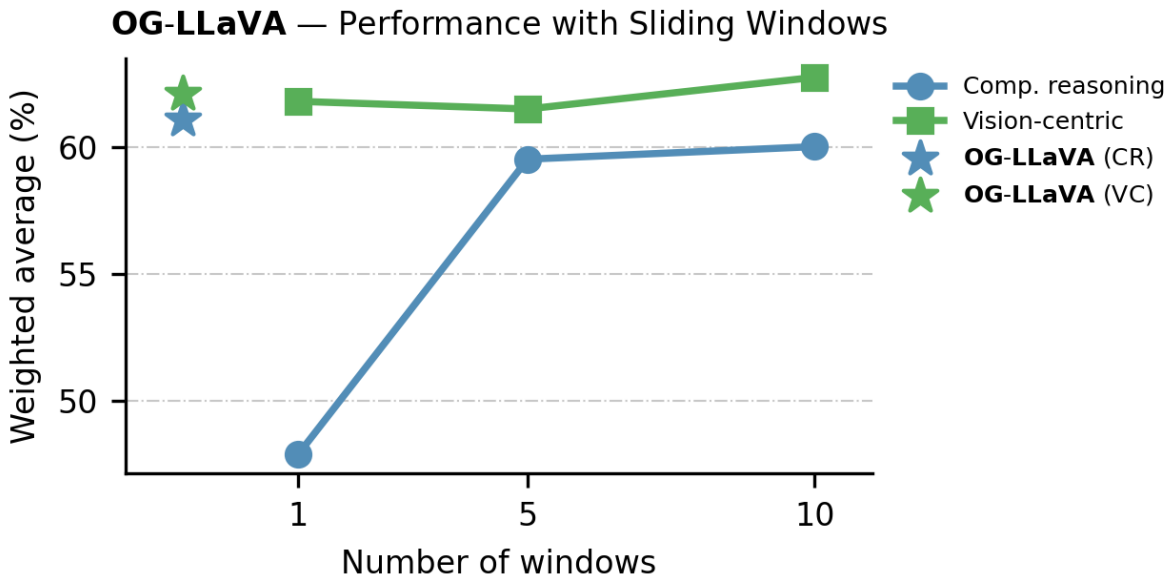
where $T_q \;=\; \sum_{q=1}^{k} t_q \;=\; V$, given each patch is not overlapping. The overall process is visualized in Figure 4.7a.

Results In Figure 4.7b, we show the performance of the *Sliding Windows* approach with 5 and 10 tiles, versus 1, meaning the standard LLaVA-1.5 pipeline described in Section 3.1, and **OG-LLaVA**.
As seen in Figure 4.7b, raising the number of sliding-window patches from one to five and then to ten systematically boosts weighted performance in Compositional Reasoning (CR), while producing an overall upward trend in Vision-Centric (VC) accuracy. The behavior in CR is almost linear: the initial single-window baseline hovers around 48%, but once the input is dissected into five windows the score climbs by more than eleven percentage points and it climbs slightly further at ten windows. VC accuracy is already high at one window ($\approx 62\%$), dips negligibly at five, and recovers at ten windows, surpassing the baseline. This pattern mirrors the intuition behind dynamic high-resolution processing: every additional window acts as an extra "sensor" that captures fine-grained spatial cues, giving the model more local context without inflating its receptive field all at once.

Figure 4.7b also highlights the surprising strength of such a straightforward mechanism. Partitioning the image into a handful of fixed, non-overlapping windows requires neither extra learnable parameters nor architectural changes, yet it narrows—indeed nearly closes—the gap to the strongest reference system in VC tasks and slashes the CR deficit by a large margin. Two complementary factors likely underpin these gains. First, windows should encourage the MLP to create features attending to local texture and shape details that would otherwise be averaged away when the entire frame is seen at once. Second, they impose an implicit form of data augmentation: by forcing the model to solve the task from partial glimpses, they might reduce over-reliance on any single region and thus improve robustness.
Despite these improvements, the two starred points corresponding to **OG-LLaVA** remain the global optimum across task families. In the CR dimension the star sits slightly above the ten-window variant, and although the VC star is narrowly edged out by the ten-window curve, our model achieves the best *combined* average because it excels in both modalities simultaneously. This balance underscores why **OG-LLaVA** remains the strongest overall choice even in the presence of window-based refinements.



(a) *Sliding Windows* approach visualization.



(b) *Sliding Windows* approach performance.

Figure 4.7: **OG-LLaVA performance change with the *Sliding Windows* approach**. Sub-figure (a) illustrates the *Sliding Windows* approach by visualizing the process from start to finish. Sub-figure (b) shows the performance as the number of windows increases from 1 to 10. It also shows-with stars-the performance of the standard **OG-LLaVA** setup. The weighted accuracies (higher is better) are reported for three macro-benchmarks—Compositional Reasoning and Vision-Centric—using the test-set item counts as weights. All the experiments in this graph are carried out with **OG-LLaVA-8B**.

4.5 Comparison to Open Source models

In this section we compare our **OG-LLaVA** with some of the most vision-interesting and state-of-the-art models to date.

LLaVA-Next-8B [42] applies a dynamic high-resolution strategy *before* its vision encoder, expanding the input to $V \times (k+1)$ visual tokens and thereby injecting a very large number of image patches. The backbones LLM is Llama3.1-8B-Instruct. Moreover, the model is exposed to a substantially larger and cleaner multimodal corpus and it is fine-tuned end-to-end during supervised fine-tuning stage.

Cambrian-1-8B [60] adopts a heterogeneous feature-fusion scheme in which representations from OpenCLIP with ConvNeXt-L@1024[15], SigLIP with ViT-SO400M/14@384 [74], DINO-v2 ViT-L/14@516 [51], and SAM2 encoder [56] as well as others, are merged. Cambrian-1-8B also employs a specific connector called spatial visual aggregator and Llama3.1-8B-Instruct as Large Language Model backbone. Furthermore, the vision encoders are unfrozen during SFT, and the model is trained on more abundant and higher-quality data relative to our setting.

Sa2Va-7B [72] leverages InternVL2.5 [12] as its Multimodal backbone, thereby natively supporting higher resolution via an InternViT encoder. Its Language backbone is InternLM [7], and its training recipe unfreezes the entire architecture on an enlarged and carefully curated dataset, with a specific training run focusing on segmentation and visual grounding tasks.

Model Details		**Comp. Reasoning**		**Vision Centric**	
		ARO	ConMe	MMVP	CVBench
		Avg. Acc.	Acc.	Acc.	2D+3D Acc.
Method	**#Vis. Tok.**				
LLaVA-NeXT-8B	$V \times (K+1)$	70.5	69.4	38.7*	65.3
Cambrian-1-8B	$>> V$	78.8	74.2	51.3*	**76.8**
Sa2Va-7B	$>> V$	**81.2**	**79.2**	**74.0**	75.0
Training data: LLaVA-1.5					
LLaVA-1.5-7B	V	31.5	57.7	33.7	60.1
LLaVA-1.5-8B	V	33.9	61.6	32.0	65.2
OG-LLaVA-8B (Ours)	$T(\approx V)$	56.6	65.2	37.0	63.5

Table 4.4: **Comparison on CR and VC tasks with open-source models**: performance is reported without masks at inference time. The ARO score is a weighted average over its four sub-tasks. Numbers marked with (*) are taken from the corresponding studies, as we were unable to exactly reproduce them. The highest value in each column is in **bold**, the second-highest is underlined. Here $T(\approx V)$ symbolizes the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for both **OG-LLaVA**. K is the number of DHR windows; the notation $>> V$ reflects uncertainty in the exact token count but guarantees it is far larger than V.

Compositional Reasoning and Vision Centric results Table 4.4 juxtaposes three recent state-of-the-art (SOTA) vision–language systems against **OG-LLaVA-8B**. Sa2Va-7B dominates the leader-board, surpassing all contenders in both ConMe and ARO and exceeding 74 % on MMVP. Cambrian-1-8B follows closely, especially on CVBench, where its spatial aggregation of heterogeneous encoders attains a notable 76.8 %. LLaVA-NeXT-8B trails the other SOTA models yet still improves markedly over the LLaVA-1.5 reference. Despite these advances, none of the systems approach ceiling performance: accuracies on compositional reasoning remain in the high seventies at best, and vision-centric scores rarely cross two-thirds of the maximum—evidence that even sophisticated multi-encoder or dynamic-resolution pipelines have

Model Details		General Purpose				
		AI2D	MME		MMStar	MMB
		Acc.	Perc.	Cogn.	Acc.	Dev. Acc.
Method	**#Vis. Tok.**					
LLaVA-NeXT-8B	$V \times (K+1)$	70.5	1562.3	307.5	41.0	70.9
Cambrian-1-8B	$>> V$	73.1	1540.4	375.7	47.7	74.8
Sa2Va-7B	$>> V$	**81.4**	**1651.5**	**587.5**	**62.1**	**82.7**
Training data: LLaVA-1.5						
LLaVA-1.5-7B	V	53.5	1479.7	323.6	34.4	62.5
LLaVA-1.5-8B	V	60.6	1515.3	292.1	40.7	71.6
OG-LLaVA-8B (Ours)	$T(\approx V)$	60.1	1551.5	317.1	38.8	67.3

Table 4.5: **Comparison on General purpose tasks with Open-Source models**: The performance is reported without masks at inference time. The numbers with (*) are taken from the corresponding study, as we were not able to exactly reproduce those results. The highest performance is in **bold** and the second highest is in underline. Here $T(\approx V)$ symbolizes the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for both **OG-LLaVA**. K is the number of DHR windows; the notation $>> V$ reflects uncertainty in the exact token count but guarantees it is far larger than V.

not cracked robust visual–logical understanding. By contrast, **OG-LLaVA** lags the SOTA range (e.g. 65.2 % on ConMe and 63.5 % on CVBench) yet achieves these numbers with a considerable *lower* number of of visual tokens $T \approx V$, eschewing the heavy token inflation of dynamic high-resolution ($V \cdot (K+1)$) or multi-patch expansions ($>> V$). The gap therefore reflects this tokens reduction, the inferior architectural complexity and the absence of additional high-quality data.

General-Purpose image understanding results A similar pattern emerges in Table 4.5. Sa2Va-7B again establishes the strongest overall showing, posting 81.4 % on AI2D and leading all metrics on MME, MMStar, and MMB. Cambrian-1-8B retains second rank, while LLaVA-NeXT-8B remains competitive yet clearly behind. Nevertheless, absolute scores underline unresolved challenges: even the front-runner attains only 62.1 % on MMStar and falls short of 85 % on any task, indicating that current open-source approaches still fall appreciably short of comprehensive visual–commonsense proficiency. In this broader evaluation, **OG-LLaVA** records 60.1 % on AI2D and a respectable 1551.5 on MME, narrowing the gap to the SOTA models while consuming an order-of-magnitude fewer vision tokens and forgoing complex encoder fusion or enlarged corpora. These observations reaffirm that efficiency and principled design choices can yield competitive performance, and they suggest that further architectural refinement and data curation could close much of the remaining distance to the leading open-source systems.

4.6 Qualitative Analysis

Building on the quantitative gains reported in Section 4.2.1, we now turn to a series of visual case studies that make the advantages (and disadvantages) of **OG-LLaVA** tangible. As in Fig. 1.1, each example juxtaposes the prediction of our model with that of the strong baseline LLaVA-1.5. The selected images span a wide spectrum of challenges—attribute distinction, subtle colour difference, depth-of-field cues, fine-grained human pose, material recognition, spatial reasoning, and small-object detection—to stress-test visual–language reasoning under diverse conditions. Crucially, these scenes are drawn *at inference time* with no additional tuning, so performance gains/drops arise solely from the Object-Guided priors baked into **OG-LLaVA**.

Highlights The narrative that follows (Figs. 4.8–4.9) highlights where those priors translate into decisive wins, revealing not only fewer outright mistakes but also more faithful alignment between textual answers and the nuanced visual evidence present in each scene. In Figure 4.8a we report four interesting examples of such cases. In the first picture, our model precisely reads player posture, placing the bat *up and behind* the batter instead of erroneously "in front," evidencing fine-grained pose understanding. The subsequent figure (picture 2) showcases reliable object-colour exclusion: it correctly concludes that *red* is absent from the umbrellas, isolating that hue to the apples and plate, whereas the baseline wrongly flags black. This example is particularly interesting as it stems directly from our *object-centric* pipeline: the segmentation mask confines colour reasoning to the precise umbrella regions instead of the global pixel distribution, preventing spurious cues (e.g., the bright red plate) from bleeding into the model's decision boundary. We continue to demonstrate the superiority of **OG-LLaVA** in depth-of-field reasoning, noting how the model understands the decrease in focus from front to back while the baseline incorrectly treats focus as shifting left-to-right (picture 3). We then highlight scene-material recognition, identifying the trick surface as *concrete*—consistent with skate-park norms—while LLaVA-1.5 mistakes it for asphalt (picture 4).

These improvements are also apparent from the examples in Figure 4.8b. Here the model correctly understands that the distant sail is *inflated with a strong breeze*, exactly aligning this visual cue with the windy surfing scene (picture 5), as well as accurately recognizing the lake's *deep shade of blue* despite the brown boat dominating the background, showing stronger colour discrimination and scene understanding over LLaVA-1.5 (picture 6). **OG-LLaVA** also shows robust small-object detection, spotting a distant *coffee maker* amid kitchen clutter that the baseline misidentifies as a blender, as well as a correct shape identification of the train tracks in the background (pictures 7 and 8).

In Figure 4.9 we report some more examples of our strong capabilities. The model properly recognizes the material of a background shower enclosure (picture 1), while also correctly identifying font characteristics, showing surprisingly good OCR capabilities (picture 2). **OG-LLaVA** can also count better, appropriately identifying giraffes (picture 4), have more nuanced spatial understanding locating people and objects within the image (pictures 3 and 6), as well as have an enhanced understanding of fashion items with a more fine-grained understanding of short vs cap sleeves (picture 7). Our model also thrives in more complex scene understanding. For instance, understanding whether the trees in the image were recently trimmed or seem in good health bearing dense leaves (picture 5), as well as precisely capturing the color reflection of a certain material (picture 8).

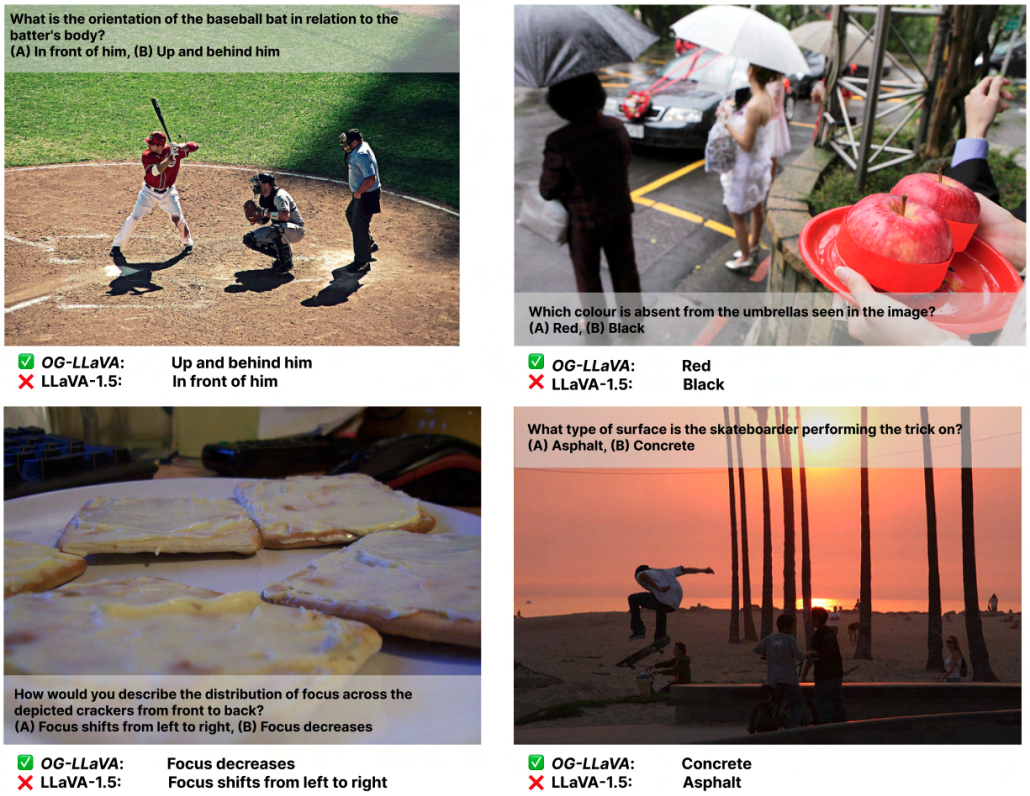
Failure Cases Figure 4.10 presents representative **OG-LLaVA** failure cases. Several errors stem from limited linguistic disambiguation rather than visual misperception. In picture 1, the model conflates edible items with decorative elements, identifying only three pieces of food instead of the five plainly visible on the plate. A comparable misinterpretation occurs in picture 3, where the term *underside* is apparently taken to mean the interior surface of the lid—an aspect obscured by the high camera angle—whereas the benchmark refers to the container's visible base. Additional failure modes arise when the object of interest is too small for reliable delineation by the segmentation model. In Picture 2, for example, the moon occupies only a few pixels in the upper-right corner, rendering it undetectable by the generated masks. Picture 4 highlights a different limitation—scene ambiguity—where the backdrop simultaneously exhibits mountainous terrain and an urban skyline, resulting in contradictory visual cues and an indeterminate model response.

Key Insights Across every instance, we observe the same pattern: once the visual field is partitioned into semantically meaningful regions, **OG-LLaVA** not only identifies each object

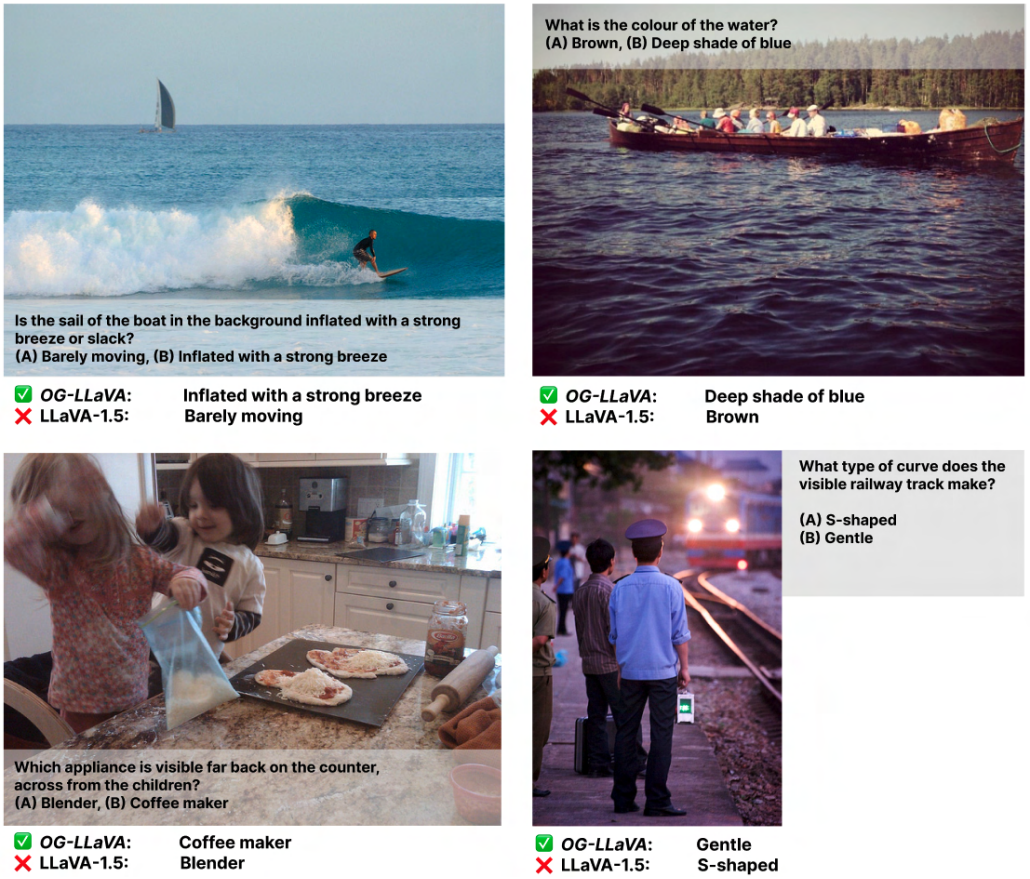
more accurately but also reasons more coherently about *how* those objects interact. Segmentation masks act as spatial priors that decouple local appearance from distracting global context (or "view"), allowing the model to ground attributes where they belong, propagate that grounding to neighboring regions, and, ultimately, build an enhanced model of the entire scene. The payoff is visible far beyond canonical "object tasks": colour disambiguation improves because an apple can no longer bias an umbrella; depth ordering becomes clearer because focus is measured within, not across, regions; material and pose cues emerge because the model attends exactly to the skateboard deck or the batter's stance instead of the surrounding background. In short, object-centric reasoning amplifies *all* facets of visual–language understanding, tightening the alignment between pixels and prose and unlocking richer relational inference than is possible with holistic, mask-free approaches.

The same analysis, however, also pinpoints residual failure modes. Linguistic ambiguities can override otherwise correct visual cues (e.g., conflating *decorations* with food or misinterpreting the term *underside*); objects occupying only a handful of pixels may evade current mask generators, as illustrated by the missed moon; and genuinely ambiguous scenes remain intrinsically challenging. Importantly, these errors are both narrower in scope and lower in frequency than the successes documented above, and many are attributable to tractable shortcomings: higher-resolution segmentation, multi-scale masking, and tighter language–vision alignment are straightforward avenues for further gains.

In aggregate, then, object-centric reasoning consistently amplifies every facet of visual–language understanding while leaving a small, well-defined set of shortcomings. The qualitative evidence presented here mirrors our quantitative results: **OG-LLaVA** delivers materially better performance than its holistic, mask-free counterpart, and the remaining gaps highlight concrete directions for future work rather than fundamental barriers.



(a) ConMe *replace-relation* examples.



(b) ConMe *replace-relation* examples

Figure 4.8: **ConMe *replace-relation* OG-LLaVA vs LLaVA-1.5**. We report two sets of four pictures on the *replace-relation* sub-task of the ConMe [29] benchmark. We highlight different settings in which **OG-LLaVA** has enhanced capabilities over LLaVA-1.5. Pictures (1-8) are referred to in order from left to right starting from the top most left.

Figure 4.9: **ConMe *replace-object* OG-LLaVA vs LLaVA-1.5**. We report two eight pictures on the *replace-object* sub-task of the ConMe [29] benchmark. We highlight different settings in which **OG-LLaVA** has enhanced capabilities over LLaVA-1.5. Pictures (1-8) are referred to in order from left to right starting from the top most left.

Figure 4.10: **ConMe *replace-relation* OG-LLaVA vs LLaVA-1.5**. We report four pictures on the *replace-relation* sub-task of the ConMe [29] benchmark. We highlight different settings in which **OG-LLaVA** has failed with respect to LLaVA-1.5. Pictures (1-4) are referred to in order from left to right starting from the top most left.

Chapter 5

Conclusions & Discussions

5.1 Final Considerations

This work set out to mitigate two long-standing obstacles in Multimodal Large-Language Models: *(i)* the ballooning sequence length that follows patch- or tile-based vision pipelines, and *(ii)* the notoriously weak spatial inductive bias of generic CLIP features. Our solution—**OG-LLaVA** equipped with the lightweight `OG-Fusion` connector—injects explicit, object-centered priors directly into the visual stream while preserving the vanilla LLaVA token budget ($T \approx V$). Across both the LLaVA-1.5 and Cambrian-1 curricula, and for backbones ranging from the compact Llama-3.2-3B to the larger Llama-3.1-8B, this design yields systematic gains: +20% on ARO sub-tasks, +2-4% on ConMe, and +3 % on MMVP, all while maintaining parity on CVBench and improving the perceptual branch of MME. These results confirm that a token-efficient, segmentation-aware adapter can unlock considerably better compositional reasoning and vision-centric grounding without resorting to heavier visual tokenization or multi-encoder fusion.

Ablation insights To understand *why* object guidance is effective—and where its limits lie—we performed a series of controlled ablations. Adding a *Global View* token, i.e. the full encoder grid, was a promising first attempt to supply holistic scene context; yet it produced a 6.4% drop on compositional reasoning and smaller but still negative shifts elsewhere. Our analysis suggests two failure modes: dilution of attention over an over-long sequence and the absence of a dedicated type embedding that distinguishes the global token from ordinary mask patches. We subsequently tested two refinements—an *End-of-Mask (EoM)* delimiter and several flavors of *positional encodings* (fixed/learnable 1-D, 2-D sinusoidal, and segmentation-aware RoPE). Neither remedy closed the gap: the EoM token lowered CR accuracy by 10% and inflated the sequence by $+N$ slots, while explicit positional cues either clashed with the mask-induced permutation or added parameters with no net benefit. These experiments confirm that `OG-Fusion`'s minimalist design already strikes an optimal balance between locality and global context for current transformer decoders.

Comparative landscape When pitted against Subobject-level Image Tokenization (SIT), a sliding-window analogue of Dynamic High Resolution, and three recent open-source MLLMs (LLaVA-NeXT-8B, Cambrian-1-8B, Sa2Va-7B), our model remains highly competitive. **OG-LLaVA-8B** outperforms SIT by $\geq$ 25% on compositional benchmarks and by $\sim$ 10% on vision-centric tasks, all without requiring masks at inference time. Although specialized pipelines such as Sa2Va still lead the absolute tables thanks to larger encoders, richer corpora and end-to-end finetuning, **OG-LLaVA** reaches respectable scores with an order-of-magnitude fewer visual tokens—highlighting the merit of a deliberately minimalist design.

Take-away By marrying off-the-shelf segmentation to a parameter-efficient connector, **OG-LLaVA** demonstrates that robust spatial reasoning need not come at the cost of quadratic token explosion. Its consistent improvements over vanilla LLaVA across data regimes and backbone scales, together with exhaustive ablation evidence, establish object-guided token fusion as a compelling direction for the next generation of vision-language models.

5.2 Limitations & Future Work

While **OG-LLaVA** narrows the gap between rapid prototyping and strong spatial reasoning, several open issues deserve attention in future iterations:

- **Multi-image and video support.** Extending **OG-LLaVA** to handle image sequences or video clips would test whether consistent object IDs across viewpoints and time can further enhance spatial reasoning and compositional understanding in three-dimensional and temporal contexts. Mapping the same object under varying angles via segmentation masks—and linking those tokens across frames—could supply richer priors on occlusion, depth, and motion while keeping the token budget tractable. Key design questions include view-invariant slot embeddings, temporal positional encodings, identity tracking, and efficient windowed attention over long sequences.

- **End-of-Mask (EoM) delimiters as vocabulary tokens.** In Section 4.3.2 we append an EoM token within the embedding space before the MLP. A further experiment would be to introduce the EoM token as a dedicated entry in the LLM's vocabulary. Promoting the delimiter to a learnable token would let the decoder explicitly reset or "re-center" its attention—much as sentence-boundary tokens (`</s>`) help language models gate context across utterances—potentially mitigating the stale-context effect observed in Section 4.3.1.

- **Richer positional encoding schemes.** We confined our analysis to fixed and learnable 1D/2D sinusoidal encodings plus segmentation-aware RoPE. Alternatives such as disentangled attention, rotary-relative hybrids, or the 2D RoPE variant employed in Pixtral-12B [1] may better reconcile locality with permutation invariance; a systematic evaluation remains pending.

- **Alternative vision backbones.** The connector was only paired with CLIP-style ViT encoders. Stronger zero-shot extractors like SigLip [74] or InternViT [50] could supply higher quality region embeddings and finer semantic granularity, but raise questions about cross-modal alignment and computational cost that have yet to be quantified.

- **Segmentation-model choice.** We relied on a single off-the-shelf mask generator (SAM2) [56]. Preliminary evidence hints that mask purity and class granularity affect downstream grounding; ablating the segmentation source—e.g., OMG-Seg [39], SEEM [79], or grounding-aware detectors—would clarify how sensitive `OG-Fusion` is to over/under-segmentation and noisy boundaries.

- **Training data and optimization regimes.** Experiments used the LLaVA-1.5 and Cambrian-1 corpora under frozen-encoder and LoRA constraints. Larger or cleaner multimodal datasets, full-model fine-tuning, and end-to-end unfreezing of the vision stack remain unexplored; early trials suggest that such regimes could bridge the residual gap to state-of-the-art systems at the cost of longer training cycles.

5.3 Broader Applicability

e-Commerce and fashion-intelligence **OG-LLaVA**'s ability to reason compositionally over fine-grained object features and spatial relations has immediate implications for on-line retail, where visual richness and accurate product understanding drive both customer experience and operational efficiency. By fusing dense segmentation masks with globally descriptive CLIP embeddings, the model can isolate individual garments (e.g., jacket, blouse, belt) within complex lifestyle shots, predict nuanced attributes such as fabric texture, pattern style, and embellishment details, and reliably map them to merchandise taxonomy and product-attribute structures, picture 7 Figure 4.9. This lays the groundwork for automated aspect prediction pipelines that populate product listings with consistent, high-recall attribute tags—reducing manual annotation costs and improving long-tail *searchability*. Furthermore, the model's compositional reasoning enables outfit compatibility and generation engines: it can infer how colors, silhouettes, and materials interact across multiple items in an image, then suggest complementary pieces or complete ensembles personalized to a shopper's style profile. Because **OG-LLaVA** achieves these gains without inflating token budgets or fine-tuning the vision backbone, it fits naturally into real-time recommendation loops and mobile AR "try-on" experiences where latency and compute are at a premium.

Robotics and embodied AI In household-robot or warehouse-automation settings, agents must identify specific objects, parse their relations (e.g., "the red cup inside the top-right bin"), and plan manipulation sequences accordingly. **OG-LLaVA**'s lightweight `OG-Fusion` lets such agents run richer visual reasoning on edge hardware, boosting success rates in multi-step fetch-and-place tasks without prohibitive inference costs.

Scientific and industrial inspection Fine-grained segmentation fused with linguistic reasoning is valuable for medical imaging (marking tumor boundaries while explaining tissue attributes), remote-sensing change detection (highlighting deforestation patches and describing land-use transitions), and quality-control pipelines (finding micro-defects on assembly lines and linking them to probable causes). Because **OG-LLaVA** leaves the vision encoder frozen, domain adaptation can happen through lightweight language-side tuning, lowering the barrier to adoption in specialized verticals.

Multimodal content moderation and accessibility Beyond retail, **OG-LLaVA** can act as a guardrail for user-generated platforms by detecting policy-violating visual content and its textual context in a single forward pass, even when harmful cues are subtle or spatially dispersed. The same dense grounding capabilities support automatic alt-text generation and scene verbalization, improving web accessibility for visually impaired users.

Collectively, these application domains underscore the broader impact of our connector design: by enriching token-efficient vision features with object-level semantics, **OG-LLaVA** unlocks practical multimodal reasoning in scenarios where both accuracy and computational frugality are non-negotiable.

Bibliography

[1] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b, 2024.

[2] Rabiul Awal, Saba Ahmadi, Le Zhang, and Aishwarya Agrawal. Vismin: Visual minimal-change understanding, 2025.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[5] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13154–13164, 2023.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[7] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze

Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024.

[8] Delong Chen, Samuel Cahyawijaya, Jianfeng Liu, Baoyuan Wang, and Pascale Fung. Subobject-level image tokenization, 2025.

[9] Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Tuomas Rintamaki, Tyler Poon, Max Ehrlich, Tuomas Rintamaki, Tyler Poon, Tong Lu, Limin Wang, Bryan Catanzaro, Jan Kautz, Andrew Tao, Zhiding Yu, and Guilin Liu. Eagle 2.5: Boosting long-context post-training for frontier vision-language models, 2025.

[10] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023.

[11] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024.

[12] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025.

[13] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language models, 2024.

[14] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[15] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2818–2829. IEEE, June 2023.

[16] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[17] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[20] Abhimanyu Dubey et al. The llama 3 herd of models, 2024.

[21] Marah Abdin et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

[22] Matt Deitke et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models, 2024.

[23] 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.

[24] Gemma-Team. Gemma 3 technical report, 2025.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36:31096–31116, 2023.

[27] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality, 2023.

[28] 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, 2021.

[29] Irene Huang, Wei Lin, M. Jehanzeb Mirza, Jacob A. Hansen, Sivan Doveh, Victor Ion Butoi, Roei Herzig, Assaf Arbelle, Hilde Kuehne, Trevor Darrell, Chuang Gan, Aude Oliva, Rogerio Feris, and Leonid Karlinsky. Conme: Rethinking evaluation of compositional reasoning for modern vlms, 2024.

[30] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[31] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[34] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[35] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024.

[36] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

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

[38] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27948–27959, June 2024.

[39] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation?, 2024.

[40] 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.

[41] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024.

[42] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[43] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[44] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024.

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[46] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. Mm1: Methods, analysis & insights from multimodal llm pre-training, 2024.

[47] Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models, 2024.

[48] Matteo Nulli, Anesa Ibrahimi, Avik Pal, Hoshe Lee, and Ivona Najdenkoska. In-context learning improves compositional understanding of vision-language models. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024.

[49] OpenAI. Gpt-4 technical report, 2024.

[50] OpenGVLab-Team. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024.

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[53] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[54] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2020.

[55] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery.

[56] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024.

[57] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International conference on machine learning*, pages 29441–29454. PMLR, 2023.

[58] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[59] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality, 2022.

[60] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms, 2024.

[61] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024.

[62] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[63] 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.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[65] Bo Wan, Michael Tschannen, Yongqin Xian, Filip Pavetic, Ibrahim Alabdulmohsin, Xiao Wang, André Susano Pinto, Andreas Steiner, Lucas Beyer, and Xiaohua Zhai. Locca: Visual pretraining with location-aware captioners, 2024.

[66] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024.

[67] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms, 2023.

[68] XAI. Grok-1.5 vision preview. real-world understanding, 2024.

[69] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2024.

[70] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014.

[71] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022, 2022.

[72] Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos, 2025.

[73] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023.

[74] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[75] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, Sam Dodge, Keen You, Zhen Yang, Aleksei Timofeev, Mingze Xu, Hong-You Chen, Jean-Philippe Fauconnier, Zhengfeng Lai, Haoxuan You, Zirui Wang, Afshin Dehghan, Peter Grasch, and Yinfei Yang. Mm1.5: Methods, analysis & insights from multimodal llm fine-tuning, 2024.

[76] Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding, 2024.

[77] Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations, 2023.

[78] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

[79] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once, 2023.

Appendix A

Appendix

A.1 Related Work & Background

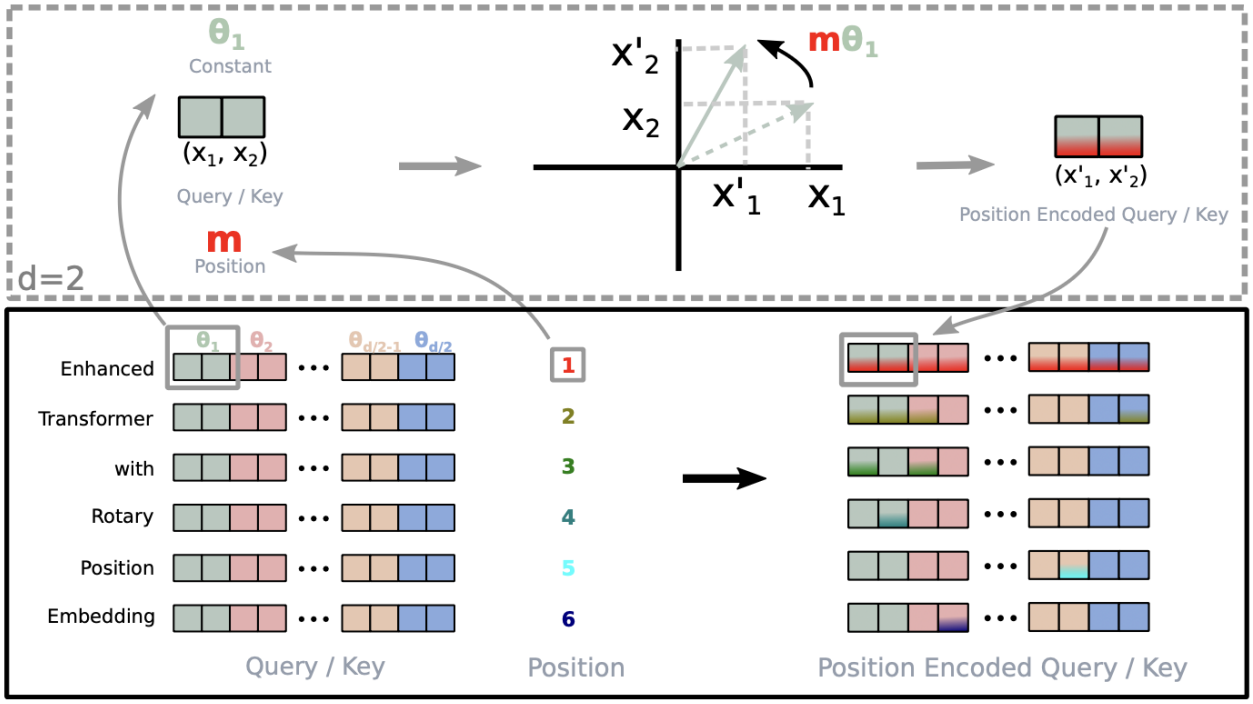


Figure A.1: **Rotary Positional Embedding (RoPE) illustration**: We show the RoPE [58] mechanism, which is used to encode positional information in the input tokens. The figure illustrates how RoPE applies a rotation to the token embeddings based on their position in the sequence.

A.2 Methodology

A.2.1 Vision Transformers

A.2.2 Implementation of Downsampling Operator Φ_α

```
def downsample_operator_phi(
    mask: torch.Tensor,
```

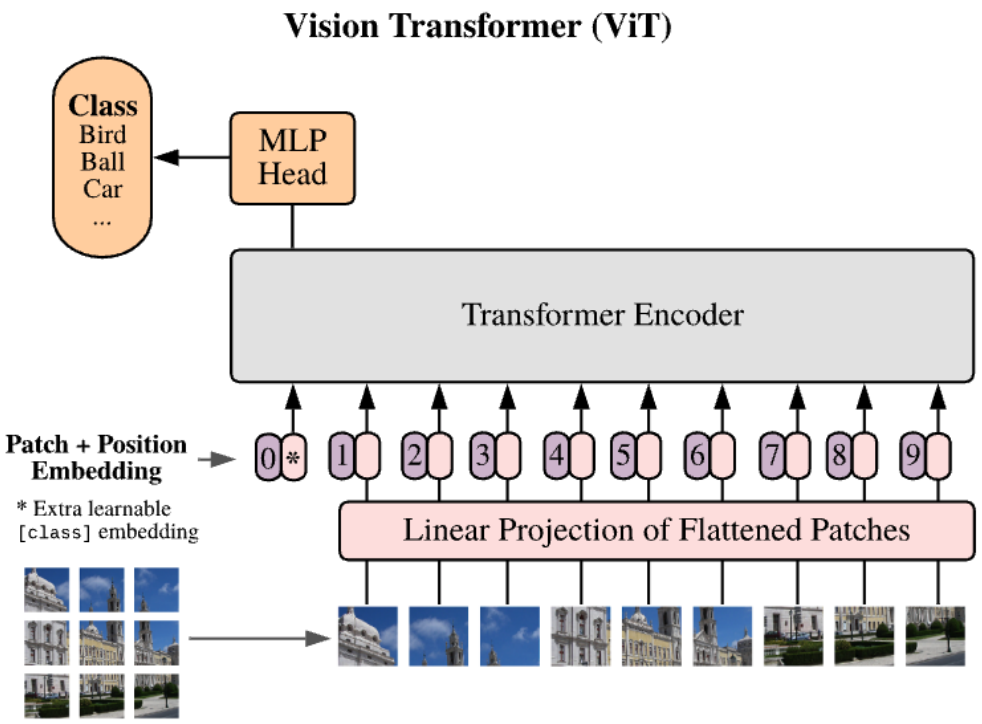


Figure A.2: **Illustration of Vision Transformer**: Figure taken from [19] showcasing vision transformers architecture.

```
    output_size: int,
    threshold_count: float = 0.5,
    device: torch.device = "cpu",
) -> torch.Tensor:
    """
    Downsamples a 2D boolean mask to a 1D boolean tensor of length 'output_size' while preserving the information of True pixels.

    This function uses adaptive average pooling to compute the average (i.e. the fraction of True
    values) in each bin, multiplies by the approximate bin size to get a count, and then marks a bin
    as True if that count is above a threshold.

    Parameters:
    mask (torch.Tensor): Input mask (2D) of booleans or 0/1 values.
    output_size (int): The desired length of the final 1D mask.
    threshold_count (float): A threshold on the
    count of True pixels per bin.
                            Default 0.5 means that if a bin receives at least 1 True pixel
                            (on average) it will be marked True.

    Returns:
    torch.Tensor: A 1D boolean tensor of length 'output_size'.
    """
    # Convert mask to float and flatten
    mask_flat = (
        mask.float().flatten().contiguous().unsqueeze(0).unsqueeze(0)
    )  # shape: (1,1,N)

```

```
    # Compute approximate number of pixels per bin.
    total_pixels = mask.numel()
    assert mask.numel() > 0, "Input mask is empty"
    bin_size = (
        total_pixels / output_size
    )  # average number of original pixels per output bin

    # Adaptive average pooling: each bin now contains the fraction of True pixels over ~bin_size pixels.
    # print("mask_flat shape device", mask_flat.shape, mask_flat.device)
    # print("output_size", output_size)
    pooled = torch.nn.functional.adaptive_avg_pool1d(
        mask_flat, output_size
    ).squeeze()  # shape: (output_size,)

    # Convert the fraction to an estimated count per bin.
    counts = pooled * bin_size

    # Binarize: mark a bin as True if the estimated count is at least threshold_count.
    downsampled_mask = counts >= threshold_count
    return downsampled_mask
```

Listing A.1: Python implementation of Φ_α operator

A.3 Experiments

A.3.1 Benchmark Examples

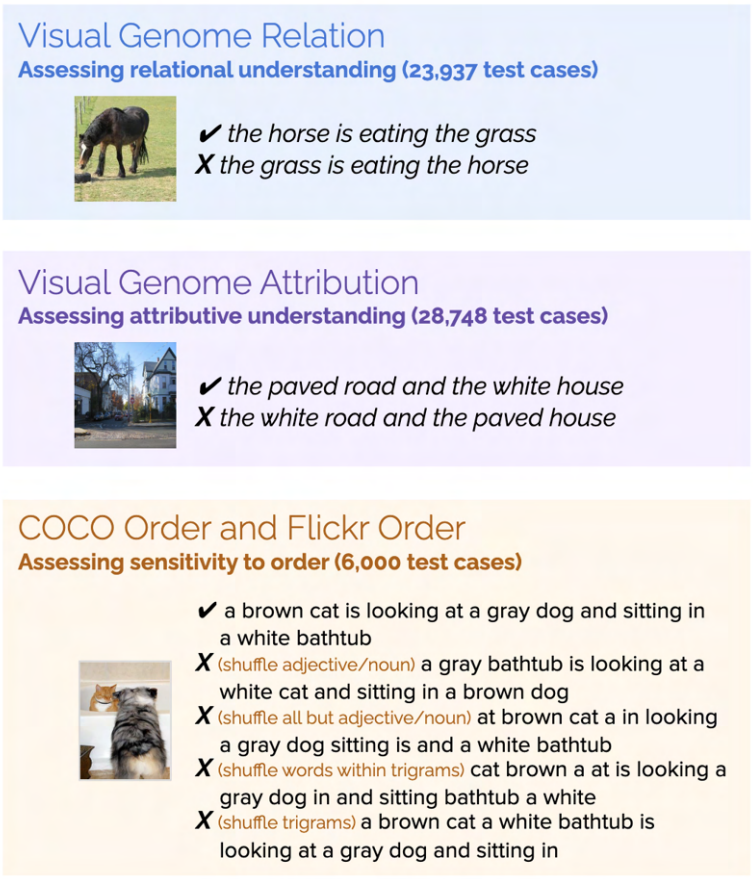


Figure A.3: **ARO benchmark examples**: The picture shows the four different subsets of ARO [73], Visual Genome Relation, Visual Genome Attribution, COCO Order and Flickr Order.

Image	Baseline	Pipeline
	Original positive: Two women are squatting down and petting brown and white long haired goats. Original negative: Two women are squatting down and petting brown and black long haired goats.	Question: How many animals are visible in the immediate group around the person in the center? Correct: At least three Negative: At least four
	Original positive: A man riding skis down a snow covered slope. Original negative: A man riding a snowboard down a snow covered slope.	Question: What is the effect of the sunlight on the snow surface? Correct: Casting shadows on the snow Negative: Fully illuminating the snow without shadows
	Original positive: a kid stands in the snow on his skiis Original negative: A kid stands in the snow next to his skis.	Question: What specific accessory does the person have around their neck and lower face region? Correct: A scarf Negative: Goggles

Figure A.4: **ConME vs SugarCrepe qualitative example**: In the center the original SugarCrepe [26] prompts, on the right the more accurate ConMe [29] questions

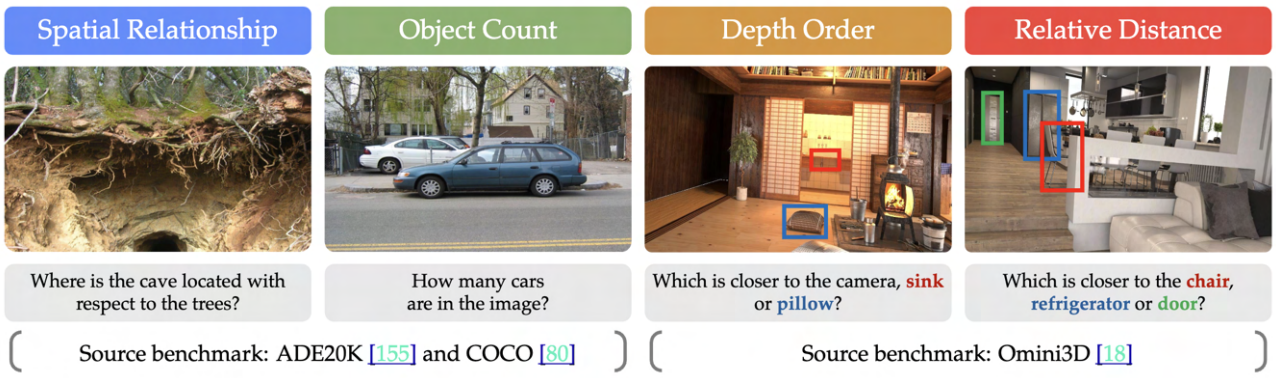


Figure A.5: **CVBench benchmark examples**: We show the four subtasks in CVBench [60], Spatial Relationship, Object Counting, Depth Order and Relative Distance.