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

Matteo Nulli^{1,2,*}**, Ivona Najdenkoska**^{1,4}**,**
Mohammad Mahdi Derakhshani^{1} **and Yuki M. Asano**^{3}
^{1}University of Amsterdam, ^{2}eBay Inc.
^{3}University of Technology Nuremberg, ^{4}Netherlands Forensics Institute - NFI

Abstract

Multimodal Large Language Models (MLLMs) employ contrastive pre-trained Vision Encoders whose performance falls short in compositional understanding and visual reasoning. This is mostly due to their pre-training objective aimed at retrieval between similar images or captions rather than an 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** (**O**bject-**G**uided **LLaVA**), a novel multimodal architecture which, through an innovative 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 representations with segmentations. By leveraging the descriptive power of advanced segmentation models, OG-LLaVA attains superior performance at tasks that require a deeper understanding of object relationships and spatial arrangements, within the domains of compositional reasoning and visual grounding. The code is available at `https://github.com/MatteoNulli/og_llava/tree/main`.

1 Introduction

Multimodal Large Language Models [3, 4, 5, 6] have made rapid gains across captioning, visual question-answering, and multi-step reasoning, yet they remain brittle on deep image understanding because they lack strong compositional reasoning [7, 8]. By compositional reasoning we mean the ability to factor a scene into objects and attributes, model their spatial relations, and recombine that structure to draw context-sensitive inferences—far beyond surface pattern matching [7, 1, 9]. Closing this gap likely requires training that rewards spatial relational correctness and architectures that tightly bind visual evidence to language, shifting emphasis toward visual-centric design [10]. Recent work encodes images with more structure: dynamic high-resolution tiling/patching, aspect-ratio–aware intake to avoid distortion, and segmentation-derived features for object-level semantics [11, 12, 13, 6, 14, 15, 16]. 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 often a Multi-Layer Perceptron (MLP) [3]. Still, funneling generic encoder features through narrow connectors creates an information bottleneck—compounded by the spatial myopia of CLIP-style representations [17], due to the documented poor retrieval-like training strategy [7, 8, 2].

[*] Correspondence to: `mnulli@ebay.com`

EurIPS 2025 Workshop on Principles of Generative Modeling (PriGM).

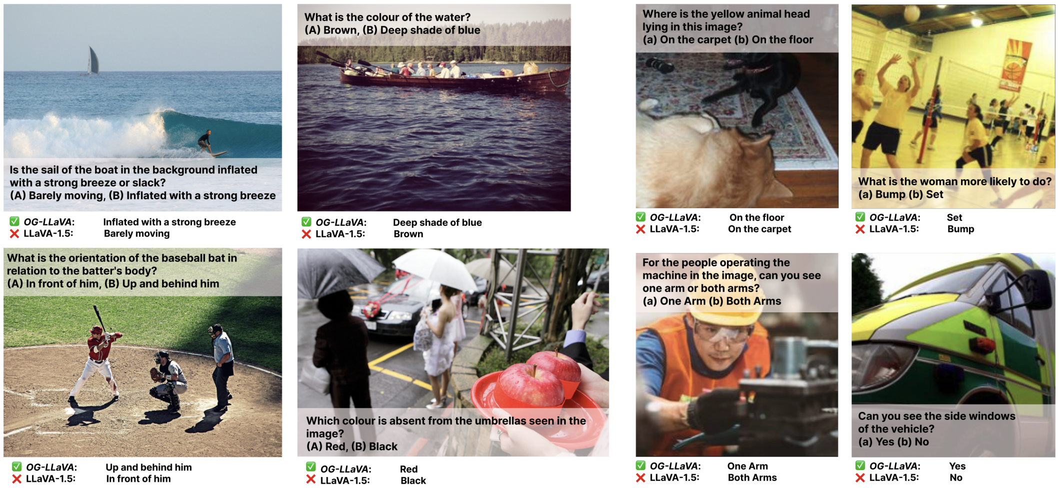

Is the sail of the boat in the background inflated with a strong breeze or slack?
(A) Barely moving, (B) Inflated with a strong breeze
✓ OG-LLaVA: Inflated with a strong breeze
✗ LLaVA-1.5: Barely moving
What is the colour of the water?
(A) Brown, (B) Deep shade of blue
✓ OG-LLaVA: Deep shade of blue
✗ LLaVA-1.5: Brown
Where is the yellow animal head lying in this image?
(a) On the carpet (b) On the floor
✓ OG-LLaVA: On the floor
✗ LLaVA-1.5: On the carpet
What is the woman more likely to do?
(a) Bump (b) Set
✓ OG-LLaVA: Set
✗ LLaVA-1.5: Bump
What is the orientation of the baseball bat in relation to the batter's body?
(A) In front of him, (B) Up and behind him
✓ OG-LLaVA: Up and behind him
✗ LLaVA-1.5: In front of him
Which colour is absent from the umbrellas seen in the image?
(A) Red, (B) Black
✓ OG-LLaVA: Red
✗ LLaVA-1.5: Black
For the people operating the machine in the image, can you see one arm or both arms?
(a) One Arm (b) Both Arms
✓ OG-LLaVA: One Arm
✗ LLaVA-1.5: Both Arms
Can you see the side windows of the vehicle?
(a) Yes (b) No
✓ OG-LLaVA: Yes
✗ LLaVA-1.5: No

Figure 1: **OG-LLaVA vs LLaVA-1.5 on compositional reasoning**. The figure shows **OG-LLaVA** strengths across the *replace-attribute* sub-task of the ConMe [1] benchmark (*left four pictures*), and the MMVP [2] benchmark (*right four pictures*).

To address (a) the ever-growing 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 [18] designed to improve spatial acuity without increasing the visual token budget. At its core is `OG-Fusion`, a lightweight connector that injects segmentation-based, object-centric priors directly into the vision stream, producing compact ***O***bject-***G***uided ***V***isual ***T***okens (***OGVT***). Our main contributions are as follows: (i) ***Object-Guided Visual Tokens.*** We present **OG-LLaVA** with `OG-Fusion` an innovative connector design fusing segmentation cues with CLIP features, reinstating spatial locality, and representing each object with a compact token block. (ii) ***Improved compositional reasoning.*** Across both LLaVA-665k [18] and Cambrian-7M [10] curricula, **OG-LLaVA** consistently outperforms vanilla counterparts. It shows a stronger understanding of object interactions, spatial relations, and layered semantics, with substantial gains on ConMe [1], ARO [7], and MMVP [2]. Qualitative improvements in Fig. 1. (iii) ***Token Efficiency.*** By representing each segmented region with a proportional token block, `OG-Fusion` maintains computational efficiency, avoiding quadratic cost growth of tiling or cropping while keeping sequence length comparable to vanilla LLaVA, maintaining scalability, unlike alternative tokenization schemes.

2 Methodology

2.1 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. In the next sections, we describe our internal process and assume a batch size of one. For a mathematical and rigourous formulation, see 5.2 in the Appendix.

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. Once the visual features $\mathbf{X}' \in \mathbb{R}^{V \times F}$ 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 a function returning a down-sampled binary mask whose entries indicate whether bin (storing the concentrated information from neighboring pixels) k contains at least a number α of foreground pixels. We apply this procedure to all N masks and obtain the set of down-sampled binary masks:
$\mathbf{M}' = \left\{ \Phi_\alpha(\mathbf{m}_i) \;\middle|\; i = 1, \ldots, N \right\} \subset \{0,1\}^V$, more on the operator implementation Φ_α in 5.1.1.

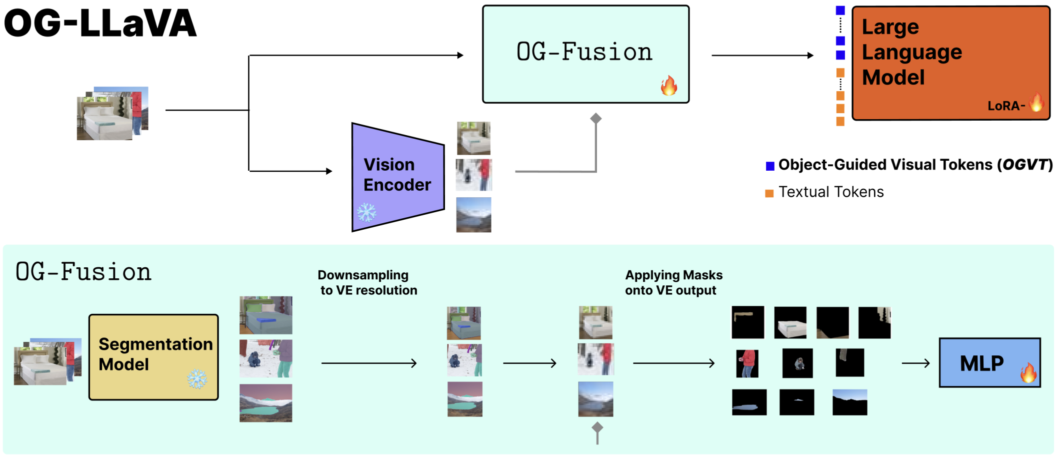


Figure 2: **OG-LLaVA with** `OG-Fusion`: The input image flows through two synchronized paths. One path uses a Vision Encoder to produce image features. In parallel, `OG-Fusion` (i) obtains object masks via a segmentation model, (ii) downsamples the resulting segmentations, and (iii) applies these masks to the encoder features. The masked embeddings are then (iv) concatenated and passed through an MLP to yield Object-Guided Visual Tokens ***(OGVT)***. The LLM consumes ***OGVT*** together with text to generate the final output.

Object-Guided Visual Tokens After these pre-processing steps, we can apply the down-sampled segmentations $\mathbf{M}'$ onto $\mathbf{X}'$, through an m-indexed row-selection matrix P_i, obtaining $\mathbf{Y}_i = P_i\,\mathbf{X}' \in \mathbb{R}^{t_i \times F}$. With these down-sampled visual fragments we define,

$$\boxed{\boldsymbol{OGVT} \;:=\; MLP(\mathbf{Y}) \;\in\; \mathbb{R}^{T\times D}} \tag{1}$$

where T equals the total number of object-bearing bins retained by the masks, its value varies *per image*—yet in expectation $T \approx V$. Note that when two masks overlap on a ViT bin, that bin appears in multiple $\mathbf{Y}_i$. Because masks are downsampled and thresholded and the LLM uses RoPE, this implies a change in the projections, meaning these repeated bins do not behave as identical keys and the overall visual sequence length remains $T \approx V$. Clearly this implies a much larger portion of the attention goes into areas with high density or large amount of object. This however is not necessarily an issue. Re-instating spatial-bias, lost due to the nature of Transformers, implies our model is more grounded by design. We go over a more rigorous definition of the effect of token duplication in our architecture in the Appendix 5.2.2.

2.2 Model Architecture, Training & Inference

To keep the same architectural structure as LLaVA-1.5, we choose as Vision Encoder CLIP ViT-L/14@336 [19, 17]. Regardless of our particular choice of CLIP, our approach applies to any kind of encoder backbone. We experiment mostly with two LLMs, Llama3.1-8B-Instruct [20] and Llama3.2-3B-Instruct [21]. In Figure 2 we show an overview of our `OG-Fusion` process. Architecturally, it is comprised of a Segment Anything Model 2 (SAM2) [12] with a frozen backbone, along with a series of detailed procedures from Sections 2.1 and a Multi-Layer Perceptron with 2 hidden layers with input size of 1024 and GeLU activations. Given these considerations, in this work we present two versions of **OG-LLaVA**: **OG-LLaVA-3B** & **OG-LLaVA-8B**: trained with Llama3.2-3B-Instruct & Llama3.1-8B-Instruct as LLM. We follow [3] and our definitions in Section 2, training our model using an auto-regressive objective, through two different stages. First, in Vision-Language Alignment only unfreezing `OG-Fusion`. Secondly, we perform Supervised Fine-Tuning, unfreezing the LLMs with LoRA [22] and `OG-Fusion`. Although during training we follow Eq. 1, the model can be evaluated both with and *without* the segmentation masks infusing. This robustness comes from using components of the original visual representations with targeted modifications, preserving semantic understanding of $\mathbf{X}'$.

3 Experiments

3.1 Experimental Setup

Training Data We employ BLIP-Laion during Vision-Language Alignment and LLaVA-665k [18] and Cambrian-7M [10] at Supervised Fine-Tuning. More information is available in the Appendix, section 5.3.1.

Evaluation Datasets & Metrics To evaluate compositional reasoning we choose two large scale evaluation sets: **ARO** [7], made up of 23,937 examples, evaluates models by asking them to pick the correct caption among five, combining data from various sources [23, 24, 25, 26], and **ConMe** [1], extends compositional QA beyond SugarCrepe [27], using negative text generation with VLMs [28, 18, 29] with 24,347 samples. To assess vision-centric performance, we consider two well established benchmarks, **MMVP** [2] targeting cases where CLIP-like models fail, using multiple-choice questions, constructed with 300 CLIP-blind pairs. And **CVBench** [10], evaluates with 2,638 samples multimodal capabilities in 2D and 3D tasks often overlooked in benchmarks. For tracking general image understanding performance we benchmark our models on these four: **AI2D** [30], **MME** [31], **MMStar** [32], **MMBench** [33], evaluating tasks from Diagram/ChartQA, Perception and Cognition to Math, Logic and Instance Reasoning.

3.2 Results

In this section we compare our **OG-LLaVA** with LLaVA-1.5 range of models. Specifically, we have: **LLaVA-1.5-3B**, **LLaVA-1.5-7B** & **LLaVA-1.5-8B** with Llama3.2-3B, Vicuna-7B & Llama3.1-8B as LLMs. All trained with both LLaVA-665k and the first one with Cambrian-7M data as well. All are trained following LLaVA-1.5 specifications and architecture components.

Table 1 highlights how **OG-LLaVA**, across both LLaVA-665k and Cambrian-7M, consistently surpasses baselines: on ARO, gains are large across datasets/backbones—about $+21pp$ on *CO* and $+16pp$ on *FO* (highest gaps 38.2 → 82.6 & 49.1 → 84.0); *VA* improves by $10pp$ on average and *VR* by about $20pp$. ConME also rises ($\geq +2pp$), reaching 65.2 in the 8B setting (+3.6 over the strongest baseline). Notably, the lightweight Llama3.2-3B backbone preserves most compositional reasoning accuracy—leading on *VA, VG* subtasks—while offering a favorable latency–accuracy trade-off for both training and inference. These leaps carry over to visual-centric evaluation: MMVP increases by roughly three points on average (e.g., 32.0 → 37.0 at 8B; 61.6 → 66.0 with Cambrian-7M), while CVBench remains essentially stable (within 1 point). On general image understanding, **OG-LLaVA** also performs strongly: under LLaVA-665k, the 8B model yields large MME gains ($+36.2pp$ perception to 1551.5; $+25.0pp$ cognition to 317.1) with small dips on AI2D and MMStar; the 3B model stays competitive on MMBench; and with Cambrian-7M, **OG-LLaVA-3B** climbs further (e.g., AI2D 66.5, MME (Perc.) 1511.7, MMBench 70.91), indicating that token-balanced training ($T \approx V$) helps object-guided tokens transfer broadly with minimal losses on outlier tasks. This negligible performance degradation on General Image Understanding multiple tasks, (MME on Perception and Cognition, MMStar with 6 core capabilities and MMB $\geq$ 20 ability axes) is an indication of good generalization capabilities. These results confirm that a token-efficient, segmentation-aware adapter can unlock considerably better compositional reasoning and visual-grounding without resorting to heavier visual tokenization.

Model Details		**Compositional Reasoning**					**Vision Centric**		**General Image Understanding**				
		ARO				ConMe	MMVP	CVBench	AI2D	MME		MMStar	MMB
		CO	FO	VA	VR	Acc.	Acc.	2D+3D Acc.	Acc.	Perc.	Cogn.	Acc.	Dev. Acc.
Method	**#Vis. Tok.**												
Training data: LLaVA-665k													
LLaVA-1.5-3B	V	63.8	70.9	60.2	52.4	59.6	**59.7**	63.3	**59.2**	**1407.1**	**330.0**	**37.7**	**67.4**
OG-LLaVA-3B (Ours)	$T(\approx V)$	**79.1**	**82.2**	**75.2**	**75.5**	**61.2**	57.3	**63.5**	56.8	1394.2	325.3	37.5	67.6
LLaVA-1.5-7B	V	36.2	44.1	28.1	28.2	57.7	33.7	60.1	53.5	1479.7	323.6	34.4	62.5
LLaVA-1.5-8B	V	38.2	49.1	28.3	29.7	61.6	32.0	**65.2**	**60.6**	1515.3	292.1	**40.7**	**71.6**
OG-LLaVA-8B (Ours)	$T(\approx V)$	**82.6**	**84.0**	**38.6**	**45.3**	**65.2**	**37.0**	62.5	60.1	**1551.5**	317.1	38.8	67.3
Training Data: Cambrian-7M													
LLaVA-1.5-3B	V	71.0	76.8	72.6	27.5	**68.6**	61.6	**66.2**	65.4	1480.9	**328.2**	**41.4**	**70.1**
OG-LLaVA-3B (Ours)	$T(\approx V)$	**73.7**	**79.5**	**79.2**	**50.8**	66.7	**66.0**	64.5	**66.5**	**1511.7**	300.6	38.3	70.9

Table 1: **OG-LLaVA vs LLaVA baselines across compositional reasoning, vision centric and general-purpose tasks.** Results are reported without segmentation masks at inference. Highest values are in **bold**; second-highest are underlined. Here $T(\approx V)$ denotes the *#Vis. Tok.* during training, with V the *#Vis. Tok.* at inference.

Furthermore, we would like to stress how these gains are not an artifact of *more mass*. In our appendix experiments 5.4.1, we show how, doubling the visual tokens by appending a global unmasked grid ("Global View") corresponds to accuracy drop (-6.4% on compositional reasoning Fig. 10, left). This implies that simply adding tokens or attention mass is counter-productive and that our improvements come from object-guided selection, rather than indiscriminate up-weighting.

4 Discussions & Conclusions

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-665k and Cambrian-7M curricula, and for backbones ranging from the compact Llama-3.2-3B to the larger Llama-3.1-8B, this design yields systematic gains, especially on ARO CONME and MMVP, all while maintaining parity on CVBENCH and improving the perceptual branch of MME. See Section 5.5 for a qualitative analysis of our gains.

Limitations Despite strong results, our approach has caveats. First, computing segmentation masks adds modest training-time overhead, which can hinder latency-critical use. At inference, masks are optional, removing the extra overhead of alternatives [34] which would not function without. Second, our experiments ran under compute constraints: we reduced both training data scale and the degree of model unfreezing. These choices, while unavoidable, disadvantage us relative to fully trained state-of-the-art models [10, 11, 16]. This limitation is largely orthogonal to the merits of **OG-LLaVA** & `OG-Fusion`. Due to the considerable performance improvements over compositional reasoning and little to no degradation on models with same backbones and training data (LLaVA-1.5), where the only difference is `OG-Fusion`, with more compute, we expect much of the gap to close. Moreover, **OG-LLaVA** performance can be limited by the capability of upstream segmentation model (see Section 5.5), similarly to how standard MLLMs like LLaVA-1.5 are limited by their Vision Encoder performance. Finally, a thorough study of how different vision encoders can be adapted to this framework is deferred to future work.

References

[1] 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. URL https://arxiv.org/abs/2406.08164.

[2] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024. URL https://arxiv.org/abs/2401.06209.

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

[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. URL https://arxiv.org/abs/2502.13923.

[5] OpenGVLab-Team. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024. URL https://internvl.github.io/blog/2024-07-02-InternVL-2.0/.

[6] Gemma-Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

[7] 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. URL https://arxiv.org/abs/2210.01936.

[8] 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. URL https://arxiv.org/abs/2407.15487.

[9] Rabiul Awal, Saba Ahmadi, Le Zhang, and Aishwarya Agrawal. Vismin: Visual minimal-change understanding, 2025. URL https://arxiv.org/abs/2407.16772.

[10] 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. URL https://arxiv.org/abs/2406.16860.

[11] 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. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

[12] 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. URL https://arxiv.org/abs/2408.00714.

[13] 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.

[14] 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. URL https://arxiv.org/abs/2504.15271.

[15] 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. URL `https://arxiv.org/abs/2406.19389`.

[16] 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. URL `https://arxiv.org/abs/2501.04001`.

[17] 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. URL `https://arxiv.org/abs/2103.00020`.

[18] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. URL `https://arxiv.org/abs/2310.03744`.

[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. URL `https://arxiv.org/abs/2010.11929`.

[20] 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.

[21] Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models, 2024. URL `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`.

[22] 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. URL `https://arxiv.org/abs/2106.09685`.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[29] 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.

[30] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016. URL https://arxiv.org/abs/1603.07396.

[31] 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. URL https://arxiv.org/abs/2306.13394.

[32] 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. URL https://arxiv.org/abs/2403.20330.

[33] 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. URL https://arxiv.org/abs/2307.06281.

[34] Delong Chen, Samuel Cahyawijaya, Jianfeng Liu, Baoyuan Wang, and Pascale Fung. Subobject-level image tokenization, 2025. URL https://arxiv.org/abs/2402.14327.

[35] 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. ISBN 9781450379984. doi: 10.1145/3394486.3406703. URL https://doi.org/10.1145/3394486.3406703.

[36] 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. doi: 10.1109/SC41405.2020.00024.

[37] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[39] 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.

[40] 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.

[41] 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. URL https://arxiv.org/abs/2410.07073.

[42] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL `https://arxiv.org/abs/2104.09864`.

[43] Abhimanyu Dubey et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

[44] 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. doi: 10.1109/cvpr52729.2023.00276. URL `http://dx.doi.org/10.1109/CVPR52729.2023.00276`.

[45] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL `https://arxiv.org/abs/2303.15343`.

[46] 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. URL `https://arxiv.org/abs/2304.07193`.

[47] 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. URL `https://arxiv.org/abs/2403.17297`.

[48] 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. URL `https://arxiv.org/abs/2401.10229`.

[49] 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. URL `https://arxiv.org/abs/2304.06718`.

5 Appendix

5.1 Methodology

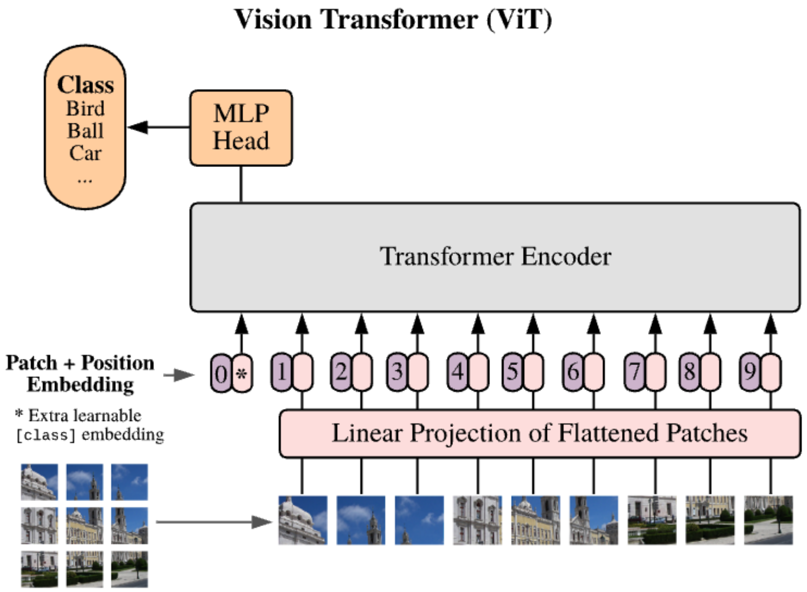

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 3: **Illustration of Vision Transformer**: Figure taken from [19] showcasing vision transformers architecture.

5.1.1 Implementation of Downsampling Operator Φ_α

```python
def downsample_operator_phi(
    mask: torch.Tensor,
    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 1: Python implementation of Φ_α operator

5.2 Object-Guided Visual Tokens with Mathematical Formulation

This is an expanded, more mathematically rigorous explanation of Section 2.1.

5.2.1 Premise: 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 the union between vision and text vocabulary spaces $\mathcal{V} = \mathcal{V}_{\text{vision}} \cup \mathcal{V}_{\text{text}}$. The Large Language Model is defined as g_ϕ mapping 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 [17], are used in MLLMs to encode an image $\mathbf{X}$ into a representation:

$$\mathbf{X}' = f_{v\theta}(\mathbf{X}) \in \mathbb{R}^{V\times F}, \tag{2}$$

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 3 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 $\mathbf{VT} = m_\gamma(\mathbf{X}') \in \mathbb{R}^{V\times D}$, 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 into textual tokens with $TT = E^{\otimes}(\mathcal{T}(t)) \in \mathbb{R}^{J\times D}$, 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 $\mathbf{T}_a = g_\phi(\mathbf{VT} \oplus TT) \in \mathcal{V}^J$.

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. During training each encoder extracts the feature representations $t' = f_{t\theta'}(t),\ \mathbf{X}' = f_{v\theta}(\mathbf{X})$. These are then normalized $\mathbf{X}'_e = t'_e \ := \ \mathbf{X}'\, t' / ||\mathbf{X}'\, t'||^2$, and used to compute the pairwise cosine similarities, *logits* $=$ $(\mathbf{X}'_e \cdot t'^{T}_e) \cdot e^t$. These logits are used to compute the joint loss function using cross-entropy (CE), with $\mathcal{L}_{\mathbf{X}} = \mathrm{CE}(\textit{logits}, \textit{labels}, \text{axis} = 0), \mathcal{L}_t = \mathrm{CE}(\textit{logits}, \textit{labels}, \text{axis} = 1)$,

$$\mathcal{L} = \tfrac{1}{2}\left(\mathcal{L}_{\mathbf{X}} + \mathcal{L}_t\right).$$

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 [7, 2]. 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.

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.

5.2.2 Detailed Process

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 ($f_{v\theta}$) following 2. 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:
$\mathcal{F} : \{0,1\}^{H \times W} \ \to \ \{0,1\}^{HW}, \quad \mathcal{F}(\mathbf{M}) \ = \ \mathrm{vec}(\mathbf{M})$
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 $\{1, \ldots, H{\cdot}W\}$ into equally-sized, contiguous blocks $B_k,\ k = 1, \ldots, V$ of length $\text{size}_b \ = \ H \cdot W/V$, corresponding to the average number of pixels per output bin. Subsequently, we define:

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

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

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

We apply this to all N masks and obtain the set of down-sampled masks:

$$\mathbf{M}' = \Big\{ \Phi_\alpha(\mathbf{m}_i) \ \Big| \ i = 1, \ldots, N \Big\} \subset \{0,1\}^V. \tag{3}$$

More information on the operator Φ_α in Appendix 5.1.1.

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*[2]

$$\begin{aligned} \mathcal{J}_i &= \big\{ j \in \{1,\dots,V\} \mid (\mathbf{m}'_i)_j = 1 \big\}, \\ t_i &= |\mathcal{J}_i| \;=\; \|\mathbf{m}'_i\|_0. \end{aligned}$$

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

$$\boxed{\mathbf{Y}_i = P_i\,\mathbf{X}' \;\in\; \mathbb{R}^{t_i \times F}} \qquad (i = 1,\dots,N). \tag{4}$$

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 $\mathbf{Y}_i \in \mathbb{R}^{t_i \times F}$ $(i = 1,\dots,N)$ derived in previous section, we can denote their *row-wise* concatenation ($\big\|$) by

$$\mathbf{Y} = \big\|_{i=1}^{N} \mathbf{Y}_i = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix} \;\in\; \mathbb{R}^{T \times F}, \tag{5}$$

with $T = \sum_{i=1}^{N} t_i$. 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 $\mathbf{Y}$

$$\boxed{\textbf{\textit{OGVT}} \;:=\; m_\gamma(\mathbf{Y}) \;\in\; \mathbb{R}^{T \times D}} \tag{6}$$

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:

$$\mathbf{T}'_a \;=\; g_\phi\big([\textbf{\textit{OGVT}}; \mathbf{TT}]\big) \tag{7}$$

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

Effect of Overlapping and Token Duplication After downsampling and masking, you concatenate the selected encoder rows, so a ViT bin that lies in two masks (e.g., "shirt" ∪ "person") appears twice in $\mathbf{Y}_i = \mathbf{Y}_1 \ldots \mathbf{Y}_N$ before projection by the MLP. Hence two rows fed to the LLM may have the same content vector but occupy different positions in the visual sequence (because they live in different mask blocks). Now, let a query token i attend to a set S of m "duplicate" visual tokens that originated from the same ViT bin. For one attention head (dimension d):

$$\text{logit } \ell_{ij} = \frac{Q_i K_j^\top}{\sqrt{d}}.$$

If keys are identical (no positional effect), then $\ell_{ij} = \ell$ for all $j \in S$. The group's softmax numerator is

$$\sum_{j \in S} e^{\ell} = m e^{\ell} \quad \Rightarrow \quad \log\left(\sum_{j \in S} e^{\ell}\right) = \ell + \log m.$$

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

So duplicating an identical key is equivalent to adding a $\log m$ bias to that group's logit mass. However, here positional difference matters because Llama3.1 uses Rotary Position Embeddings (RoPE) inside attention. As a result, keys (and queries) are rotated by their positions, resulting in

$$Q_i = R(\theta_i)\,\tilde{Q}_i, \qquad K_j = R(\theta_j)\,\tilde{K},$$

where $R(\cdot)$ is block-diagonal with 2D rotations at many angular frequencies. The logit becomes

$$\ell_{ij} = \frac{\tilde{Q}_i^\top R(\theta_i)^\top R(\theta_j)\tilde{K}}{\sqrt{d}} = \frac{\tilde{Q}_i^\top R(\theta_j - \theta_i)\tilde{K}}{\sqrt{d}}.$$

Even though the content $\tilde{K}$ is the same, the relative phase

$$\Delta_{ij} = \theta_j - \theta_i$$

generally differs across the m copies, so their logits $\{\ell_{ij}\}_{j\in S}$ are not equal.

5.3 Experiments

5.3.1 Training Setup

Following the training setup of LLaVA1.5 [18], we employ the deepspeed library [35] with ZeRO-3 [36] 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 [37, 38] 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 [22] layers in SFT, which we unfreeze only during SFT with a LoRA rank of 128 and a LoRA alpha of 256. Table 2 summarizes the training setup used in our experiments.

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

5.3.2 Benchmark Examples

5.3.3 Comparison to SIT

In a recent study, the authors of [34] propose a method for Subobject-level Image Tokenization (SIT). While their main contributions revolve around different areas, among the many experiments, they show how to leverage segmentation masks to create a more efficient visual tokenization process within MLLMs. For a more detailed process description [34]. 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.

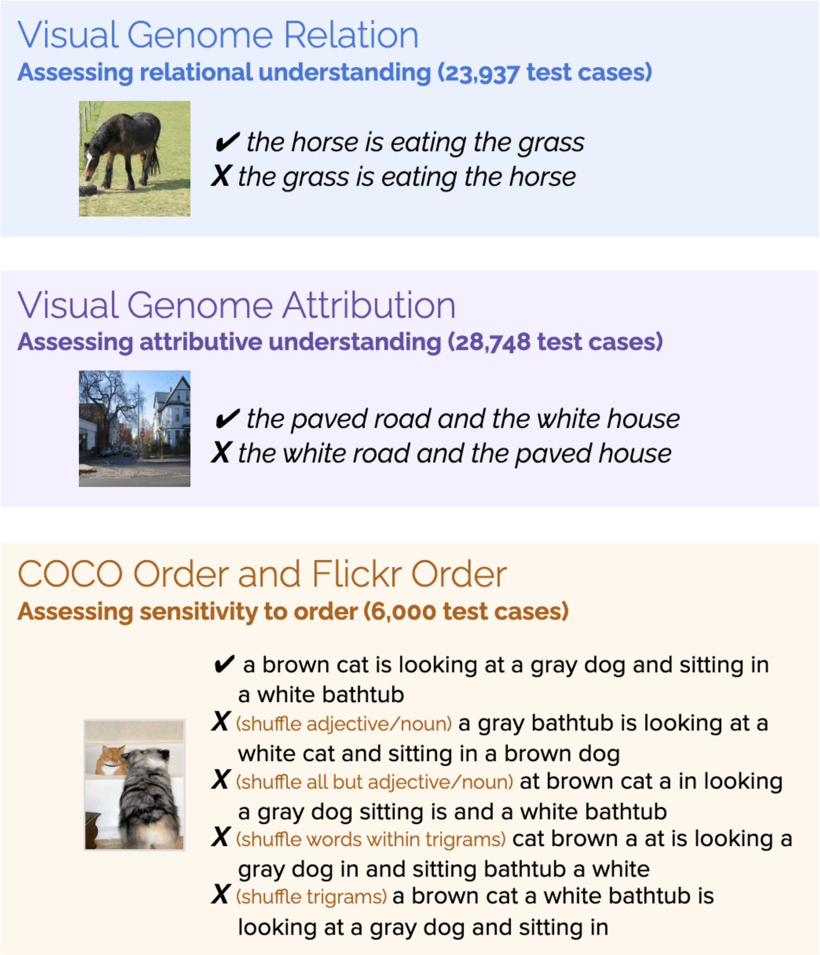

Visual Genome Relation
Assessing relational understanding (23,937 test cases)
✔ the horse is eating the grass
X the grass is eating the horse
Visual Genome Attribution
Assessing attributive understanding (28,748 test cases)
✔ the paved road and the white house
X the white road and the paved house
COCO Order and Flickr Order
Assessing sensitivity to order (6,000 test cases)
✔ a brown cat is looking at a gray dog and sitting in a white bathtub
X (shuffle adjective/noun) a gray bathtub is looking at a white cat and sitting in a brown dog
X (shuffle all but adjective/noun) at brown cat a in looking a gray dog sitting is and a white bathtub
X (shuffle words within trigrams) cat brown a at is looking a gray dog in and sitting bathtub a white
X (shuffle trigrams) a brown cat a white bathtub is looking at a gray dog and sitting in

Figure 4: **ARO benchmark examples**: The picture shows the four different subsets of ARO [7], Visual Genome Relation, Visual Genome Attribution, COCO Order and Flickr Order.

Comparison with our OG-LLaVA In Figure 9 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.

5.4 Ablation Studies

In this section we present ablation studies to disentangle the impact of key design choices in our framework. We first test the visual masking scheme, asking whether the default `OG-Fusion` setup provides enough context or if exposing more image content improves results. We then examine the role of explicit structure in the visual prompt by introducing specialized tokens to convey finer-grained spatial information to the LLM.

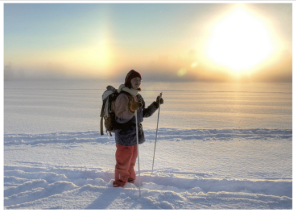
Image	Baseline	Pipeline
	Original positive: Two women are squatting down and petting brown and white long haired goats. Original negative: Two women are squatting down and petting brown and black long haired goats.	Question: How many animals are visible in the immediate group around the person in the 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 5: **ConME vs SugarCrepe qualitative example**: In the center the original SugarCrepe [27] prompts, on the right the more accurate ConMe [1] questions

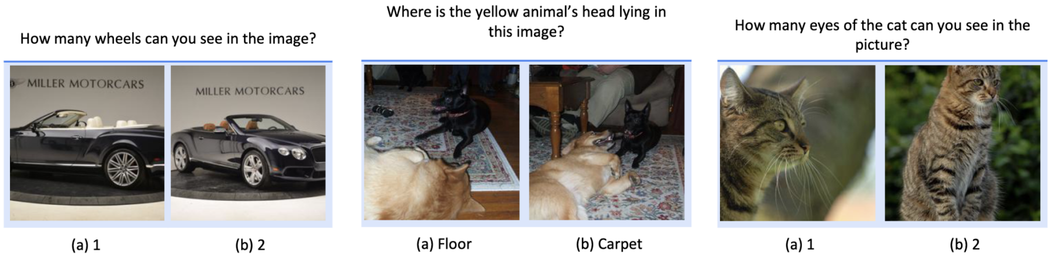

How many wheels can you see in the image?
MILLER MOTORCARS
MILLER MOTORCARS
(a) 1
(b) 2
Where is the yellow animal's head lying in this image?
(a) Floor
(b) Carpet
How many eyes of the cat can you see in the picture?
(a) 1
(b) 2

Figure 6: **MMVP benchmark examples**: Showing three qualitative examples from the MMVP [2] benchmark.

5.4.1 Masking Approach with Global View

We analyze how different masking strategies affect information flow in `OG-Fusion` (Section 2.1). In the default setup, down-sampled masks ($\mathbf{M}'$) are applied one by one onto the vision encoder output ($\mathbf{X}'$). The resulting features are concatenated and projected to the LLM input space (Eq. 1), yielding ***OGVT***. While this produces object-centric tokens, it may overlook holistic scene context, since the pipeline isolates regions independently. To test whether a global embedding improves relational understanding across masks, we add back the full-image representation $\mathbf{X}'$, concatenated before the masked features, changing Eq. 1 into:

$$\boldsymbol{OGVT}_{gv} := m_\gamma\Big(\mathbf{X}' \oplus \big\|_{i=1}^{N} \mathbf{Y}_i\Big) \in \mathbb{R}^{(V+T)\times D}, \tag{8}$$

where $\oplus$ denotes concatenation. This increases the number of visual tokens from T to $(V+T)$. We refer to this variant as the "*Global View*" since it retains both object-level cues and the overall image representation.

Results The *right* panel of Fig. 10 shows that adding a global feature is counter-productive. The variant drops 6.4% on compositional reasoning and slightly on vision-centric (-0.8%) and general-purpose (-1.7%) benchmarks, with no compensating gains. Worse, concatenating the encoder grid

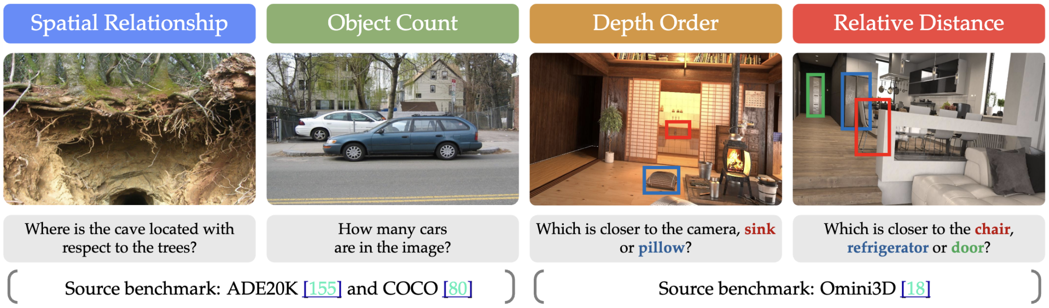

Spatial Relationship
Object Count
Depth Order
Relative Distance
Where is the cave located with respect to the trees?
How many cars are in the image?
Which is closer to the camera, sink or pillow?
Which is closer to the chair, refrigerator or door?
Source benchmark: ADE20K [155] and COCO [80]
Source benchmark: Omini3D [18]

Figure 7: **CVBench benchmark examples**: We show the four subtasks in CVBench [10], Spatial Relationship, Object Counting, Depth Order and Relative Distance.

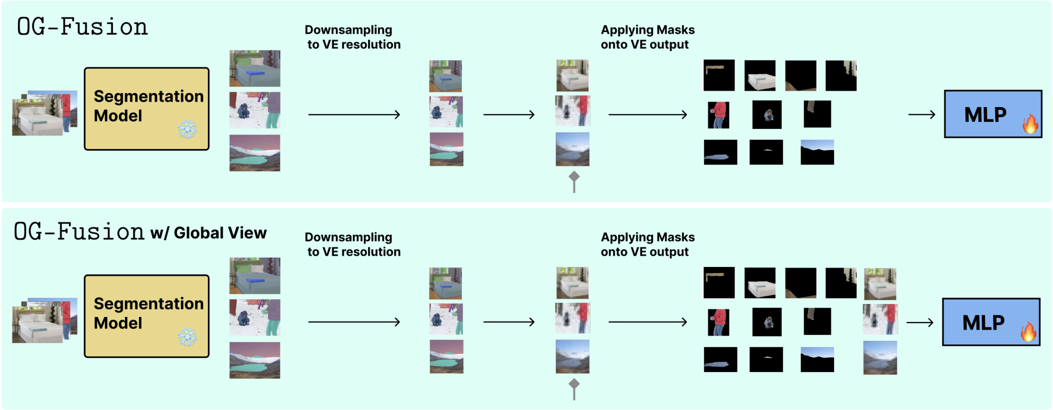

OG-Fusion
Segmentation Model
Downsampling to VE resolution
Applying Masks onto VE output
MLP
OG-Fusion w/ Global View
Segmentation Model
Downsampling to VE resolution
Applying Masks onto VE output
MLP

Figure 8: `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).

doubles the visual tokens ($T \approx V \ \rightarrow \ 2V$), and thus the quadratic self-attention cost. Given lower accuracy and higher compute, we discard the global-view extension and adopt the lean ***OGVT*** design. Why does this global view backfire? We conjecture two failure modes:

(1) Dilution in an over-long context. Doubling sequence length enlarges the attention space, forcing tokens to compete harder for relevance. With fixed heads, coverage degrades and object evidence is mixed or discarded.

(2) Lack of disambiguating priors. The global embedding is treated as an ordinary token. Without a distinct type or positional encoding, the model cannot recognize it as summarizing the full image. A special ”GLOBAL” token, similar to BERT's CLS [39], may help.

5.4.2 End-of-Mask Token

Building on these findings, we next explore a complementary strategy that makes the end of every mask explicit to the decoder. We do so by appending a dedicated End-of-Mask (EoM) token after each object mask. The EoM acts as a lightweight delimiter, giving the decoder a clear boundary between objects even when all preceding tokens share the same type embedding. We ablate `OG-Fusion` with and without this token. Compared to the standard ***OGVT***, Eq. 1 is modified as:

$$\boldsymbol{OGVT}_{\text{EoM}} := m_\gamma\left(\big\|_{i=1}^{N}\,[\mathbf{Y}_i;\mathbf{e}_i]\right) \ \in \ \mathbb{R}^{(T+N)\times D}, \tag{9}$$

where $\mathbf{e}_i \in \mathbb{R}^{1\times D}$ is the End-of-Mask token, “;” denotes concatenation, and $T = \sum_{i=1}^{N} t_i$.

Results Figure 10 (*right*) compares **OG-LLaVA** with and without the EoM token. Despite its intended structural disambiguation, the EoM variant loses significant performance in compositional reasoning (61.0 $\rightarrow$ 52.1, -10%). The extra EoM tokens elongate sequences, disrupting cross-mask attention or confusing the model. Vision-centric tasks show only minor gains ($+ \leq 3\%$), while

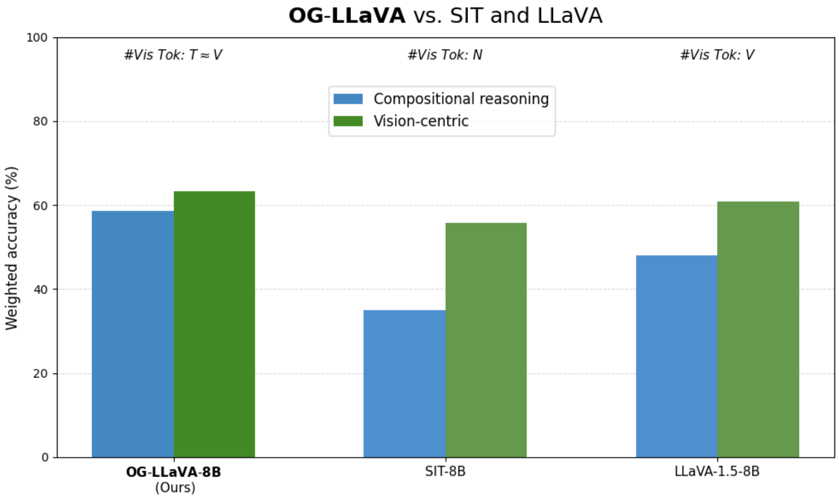


Figure 9: **Comparison between our OG-LLaVA, Subobject level Image Tokenization and LLaVA-1.5**: We report the performance of our **OG-LLaVA-8B** with `OG-Fusion` (darker bars) against the Subobject Level Image Tokenization (SIT-8B) [34] approach and LLaVA-1.5-8B. The weighted accuracies (higher is better) are reported for three macro-benchmarks, with the test-set item counts as weights.

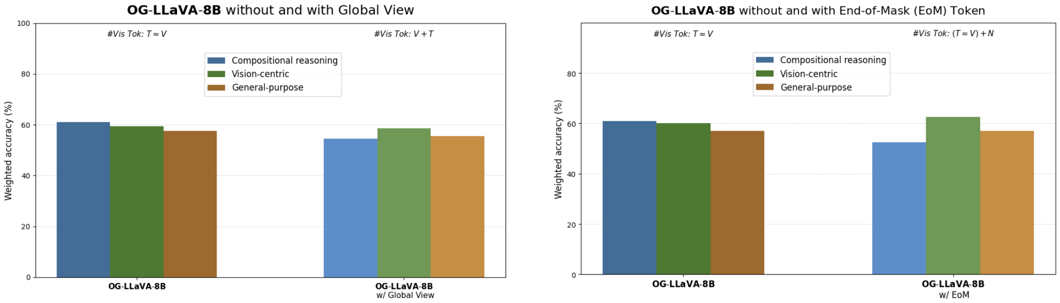


Figure 10: **Ablation results on OG-LLaVA-8B**: The *left* figure reports the performance difference without and with the Global View approach. On the *right* we show the performance of our method with and without End of Mask token. In both the weighted accuracies (higher is better) are reported for the three macro-benchmark categories, using the test-set item counts as weights.

General-purpose remains unchanged. Since the delimiter both degrades multi-object reasoning and inflates the token budget ($T \to T + N$), we adopt the lean, delimiter-free `OG-Fusion` of section 2.1 for all subsequent experiments.

5.4.3 Positional Encoding

Positional encodings (PE) are a cornerstone of Transformer-based models [40], especially in the vision domain [19, 41]. Furthermore, positional embeddings may also address the concerns raised in Section 5.4.1, 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**.

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 2.1 by assigning every mask patch i a discrete segment label $s_i \in \{0, \dots, K\}$. We then compute a deterministic vector $\text{PE}(s_i) \in \mathbb{R}^D$

whose even and odd coordinates follow the classical scheme of [40]:

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

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

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

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:

$$\mathrm{PE}(s) \;:=\; E_s, E = \big[E_0^\top E_1^\top \dots E_K^\top\big]^\top \in \mathbb{R}^{(K+1)\times D},$$

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

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

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:

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

Then we concatenate the two halves to obtain $\mathrm{PE}^{\text{2D}}(x, y) \in \mathbb{R}^{V \times D}$, 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. 1 to:

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

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) [42] within its attention mechanism, see [43]. 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

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

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:

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

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 11 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.

5.4.4 Additional Experiments

***Sliding Windows* approach** Given new visual feature encodings like Dynamic High Resolution (DHR) [11], 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

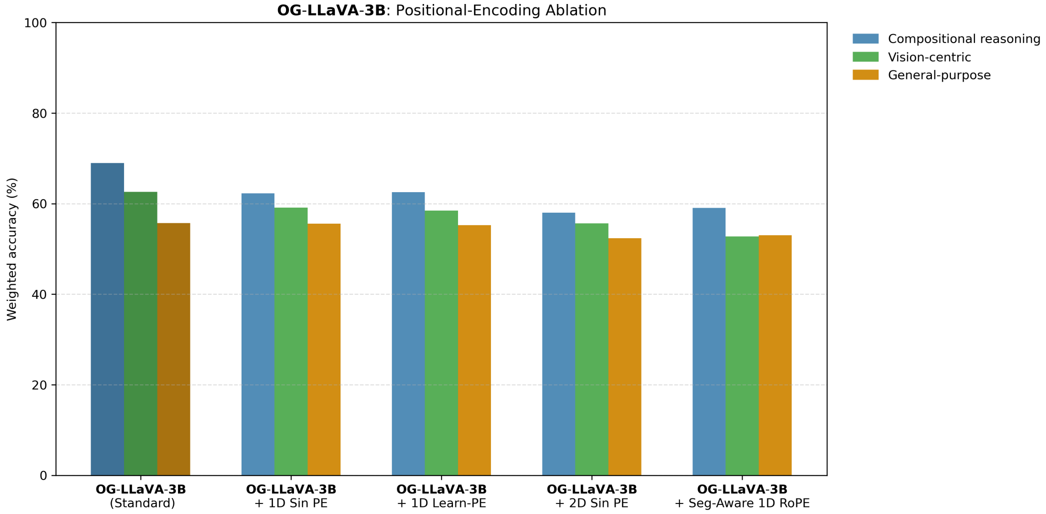


Figure 11: **Positional encoding ablation on OG-LLaVA-3B**: The weighted accuracies (higher is better) are reported for three macro-benchmarks—Compositional Reasoning, Vision-Centric, and General-Purpose—using the test-set item counts as weights. The darker bars correspond to the baseline **OG-LLaVA** 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.

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

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

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 1 becomes:

$$\boldsymbol{OGVT}_{sw} \ := \ m_\gamma \Big(\mathbf{Y}^{(q)} \Big) \ \in \ \mathbb{R}^{T_q \times D} \tag{10}$$

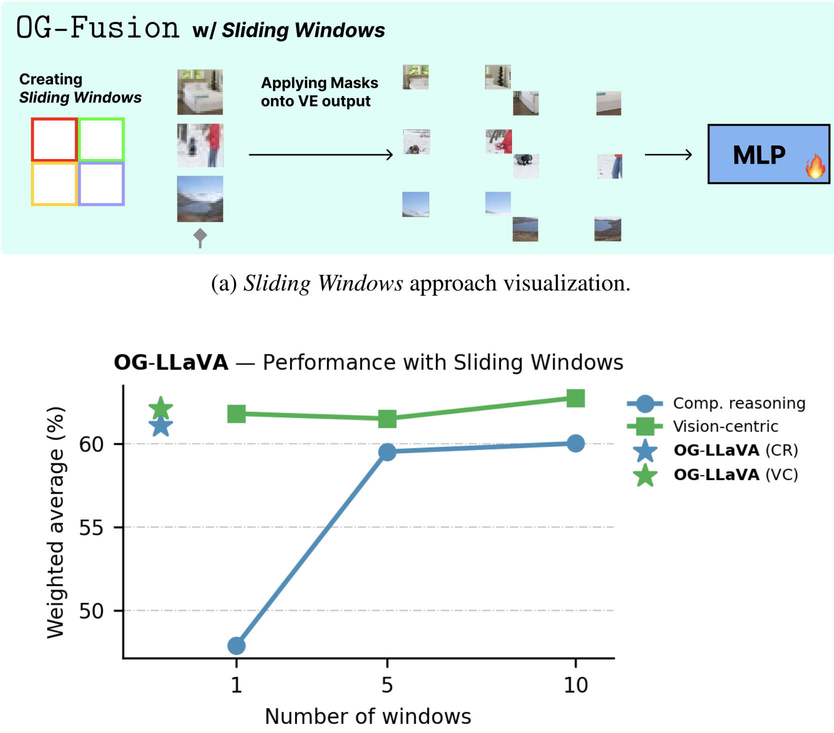
where $T_q \ = \ \sum_{q=1}^{k} t_q \ = \ V$, given each patch is not overlapping. The overall process is visualized in Figure 12a.

Results In Figure 12b, 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 5.2.1, and **OG-LLaVA**.

As seen in Figure 12b, 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 12b 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 12: **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**.

5.4.5 Comparison to Open Source Models

We compare **OG-LLaVA** against recent state-of-the-art multimodal systems, including LLaVA-Next-8B [11], Cambrian-1-8B [10], and Sa2Va-7B [16]. These baselines differ widely in vision encoders ([44, 45, 46, 12]), training scale ([5, 10]), data composition ([5]) and LM backbones ([47]) providing a broad and challenging reference point for evaluating our approach.

Model Details		Comp. Reasoning		Vision Centric		General Purpose				
		ARO	ConMe	MMVP	CVBench	AI2D	MME		MMStar	MMB
		Avg. Acc.	Acc.	Acc.	2D+3D Acc.	Acc.	Perc.	Cogn.	Acc.	Dev. Acc.
Method	**#Vis. Tok.**									
LLaVA-NeXT-8B	$V \times (K+1)$	70.5	69.4	38.7*	65.3	70.5	1562.3	307.5	41.0	70.9
Cambrian-1-8B	$>> V$	78.8	74.2	51.3*	**76.8**	73.1	1540.4	375.7	47.7	74.8
Sa2Va-7B	$>> V$	**81.2**	**79.2**	**74.0**	75.0	**81.4**	**1651.5**	**587.5**	**62.1**	**82.7**
Training data: LLaVA-1.5										
LLaVA-1.5-7B	V	31.5	57.7	33.7	60.1	53.5	1479.7	323.6	34.4	62.5
LLaVA-1.5-8B	V	33.9	61.6	32.0	65.2	60.6	1515.3	292.1	40.7	71.6
OG-LLaVA-8B (Ours)	$T(\approx V)$	56.6	65.2	37.0	63.5	60.1	1551.5	317.1	38.8	67.3

Table 3: **Open-Source comparison across Compositional Reasoning, Vision Centric and General-Purpose tasks.** Performance is reported without segmentation masks at inference time. ARO is a weighted average over its four sub-tasks. Numbers marked with (*) are taken from the corresponding studies when exact reproduction was not possible. Highest values are in **bold**; second-highest are underlined. Here $T(\approx V)$ denotes the *#Vis. Tok.* during training, with V being the inference *#Vis. Tok.* for **OG-LLaVA**. K is the number of DHR windows; $>> V$ indicates a token count much larger than V.

Results Table 3 compares **OG-LLaVA-8B** against recent open-source SOTA models. Sa2Va-7B leads overall, topping ConMe, ARO, and MMVP (74%), and also posting the best results on AI2D, MME, MMStar, and MMBench. Cambrian-1-8B consistently ranks second, excelling on CVBench (76.8%), while LLaVA-NeXT-8B improves over LLaVA-1.5 but lags behind the other SOTAs. Yet even these strong systems fall short of robust multimodal reasoning: compositional accuracies remain in the 70s, vision-centric scores rarely exceed two-thirds, and general-purpose metrics plateau below 85%.

By contrast, **OG-LLaVA** trails the SOTA range (65.2% on ConMe, 63.5% on CVBench, 60.1% on AI2D, 1551.5 on MME) but does so with far fewer visual tokens ($T \approx V$)—avoiding the token inflation of dynamic-resolution ($V \cdot (K+1)$) and multi-patch schemes ($>> V$)—and without complex encoder fusion or extra curated data. This efficiency highlights the strength of object-guided tokens and suggests that further architectural refinement and data scaling could close much of the gap to leading open-source systems.

5.5 Qualitative Analysis

Building on the quantitative gains reported in Section 3.2, we now turn to a series of visual case studies that make the advantages (and disadvantages) of **OG-LLaVA** tangible. As in Fig. 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. 13–15) 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 13a 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 13b. 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).

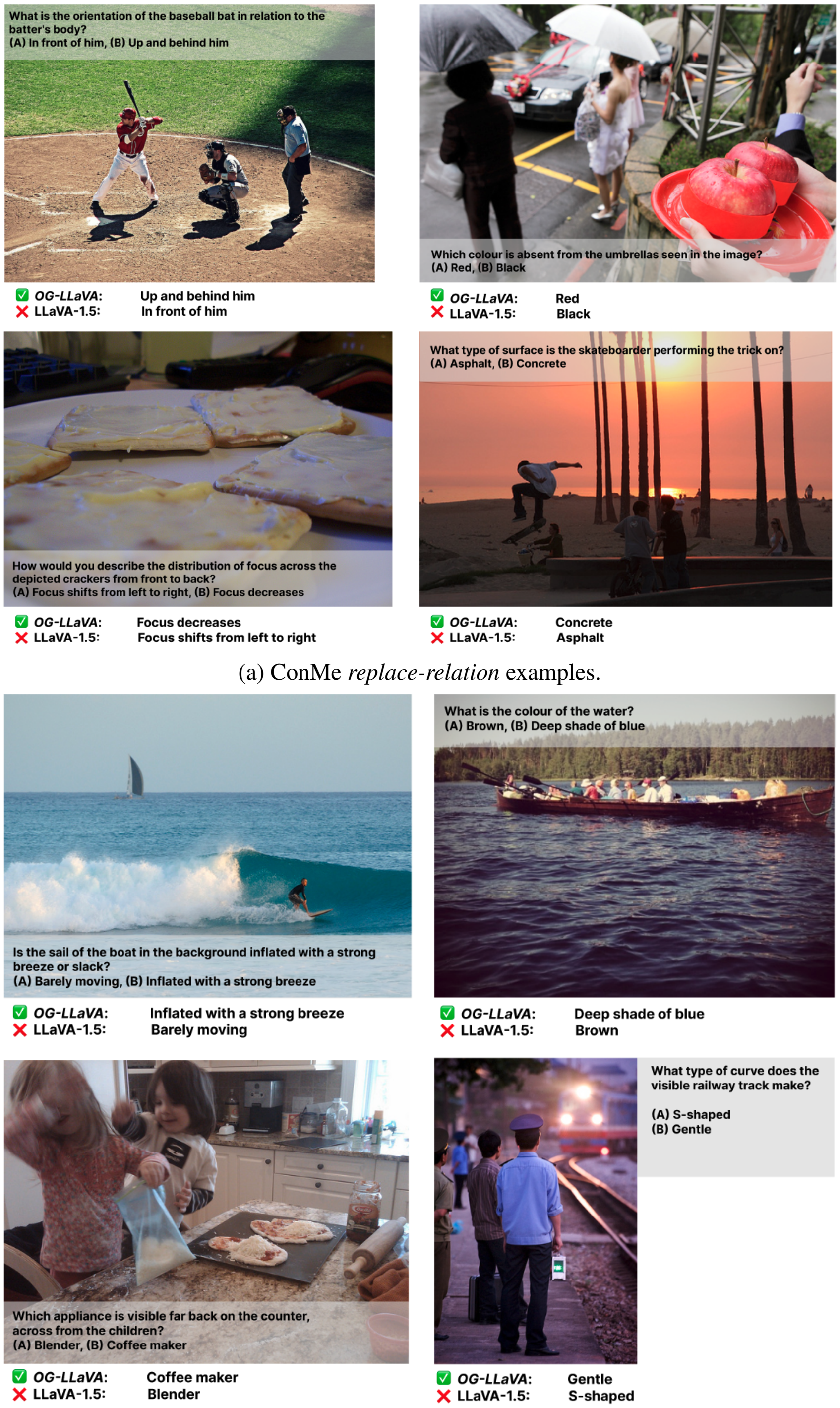
In Figure 15 we report some more examples of our strong capabilities. The model properly recognizes the material of a background shower enclosure (picture 1), while also correctly identifying font characteristics, showing surprisingly good OCR capabilities (picture 2). **OG-LLaVA** can also count better, 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 16 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.

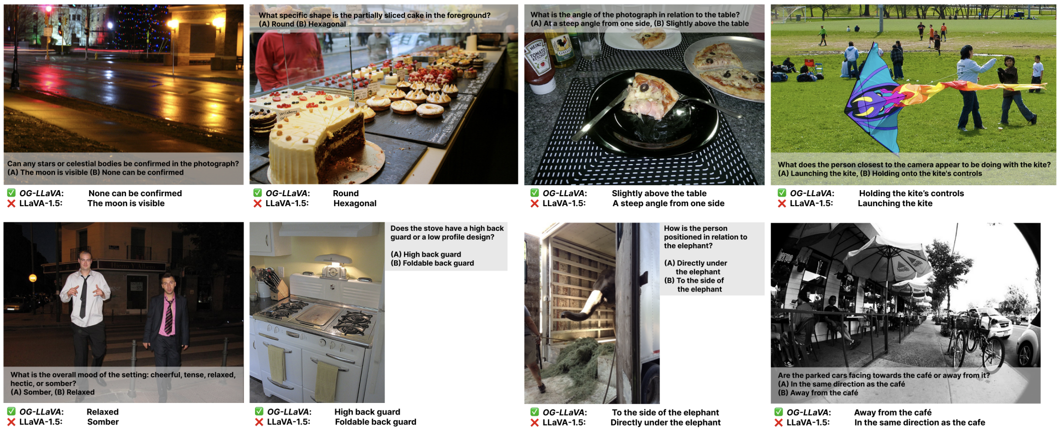
In aggregate, then, object-centric reasoning consistently amplifies every facet of visual–language understanding while leaving a small, well-defined set of shortcomings. The qualitative evidence presented here mirrors our quantitative results: **OG-LLaVA** delivers materially better performance than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


What is the orientation of the baseball bat in relation to the batter's body?
(A) In front of him, (B) Up and behind him
✓ OG-LLaVA: Up and behind him
✗ LLaVA-1.5: In front of him
Which colour is absent from the umbrellas seen in the image?
(A) Red, (B) Black
✓ OG-LLaVA: Red
✗ LLaVA-1.5: Black
How would you describe the distribution of focus across the depicted crackers from front to back?
(A) Focus shifts from left to right, (B) Focus decreases
✓ OG-LLaVA: Focus decreases
✗ LLaVA-1.5: Focus shifts from left to right
What type of surface is the skateboarder performing the trick on?
(A) Asphalt, (B) Concrete
✓ OG-LLaVA: Concrete
✗ LLaVA-1.5: Asphalt
Is the sail of the boat in the background inflated with a strong breeze or slack?
(A) Barely moving, (B) Inflated with a strong breeze
✓ OG-LLaVA: Inflated with a strong breeze
✗ LLaVA-1.5: Barely moving
What is the colour of the water?
(A) Brown, (B) Deep shade of blue
✓ OG-LLaVA: Deep shade of blue
✗ LLaVA-1.5: Brown
Which appliance is visible far back on the counter, across from the children?
(A) Blender, (B) Coffee maker
✓ OG-LLaVA: Coffee maker
✗ LLaVA-1.5: Blender
What type of curve does the visible railway track make?
(A) S-shaped
(B) Gentle
✓ OG-LLaVA: Gentle
✗ LLaVA-1.5: S-shaped

(a) ConMe *replace-relation* examples.

(b) ConMe *replace-relation* examples

Figure 13: **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 [1] 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.



Can any stars or celestial bodies be confirmed in the photograph?
(A) The moon is visible (B) None can be confirmed

✅ *OG-LLaVA*: None can be confirmed
❌ LLaVA-1.5: The moon is visible

What specific shape is the partially sliced cake in the foreground?
(A) Round (B) Hexagonal

✅ *OG-LLaVA*: Round
❌ LLaVA-1.5: Hexagonal

What is the angle of the photograph in relation to the table?
(A) At a steep angle from one side, (B) Slightly above the table

✅ *OG-LLaVA*: Slightly above the table
❌ LLaVA-1.5: A steep angle from one side

What does the person closest to the camera appear to be doing with the kite?
(A) Launching the kite, (B) Holding onto the kite's controls

✅ *OG-LLaVA*: Holding the kite's controls
❌ LLaVA-1.5: Launching the kite

What is the overall mood of the setting: cheerful, tense, relaxed, hectic, or somber?
(A) Somber, (B) Relaxed

✅ *OG-LLaVA*: Relaxed
❌ LLaVA-1.5: Somber

Does the stove have a high back guard or a low profile design?

(A) High back guard
(B) Foldable back guard

✅ *OG-LLaVA*: High back guard
❌ LLaVA-1.5: Foldable back guard

How is the person positioned in relation to the elephant?

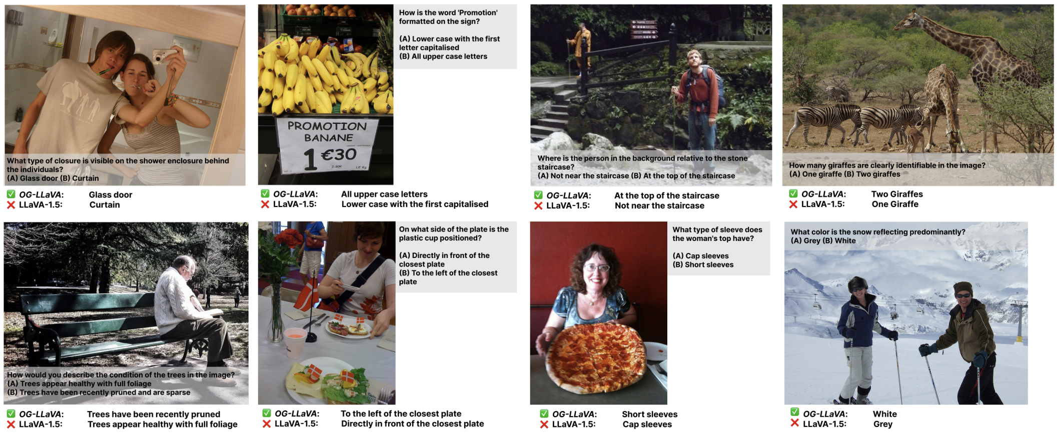
(A) Directly under the elephant
(B) To the side of the elephant

✅ *OG-LLaVA*: To the side of the elephant
❌ LLaVA-1.5: Directly under the elephant

Are the parked cars facing towards the café or away from it?
(A) In the same direction as the café
(B) Away from the café

✅ *OG-LLaVA*: Away from the café
❌ LLaVA-1.5: In the same direction as the cafe

Figure 14: **ConMe *replace-object* OG-LLaVA vs LLaVA-1.5**. We report two eight pictures on the *replace-object* sub-task of the ConMe [1] 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.



What type of closure is visible on the shower enclosure behind the individuals?
(A) Glass door (B) Curtain

✅ *OG-LLaVA*: Glass door
❌ LLaVA-1.5: Curtain

How is the word 'Promotion' formatted on the sign?

(A) Lower case with the first letter capitalised
(B) All upper case letters

✅ *OG-LLaVA*: All upper case letters
❌ LLaVA-1.5: Lower case with the first capitalised

Where is the person in the background relative to the stone staircase?
(A) Not near the staircase (B) At the top of the staircase

✅ *OG-LLaVA*: At the top of the staircase
❌ LLaVA-1.5: Not near the staircase

How many giraffes are clearly identifiable in the image?
(A) One giraffe (B) Two giraffes

✅ *OG-LLaVA*: Two Giraffes
❌ LLaVA-1.5: One Giraffe

How would you describe the condition of the trees in the image?
(A) Trees appear healthy with full foliage
(B) Trees have been recently pruned and are sparse

✅ *OG-LLaVA*: Trees have been recently pruned
❌ LLaVA-1.5: Trees appear healthy with full foliage

On what side of the plate is the plastic cup positioned?

(A) Directly in front of the closest plate
(B) To the left of the closest plate

✅ *OG-LLaVA*: To the left of the closest plate
❌ LLaVA-1.5: Directly in front of the closest plate

What type of sleeve does the woman's top have?

(A) Cap sleeves
(B) Short sleeves

✅ *OG-LLaVA*: Short sleeves
❌ LLaVA-1.5: Cap sleeves

What color is the snow reflecting predominantly?
(A) Grey (B) White

✅ *OG-LLaVA*: White
❌ LLaVA-1.5: Grey

Figure 15: **ConMe *replace-object* OG-LLaVA vs LLaVA-1.5**. We report two eight pictures on the *replace-object* sub-task of the ConMe [1] 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.

5.6 Conclusions

5.6.1 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.

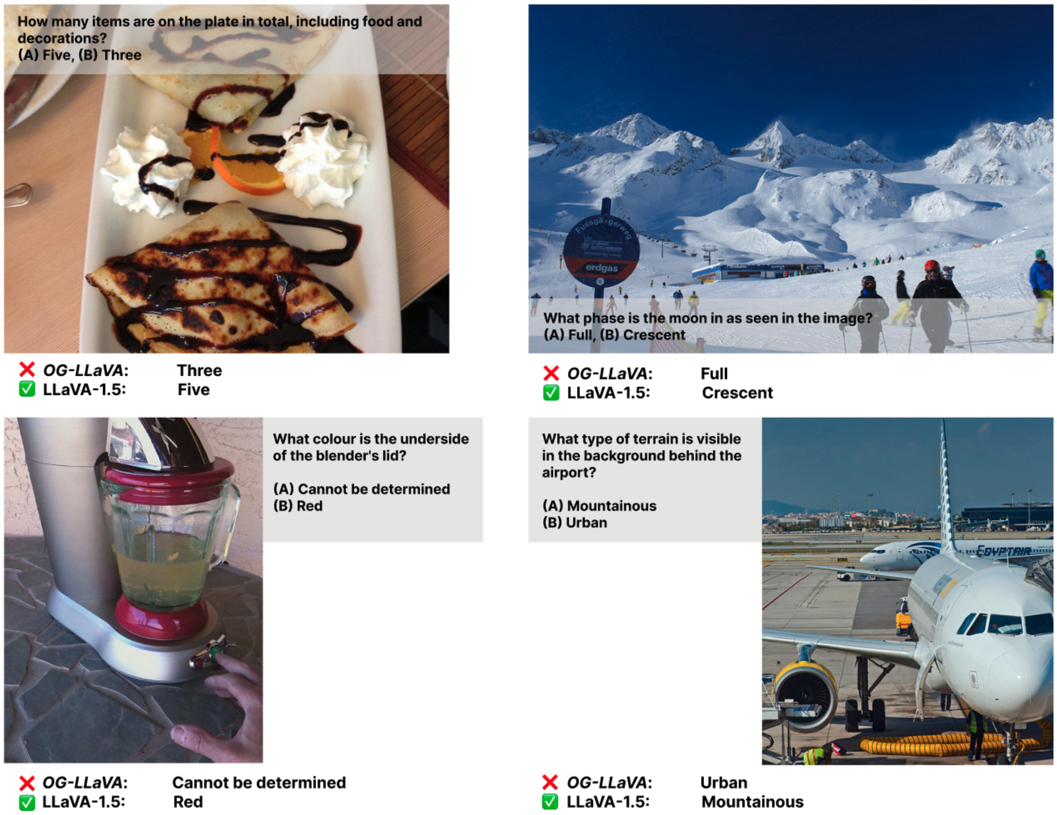

How many items are on the plate in total, including food and decorations?
(A) Five, (B) Three
✗ OG-LLaVA: Three
✅ LLaVA-1.5: Five
What phase is the moon in as seen in the image?
(A) Full, (B) Crescent
✗ OG-LLaVA: Full
✅ LLaVA-1.5: Crescent
What colour is the underside of the blender's lid?
(A) Cannot be determined
(B) Red
✗ OG-LLaVA: Cannot be determined
✅ LLaVA-1.5: Red
What type of terrain is visible in the background behind the airport?
(A) Mountainous
(B) Urban
✗ OG-LLaVA: Urban
✅ LLaVA-1.5: Mountainous

Figure 16: **ConMe *replace-relation* OG-LLaVA vs LLaVA-1.5**. We report four pictures on the *replace-relation* sub-task of the ConMe [1] 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.

5.7 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 5.4.2 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 5.4.1.
- **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 [41] 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 [45] or InternViT [5] 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) [12]. Preliminary evidence hints that mask purity and class granularity affect downstream grounding; ablating the segmentation source—e.g., OMG-Seg [48], SEEM [49], 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-7M 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.

5.8 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 15. 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.