LOOK CLOSER, SEGMENT BETTER: QUERY-GUIDED VISUAL TOKEN UNFOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) demonstrate strong vision–language understanding capabilities, yet precise pixel-level segmentation remains challenging. Existing methods primarily focus on mask representation and generation, while the visual tokens fed into the large language model are typically spatially compressed, weakening local details and limiting segmentation accuracy for complex objects. To address this issue, we propose query-guided ***V***isual ***T***oken ***U***nfolding (***VTU***), which enables fine-grained visual tokens to directly perform in vision–language interaction before mask prediction. ***VTU*** first processes coarse-grained visual tokens to achieve a global understanding of the image and the text query, and employs a lightweight router to select tokens for unfolding. The corresponding tokens are inserted before the segmentation token, allowing the segmentation representation to integrate global context and local details. We further introduce router supervision targeting mixed-region and a simple-sample skipping strategy to incorporate informative tokens while reducing unnecessary unfolding. Extensive experiments demonstrate that ***VTU*** achieves state-of-the-art performance across multiple segmentation tasks and different model scales with only a small number of additional visual tokens, while preserving general visual understanding capabilities. Anonymous link: https://anonymous.4open.science/r/VTU.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) integrate visual perception with the language understanding and reasoning capabilities of Large Language Models (LLMs), achieving substantial progress in image captioning, visual question answering, and instruction following (Liu et al., 2024; Chen et al., 2024c; Shi et al., 2025). As research advances from image-level understanding toward fine-grained visual interaction, recent studies have explored the application of MLLMs to language-guided image segmentation (Lan et al., 2025; Wang et al., 2025c; Qian et al., 2025). However, MLLMs were originally designed primarily for text generation, and extending their capabilities to dense pixel-level prediction with precise spatial structure remains a significant challenge.

Recent MLLM-based segmentation methods generally follow two paradigms. As shown in Figure 1(a), some works encode masks as structured sequences (*e.g.*, boundary coordinate, mask tokens) and learn to generate them through *next-token-prediction*, naturally integrating segmentation into the language modeling objective (Pramanick et al., 2024; Wang et al., 2025b; Zhou et al., 2026; Liu et al., 2026a). Other methods introduce special segmentation tokens and use their hidden states to predict masks through external mask decoders or the inherent visual features of the MLLM (Lai et al., 2024; Qian et al., 2025; Tang et al., 2025; Zhang et al., 2023; 2026). While these methods primarily focus on mask representation and generation, the exploration of visual features fed into LLMs to support precise segmentation remains insufficient. Prior studies indicate that MLLMs still fail to fully exploit fine-grained and multi-level information within visual encoders, and their visual representations suffer from limited spatial precision in dense prediction tasks (Liu et al., 2026b; Wang et al., 2025a; Yao et al., 2024; Cao et al., 2026). To reduce the computational overhead, MLLMs compress the spatial resolution of visual features extracted by the visual encoder through pixel shuffle or feature projection, thereby decreasing the number of visual tokens fed into the LLM (Chen et al., 2024b; Wang et al., 2024). Although such coarse-grained tokens might support the segmen-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

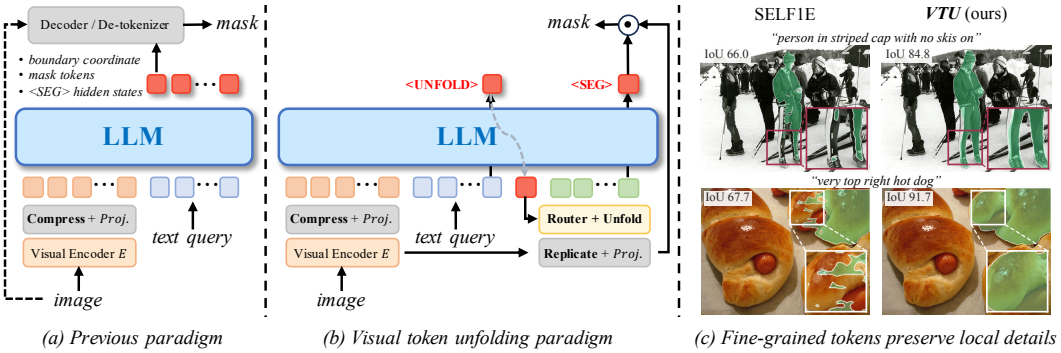


Figure 1: (a) Existing MLLM-based segmentation with coarse-grained visual tokens. (b) Our method unfolds fine-grained visual tokens and introduces them into the LLM before mask generation. (c) Incorporating these fine-grained tokens improves the segmentation of local details, producing more complete masks as illustrated in the zoomed-in regions.

tation of simple objects, the degradation of local details often hinders precise boundary delineation when dealing with complex contours and regions with mixed foreground and background.

A closely related work is SELF1E (Zhang et al., 2026), which preserves and enhances uncompressed fine-grained tokens from the visual encoder to strengthen the model's perceptual capability, and then generates masks by computing dot products between these features and the hidden representation of a special segmentation token. However, the visual representations fed into the LLM remain spatially compressed coarse-grained visual tokens. As shown in Figure 1(c), the resulting segmentation tokens inevitably suffer from insufficient perception of local details, which limits the model's capacity to perform precise segmentation on complex regions.

This motivates the following question: ***Can the LLM selectively access fine-grained visual evidence conditioned on the query before predicting a mask?*** To this end, we propose query-guided ***V***isual ***T***oken ***U***nfolding (***VTU***), as illustrated in Figure 1(b). Specifically, we first allow MLLMs to establish a global semantic understanding based on coarse-grained visual tokens and the text query, and to generate a newly designed special token, namely the unfolding token `<UNFOLD>`. We then design a lightweight router module that uses the hidden state of `<UNFOLD>` as a query to compute relevance scores for all coarse-grained visual tokens, and applies threshold filtering to obtain a set of candidate tokens. We further apply top-K selection to filter the final tokens to unfold and control the number of additional visual tokens. Next, we retrieve the fine-grained visual tokens corresponding to these selected tokens and insert them after `<UNFOLD>`. The model then generates the segmentation token `<SEG>`, allowing its hidden representation to incorporate local details from the selected regions for mask prediction. This mechanism preserves global context while enabling the model to adaptively select and utilize fine-grained visual evidence based on the text query. Extensive experimental results demonstrate that our method achieves consistent performance improvements across multiple segmentation tasks by introducing only a small number of additional visual tokens, with negligible additional inference latency.

The contributions of our work are summarized as follows:

- We identify visual feature utilization limitations in MLLM-based segmentation methods, showing that relying only on compressed coarse-grained visual representations constrains complex-object segmentation.
- We propose ***VTU***, a query-guided visual token unfolding mechanism that uses a lightweight router to select relevant coarse-grained visual tokens and injects their corresponding fine-grained features before generating the segmentation token, enabling local visual details to inform mask prediction while preserving global context.
- Extensive experiments demonstrate that our method achieves state-of-the-art (*SOTA*) performance across multiple segmentation tasks and model scales with only a small number of additional visual tokens and negligible additional inference latency, while preserving general image understanding capabilities.

2 Related Work

MLLM-based Image Segmentation. Recent studies have extended MLLMs to image segmentation, drawing on their language understanding and multimodal reasoning capabilities for pixel-level prediction (Yuan et al., 2026). Several works represent *sequence-as-mask* paradigm. For example, VisionLLM (Wang et al., 2023) and VistaLLM (Pramanick et al., 2024) represent object contours as coordinate sequences, Text4Seg (Lan et al., 2025) converts masks into textual semantic descriptions, and HiMTok (Wang et al., 2025c) and ALTo (Wang et al., 2025b) employ hierarchical and adaptive-length discrete mask tokens, respectively. Other works introduce special segmentation tokens. The pioneering work LISA (Lai et al., 2024) proposes the *embedding-as-mask* paradigm, using the hidden representation of <SEG> to prompt SAM (Kirillov et al., 2023) to generate the target mask. Subsequent studies extend this paradigm, PixelLM (Ren et al., 2024) supports multi-object segmentation through multi-granularity segmentation tokens and a lightweight pixel decoder, while PSALM (Zhang et al., 2024b) accommodates diverse segmentation tasks through a unified input schema and mask queries. However, these studies primarily focus on mask representation and generation, leaving the visual features that are fed into LLMs for precise segmentation insufficiently explored.

Visual Representation in MLLM. Some MLLMs compress encoded visual features before feeding them into the LLM, thereby reducing visual sequence length and computational cost. These coarse-grained representations perform well on tasks that rely on global semantic information, such as image captioning, but may not preserve sufficient spatial detail for accurate segmentation when applied to mask prediction. For example, VersaViT (Liu et al., 2026b) demonstrates limitations of the MLLM vision encoders on semantic segmentation and depth estimation, while Dense Connector (Yao et al., 2024) and MMFuser (Cao et al., 2026) highlight that relying solely on high-level features of the vision encoder fails to fully exploit the rich visual information in other layers. MLLMSeg (Wang et al., 2025a) fuses detailed features from the vision encoder with semantic features produced by the LLM, whereas SELF1E (Zhang et al., 2026) preserves high-resolution features and incorporates LLM residual features to mitigate the effects of spatial compression on mask prediction. Nevertheless, they overlook that the visual features input to the LLM undergo compression, which inevitably degrades the capacity of segmentation representations for local-region segmentation. In contrast, our work focuses on the enhancement of visual information fed into the LLM. By selecting coarse-grained visual tokens and retrieving their corresponding fine-grained representations, we allow these representations to directly engage in subsequent vision-language interactions and furnish fine-grained visual evidence for the construction of segmentation representations.

3 Method

We propose ***VTU***, which enables MLLMs to selectively introduce fine-grained visual features for image–language interaction. We first introduce the construction of coarse- and fine-grained visual representations (§3.1). Subsequently, we elaborate on global visual understanding and token routing mechanism (§3.2), followed by fine-grained visual token unfolding for mask prediction (§3.3). The overview of our method is illustrated in Figure 2. Finally, we introduce the construction of supervision labels for the router and the joint optimization objective (§3.4).

3.1 Preliminary

Most MLLMs consist of a visual encoder E, a visual projector P, and an LLM. To reduce the computational overhead, MLLMs typically compress visual features spatially before feeding them into the LLM. Specifically, given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the visual encoder E first encodes it into visual features $F \in \mathbb{R}^{H_f\times W_f\times d_v}$, where H_f and W_f denote the spatial dimensions of the visual features, and d_v is the embedding dimension.

Subsequently, a spatial compression operation (*e.g.*, pixel shuffle (Chen et al., 2024b)) concatenates the visual features F from $r \times r$ neighboring locations along the channel dimension. This operation reduces the spatial dimensions to $(H_f/r)\times(W_f/r)$ while increasing the channel dimension to $r^2 d_v$:

$$F \in \mathbb{R}^{H_f\times W_f\times d_v} \rightarrow F_c \in \mathbb{R}^{(H_f/r)\times(W_f/r)\times r^2 d_v}, \tag{1}$$

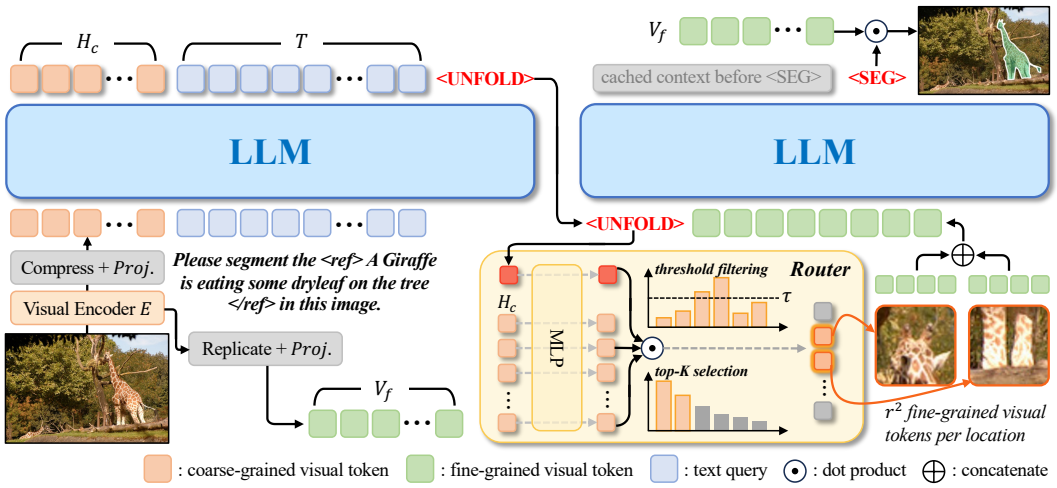


Figure 2: Overview of our ***VTU***. The MLLM processes coarse-grained visual tokens and the text query to generate <UNFOLD>. Next, a lightweight router selects coarse-grained visual tokens through threshold filtering and top-K selection. The corresponding fine-grained visual tokens are inserted before <SEG>, enabling the segmentation representation to integrate global context and local details for mask prediction.

where r represents the spatial downsampling factor and F_c denotes the compressed visual features. The projector P then maps F_c to the semantic space of the LLM:

$$V_c = P(F_c) \in \mathbb{R}^{N_c \times d}, \quad N_c = \frac{H_f W_f}{r^2}, \tag{2}$$

where N_c indicates the number of coarse-grained visual tokens. To mitigate the limitations caused by the reduction in spatial resolution, we follow SELF1E to preserve the original visual features F produced by the visual encoder E. First, the visual feature at each spatial location is replicated r^2 times along the channel dimension, expanding the embedding dimension from d_v to $r^2 d_v$:

$$F \in \mathbb{R}^{H_f \times W_f \times d_v} \rightarrow F_r \in \mathbb{R}^{H_f \times W_f \times r^2 d_v}, \tag{3}$$

where F_r denotes the replicated visual features. We then apply the same projector P to obtain the fine-grained visual tokens:

$$V_f = P(F_r) \in \mathbb{R}^{N_f \times d}, \quad N_f = H_f W_f = r^2 \times N_c. \tag{4}$$

With this construction, each coarse-grained visual token corresponds to r^2 fine-grained visual tokens. It is worth noting that we reuse the pretrained visual projector P to map the fine-grained visual features into the semantic space of the LLM. This enables these features to participate in the subsequent image–language interaction without requiring an additional adapter.

3.2 Visual Token Routing

In prior MLLM-based segmentation methods, the LLM receives spatially compressed visual features and a text query, and then directly generates special segmentation tokens or structured sequences. However, spatial compression may obscure local details, making it difficult for the LLM to segment complex objects accurately. To alleviate this issue, we introduce a special token <UNFOLD>, along with a lightweight router that enables the LLM to adaptively select the required fine-grained visual tokens for generating segmentation tokens.

Specifically, the LLM generates the <UNFOLD> token after processing the coarse-grained tokens V_c and the text query T. We denote the final-layer hidden states at the coarse-grained visual positions as $H_c = [h_1, \ldots, h_{N_c}]^\top \in \mathbb{R}^{N_c \times d}$, and the hidden state corresponding to <UNFOLD> as $h_u \in \mathbb{R}^d$. We introduce two MLPs, MLP_q and MLP_k, to project the normalized hidden states into query and key representations, respectively. The router then computes an unfolding score for each position:

$$q = \mathrm{MLP}_q(\mathrm{LN}_q(h_u)), \quad k_i = \mathrm{MLP}_k(\mathrm{LN}_k(h_i)), \quad p_i = \sigma\left(\frac{q^\top k_i}{\sqrt{d}}\right), \tag{5}$$

where LN denotes layer normalization, and σ represents the sigmoid function. We adopt threshold filtering to select highly relevant candidate tokens. To control the number of additional visual tokens, we further apply top-K selection on the filtered visual tokens and obtain the set of position indices $\mathcal{S}$ for the coarse-grained visual tokens selected for unfolding:

$$\mathcal{S} = \left\{ i \in \mathrm{TopK}\Big(\{p_j\}_{j=1}^{N_c}, K \Big) \;\Big|\; p_i \geq \tau \right\}. \tag{6}$$

where K represents the maximum number of visual tokens selected for unfolding, and τ denotes the filtering threshold. The router then retrieves the fine-grained visual tokens corresponding to $\mathcal{S}$ and incorporates them into subsequent segmentation representation generation.

3.3 Visual Token Unfolding and Mask Prediction

Unlike prior methods, we insert the fine-grained visual tokens corresponding to $\mathcal{S}$ before generating the special segmentation token <SEG>. This enables the LLM to incorporate both global context and local details. Specifically, given the set of position indices $\mathcal{S}$, we exploit the spatial correspondence between the coarse- and fine-grained visual features to extract the corresponding visual tokens from V_f. Let $\Omega(i)$ denote the $r \times r$ fine-grained spatial locations associated with the i-th coarse-grained location, and let v_j^f represent the j-th token in V_f. The sequence of unfolded visual tokens is concatenated as follows:

$$V_{\text{unfold}} = \mathrm{Concat}_{i \in \mathcal{S}} \left([v_j^f]_{j \in \Omega(i)} \right) \in \mathbb{R}^{L_{\text{unfold}} \times d}, \quad L_{\text{unfold}} = r^2 |\mathcal{S}| \leq r^2 K, \tag{7}$$

The selected coarse-grained tokens are sorted in descending order based on their unfolding scores, while the fine-grained tokens corresponding to each coarse-grained token are arranged in a row-major spatial order. Subsequently, we insert V_{unfold} after the <UNFOLD> token. The whole sequence can be formalized as follows (other identifiers in the conversation template are omitted for brevity):

$$[V_c,\ T,\ \texttt{<UNFOLD>},\ V_{\text{unfold}},\ \texttt{<SEG>}], \tag{8}$$

The LLM processes the sequence to generate the <SEG> token for mask prediction. We denote the hidden feature of this token as $h_{\text{seg}} \in \mathbb{R}^d$. Next, we rearrange the fine-grained visual features V_f into a spatial grid format and compute the mask logits M via a scaled dot product with h_{seg}. Finally, we upsample M to the original image resolution to obtain the pixel-level mask:

$$M = \mathrm{Reshape}\left(\frac{h_{\text{seg}} V_f^T}{\sqrt{d}} \right) \in \mathbb{R}^{H_f \times W_f}. \tag{9}$$

3.4 Training Objectives

Router Supervision. Our goal is to guide the router toward visual tokens that benefit segmentation. We observe that the limited spatial resolution of coarse-grained features often hinders precise segmentation in mixed-region with foreground and background. We therefore design a mixed-region supervision method that directs the router toward these regions. As illustrated in Figure 3, we partition the ground-truth mask M_{gt} according to the spatial resolution of the coarse-grained feature map and compute the foreground occupancy of each cell through average pooling, yielding an occupancy map A_{gt}. Locations satisfying $0.1 < A_{\text{gt}}(i) < 0.9$ contain substantial proportions of both foreground and background, making them difficult to capture precisely with a single coarse-grained representation. We label these locations positive and all remaining locations negative. However, applying this supervision to every sample may encourage unnecessary unfolding for simple targets that can already be segmented accurately using coarse-grained features. To address this, we propose a simple-sample skipping strategy, which upsamples occupancy map A_{gt} to the original resolution to obtain a reconstructed mask $\widetilde{M}_{\text{gt}}$ and compute its IoU score with the ground-truth mask. If $\mathrm{IoU}(\widetilde{M}_{\text{gt}}, M_{\text{gt}}) > \tau_{\text{simple}}$, all locations

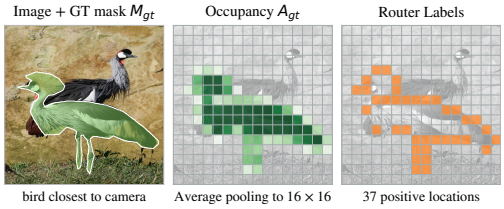


Figure 3: Construction of router labels.

in the sample are labeled negative to discourage unnecessary unfolding. The router is trained using weighted binary cross-entropy loss $\mathcal{L}_{\text{router}}$ to mitigate class imbalance. The complete router label construction procedure is provided in Appendix A.1.

We jointly optimize text generation, mask prediction, and router token selection. The autoregressive text generation loss $\mathcal{L}_{\text{text}}$ is computed using cross-entropy over text tokens and special tokens, excluding the inserted fine-grained visual tokens. The mask loss $\mathcal{L}_{\text{mask}}$ combines binary cross-entropy (BCE) and Dice losses to enforce consistency between predicted and ground-truth masks. The overall training objective is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{router}}$.

4 Experiment

4.1 Experimental Setup

Implementation Details. We build ***VTU*** upon the 2B and 8B variants of InternVL3 (Zhu et al., 2025), initialized from their corresponding pretrained weights, while the router is randomly initialized. We set the top-K selection to $K = 8$, the filtering threshold to $\tau = 0.5$, and the simple-sample threshold to $\tau_{\text{simple}} = 0.9$. InternVL3 uses a spatial downsampling factor of $r = 2$, so each selected coarse-grained token corresponds to 2×2 fine-grained tokens. We fine-tune the visual encoder and LLM using LoRA (Hu et al., 2022) and jointly train the newly introduced router. The LoRA ranks are set to 128 and 64 for the 2B and 8B models, respectively. We use AdamW (Loshchilov & Hutter, 2017) with a cosine learning rate schedule, setting the initial learning rates to $1e{-4}$ and $6e{-5}$ for the 2B and 8B models, respectively. Both models are trained for one epoch on eight NVIDIA RTX 4090 GPUs with BF16 mixed precision, gradient checkpointing, and gradient accumulation, yielding a global batch size of 128 for each model. More training details are provided in Appendix A.2.

Training Datasets and Baselines. We conduct joint training and evaluation on multiple datasets, including **RefCOCO** (Yu et al., 2016), **RefCOCO+** (Yu et al., 2016), **RefCOCOg** (Mao et al., 2016), and **RefCLEF** (Kazemzadeh et al., 2014) for referring expression segmentation, **gRefCOCO** (Liu et al., 2023a) for generalized referring expression segmentation, **ReasonSeg** (Lai et al., 2024) for reasoning segmentation, and **ADE20K** (Zhou et al., 2017), **COCOStuff** (Caesar et al., 2018), **PASCAL Context (PC-59)** (Mottaghi et al., 2014), and **PASCAL VOC (PAS-20)** (Everingham et al., 2010) for semantic segmentation. We additionally incorporate several VQA datasets to preserve the model's visual understanding capabilities. Given the substantial differences in dataset scales, we adjust their sampling proportions to balance contributions during training. For evaluation, we compare ***VTU*** with representative MLLM-based segmentation methods under the *sequence-as-mask* and *embedding-as-mask* paradigms. More datasets and baseline settings are provided in Appendix A.3.

4.2 Comparison with State-of-the-Arts

Referring Expression Segmentation. We conduct experiments on three classic referring segmentation benchmarks. Table 1 presents the evaluation results of ***VTU*** on RefCOCO, RefCOCO+, and RefCOCOg. Compared with other representative methods, ***VTU-8B*** achieves the best results across all evaluation splits, with an average cIoU of 85.6%. Notably, the smaller ***VTU-2B*** also achieves an average cIoU of 83.2%, outperforming all compared methods, including ALToLLM-8B and SELF1E-SEG-8B, which adopt larger MLLMs. Crucially, our method does not rely on any external mask decoder yet outperforms decoder-based methods in average cIoU. These results highlight the value of our method in improving segmentation performance by selectively incorporating local fine-grained visual information before generating the segmentation token.

Generalized Referring Expression Segmentation. We evaluate ***VTU*** on gRefCOCO, which includes expressions referring to multiple or no targets, to validate its performance in more general referring scenarios. As shown in Table 2, we conduct comparisons under both zero-shot and dataset-specific fine-tuning settings. In the zero-shot setting, ***VTU-8B*** achieves the best cIoU and gIoU across three splits, with an average score of 55.6%, surpassing the prior *SOTA* PSALM-1.3B by 6.4%. Moreover, ***VTU-2B*** also achieves competitive performance. For the fine-tuning setting, two ***VTU*** variants achieve average scores of 76.4% and 78.6%, respectively, both surpassing prior methods. This demonstrates that our method remains effective in both multiple or no target scenarios.

Table 1: Referring expression segmentation results in cIoU (%). ***Avg.*** is the mean cIoU over eight evaluation splits. Best results are shown in **bold**. "(ft)" denotes fine-tuning on the training split of the corresponding dataset. Mask Decoder indicates the use of an external decoder.

Methods	MLLM	Mask Decoder	RefCOCO			RefCOCO+			RefCOCOg		*Avg.*
			val	testA	testB	val	testA	testB	val	test	
Paradigm: sequence-as-mask											
VistaLLM-13B (CVPR'24)	EVA-CLIP+Vicuna-13B	✗	77.2	78.7	73.9	71.8	74.4	65.6	69.8	71.9	72.9
Text4Seg-13B (ICLR'25)	LLaVA1.5-13B	✗	71.3	72.9	70.3	65.9	70.0	61.8	66.8	67.6	68.3
Text4Seg-13B (ICLR'25)	LLaVA1.5-13B	✓	80.2	82.7	77.3	73.7	78.6	67.6	74.0	75.1	76.2
Text4Seg-8B (ICLR'25)	InternVL2-8B	✗	70.3	71.9	68.7	65.0	68.9	60.8	66.7	67.6	67.5
Text4Seg-8B (ICLR'25)	InternVL2-8B	✓	79.2	81.7	75.6	72.8	77.9	66.5	74.0	75.3	75.4
HiMTok-8B (ft) (ICCV'25)	InternVL2.5-8B	✓	85.9	86.3	83.9	80.5	83.7	76.4	80.1	80.9	82.2
ALToLLM-8B (NeurIPS'25)	InternVL2.5-8B	✓	85.8	86.6	84.7	81.3	83.8	77.0	80.6	81.4	82.7
STAMP-7B (CVPR'26)	Qwen2VL-7B	✗	78.1	79.2	76.8	74.7	77.6	70.9	75.7	76.2	76.2
STAMP-7B (CVPR'26)	Qwen2VL-7B	✓	83.1	84.5	80.8	79.4	82.8	74.6	79.9	80.4	80.7
Paradigm: embedding-as-mask											
LISA-7B (ft) (CVPR'24)	LLaVA1.1-7B	✓	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6	69.9
PixelLM-7B (CVPR'24)	LLaVA-7B	✓	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5	69.2
LaSagnA-7B (arXiv'24)	LLaVA-7B	✓	76.8	78.7	73.8	66.4	70.6	60.1	70.6	71.9	71.1
GSVA-13B (ft) (CVPR'24)	LLaVA-13B	✓	79.2	81.7	77.1	70.3	73.8	63.9	75.7	77.0	74.8
u-LLaVA-7B (ECAI'24)	CLIP+Vicuna1.1-7B	✓	83.0	85.1	80.5	77.1	81.7	70.6	77.1	78.0	79.1
GLaMM-7B (CVPR'24)	CLIP+Vicuna-7B	✓	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9	75.6
VisionLLM v2-7B (NeurIPS'24)	CLIP+Vicuna-7B	✓	76.6	79.3	74.3	64.5	69.8	61.5	70.7	71.2	71.0
PSALM-1.3B (ECCV'24)	Swin-B+Phi1.5-1.3B	✓	83.6	84.7	81.6	72.9	75.5	70.1	73.8	74.4	77.1
OMG-LLaVA-7B (ft) (NeurIPS'24)	OMG-Seg+InternLM2-7B	✓	78.0	80.3	74.1	69.1	73.1	63.0	72.9	72.9	72.9
HyperSeg-3B (CVPR'25)	Mipha-3B	✓	84.8	85.7	83.4	79.0	83.5	75.2	79.4	78.9	81.2
UFO-8B (ft) (NeurIPS'25)	InternVL2.5-8B	✗	81.0	82.6	78.6	77.1	80.4	72.6	76.7	77.3	78.3
SELF1E-2B (CVPR'26)	InternVL3-2B	✗	80.2	82.1	77.6	74.6	79.1	69.2	77.0	77.8	77.2
SELF1E-SEG-2B (CVPR'26)	InternVL3-2B	✗	84.3	85.4	82.3	78.9	83.5	75.1	80.4	81.0	81.4
SELF1E-8B (CVPR'26)	InternVL3-8B	✗	82.5	83.8	80.1	77.6	81.6	73.7	79.1	79.8	79.8
SELF1E-SEG-8B (CVPR'26)	InternVL3-8B	✗	84.7	86.2	83.4	80.2	84.2	77.0	82.1	82.8	82.6
VTU-2B	InternVL3-2B	✗	85.0	86.8	84.0	81.3	84.7	78.2	82.8	82.9	83.2
VTU-8B	InternVL3-8B	✗	**87.3**	**88.2**	**86.5**	**83.8**	**87.0**	**81.8**	**84.9**	**84.9**	**85.6**

Reasoning Segmentation. Reasoning Segmentation is a benchmark introduced by LISA for reasoning-based segmentation, which requires models to understand complex or implicit language instructions in the context of an image and infer the intended targets. We conduct experiments on ReasonSeg. As shown in Table 3, ***VTU-8B*** achieves the best performance on both the validation and test sets, reaching cIoU / gIoU scores of 73.0% / 70.5% and 71.0% / 69.5%, respectively. The smaller ***VTU-2B*** also exhibits competitive potential. Compared with SELF1E-8B, the two metrics differ by only approximately 2%. Notably, our method achieves the best performance without explicitly employing chain-of-thought (CoT) (Wei et al., 2023) techniques. These results demonstrate that our method preserves the model's strong reasoning capabilities while enabling segmentation.

Open-Vocabulary Semantic Segmentation. We evaluate our ***VTU*** on open-vocabulary semantic segmentation. Following common practice, we train on COCOStuff and evaluate on ADE20K (A-150), PASCAL Context (PC-59), and PASCAL VOC (PAS-20), using mIoU as the evaluation metric. To obtain a multi-class segmentation map, we perform inference independently for each category to produce a logits map and assign each pixel to the category with the highest score. As shown in Table 4, our method achieves promising performance with an average mIoU of 53.0% and 57.3%, which is only lower than HyperSeg-3B equipped with an additional mask decoder. Our method achieves the best result on A-150 and remains competitive on PC-59 and PAS-20. Notably, some methods are fine-tuned using the evaluation datasets. For a fair comparison, we follow the training setting of SELF1E and further fine-tune our method on these training sets. The results show that ***VTU-2B*** and ***VTU-8B*** achieve average mIoU scores of 77.6% and 79.2%, respectively, surpassing HiMTok-8B* and SELF1E-2B*. These results show that the substantial potential of our method for open-vocabulary semantic segmentation.

General Visual Understanding. We further evaluate ***VTU*** on eight visual understanding benchmarks to validate whether it retains general visual understanding capabilities. As shown in Table 5, ***VTU*** achieves competitive performance while attaining strong segmentation capabilities. Our ***VTU*** outperforms SELF1E of the same model scale across all eight benchmarks and matches or exceeds InternVL3 on GQA and POPE benchmarks. These results demonstrate that our method enhances segmentation performance without compromising the model's generalization capabilities.

Table 2: Results for generalized referring segmentation on gRefCOCO. We compare results under both zero-shot and dataset-specific fine-tuning settings.

Methods	MLLM	Mask Decoder	val		testA		testB		*Avg.*
			cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	
Zero-Shot									
LaSagnA-7B (arXiv'24)	LLaVA-7B	✓	38.1	32.4	50.4	47.3	42.1	38.9	41.5
PSALM-1.3B (ECCV'24)	Swin-B+Phi1.5-1.3B	✓	42.0	43.3	52.4	54.5	50.6	52.5	49.2
OMG-LLaVA-7B (NeurIPS'24)	OMG-Seg+InternLM2-7B	✓	39.3	36.1	52.4	50.1	43.7	42.2	44.0
LIRA-8B (ICCV'25)	InternVL2-8B	✓	40.9	36.7	52.4	50.4	44.9	42.4	44.6
SELF1E-2B (CVPR'26)	InternVL3-2B	✗	41.4	35.2	55.6	51.4	47.0	42.7	45.6
SELF1E-8B (CVPR'26)	InternVL3-8B	✗	44.4	37.5	57.5	53.2	50.9	45.6	48.2
VTU-2B	InternVL3-2B	✗	45.3	40.3	51.1	54.3	48.5	51.9	48.6
VTU-8B	InternVL3-8B	✗	**50.9**	**45.9**	**60.8**	**61.6**	**56.5**	**57.9**	**55.6**
Fine-tuned on gRefCOCO									
LISA-13B (ft) (CVPR'24)	LLaVA-13B	✓	64.0	65.2	71.0	70.0	62.3	62.1	65.8
GSVA-13B (CVPR'24)	LLaVA-13B	✓	66.4	70.0	72.8	73.3	63.2	65.5	68.5
SAM4MLLM-8B (ECCV'24)	LLaVA1.6-8B	✓	67.8	71.9	72.2	74.2	63.4	65.3	69.1
Text4Seg-8B (ICLR'25)	InternVL2-8B	✓	69.1	74.4	73.8	75.1	66.6	67.3	71.1
Text4Seg-13B (ICLR'25)	LLaVA1.5-13B	✓	69.8	74.8	74.3	75.1	67.1	68.0	71.5
HiMTok-8B (ft) (ICCV'25)	InternVL2.5-8B	✓	70.4	72.1	74.9	73.5	72.0	71.7	72.4
STAMP-7B (CVPR'26)	Qwen2VL-7B	✗	70.9	74.4	74.7	73.8	69.1	68.1	71.8
STAMP-7B (CVPR'26)	Qwen2VL-7B	✓	73.6	**77.6**	77.2	77.6	71.6	71.4	74.8
VTU-2B	InternVL3-2B	✗	79.5	72.8	79.3	79.0	74.0	73.6	76.4
VTU-8B	InternVL3-8B	✗	**81.2**	75.3	**81.7**	**81.1**	**76.7**	**75.8**	**78.6**

Table 3: Results for reasoning segmentation on the ReasonSeg validation and test sets. We report cIoU and gIoU (%).

Methods	MLLM	Mask Decoder	val		test	
			cIoU	gIoU	cIoU	gIoU
LISA-7B (ft)	LLaVA1.5-7B	✓	62.9	61.3	56.9	55.6
LISA-13B (ft)	LLaVA1.5-13B	✓	72.9	65.0	62.2	61.3
SAM4MLLM-8B	LLaVA1.6-8B	✓	60.4	58.4	-	-
READ-7B	LLaVA1.5-7B	✓	67.6	59.8	58.6	58.5
HiMTok-8B	InternVL2.5-8B	✓	67.0	60.7	66.2	60.8
ALToLLM-8B	InternVL2.5-8B	✓	67.3	62.8	-	-
STAMP-2B	Qwen2VL-2B	✗	63.9	65.1	60.9	62.7
SELF1E-2B	InternVL3-2B	✗	68.8	59.4	57.6	57.3
SELF1E-8B	InternVL3-8B	✗	69.7	65.9	67.0	65.7
VTU-2B	InternVL3-2B	✗	70.2	64.2	65.9	64.6
VTU-8B	InternVL3-8B	✗	**73.0**	**70.5**	**71.0**	**69.5**

Table 4: Result for open-vocabulary semantic segmentation in mIoU (%). * denotes fine-tuning on the corresponding dataset.

Methods	MLLM	Mask Decoder	A-150	PC-59	PAS-20	*Avg.*
PSALM-1.3B	Swin-B+Phi1.5-1.3B	✓	18.2	48.5	81.3	49.3
LaSagnA-7B	LLaVA-7B	✓	14.3	46.1	69.8	43.4
HyperSeg-3B	Mipha-3B	✓	22.3	64.6	**92.1**	**59.7**
Text4Seg-7B	LLaVA1.5-7B	✗	16.5	52.5	76.5	48.5
SELF1E-2B	InternVL3-2B	✗	24.0	36.9	44.8	35.2
VTU-2B	InternVL3-2B	✗	27.4	59.0	72.6	53.0
VTU-8B	InternVL3-8B	✗	**29.5**	**65.4**	77.1	57.3
HiMTok-8B*	InternVL2.5-8B	✓	25.0	43.9	82.0	50.3
SELF1E-2B*	InternVL3-2B	✗	50.2	64.6	89.9	68.2
VTU-2B*	InternVL3-2B	✗	59.4	77.8	95.5	77.6
VTU-8B*	InternVL3-8B	✗	**63.0**	**78.3**	**96.3**	**79.2**

4.3 Ablation Study

In this section, we conduct ablation studies on the core designs of ***VTU***. We compare the impacts of different token selection and router supervision strategies on segmentation performance, and analyze which visual regions can provide valid visual evidence. We then evaluate performance and computational overhead under different top-K token length, demonstrating that superior performance can be attained by introducing only a small number of effective visual tokens. All ablation experiments are performed based on ***VTU-2B***, with SELF1E-SEG-2B serving as the baseline.

Impact of Token Selection Strategies. To validate the impact of our router selection strategy, we compare two common token selection methods: ❶ random selection, which randomly selects tokens from the coarse-grained visual grid. ❷ uniform selection, which selects spatially distributed tokens uniformly across the grid. We evaluate on RefCOCO/+/g validation sets using cIoU. As shown in Table 6(a), our strategy achieves the best results across three datasets, whereas both non-adaptive methods perform worse than the baseline. These results indicate that introducing arbitrary visual tokens does not universally yield improvement. Random selection may miss critical target tokens, while uniform selection provides broad spatial coverage without accounting for the relevance of each token to the text query. The experimental results validate the necessity of introducing visual details in a targeted manner.

Impact of Router Supervision. We investigate the impact of different router supervision strategies on the final performance, including full foreground supervision, mixed-region supervision, and mixed-region supervision with simple-sample skipping. We also report the number of candidate coarse-grained tokens selected by the router. As shown in Table 6(b), compared with full foreground supervision, our mixed-strategy supervision reduces the average number of candidate tokens by approximately 44%∼48% across three datasets while achieving superior segmentation performance. This indicates that visual information from mixed regions can provide more effective visual

Table 5: Comparison of general visual understanding performance.

Methods	VQAv2	OKVQA	GQA	AI2D	TextVQA	DocVQA	POPE	MME
InternVL3-2B	80.1	56.1	60.7	77.6	77.0	85.9	89.6	2221.2
InternVL3-8B	**81.8**	**61.9**	63.2	**83.9**	**78.9**	**91.2**	91.1	**2415.4**
SELF1E-2B	77.7	48.5	61.5	71.9	71.7	79.8	89.4	2014.6
SELF1E-8B	80.2	54.0	64.1	77.2	72.3	84.4	91.1	2265.1
VTU-2B	77.9	49.0	61.9	73.1	72.2	79.9	91.2	2084.4
VTU-8B	81.6	55.8	**67.1**	82.9	74.7	88.5	**92.3**	2301.4

Table 6: Effectiveness of token selection, router supervision, and router variants. #Cand. is the mean number of coarse-grained candidate visual tokens selected by the router.

Components	RefCOCO		RefCOCO+		RefCOCOg	
	cIoU	#Cand.	cIoU	#Cand.	cIoU	#Cand.
Baseline	84.3	-	78.9	-	80.4	-
(a) Token Selection						
Random	80.7	-	77.3	-	79.0	-
Uniform	82.1	-	78.6	-	79.8	-
Router (ours)	**85.0**	28.1	**81.3**	28.2	**82.8**	26.4
(b) Router Supervision						
Full foreground	82.7	50.3	79.8	53.2	80.8	50.6
Mixed-region	84.8	30.5	81.1	30.7	82.4	30.0
Mixed-region + Skip (ours)	**85.0**	28.1	**81.3**	28.2	**82.8**	26.4
(c) Router Variants						
Router (MLP×1)	**85.0**	27.8	81.0	28.0	81.9	27.5
Router (MLP×2) (ours)	**85.0**	28.1	**81.3**	28.2	**82.8**	26.4
Router (MLP×3)	84.6	29.1	80.7	28.1	81.8	26.8

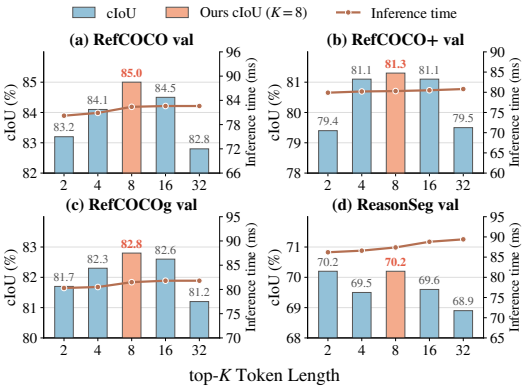


Figure 4: Impact of the top-K token length on performance and inference latency.

evidence to yield better segmentation results. For homogeneous foreground regions, incorporating extra tokens brings marginal gains. Furthermore, the simple-sample skip strategy further cuts down the number of candidates while retaining strong performance, which demonstrates the flexibility of our method across various scenarios.

Router Variants. We compare three router variants in RefCOCO/+/g datasets, and the results are shown in Table 6(c). MLP×2 achieves the best performance across all three datasets. Compared with MLP×1, it maintains the same cIoU on RefCOCO while improving cIoU by 0.3% and 0.9% on RefCOCO+ and RefCOCOg, respectively. In contrast, MLP×3 reduces the average cIoU across the three datasets by approximately 0.7%. This validates the effectiveness of the MLP×2 router in selecting fine-grained visual tokens that are beneficial for segmentation, while showing that further increasing router complexity yields no additional performance gains.

Impact of the top-K Token Length. We analyze the segmentation performance and inference latency under different top-K settings ($K \in \{2, 4, 8, 16, 32\}$), with the results shown in Figure 4. Our method achieves the best performance at $K = 8$ across all four benchmarks, indicating that introducing only a small number of effective visual tokens is sufficient to obtain the best results. A smaller K limits the visual details available to the model, potentially excluding some target-relevant tokens from subsequent segmentation. However, performance declines on all four benchmarks when $K > 8$, indicating that increasing the number of visual tokens does not yield sustained performance gains. We attribute this phenomenon to excessive tokens containing redundant or noisy information, which interferes with the model's exploitation of critical details. Meanwhile, increasing K adds no more than 3.2 ms to inference time. Therefore, introducing a small number of effective visual tokens can balance segmentation accuracy and inference efficiency.

5 Conclusion

In this paper, we propose ***VTU***, a query-guided visual token unfolding method. Our method selectively introduces fine-grained visual tokens to directly perform in image-language interactions before mask prediction. Extensive experiments demonstrate that our method achieves *SOTA* performance on diverse segmentation tasks across different model scales, with only a small number of additional visual tokens, while preserving general visual understanding capabilities. We hope our work paves the way for enhancing the precise segmentation capacity of MLLMs.

AI USE STATEMENT

We used generative AI tools to assist with dataset preparation and reformatting, manuscript organization, drafting, translation, language editing, and LaTeX typesetting. We carefully reviewed all AI-assisted content and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We describe the ***VTU*** architecture, visual token unfolding mechanism, and training objectives in the main paper. Additional details on router label construction are provided in Appendix A.1, while instruction templates, trainable modules, and optimization settings are detailed in Appendix A.2. Datasets splits and sampling strategies are documented in Appendix A.3.

REFERENCES

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.

Yue Cao, Yong Huang, Wei Zhu, Yangzhou Liu, Zhe Chen, Guangchen Shi, Yong Fa, Yujie Yang, Song Mei, and Tong Lu. Mmfuser: Multimodal multi-layer feature fuser for fine-grained vision-language understanding. In *International Conference on Pattern Recognition*, pp. 406–421. Springer, 2026.

Yi-Chia Chen, Wei-Hua Li, Cheng Sun, Yu-Chiang Frank Wang, and Chu-Song Chen. Sam4mllm: Enhance multi-modal large language model for referring expression segmentation. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024a.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67 (12):220101, 2024b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, June 2024c.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2025. URL https://arxiv.org/abs/2306.13394.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2017. URL https://arxiv.org/abs/1612.00837.

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. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. URL https://arxiv.org/abs/1902.09506.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

Aniruddha Kembhavi, Michael Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. *ArXiv*, abs/1603.07396, 2016. URL https://api.semanticscholar.org/CorpusID:2682274.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589, June 2024.

Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wei Zhang. Text4seg: Reimagining image segmentation as text generation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 1634–1661, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/04d8c25cbf0d5e1b70d64e672ac92637-Paper-Conference.pdf.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models, 2023. URL https://arxiv.org/abs/2305.10355.

Zhang Li, Biao Yang, Qiang Liu, Shuo Zhang, Zhiyin Ma, Liang Yin, Linger Deng, Yabo Sun, Yuliang Liu, and Xiang Bai. Lira: Inferring segmentation in large multi-modal models with local interleaved region assistance. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24056–24067. IEEE, 2025.

Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23592–23601. IEEE, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023b. doi: 10.52202/075280-1516. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, June 2024.

Jiazhen Liu, Mingkuan Feng, and Long Chen. Better, stronger, faster: Tackling the trilemma in mllm-based segmentation with simultaneous textual mask prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33121–33130, 2026a.

Yikun Liu, Yuan Liu, Shangzhe Di, Haicheng Wang, Zhongyin Zhao, Le Tian, Xiao Zhou, Jie Zhou, Jiangchao Yao, Yanfeng Wang, et al. Versavit: Enhancing mllm vision backbones via task-guided optimization. *arXiv preprint arXiv:2602.09934*, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge, 2019. URL https://arxiv.org/abs/1906.00067.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021. URL https://arxiv.org/abs/2007.00398.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 891–898, 2014.

Shraman Pramanick, Guangxing Han, Rui Hou, Sayan Nag, Ser-Nam Lim, Nicolas Ballas, Qifan Wang, Rama Chellappa, and Amjad Almahairi. Jack of all tasks master of many: Designing general-purpose coarse-to-fine vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14076–14088, June 2024.

Rui Qian, Xin Yin, and Dejing Dou. Reasoning to attend: Try to understand how <seg> token works. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24722–24731, June 2025.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13009–13018. IEEE, 2024.

Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26364–26373. IEEE, 2024.

Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, Yilin Zhao, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. Eagle: Exploring the design space for multimodal llms with mixture of encoders. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 92628–92646, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/e78457d4a04b8565f1fe5077df13cddb-Paper-Conference.pdf.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Hao Tang, Chen-Wei Xie, Haiyang Wang, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. Ufo: A unified approach to fine-grained visual perception via open-ended language interface. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 83761–83791. Curran Associates, Inc., 2025. doi: 10.52202/085713-2808. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/78a241e1d2ff2366af8cc9a459875a01-Paper-Conference.pdf.

Jingchao Wang, Zhijian Wu, Dingjiang Huang, Yefeng Zheng, and Hong Wang. Unlocking the potential of mllms in referring expression segmentation via a light-weight mask decoder. *arXiv preprint arXiv:2508.04107*, 2025a.

Lingfeng Wang, Hualing Lin, Senda Chen, Tao Wang, Changxu Cheng, Yangyang Zhong, Dong Zheng, and Wuyue Zhao. Alto: Adaptive-length tokenizer for autoregressive mask generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 16054–16078. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0542. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/17a9ab4190289f0e1504bbb98d1d111a-Paper-Conference.pdf.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Tao Wang, Changxu Cheng, Lingfeng Wang, Senda Chen, and Wuyue Zhao. Himtok: Learning hierarchical mask tokens for image segmentation with large multimodal model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23267–23278, October 2025c.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *Advances in Neural Information Processing Systems*, 36:61501–61513, 2023.

Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. Lasagna: Language-based segmentation assistant for complex queries. *arXiv preprint arXiv:2404.08506*, 2024.

Cong Wei, Yujie Zhong, Haoxian Tan, Yong Liu, Jie Hu, Dengjie Li, Zheng Zhao, and Yujiu Yang. Hyperseg: Hybrid segmentation assistant with fine-grained visual perceiver. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8931–8941, June 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, et al. Visionllm v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. *Advances in Neural Information Processing Systems*, 37:69925–69975, 2024.

Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3858–3869. IEEE, 2024.

Jinjin Xu, Liwu Xu, Yuzhe Yang, Xiang Li, Fanyi Wang, Yanchun Xie, Yi-Jie Huang, and Yaqian Li. u-llava: Unifying multi-modal tasks via large language model. In *Proceedings of the 27th European Conference on Artificial Intelligence*, 2024.

Huanjin Yao, Wenhao Wu, Taojiannan Yang, Yuxin Song, Mengxi Zhang, Haocheng Feng, Yifan Sun, Zhiheng Li, Wanli Ouyang, and Jingdong Wang. Dense connector for mllms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 33108–33140. Curran Associates, Inc., 2024. doi: 10.52202/079017-1043. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3a10c46572628d58cb44fb705f25cbbf-Paper-Conference.pdf.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Yuqian Yuan, Wenqiao Zhang, Juekai Lin, Yu Zhong, Mingjian Gao, Binhe Yu, Yunqi Cao, Wentong Li, Yueting Zhuang, and Beng Chin Ooi. Lmms meet object-centric vision: Understanding, segmentation, editing and generation. *arXiv preprint arXiv:2604.11789*, 2026.

Anqi Zhang, Xiaokang Ji, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. Rethinking mllm itself as a segmenter with a single segmentation token. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ao Zhang, Yuan Yao, Wei Ji, Zhiyuan Liu, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation. *arXiv preprint arXiv:2311.04498*, 2023.

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. *Advances in neural information processing systems*, 37:71737–71767, 2024a.

Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. In *European Conference on Computer Vision*, pp. 74–91. Springer, 2024b.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Yikang Zhou, Tao Zhang, Dengxian Gong, Yuanzheng Wu, Ye Tian, Haochen Wang, Haobo Yuan, Jiacong Wang, Lu Qi, Hao Fei, Shunping Ji, Anran Wang, Zhuochen Wang, Yujing Wang, Cheng Chen, and Xiangtai Li. Samtok: Representing any mask with two words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37852–37863, June 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Implementation and Architectural Details **15**
A.1 Router Label Construction 15
A.2 Training Details 16
A.3 Datasets and Baselines 17

B Additional Experimental Analysis and Discussion **18**
B.1 Model-based Supervision Strategy Analysis 18
B.2 More Analysis on VQA 19
B.3 Efficiency Analysis 20
B.4 Limitations and Broader Impacts 20

C Expanded Qualitative Analysis **20**
C.1 Visualization of Router Predictions 20
C.2 More Segmentation Comparisons 21

A Implementation and Architectural Details

A.1 Router Label Construction

Router supervision aims to guide MLLMs toward local details that are difficult to preserve in coarse-grained representations while reducing unnecessary unfolding for simple targets. To this end, we introduce mixed-region supervision and a simple-sample skipping strategy, as shown in Figure 5. We show the ground-truth mask, foreground occupancy on the coarse-grained grid, mixed-region labels, the reconstructed mask, and the final router labels (from left to right). Shades of green indicate foreground occupancy, while orange highlights positive locations designated for unfolding. For the giraffe (first row), mixed-region labels are concentrated around the object contour and thin structures, whereas homogeneous foreground interiors remain unmarked. For the target in the second row, the reconstruction IoU exceeds the threshold of 0.9, indicating that the coarse-grained grid already preserves its overall shape well. All router labels for this sample are set to zero to discourage unnecessary unfolding. The pseudocode for router label construction is illustrated in Algorithm 1.

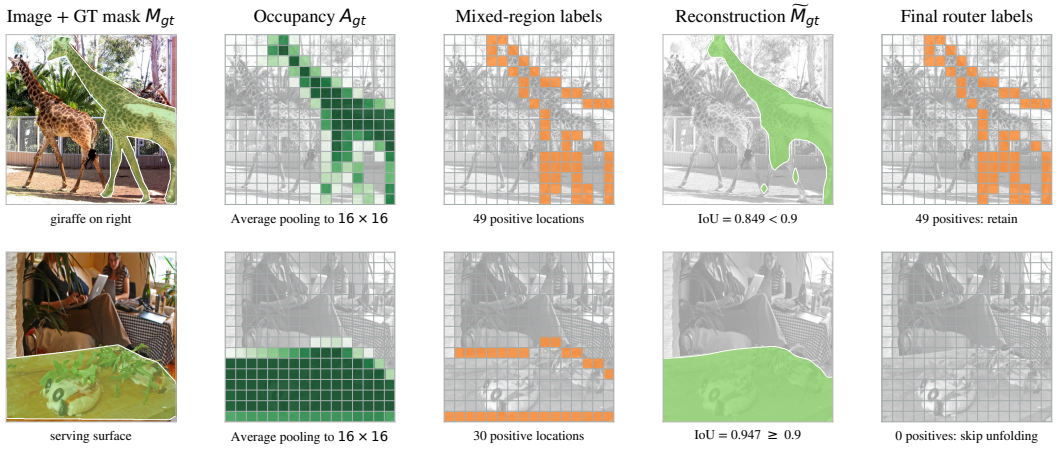


Figure 5: Construction of router labels. We propose a mixed-region supervision and simple-sample skipping strategy.

Algorithm 1 Router Label Construction

Require: Binary target mask M_{gt}; coarse-grid size (H_c, W_c); occupancy thresholds $(\alpha, \beta) = (0.1, 0.9)$; simple-sample threshold $\tau_{\mathrm{simple}} = 0.9$
Ensure: Binary router labels $\mathbf{y}$

Align M_{gt} to the image input resolution using nearest-neighbor interpolation
$Y \leftarrow \mathbf{0}_{H_c \times W_c}$
if $\sum M_{\mathrm{gt}} = 0$ **then**
  return $\mathrm{Flatten}(Y)$ ▷ Empty target
end if
$A_{\mathrm{gt}} \leftarrow \mathrm{AdaptiveAvgPool}(M_{\mathrm{gt}}, (H_c, W_c))$
$Y \leftarrow \mathbb{I}[\alpha < A_{\mathrm{gt}} < \beta]$ ▷ Mixed-region labels
$\widetilde{M}_{\mathrm{gt}} \leftarrow \mathbb{I}[\mathrm{BilinearUpsample}(A_{\mathrm{gt}}, \mathrm{size}(M_{\mathrm{gt}})) \geq 0.5]$
$q \leftarrow \mathrm{IoU}(\widetilde{M}_{\mathrm{gt}}, M_{\mathrm{gt}})$
if $q > \tau_{\mathrm{simple}}$ **then**
  $Y \leftarrow \mathbf{0}_{H_c \times W_c}$ ▷ Skip unfolding for simple targets
end if
$\mathbf{y} \leftarrow \mathrm{Flatten}(Y)$ ▷ Follow the coarse visual token order
return $\mathbf{y}$

Table 7: Trainable components and training settings of ***VTU***. Parameter counts are rounded to millions (M).

(a) Trainable components

Component	Parameters / Target Layers	Params. (2B / 8B)
Visual encoder: attention	W_{qkv}, W_{proj}	19M / 9M
Visual encoder: feed-forward	W_{fc1}, W_{fc2}	31M / 16M
Language model: attention	W_q, W_k, W_v, W_o	35M / 40M
Language model: feed-forward	W_{gate}, W_{up}, W_{down}	113M / 121M
Shared vision–language projector	P	9M / 28M
Input token embeddings	$E \in \mathbb{R}^{\lvert\mathcal{V}\rvert \times d}$	233M / 544M
Language model output head	$W_{\mathrm{head}} \in \mathbb{R}^{\lvert\mathcal{V}\rvert \times d}$	233M / 544M
Router: query/key MLPs	W_q^{router}, W_k^{router}	5M / 26M
Router: normalization	LN_q, LN_k	$<$ 1M / $<$ 1M
Overall	-	**∼677M / ∼1327M**

(b) Training settings

Hyperparameter	2B	8B
Backbone	InternVL3-2B	InternVL3-8B
Backbone initialization	Official pretrained	Official pretrained
Router initialization	Random	Random
LoRA rank (vision / language)	128 / 128	64 / 64
Optimizer	AdamW	AdamW
Initial learning rate	1×10^{-4}	6×10^{-5}
Learning rate schedule	Cosine	Cosine
Weight decay	0	0
Adam (β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.95)$
Adam ϵ	10^{-8}	10^{-8}
Maximum gradient norm	1.0	1.0
Training GPUs	8× RTX 4090	8× RTX 4090
Training epochs	1	1
Effective global batch size	128	128
Gradient accumulation	Enabled	Enabled
Mixed precision	BF16	BF16
Gradient checkpointing	Enabled	Enabled
top-K token length K	8	8
Filtering threshold τ	0.5	0.5
Simple-sample threshold τ_{simple}	0.9	0.9
Spatial downsampling factor r	2	2
Maximum additional visual tokens r^2K	32	32

A.2 Training Details

Training Settings. We jointly train the MLLM and the proposed router. For the visual encoder, we apply LoRA to the joint QKV projection, the attention output projection, and the two linear layers of the feed-forward network. For the language model, LoRA is applied to the query, key, value, and output projections of self-attention, as well as the gate, up, and down projections of the feed-forward network in each Transformer layer. The pretrained weights of these modules remain

frozen, and only the low-rank adaptation parameters are updated. We additionally fully fine-tune the shared vision–language projector, input token embeddings, and language modeling head. Embedding updates cover the entire vocabulary rather than only <UNFOLD> and <SEG>, supporting joint learning of segmentation and visual question answering. All router parameters are trainable. The mask loss backpropagates through the segmentation representation into the language model and the selected visual feature pathways. For discrete routing decisions, we adopt a straight-through estimator (STE), allowing the router to be jointly optimized through explicit routing supervision and downstream task losses. Table 7 summarizes the parameter update settings for each module.

Instruction Construction. We organize segmentation and VQA datasets into text-query pairs. As shown in Example 1, for referring segmentation, we use the original referring expressions to describe the targets. For generalized referring segmentation, masks associated with the same expression are merged, while expressions referring to absent targets are paired with empty masks. For reasoning segmentation, we retain the original implicit instructions and append an explicit segmentation request. During training, we randomly select from the corresponding segmentation prompt templates to introduce linguistic diversity. For open-vocabulary semantic segmentation, we convert semantic annotations into class-conditioned binary segmentation samples. Each query specifies a single category: all valid pixels belonging to that category are treated as foreground, and the remaining valid pixels as background. Ignored regions are excluded from mask supervision. We additionally construct negative queries using categories absent from the image and supervise them with empty masks. Positive and negative queries share the same instruction format, without explicitly indicating whether the target is present. Segmentation responses use <UNFOLD><SEG>, with selected fine-grained visual tokens inserted between these identifiers during model execution rather than included in the textual annotations. For VQA, we retain the original questions and answers, append a short-answer instruction to open-ended questions, and request an option letter for multiple-choice questions. We preserve the original multi-turn conversations in LLaVA-Instruct-150K. VQA responses do not contain segmentation identifiers.

Referring segmentation and reasoning segmentation

- Can you segment the {expression} in this image?
- Please segment the {expression} in this image.
- What is {expression} in this image? Please respond with a segmentation mask.
- What is {expression} in this image? Please output segmentation mask.

Response: <UNFOLD><SEG>

Open-vocabulary semantic segmentation

- Can you segment the {class} in this image?
- Please segment the {class} in this image.
- What is {class} in this image? Please respond with a segmentation mask.
- What is {class} in this image? Please output segmentation mask.

Response: <UNFOLD><SEG>

Visual Question Answering

- **Q**: What color is the car? Answer the question using a single word or short phrase. **A**: red.
- **Q**: Which part of the plant absorbs water? A. Root B. Flower C. Leaf D. Fruit. Answer with the option letter only. **A**: A.
- **Q**: What is the invoice number? Answer the question using a single word or short phrase. **A**: 1042.
- **Q**: How many people are in the image? Answer the question using a single word or short phrase. **A**: three.

A.3 Datasets and Baselines

Datasets. We jointly train and evaluate on multiple datasets, with the splits and sampling settings summarized in Table 8. Given the substantial differences in dataset sizes, we adopt task-level resampling to balance their contributions during training. Taking the sampling multiplier for VQA data as 1, we use multipliers of 3, 6, and 20 for referring, semantic, and reasoning segmentation, respectively. This increases the frequency of segmentation samples during training, with greater

Table 8: Training and evaluation datasets. Training counts and sample visits per epoch are reported in K (10^3) or M (10^6), rounded independently. Sampling multipliers apply only to training.

(a) Training datasets

Task	Dataset	Split	Original	Multiplier	Sampled
Referring segmentation	RefCOCO (Yu et al., 2016)	train	120.62K	3×	361.87K
	RefCOCO+ (Yu et al., 2016)	train	120.19K	3×	360.57K
	RefCOCOg (Mao et al., 2016)	train	80.51K	3×	241.54K
	RefCLEF (Kazemzadeh et al., 2014)	train	108.65K	3×	325.96K
Reasoning segmentation	ReasonSeg (Lai et al., 2024)	train	0.239K	20×	4.78K
Semantic segmentation	COCO-Stuff (Caesar et al., 2018)	train	118.29K	6×	709.72K
	ADE20K (Zhou et al., 2017)	train	20.21K	6×	121.26K
	PASCAL Context (PC-59) (Mottaghi et al., 2014)	train	5.00K	6×	29.99K
	PASCAL VOC (PAS-20) (Everingham et al., 2010)	train	1.46K	6×	8.78K
VQA	VQAv2 (Goyal et al., 2017)	train	443.76K	1×	443.76K
	GQA (Hudson & Manning, 2019)	train	943.00K	1×	943.00K
	OKVQA (Marino et al., 2019)	train	9.01K	1×	9.01K
	TextVQA (Singh et al., 2019)	train	34.60K	1×	34.60K
	AI2D (Kembhavi et al., 2016)	train	12.41K	1×	12.41K
	DocVQA (Mathew et al., 2021)	train	39.46K	1×	39.46K
	LLaVA-Instruct-150K (Liu et al., 2023b)	train	157.71K	1×	157.71K
Overall	-	-	**2.215M**	-	**3.804M**

(b) Evaluation datasets

Task	Dataset	Evaluation splits / subsets
Referring segmentation	RefCOCO (Yu et al., 2016)	val, testA, testB
	RefCOCO+ (Yu et al., 2016)	val, testA, testB
	RefCOCOg (Mao et al., 2016)	val, test
Generalized referring segmentation	gRefCOCO (Liu et al., 2023a)	val, testA, testB
Reasoning segmentation	ReasonSeg (Lai et al., 2024)	val, test
Open-vocabulary segmentation	ADE20K (A-150) (Zhou et al., 2017)	val
	PASCAL Context (PC-59) (Mottaghi et al., 2014)	val
	PASCAL VOC (PAS-20) (Everingham et al., 2010)	val
VQA	VQAv2 (Goyal et al., 2017)	val
	OKVQA (Marino et al., 2019)	val
	GQA (Hudson & Manning, 2019)	test-dev
	AI2D (Kembhavi et al., 2016)	test
	TextVQA (Singh et al., 2019)	val
	DocVQA (Mathew et al., 2021)	val
	POPE (Li et al., 2023)	random, popular, adversarial
	MME (Fu et al., 2025)	perception and cognition (14 tasks)

emphasis on the smaller reasoning segmentation dataset. Under this configuration, each training epoch comprises approximately 3.804M sample visits, with segmentation and VQA accounting for approximately 57% and 43%, respectively.

Baselines. For a comprehensive comparison, we consider representative methods from two paradigms: ❶ *Sequence-as-mask*: This paradigm represents target regions as structured token sequences, such as mask tokens or boundary coordinates, and generates these representations autoregressively. The resulting sequences are decoded directly into masks. We compare against VistaLLM (Pramanick et al., 2024), Text4Seg (Lan et al., 2025), HiMTok (Wang et al., 2025c), ALToLLM (Wang et al., 2025b), STAMP (Liu et al., 2026a), and SAM4MLLM (Chen et al., 2024a). ❷ *Embedding-as-mask*: This paradigm represents targets using the hidden states of special segmentation tokens, which produce pixel-level predictions through a mask decoder or interactions with visual features. We compare against LISA (Lai et al., 2024), PixelLM (Ren et al., 2024), GSVA (Xia et al., 2024), u-LLaVA (Xu et al., 2024), LaSagnA (Wei et al., 2024), GLaMM (Rasheed et al., 2024), VisionLLM v2 (Wu et al., 2024), PSALM (Zhang et al., 2024b), HyperSeg (Wei et al., 2025), OMG-LLaVA (Zhang et al., 2024a), UFO (Tang et al., 2025), READ (Qian et al., 2025), LIRA (Li et al., 2025), and SELF1E (Zhang et al., 2026).

B Additional Experimental Analysis and Discussion

B.1 Model-based Supervision Strategy Analysis

We further investigate a model-based router supervision strategy. Discrepancies between an existing segmentation model's predictions and the ground-truth masks reveal regions where segmentation is inaccurate and may benefit from additional visual detail. We use SELF1E as the reference model to

Table 9: Comparison of model-based and mixed-region router supervision with and without simple-sample skipping. We report cIoU (%) on the RefCOCO/+/g validation sets. We use SELF1E as an offline reference model for label construction. Best results are shown in **bold**.

Supervision Strategy	Reference model	Skipping	RefCOCO	RefCOCO+	RefCOCOg
Model-based	Required	✗	85.0	80.7	82.5
Model-based	Required	✓	**85.1**	81.0	82.3
Mixed-region	Not required	✗	84.8	81.1	82.4
Mixed-region (ours)	Not required	✓	85.0	**81.3**	**82.8**

Table 10: Performance comparison of MME benchmark.

Perception										
Methods	Existence	Count	Position	Color	Posters	Celebrity	Scene	Landmark	Artwork	OCR
InternVL3-2B	**195.00**	**165.00**	136.67	170.00	157.48	155.00	156.25	**163.75**	**156.75**	155.00
SELF1E-2B	190.00	140.00	**153.33**	**175.00**	**164.29**	159.41	160.25	156.50	140.50	102.50
VTU-2B	190.00	148.33	130.00	**175.00**	151.36	**159.71**	**166.25**	149.00	144.75	**162.50**

Cognition and Overall Score					
Methods	Commonsense reasoning	Numerical calculation	Text translation	Code reasoning	**Overall score**
InternVL3-2B	115.71	**105.00**	**185.00**	**147.50**	**2221.2**
SELF1E-2B	97.86	100.00	162.50	112.50	2014.6
VTU-2B	**125.00**	**105.00**	**185.00**	92.50	2084.4

generate a binary mask $\widehat{M}$ for each training image–query pair. The pixel-wise disagreement between $\widehat{M}$ and the ground-truth mask M_{gt} defines an error map:

$$E(x,y) = \mathbb{I}[\widehat{M}(x,y) \neq M_{\text{gt}}(x,y)], \qquad e_i = \frac{1}{|\mathcal{P}_i|} \sum_{(x,y)\in\mathcal{P}_i} E(x,y), \qquad y_i^{\text{err}} = \mathbb{I}[e_i > \eta], \quad (10)$$

Here, $\mathcal{P}_i$ denotes the image region corresponding to the i-th coarse-grained location, and η is the threshold on the fraction of misclassified pixels. These labels capture both false-negative and false-positive foreground predictions. The reference model is used only for offline training-label construction; neither ground-truth masks nor the reference model is required during evaluation.

We evaluate model-based and mixed-region supervision both with and without the simple-sample skipping strategy. As shown in Table 9, both supervision sources achieve comparable performance. With skipping enabled, mixed-region supervision outperforms model-based supervision by 0.3% and 0.5% on RefCOCO+ and RefCOCOg, respectively, while trailing by only 0.1% on RefCOCO. Skipping consistently improves mixed-region supervision across all three datasets, whereas its effect on model-based supervision is mixed, slightly reducing performance on RefCOCOg. These results support mixed-region supervision with simple-sample skipping as an effective choice that avoids the additional offline inference required by a reference model.

B.2 More Analysis on VQA

We further present detailed results of ***VTU-2B*** on MME and POPE to examine its general visual understanding capabilities and object hallucination behavior. As shown in Table 10, ***VTU-2B*** achieves an overall MME score of 2084.4, exceeding SELF1E-2B by 69.8 points, with improvements in scene recognition, OCR, and commonsense reasoning. In particular, its OCR and commonsense reasoning scores reach 162.50 and 125.00, respectively, surpassing both baselines. Nevertheless, its overall performance remains below the original InternVL3-2B, with degradation on tasks such as position recognition and code reasoning. These results indicate that improved segmentation does not necessarily translate into consistent gains across all general-purpose capabilities.

As shown in Table 11, ***VTU-2B*** achieves an overall POPE score of 91.2, compared with 89.6 for InternVL3-2B and 89.4 for SELF1E-2B. Across individual subsets, it obtains F1 scores of 93.68 and 92.04 on Random and Popular, respectively, and 88.09 on the more challenging Adversarial subset. Recall remains at 90.93% across all three subsets, whereas precision drops to 85.41% on Adversarial. This suggests that the model consistently identifies existing objects but remains susceptible to falsely predicting object presence under challenging negative queries. Overall, ***VTU*** retains competitive visual understanding performance alongside its segmentation capabilities, while challenging negative queries and certain cognitive tasks remain areas for improvement.

Table 11: Performance comparison of POPE benchmark.

Methods	Subset	Acc	Prec	Rec	F1	Overall score
InternVL3-2B	Overall	-	-	-	-	89.6
SELF1E-2B	Overall	-	-	-	-	89.4
VTU-2B	Random	93.87	96.60	90.93	93.68	**91.2**
	Popular	92.13	93.17	90.93	92.04	
	Adversarial	87.70	85.41	90.93	88.09	
	Overall	**91.23**	**91.48**	**90.93**	**91.21**	

Table 12: Inference efficiency comparison on a single NVIDIA RTX 4090 GPU.

Method	Inference (ms)	FPS	Memory (GB)
LISA-7B†	250.0	4.00	19.2
UFO-7B†	961.5	1.04	14.4
SELF1E-8B†	105.0	9.52	17.7
VTU-8B ($K = 2$)	111.5	8.97	16.6
VTU-8B ($K = 4$)	112.6	8.88	16.6
VTU-8B ($K = 8$)	114.1	8.77	16.6
VTU-8B ($K = 16$)	114.9	8.71	16.6
VTU-8B ($K = 32$)	125.1	7.99	16.6

B.3 Efficiency Analysis

We summarize inference efficiency on a single NVIDIA RTX 4090. As shown in Table 12, with the default budget $K = 8$, ***VTU-8B*** achieves an average latency of 114.1 ms (8.77 FPS), with a peak memory footprint of 16.6 GB. For reference, SELF1E reports a latency of 105.0 ms, compared with 250.0 ms for LISA and 961.5 ms for UFO. Under our fixed local evaluation settings, increasing K from 2 to 32 raises latency from 111.5 ms to 125.1 ms, while memory consumption remains at 16.6 GB at the reported precision. Thus, larger unfolding budgets introduce modest memory overhead but increase inference time. Our default $K = 8$ adds at most 32 fine-grained visual tokens, providing additional local visual evidence while maintaining a compact token budget.

B.4 Limitations and Broader Impacts

Limitations. Our method constructs router supervision from ground-truth segmentation masks, but these spatial labels do not directly measure how much the fine-grained information at each location contributes to the final prediction. Learning more effective unfolding strategies therefore remains an open direction. Moreover, our experiments primarily use the 2B and 8B variants of InternVL3, applicability to other multimodal architectures and dynamic settings such as video requires further investigation.

Broader Impacts. ***VTU*** has the potential to improve language-guided segmentation for applications such as interactive image editing, visual assistance, and robotic perception. However, it may inherit biases from pretrained models and produce incorrect masks under ambiguous instructions or distribution shifts, affecting the reliability of downstream actions. Fine-grained localization could also be misused for unauthorized surveillance. Practical deployment should therefore incorporate privacy safeguards, appropriate use restrictions, and human verification where necessary, particularly in high-risk settings.

C Expanded Qualitative Analysis

C.1 Visualization of Router Predictions

We visualize router labels and selected visual tokens for different queries. As shown in Figure 6, orange indicates the supervision labels, while red denotes the predicted selections. The selected locations primarily concentrate on the queried targets and their surrounding regions, covering local structures such as the cat's limbs, the giraffe's slender neck and legs, and the vase's handles and contours. In scenes with overlapping people or partially visible targets, the router also focuses on the person or clothing region specified by the query rather than other visually salient subjects. Predicted selections do not exactly coincide with the supervision labels, and some extend into nearby background regions, revealing remaining localization errors. These examples illustrate query-guided

region selection, allowing fine-grained visual information to be introduced at target-related locations without unfolding the entire image.

C.2 More Segmentation Comparisons

We present additional qualitative results of ***VTU-2B*** on referring segmentation, reasoning segmentation, and open-vocabulary semantic segmentation, alongside ground-truth masks. As shown in Figure 7, 8 and 9, for referring segmentation, the model distinguishes targets using spatial and appearance cues, such as the boy on the left, the partially occluded truck on the right, and both striped socks. For reasoning segmentation, it identifies targets from functional descriptions and contextual relationships, locating the pen described as an object for writing handwritten notes or letters and distinguishing the cut slice of cake from the remaining whole. For open-vocabulary semantic segmentation, the predictions largely capture the semantic layout of indoor and outdoor scenes, covering objects such as people and cars, as well as regions such as walls, grass, and sky. Nevertheless, discrepancies remain around small structures, occlusion boundaries, and local contours. These samples demonstrate that ***VTU*** can translate various language instructions into pixel-level predictions, and also reveal that there remains room for further improvement in the model's fine boundary reconstruction.

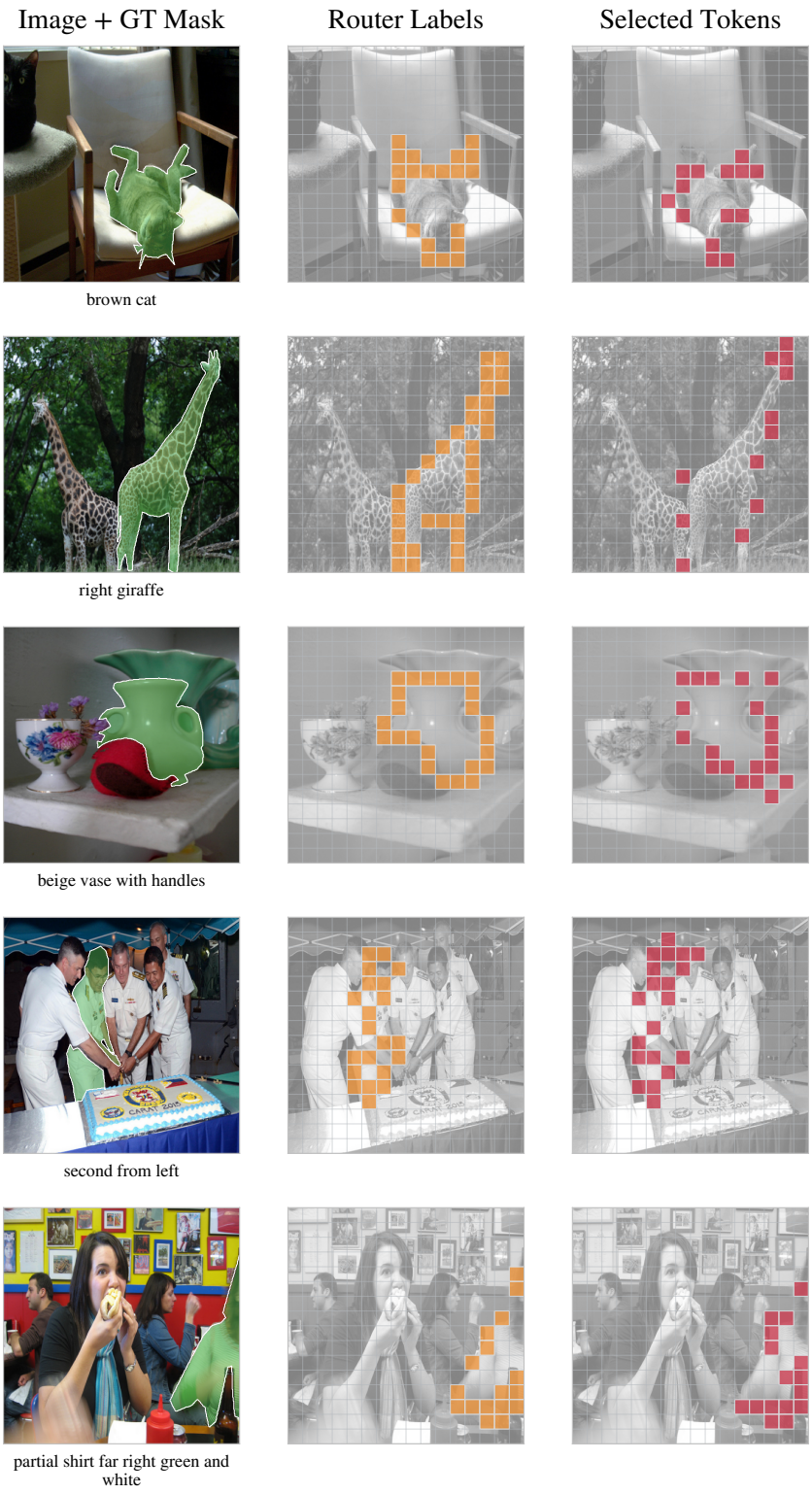
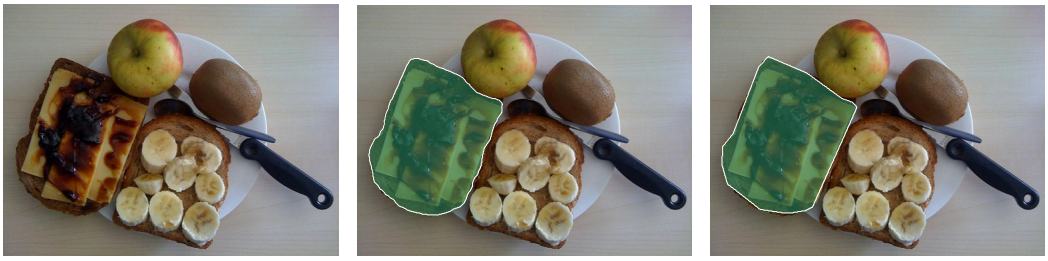


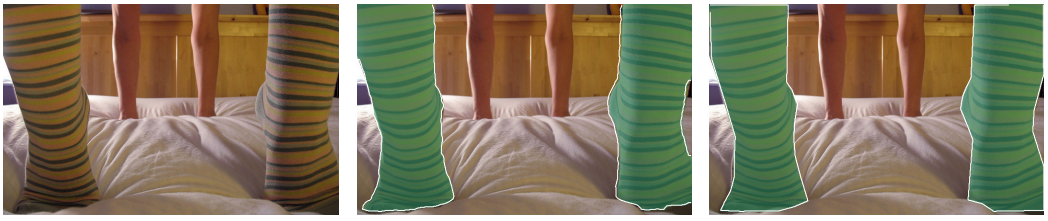
Figure 6: Visualization of router predictions and router labels.



Query: What is boy on left in this image? Please output segmentation mask.



Query: What is left most food in this image? Please output segmentation mask.



Query: What is striped socks either in this image? Please output segmentation mask.



Query: What is truck on right in this image? Please output segmentation mask.

Figure 7: Visualization of referring segmentation.



Query: In a world where digital communication dominates, people may still enjoy writing by hand. What object in the picture can be used to make handwritten notes or letters? Please output segmentation mask.



Query: What is the part of cake that was cut down in this image? Please output segmentation mask.



Query: In the picture, there are various containers used to hold fruit. Which container would be most suitable for serving and holding sliced apples? Please output segmentation mask.



Query: What is the owner of this car in this image? Please output segmentation mask.

Figure 8: Visualization of reasoning segmentation.

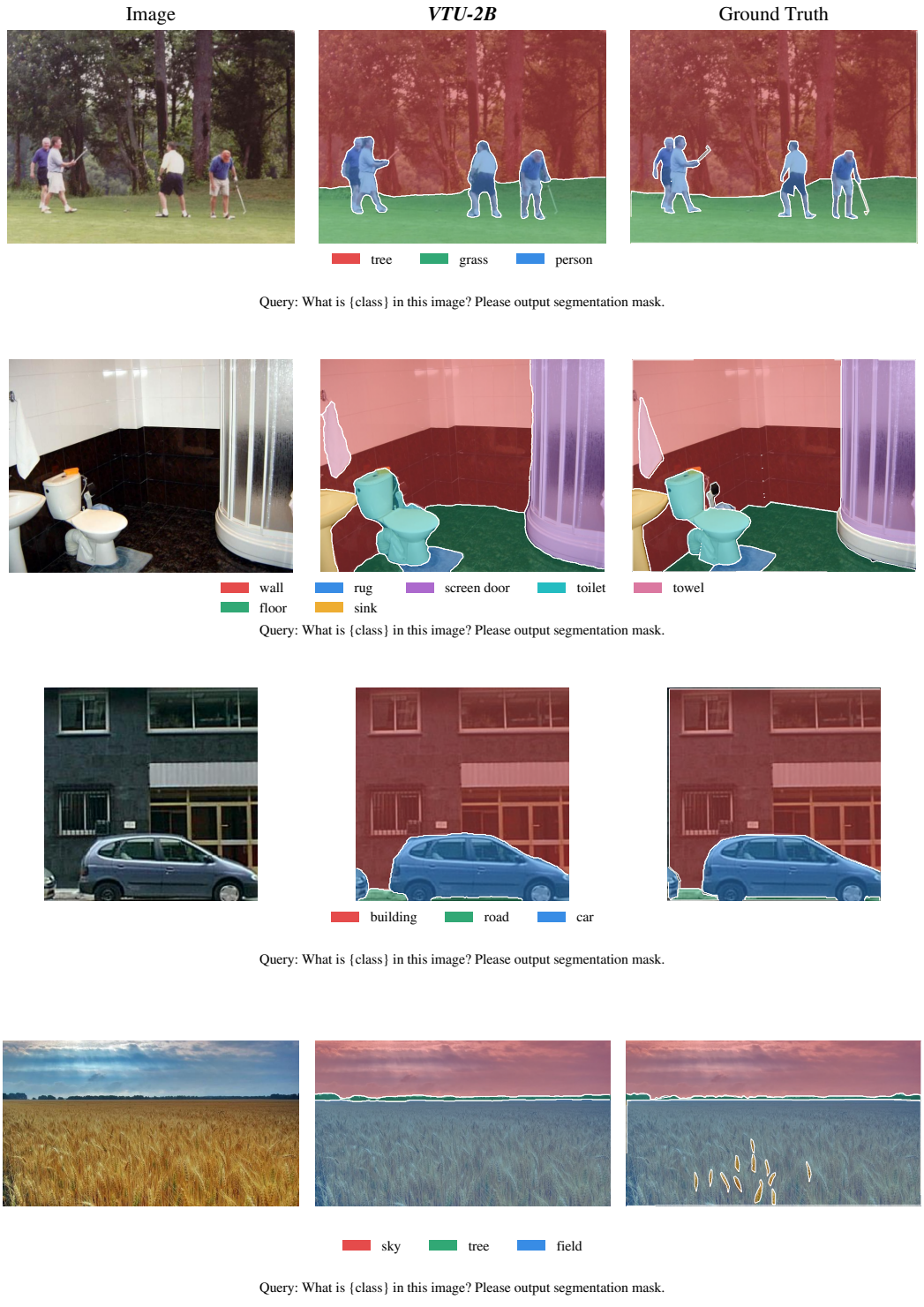


Figure 9: Visualization of open-vocabulary semantic segmentation.