ARGenSeg: Image Segmentation with Autoregressive Image Generation Model

Xiaolong Wang, Lixiang Ru, Ziyuan Huang, Kaixiang Ji, Dandan Zheng
Jingdong Chen, Jun Zhou

Ant Group
{xiaowang.wxl, rulixiang.rlx, pishi.hzy, kaixiang.jkx, yuandan.zdd}@antgroup.com
{jingdongchen.cjd, jun.zhoujun}@antgroup.com

Figure 1: ARGenSeg is a unified framework for visual understanding, segmentation, and generation. It supports semantic, instance, interactive, and zero-shot reasoning segmentation, as well as anomaly detection, by leveraging strong visual understanding capabilities.

Abstract

We propose a novel **A**uto**R**egressive **Gen**eration-based paradigm for image **Seg**mentation (ARGenSeg), achieving multimodal understanding and pixel-level perception within a unified framework. Prior works integrating image segmentation into multimodal large language models (MLLMs) typically employ either boundary points representation or dedicated segmentation heads. These methods rely on discrete representations or semantic prompts fed into task-specific decoders, which limits the ability of the MLLM to capture fine-grained visual details. To address these challenges, we introduce a segmentation framework for MLLM based on image generation, which naturally produces dense masks for target objects. We leverage MLLM to output visual tokens and detokenize them into images using an universal VQ-VAE, making the segmentation fully dependent

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

on the pixel-level understanding of the MLLM. To reduce inference latency, we employ a next-scale-prediction strategy to generate required visual tokens in parallel. Extensive experiments demonstrate that our method surpasses prior state-of-the-art approaches on multiple segmentation datasets with a remarkable boost in inference speed, while maintaining strong understanding capabilities.

1 Introduction

The emergence of large language models (LLMs) [8, 17, 52] has significantly accelerated the development of artificial general intelligence (AGI) [9]. Breakthroughs like ChatGPT [40] enable the transformer-based [54] autoregressive framework to unify diverse tasks of natural language processing [1, 4]. As for multimodal task, LLaVA [34] employs visual adaptor to map visual features into the embedding space of LLMs, establishing a universal paradigm for multimodal large language models [33, 5, 15, 56]. Recent studies [48, 64, 49, 67, 51, 61, 2, 3, 22] explore the unified framework for multimodal understanding and generation. However, integrating fundamental visual perception tasks into a unified AGI framework remains an open challenge. While sparse-output tasks such as visual grounding can be directly addressed via text expression [14], tasks requiring dense outpus like image segmentation are inherently difficult to represent through natural language.

Previous methods that incorporate image segmentation into MLLMs typically fall into two categories. The first discretizes dense masks into boundary point sequences [11, 42, 58], which inevitably leads to incomplete segmentation masks and unnatural object boundaries. The second achieves segmentation through downstream dedicated decoders (*e.g.*, SAM [27], Mask2Former [16]), which are conditioned on either textual prompts [12] or hidden states [29, 45, 75] generated by MLLMs. This not only results in complex model architectures, but also leads to insufficient understanding of pixel-level information for LLM due to its reliance on specialized task head.

To address the above challenges, we propose ARGenSeg, which leverages the image generation-based paradigm to integrate image segmentation into a unified MLLM framework. To retain the strong understanding capability of MLLMs, we use continuous image features as the input. For the generation output, we train the model to directly predict quantized image tokens, aligning with the next-token autoregressive prediction mechanism of language models. We use a pre-trained VQ-VAE as image tokenizer to quantize and detokenize images, with its visual tokens added to the codebook of MLLM. By leveraging the understanding ability of MLLM, ARGenSeg is capable of additional complex reasoning segmentation [29], anomaly detection [7, 6] and other image segmentation tasks [75] as shown in Fig. 1. The image tokenizer is kept frozen throughout training, thereby avoiding the dependence of LLM on subsequent decoders when learning pixel-level information.

In real-world application, image segmentation often requires fast response times. For this purpose, we adopt a next-scale prediction strategy for image generation. On one hand, the multi-scale mask generation process aligns with the intuitive process of object segmentation, which typically involves coarse localization followed by fine-grained boundary refinement. On the other hand, generating visual tokens in parallel provides a significant efficiency advantage, achieving over $4\times$ speedup compared to sequential generation methods [19, 59].

Some methods also propose to use image generation for image segmentation. UniGS [43] uses diffusion model [21, 46] to achieve image segmentation. However, its U-Net structure causes lack of understanding ability. HiMTok [57] proposes an innovative mask tokenizer that enables decoding discrete outputs from the MLLM into binary masks via image generation. However, the task-specific tokenizer limits its generality and extensibility. Moreover, both of these methods suffer from significant disadvantages in inference speed.

Extensive experiments demonstrate that the proposed ARGenSeg outperforms existing MLLM-based segmentation methods, while also achieving significantly faster inference. Notably, our method achieves superior performance using substantially less segmentation data compared to prior state-of-the-art approach [57]. In addition, the use of a general-purpose visual tokenizer provides the flexibility to extend the framework to additional tasks. As a demonstration, by fine-tuning on a small amount of image generation data, we successfully unlock the image generation capability of our framework, as illustrated in Fig. 1.

The main contributions of this paper include:

- We propose a novel image segmentation framework based on a unified multimodal understanding and generation paradigm. To our knowledge, we are the first to show that unified MLLMs can achieve SOTA segmentation results without any extra segmentation heads.
- We leverage a universal image tokenizer, allowing segmentation to fully rely on the pixel-level visual understanding of the MLLM. We further show that direct image token prediction by the MLLM is important for achieving high segmentation accuracy.
- We propose to use next-scale prediction to speed up inference. And we observe that the coarse-to-fine multi-scale mask generation process also boosts segmentation robustness.

2 Related Work

Integrating image segmentation into MLLMs not only equips them with fine-grained visual perception, but also enables more complex reasoning-based segmentation tasks by leveraging understanding capabilities. However, representing segmentation masks within the MLLM framework remains a significant challenge. PolyFormer [35] and VistaLLM [42] represent masks as polygons using point sequences, which are easy to express but struggle with complex shapes. LISA [29] aggregates segmentation information using special tokens and predicts masks through a SAM [27] decoder. Subsequent works such as GLaMM [44], PixelLM [45], GSVA [65], and PSALM [75] build upon this paradigm, and still rely on special tokens and dedicated segmentation decoders. These methods essentially aim to extract semantic embeddings of target objects and then obtain dense segmentation masks by ***computing similarity with image features***. Such representations tend to emphasize high-level semantics rather than true pixel-level understanding. HiMTok [57] explores an alternative that removes the reliance on special tokens and SAM-like decoders. However, it still depends on a dedicated mask tokenizer trained on binary masks. Moreover, the expressiveness of the tokenizer is limited and cannot be extended to support other tasks such as image generation. This suggests that segmentation representation in MLLMs remains an open challenge, which we think can be effectively addressed through autoregressive image generation.

Unified multimodal understanding and generation models have recently attracted increasing attention for their ability to seamlessly perform both understanding and generation tasks within a single framework. Several works [48, 20, 63, 51] leverage diffusion models for image generation by regressing visual embeddings from MLLM outputs and using them as conditional inputs. TransFusion [77] and Show-O [67] unify next-token prediction and diffusion-based generation within a single transformer framework. Chameleon [49] and Emu3 [59] adopt a shared discrete visual embedding space for both understanding and generation, decoding images through VQ-based tokenizers [19, 71]. Janus [61] decouples the encoder for multimodal understanding and generation, using discrete visual tokens for generation while retaining continuous visual features for better understanding accuracy. VARGPT [78] proposes next-token prediction for understanding and next-scale prediction for image generation, but relies on an additional transformer-based visual decoder.

Image tokenization enables discrete outputs from autoregressive models to be reconstructed into images. VQ-VAE [53] encodes images into a downsampled latent space and quantizes the features into discrete token IDs, simplifying the learning process for generative models. VQGAN [19] improves reconstruction quality and training efficiency through adversarial training. TiTok [72] significantly reduces the number of tokens required for image representation, improving generation speed, and further shows that increasing the number of latent tokens consistently enhances reconstruction quality. VAR [50] reformulates visual autoregressive generation as a next-scale prediction task, achieving high efficiency while maintaining a relatively large number of visual tokens.

3 Method

In this paper, we propose a novel image segmentation framework based on autoregressive image generation model, using a Vector-Quantized (VQ) autoencoder [53, 19] to tokenize images into discrete tokens and reconstruct them from generated outputs. To address the unique challenges of segmentation, we introduce two key designs. (1) **The MLLM is trained to directly output image tokens**, which is crucial for achieving high pixel-level accuracy. (2) **We utilize a multi-scale generation process that performs coarse-to-fine refinement.** This not only enhances segmentation robustness but also improves inference efficiency. This section first presents the background of the

image tokenizer (Sec. 3.1), then details the architecture (Sec. 3.2), training procedure (Sec. 3.3), and inference process (Sec. 3.4) of our proposed model.

3.1 Preliminary

Vector-Quantized Autoencoder The standard VQ model learns to encode images into a latent space and reconstruct them from discrete tokens. Given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, the encoder $\mathcal{E}$ maps it to a latent feature space:

$$f = \mathcal{E}(\mathbf{I}), \quad f \in \mathbb{R}^{\frac{H}{l}\times\frac{W}{l}\times D}, \tag{1}$$

where l is the spatial downsampling factor and D denotes the feature dimesion. The latent features f are then quantized by a vector quantizer $\mathcal{Q}$ into discrete token indices $q \in [V]^{\frac{H}{l}\times\frac{W}{l}}$:

$$q = \mathcal{Q}(f), \quad q^{(i,j)} = \underset{v\in[V]}{\arg\min}\|f^{(i,j)} - c^v\|_2, \tag{2}$$

where c^v is the v-th embedding vector in the visual codebook $\mathbb{C} \in \mathbb{R}^{V\times D}$, and $[V]$ denotes the set of codebook indices $\{1, 2, \ldots, V\}$.

The reconstruction of the image can be interpreted as detokenizing discrete visual tokens into an image. In this procedure, the quantized indices q are used to index the corresponding embedding from the visual codebook $\mathbb{C}$, producing the estimated latent feature map $\hat{f}$. The estimated feature map is then passed through the decoder $\mathcal{D}$ to generate the reconstructed image $\hat{\mathbf{I}}$:

$$\hat{f} = \textbf{lookup}(\mathbb{C}, q), \quad \hat{\mathbf{I}} = \mathcal{D}(\hat{f}). \tag{3}$$

Multi-Scale VQ Autoencoder When using VQ-VAE for autoregressive image generation, the inference process typically requires $\mathcal{O}(n^2)$ steps. To address this inefficiency, VAR[50] introduces a next-scale prediction paradigm for visual token generation. Specifically, the feature map f is quantized into K multi-scale token maps $(r_1, r_2, \ldots, r_K)$, where each map corresponds to a different resolution. At each inference step, the model generates all $h_k \times w_k$ tokens required for the current scale r_k in parallel, repeating this process until r_K reaches the target resolution of $\frac{H}{l} \times \frac{W}{l}$. Moreover, the coarse-to-fine predictions can enhance the generation quality. Based on this paradigm, an image of resolution 256×256 can be represented using 680 visual tokens, while requiring just K autoregressive inference steps, significantly improving generation efficiency. Given the fast response requirements of image segmentation tasks, we adopts this paradigm to enable efficient autoregressive image generation.

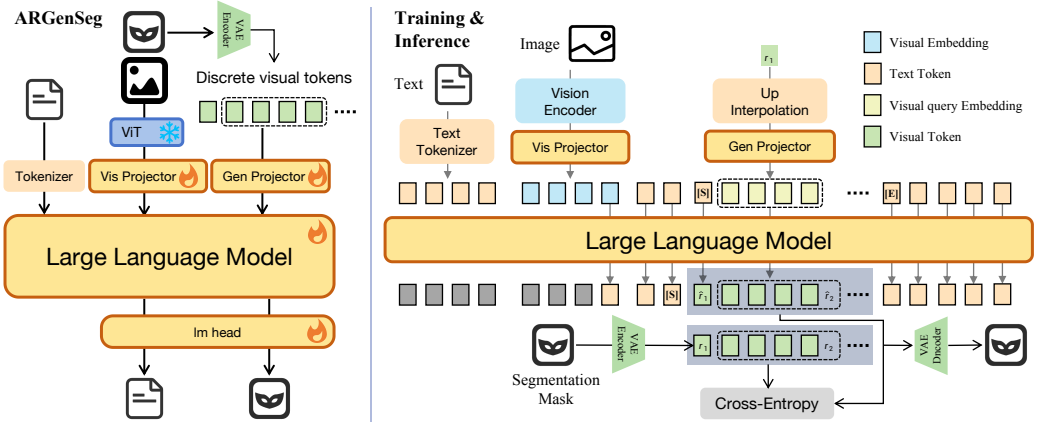


Figure 2: The architecture of ARGenSeg and its training and inference procedures. **Left:** ARGenSeg integrates image segmentation into the MLLM via an autoregressive image generation paradigm. A unified classification prediction head is used to generate both text and visual tokens. **Right:** Visual tokens are generated in parallel using the next-scale prediction strategy. During training, a VAE encoder is used to construct supervision for cross-entropy loss. During inference, the VAE decoder reconstructs the image from the predicted visual tokens. `[S]`/`[E]` denotes `<gen_start>`/`<gen_end>`.

3.2 Architecture

Multimodal Understanding ARGenSeg uses a unified autoregression framework for image understanding and generation as shown in Fig. 2. Our framework employs the built-in tokenizer of the LLM to convert text input into discrete token IDs and corresponding embeddings. For image input, a vision encoder is used to extract features, which are then mapped to the LLM's embedding space via a vision projector. After the concatenated embeddings are fed into the LLM, the model performs next-token prediction to sequentially generate token embeddings. These embeddings are then passed through a classification head to sample discrete token IDs, which are subsequently detokenized into meaningful text. For multimodal understanding tasks, decoupling the framework from image generation preserves the native understanding capabilities of the LLM.

Image Generation To integrate image generation into the framework, we introduce special tokens `<gen_start>` and `<gen_end>` to mark the beginning and end of the generation process. Additionally, the visual token IDs from the visual tokenizer are added to the LLM's vocabulary in the form of `<visual_token_ID>`. When image generation is required, the framework autonomously determines whether to initiate generation based on the input instruction. Upon encountering the `<gen_start>` token, multi-scale image generation begins, where visual tokens for each scale are predicted in parallel. At k-th scale, the visual feature corresponding to the visual token map from the previous scale is retrieved by looking up the visual codebook $\mathbb{C}$ and then upsampled to match the resolution of the current scale. A lightweight linear layer, referred to as the generation projector, maps these upsampled visual features into the embedding space of LLM, serving as input for the next scale. This design allows one-step parallel inference to obtain all visual tokens at the current scale. Importantly, the unified prediction head is used to generate visual tokens, which are then directly converted to the corresponding index IDs in the codebook $\mathbb{C}$. Once all visual tokens across scales are generated, they are detokenized by the visual tokenizer to reconstruct the final image.

3.3 Training Procedure

Training Strategy In our framework, the vision encoder, large language model, vision projector and classification prediction head are initialized using InternVL 2.5[13], while the multi-scale visual tokenizer is initialized from VAR [50]. During training, the vision encoder and visual tokenizer are kept frozen to reduce the model's reliance on dedicated decoders for pixel-level understanding. By leveraging pre-trained multimodal understanding, the framework converges rapidly when training on image segmentation data. Thus, we employ a single-stage supervised finetuning (SFT) strategy, jointly optimizing both image segmentation and multimodal understanding data. For image generation, we further finetune the pre-trained ARGenSeg model using image generation data to unlock its text-to-image generation capabilities.

Training Objective Since our framework unifies both text and image generation outputs within the LLM codebook, the entire training process is directly supervised using cross-entropy loss, as shown in Fig. 2. During supervision construction, the `<gen_start>` token is added as a marker before image generation begins. The model is expected to **learn both when to initiate image generation and how to generate all the required visual tokens.** The ground-truth visual tokens are obtained using the encoder and quantizer of the VQ-VAE. When constructing input embeddings, the visual tokens for the first scale are obtained by using the `<gen_start>` token as the query. For each subsequent scale, the input embeddings are derived by upsampling the visual token map r_{k-1} of the previous scale to match the size of the current scale. Finally, the `<gen_end>` token is added to ensure the proper progression of subsequent predictions.

3.4 Inference

During inference, our model follows a next-token prediction strategy, generating outputs sequentially until the `<gen_start>` token is produced. This token then serves as a query to initiate the generation of visual tokens for the first scale. For the subsequent $K-1$ scales, query embeddings of size $h_k \times w_k$ are obtained by upsampling and projecting the visual token map $\hat{r}_{k-1}$ predicted at the previous scale, enabling parallel generation of all visual tokens at the current scale. Since the upsampling process determines the number of queries, our framework naturally ensures alignment between the number of generated tokens and the input size required by the VQ-VAE decoder. Once the visual tokens for

Table 1: Performance comparison with state-of-the-art methods on three referring image segmentation benchmarks using cIoU. (ft) indicates models further finetuned on RefCOCO/+/g after mixed training.

Paradigm	Method	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val	test
Boundary Point-based	PolyFormer-B [42]	74.8	76.6	71.1	67.6	72.9	59.3	67.8	69.1
	VistaLLM-7B [42]	74.5	76.0	72.7	69.1	73.7	64.0	69.0	70.9
Dedicated Segmentation Head-based	LISA-7B(ft) [29]	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
	PixelLM-7B [45]	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5
	GSVA-7B [65]	76.4	77.4	72.8	64.5	67.7	58.6	71.1	72.0
	GSVA-7B(ft)	77.2	78.9	73.5	65.9	69.6	59.8	72.7	73.3
	LaSagnA-7B [60]	76.8	78.7	73.8	66.4	70.6	60.1	70.6	71.9
	VisionLLM v2 [62]	76.6	79.3	74.3	64.5	69.8	61.5	70.7	71.2
	OMG-LLAVA [73]	75.6	77.7	71.2	65.6	69.7	58.9	70.7	70.2
	OMG-LLAVA(ft)	78.0	80.3	74.1	69.1	73.1	63.0	72.9	72.9
	GLaMM [44]	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9
	u-LLAVA [68]	83.0	85.1	80.5	77.1	81.7	70.6	77.1	78.0
	PSALM [75]	83.6	84.7	81.6	72.9	75.5	70.1	73.8	74.4
	GroundHog-7B [74]	78.5	79.9	75.7	70.5	75.0	64.9	74.1	74.6
	SAM4MLLM-8B [42]	79.8	82.7	74.7	74.6	80.0	67.2	75.5	76.4
	$\text{LMM}_{\text{HiMTok}}$-8B [57]	81.1	81.2	79.2	77.1	78.8	71.5	75.8	76.7
	$\text{LMM}_{\text{HiMTok}}$-8B(ft)	85.0	85.2	**83.5**	79.7	82.7	76.0	80.0	80.6
Generation based	ARGenSeg	82.2	84.0	80.1	77.9	81.8	73.3	78.4	79.6
	ARGenSeg (ft)	**86.3**	**87.5**	82.7	**82.3**	**85.8**	**77.0**	**81.7**	**83.5**

all K scales are obtained, the VAR tokenizer decodes them into the final image. To ensure smooth progression of subsequent inference, the `<gen_end>` token is manually added.

4 Experiments

4.1 Experimental Setup

Datasets As described in Sec. 3.3, we perform a single-stage supervised finetuning to jointly train on both image segmentation and multimodal understanding data. Details of all datasets used are provided in Appendix A. The training of ARGenSeg relies entirely on publicly available external datasets. Specifically, we use 402K image segmentation samples, which are significantly fewer than the 2.91M samples used by HiMTok[57] and constitute a strict subset of their data. For multimodal understanding, we use 1.25M samples derived from the open-source dataset of InternVL 1.2 [14].

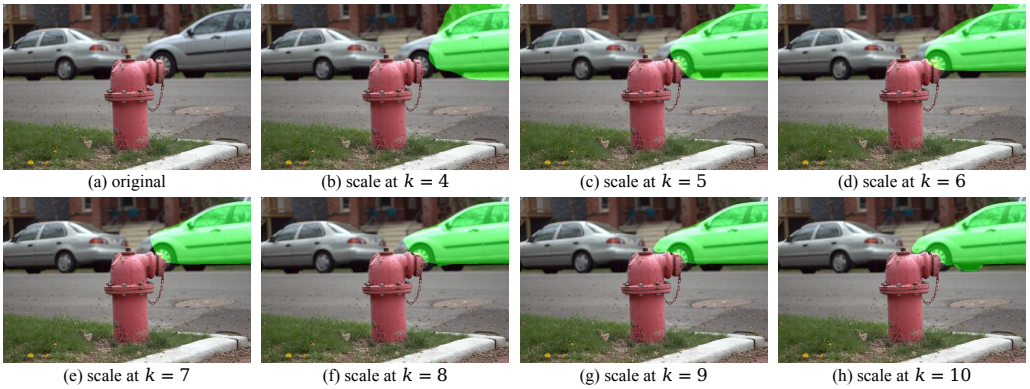
Implementation Details Our model accepts input images of arbitrary resolutions, while the output images are generated at the resolution of 256×256. The image tokenizer uses a downsampling ratio $l = 16$, with a feature dimension $D = 32$ and a visual codebook size $V = 4096$. The model operates with $K = 10$ scales. During training, we use the AdamW [36] optimizer with a maximum learning rate of 4×10^{-5} and employ cosine learning rate scheduling. The batch size is set to 128.

4.2 Referring Segmentation

Referring Expression Segmentation Recent works have increasingly focused on equipping multimodal large language models with image segmentation capabilities, aiming to leverage their strong language understanding for more complex segmentation tasks. Referring Expression Segmentation (RES) requires models to segment target objects in an image based on natural language descriptions. We evaluate our approach on standard RES benchmarks RefCOCO/+/g [37, 70]. Following prior works [29, 57], we assess two versions of our model: one trained on the mixed dataset, and another further finetuned on the in-domain training sets of RefCOCO/+/g. As shown in Tab. 1, our method consistently outperforms the previous state-of-the-art, HiMTok [57], across both versions, despite training on fewer segmentation data. It is worth noting that, our approach achieves superior results without relying on a dedicated segmentation head, demonstrating the effectiveness of our unified multimodal understanding and generation framework.

Table 2: Performance comparison with state-of-the-art methods on generalized referring expression segmentation. * indicates zero-shot performance.

Method	val		testA		testB		Average
	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	
LISA-7B [29]	38.7	32.2	52.6	48.5	44.8	39.7	42.8
LISA-7B(ft)	61.8	61.6	68.5	66.3	60.6	58.8	62.9
GSVA-7B [65]	61.7	63.3	69.2	70.1	60.3	61.3	64.3
GSVA-7B(ft)	63.3	66.5	69.9	71.1	60.5	62.2	65.6
LaSagnA* [60]	38.1	32.4	50.4	47.3	42.1	38.9	41.5
PSALM* [75]	42.0	43.3	52.4	54.5	50.6	52.5	49.2
GroundHog-7B [74]	-	66.7	-	-	-	-	66.7
SAM4MLLM-8B [42]	67.8	71.9	72.2	**74.2**	63.4	65.3	69.1
$\text{LMM}_{\text{HiMTok}}$-8B [57]	66.8	68.7	68.6	67.6	65.8	64.1	66.9
ARGenSeg	**72.2**	**74.7**	**73.6**	73.7	**70.0**	**70.4**	**72.4**



(a) original (b) scale at $k = 4$ (c) scale at $k = 5$ (d) scale at $k = 6$

(e) scale at $k = 7$ (f) scale at $k = 8$ (g) scale at $k = 9$ (h) scale at $k = 10$

Figure 3: Multi-scale generation process of the segmentation mask. The model first localizes the target object and then progressively refines its boundaries.

Fig. 3 illustrates the multi-scale mask generation process of ARGenSeg. The model first locates the target object and then progressively refines the segmentation boundaries. This coarse-to-fine reasoning process aligns with human intuition and enhances the robustness of image segmentation.

Generalized Referring Expression Segmentation We further evaluate our model on the more challenging gRefCOCO benchmark [32], where segmentation instructions may refer to multiple objects or none at all. As shown in Tab. 2, our method outperforms all prior approaches that rely on dedicated segmentation heads, highlighting the strong understanding and segmentation capabilities of our unified framework.

4.3 Multumodal Understanding

Our model adopts InternVL 2.5 [13] as the underlying MLLM and is finetuned on both understanding and segmentation data. To fairly assess the effect of adding segmentation supervision on the model's understanding capability, we finetune a baseline using only understanding data. We evaluate the model's understanding performance using two tasks. The first task is visual grounding, where we use the RefCOCO/+/g datasets for referring expression comprehension (REC). As shown in Tab. 3, our model successfully retains and even slightly enhances its grounding ability while acquiring segmentation capabilities. The second task evaluates object hallucination in MLLMs using POPE [30] as the benchmark. Results in Tab. 3 also demonstrate a performance improvement of our model compared to the baseline. These results highlight the effectiveness of our proposed framework in unifying understanding and segmentation tasks. A further discussion on the understanding performance is provided in Appendix C.1.

Table 3: Multimodal understanding performance compared with the baseline. * indicates further finetuning on understanding data.

Method	RefCOCO			RefCOCO+			RefCOCOg		POPE
	val	testA	testB	val	testA	testB	val	test	
InternVL2.5-8B* [13]	89.0	92.6	84.3	83.4	**89.1**	**76.5**	83.5	85.0	86.73
ARGenSeg	**89.6**	**92.8**	**84.4**	**83.8**	88.8	**76.5**	**86.1**	**85.6**	**87.57**

4.4 Function Extension

Interactive Segmentation Interactive segmentation allows users to provide diverse input prompts during segmentation tasks to meet varying application needs. We finetune ARGenSeg on the COCO-Interactive dataset [75] to unlock its interactive segmentation capabilities. During training, various forms of interactive prompts are used, including **points, scribbles, and bounding boxes.** Bounding boxes are provided as textual input to the MLLM, while points and scribbles are represented as binary masks and fed in as additional visual inputs. We observe that, building upon pre-trained segmentation capabilities, the model quickly adapts to interactive segmentation tasks. Qualitative results are shown in the top portion of Fig. 4, while the quantitative evaluation can be found in the Appendix C.2.

Image Generation Our model leverages a universal image tokenizer, enabling the potential for image generation. We finetune ARGenSeg on 1.28M class-based samples from the ImageNet-Instruct-class dataset [78], using a batch size of 512 for $20k$ iterations. This successfully enables class-conditional image generation, as illustrated in Fig. 1. We then continue training for an additional $30k$ iterations with a batch size of 256 on the ImageNet-Instruct1270K dataset [78], which is based on instruction-conditioned generation. The results of instruction-based image generation are shown in the bottom of Fig. 4. Notably, our model achieves these results **without relying on pre-trained generation model**, using only a small amount of data and training iterations.

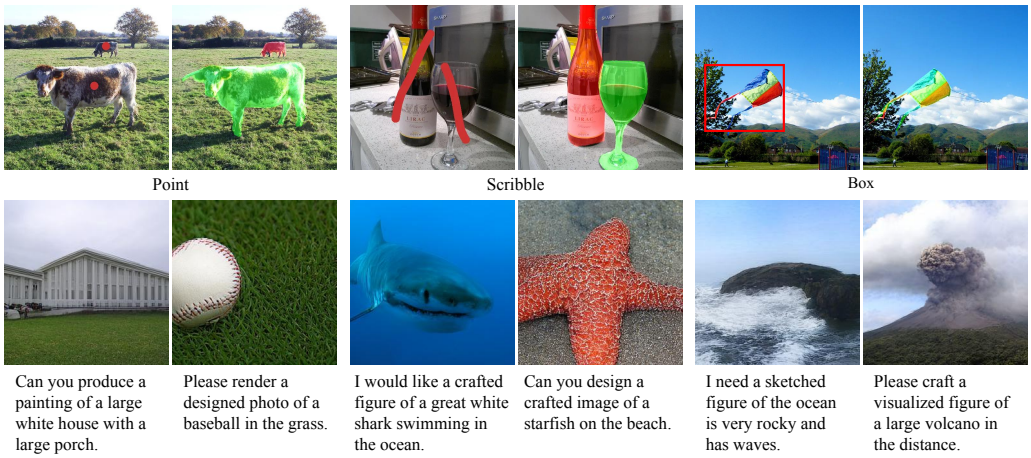


Figure 4: **Top:** Visualization of interactive segmentation. Points and scribbles are provided as visual prompts, while bounding boxes are input via text. **Bottom:** Visualization results of instruction-based image generation. The model is trained on image generation data for only $50k$ iterations.

4.5 Efficiency Analysis

We compare ARGenSeg with previous autoregressive generation models and MLLM-based segmentation methods in terms of inference time required to generate a 256×256 image or mask. All experiments are conducted using official implementations on an NVIDIA A100 GPU. Segmentation performance is evaluated using cIoU on RefCOCO-val. Detailed results are provided in Tab. 4.

Compared to sequential token generation approaches such as Emu3 [59], our parallel inference achieves more than $10\times$ speedup. While VARGPT [78] also employs VAR as its visual tokenizer,

Table 4: Computational efficiency comparison. "Num." represents the number of required tokens. Time is tested by seconds per image.

Method	Paradigm	Num.	cIoU	Time
Emu3 [59]	VQ-GAN[19]	1024	-	59.4
VARGPT [78]	VAR + Vis.Dec	680	-	2.64
PixelLM [45]	Query + Seg.Dec	6	73.0	0.91
HiMTok [57]	Mask Tokenizer	32	81.1	1.89
ARGenSeg	VAR Tokenizer	680	82.2	1.28

Table 5: Ablation study on the impact of understanding capability, pretraining-stage, and generation projector on segmentation performance.

Experiment	Ref.	Ref.+	Ref.g	Average
Baseline	82.2	77.9	78.4	79.5
Only-Seg.	80.5	73.8	73.2	75.8
Gen Projetctor	80.5	73.7	73.4	75.9
Pretrain	80.8	74.9	74.1	76.6

Table 6: Ablation study on the visual tokenizer. All results are reported on the val splits, using gIoU (per-sample IoU averaged over the dataset) as the segmentation metric.

Tokenizer Type	Prediction	RefCOCO	RefCOCO+	RefCOCOg	Average	Time
Single-scale	next-token	82.1	71.8	65.8	73.23	5.50 s
Multi-scale	next-scale	80.5	76.7	70.4	75.87	1.28 s

our method is approximately $2\times$ more efficient, due to its simplified architecture. In contrast to VARGPT, our model directly uses the classification head to predict token IDs from the VAR codebook, eliminating the need for an additional transformer-based visual decoder. PixelLM [45], a identifier-based approach, uses only six tokens and a dedicated segmentation decoder, making it slightly faster than ARGenSeg. However, its segmentation performance is significantly lower. While HiMTok [57] employs a dedicated mask tokenizer to achieve notable segmentation performance using only 32 visual tokens for efficiency, our method achieves superior performance while offering a clear advantage in inference speed.

4.6 Ablation Study

Ablation on Understanding Data We compare our baseline, fine-tuned on both understanding and segmentation data, against a counterpart trained solely on segmentation data. As shown in Tab. 5, incorporating understanding data significantly improves performance on reasoning-based segmentation, particularly on the semantically challenging RefCOCO+/g dataset. This highlights the value of unifying segmentation with a multimodal large language model.

Ablation on Model Architecture and Training Strategy We analyze the effects of model architecture and training strategy. First, to ablate the architecture, we replace our default single-layer generation projector with a two-layer variant. Results indicate that the simpler design is sufficient. Second, to assess the training strategy, we introduce a pre-training phase where only the generation projector is trained, followed by a full fine-tuning stage. As shown in Tab. 5, this two-stage approach offers only marginal gains on RefCOCO+/g and little impact on RefCOCO, while increasing training complexity. Therefore, for efficiency, our final model adopts a direct, single-stage fine-tuning strategy.

Ablation on Visual Tokenizer We ablate our multi-scale visual tokenizer by comparing it against a single-scale tokenizer, for which we adopt the pre-trained VQ-GAN [59] from Janus [61]. As shown in Tab. 6, using multi-scale scheme not only demonstrates a clear speed advantage but also improves robustness through its inherent coarse-to-fine refinement process. Further ablations, including an analysis of using semantic embeddings instead of visual tokens, are provided in Appendix D.

5 Conclusion

In this paper, we present ARGenSeg, a unified framework that integrates image segmentation into multimodal large language models through an image generation paradigm. To address the unique challenges of segmentation, we design the framework so that the MLLM directly outputs image tokens for pixel-level accuracy and utilizes multi-scale image generation for high responsiveness and robustness through coarse-to-fine refinement. Our experiment results are the first to show that unified MLLM models can perform state-of-the-art segmentation without any extra task-specific segmentation heads, providing an effective technical pathway for unified AGI.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Inclusion AI, Biao Gong, Cheng Zou, Chuanyang Zheng, Chunluan Zhou, Canxiang Yan, Chunxiang Jin, Chunjie Shen, Dandan Zheng, Fudong Wang, et al. Ming-omni: A unified multimodal model for perception and generation. *arXiv preprint arXiv:2506.09344*, 2025.

[3] Inclusion AI, Biao Gong, Cheng Zou, Dandan Zheng, Hu Yu, Jingdong Chen, Jianxin Sun, Junbo Zhao, Jun Zhou, Kaixiang Ji, et al. Ming-lite-uni: Advancements in unified architecture for natural multimodal interaction. *arXiv preprint arXiv:2505.02471*, 2025.

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[6] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

[7] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[10] Jie Cao and Jing Xiao. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th international conference on computational linguistics*, pages 1511–1520, 2022.

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

[12] 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*, pages 323–340. Springer, 2024.

[13] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[14] 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, 2024.

[15] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pages 24185–24198, 2024.

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

[17] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[18] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehension. *arXiv preprint arXiv:1710.10723*, 2017.

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[20] Yuying Ge, Sijie Zhao, Ziyun Zeng, Yixiao Ge, Chen Li, Xintao Wang, and Ying Shan. Making llama see and draw with seed tokenizer. *arXiv preprint arXiv:2310.01218*, 2023.

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[22] Ziyuan Huang, DanDan Zheng, Cheng Zou, Rui Liu, Xiaolong Wang, Kaixiang Ji, Weilong Chai, Jianxin Sun, Libin Wang, Yongjie Lv, et al. Ming-univision: Joint image understanding and generation with a unified continuous tokenizer. *arXiv preprint arXiv:2510.06590*, 2025.

[23] 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.

[24] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2018.

[25] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.

[26] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022.

[27] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[28] 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.

[29] 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*, pages 9579–9589, 2024.

[30] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[31] 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.

[32] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023.

[33] 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*, pages 26296–26306, 2024.

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[35] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18653–18663, 2023.

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

[37] 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*, pages 11–20, 2016.

[38] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[39] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019.

[40] OpenAI. Chatgpt. `https://chat.openai.com/`, 2023. Accessed: 2023.

[41] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

[42] 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*, pages 14076–14088, 2024.

[43] Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6305–6315, 2024.

[44] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024.

[45] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[47] 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*, pages 8317–8326, 2019.

[48] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

[49] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[50] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[51] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.

[52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[53] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[54] 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.

[55] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[56] 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.

[57] Tao Wang, Changxu Cheng, Lingfeng Wang, Senda Chen, and Wuyue Zhao. Himtok: Learning hierarchical mask tokens for image segmentation with large multimodal model. *arXiv preprint arXiv:2503.13026*, 2025.

[58] 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.

[59] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[60] 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.

[61] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[62] 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.

[63] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. In *Forty-first International Conference on Machine Learning*, 2024.

[64] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[65] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3858–3869, 2024.

[66] Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, et al. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer, 2025.

[67] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[68] 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 *ECAI 2024*, pages 618–625. IOS Press, 2024.

[69] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. An improved baseline for reasoning segmentation with large language model. *CoRR*, 2023.

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

[71] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[72] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.

[73] 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, 2024.

[74] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14227–14238, 2024.

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

[76] 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*, pages 633–641, 2017.

[77] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

[78] Xianwei Zhuang, Yuxin Xie, Yufan Deng, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. Vargpt: Unified understanding and generation in a visual autoregressive multimodal large language model. *arXiv preprint arXiv:2501.12327*, 2025.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The contents in the abstract and Instruction clearly reflect the contribution of the paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We analyzed the computational efficiency and included the Limitations subsection in the appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: We do not have theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We clearly explain the structure, dataset and training details of the method.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The data and the underlying model used in this article have provided detailed information and are publicly accessible.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The details of the training and test have been explained in the Experiment section of the text.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the resource limitation, we do not report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The platform and running time for the experiment are given in the main paper.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research in this article is in line with NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss societal impact of the work in the appendix.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [No]

    Justification: We do not plan to release any data. For the model, we currently do not have safeguards for releasing it. We will make sure that the guidelines and instructions are in place when we release the model.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes, we have.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [NA]

    Justification: No new assets are provided.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA] .

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix of ARGenSeg

A Implementation Details

Datasets The datasets used for image segmentation, multimodal understanding, and image generation are listed in Tab. 7. To ensure a fair comparison, we exclusively use subsets of the data employed by the previous state-of-the-art method, HiMTok [57]. Specifically, we train on 402K segmentation samples compared to HiMTok's 2.91M, and 1.25M multimodal understanding samples compared to HiMTok's 4.2M. Image generation data are used only in the optional function-extension stage.

Table 7: Training data used in our experiments.

Task	Datasets
Image Segmentation	ADE20K(20K) [76], COCO-Panoptic(118K) [31], gRefCOCO (79K) [32], RefCOCO/+/g(127K) [37, 70], LISA++ Inst.Seg(58K) [69]
Multimodal Understanding	AI2D [25], ChartQA[38], COCO-Text[55], DocVQA[18], LLaVA-150K[34], GQA[23], DVQA[24], OCR-VQA[39], TextVQA[47], SynthDoG-EN [26], InternVL-SA1B-Caption [14], VisualGenome [28], GeoQA+[10]
Image Generation	ImageNet-Instruct-class [78], ImageNet-Instruct1270K [78]

Inference Details During inference, we get visual outputs exclusively from the logits corresponding to visual tokens in the MLLM codebook. This constraint ensures compatibility with the visual tokenizer and enables successful reconstruction of the image. For image segmentation tasks, we adopt a deterministic **argmax** sampling strategy to obtain the predicted visual tokens. For image generation tasks, we apply classifier-free guidance (CFG) to compute the output distribution over visual tokens, followed by **top-k** sampling to enhance the diversity and quality of generated images.

B Additional Qualitative Results

Multi-scale Image Generation We provide visualization of segmenting similar objects in the same image using different instructions, as shown in Fig.5. From the multi-scale mask generation process, it is evident that our model can correctly understand and localize the target based on the given instructions. The ability to correctly follow distinct segmentation commands indicates that ARGenSeg possesses a robust understanding of both spatial positions and semantic relationships.

Comparison with Single-scale Generation We compare our method with HiMTok [57], treating it as a representative single-scale generative segmentation approach. We conducted a thorough evaluation on the test set and visualized cases where ARGenSeg succeeds while HiMTok fails. As shown in Fig. 6, these cases reveal two primary advantages of our coarse-to-fine, multi-scale generation scheme: **(1) Robust Target Identification in Multi-object Scenarios.** The initial coarse localization stage effectively identifies the target object even when multiple similar objects are present. **(2) Enhanced Mask Quality through Progressive Refinement.** Following target identification, the multi-scale refinement process progressively improves mask precision for higher-quality segmentation. For instance, in the case of a partially occluded teddy bear, both HiMTok and our coarse localization stage initially segment only a visible part. However, our model's subsequent fine-grained refinement successfully reconstructs the entire object while correctly excluding the occluder.

C Additional Quantitative Results

C.1 Performance on Multimodal Understanding

We further assess the multimodal understanding capabilities of ARGenSeg. As shown in Tab. 8, the inclusion of segmentation data does not cause the model to lose its reasoning capability. While we

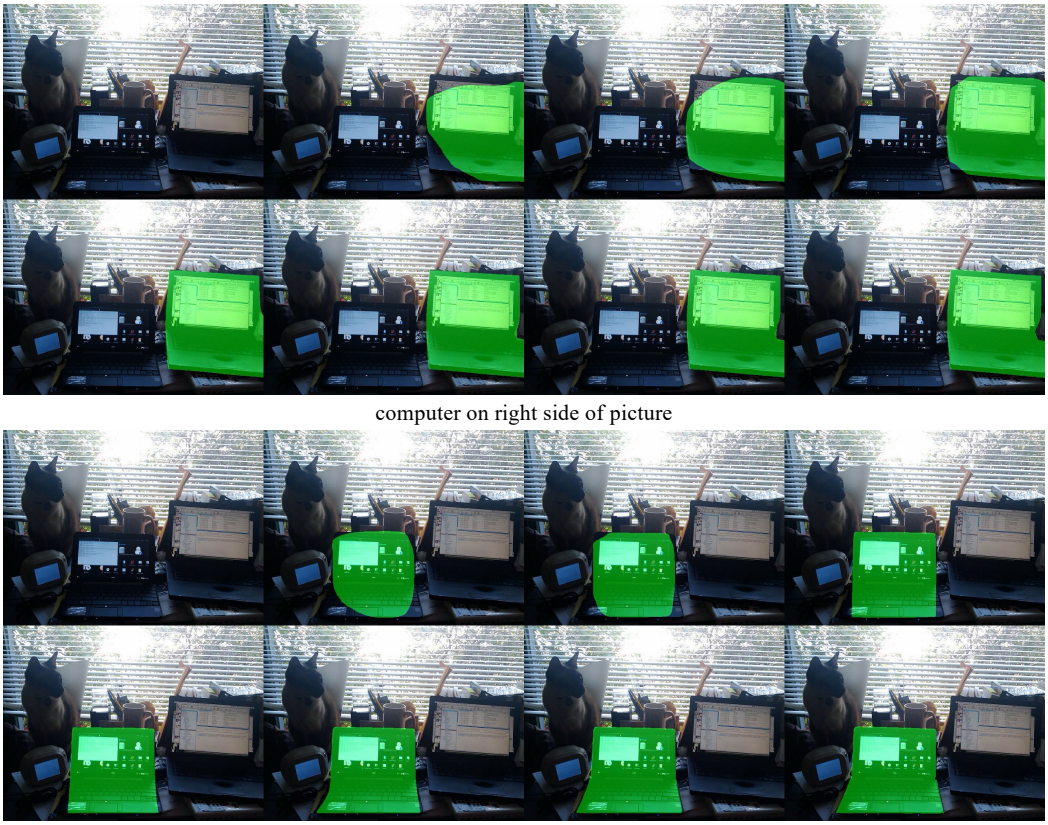


Figure 5: Visualization of using different segmentation instructions in the same image.

Figure 6: Comparison between multi-scale and single-scale generative segmentation approach. The examples highlight scenarios where the multi-scale approach excels.

observe slight performance drops on some benchmarks, we attribute this minor degradation not to the segmentation task itself, but to the significantly smaller and lower-quality understanding corpus used for fine-tuning (1.25M vs. the 16.3M samples used for InternVL-2.5 [13]). To validate this hypothesis, we conducted a control experiment: fine-tuning InternVL-2.5 solely on the same understanding data for an increasing number of steps. The performance declined monotonically, mirroring the trend observed with joint segmentation training and thus confirming our attribution.

Table 8: Multimodal understanding results across benchmarks.

Method	POPE	TextVQA	VQAv2	MMMU-val	AI2D
InternVL2.5-ft-1ep	86.73	63.54	80.40	43.7	78.7
InternVL2.5-ft-4ep	86.01	59.73	79.28	36.8	74.8
ARGenSeg	87.57	56.98	77.87	33.4	69.6

C.2 Results on Interactive Segmentation

To ensure a fair comparison with HiMTok, which was not trained on interactive-segmentation data, we omitted this task from our main experiments. Here, we evaluate our model on the COCO-Interactive benchmark [75], reporting the cIoU metric. It is worth noting that while PSALM [75] was fine-tuned for 10 epochs according to its official implementation, our model is fine-tuned for only a single epoch due to computational constraints. As shown in Tab. 9, ARGenSeg significantly outperforms SAM [27] in interactive segmentation. Moreover, it achieves performance comparable to PSALM with substantially less fine-tuning, which underscores the strong generalization capabilities of our model.

Table 9: Quantitative results on interactive segmentation. The results for SAM and PSALM are sourced directly from the PSALM paper.

Method	Point	Scribble	Box
SAM-B [27]	33.6	–	68.7
SAM-L [27]	37.7	–	71.6
PSALM [75]	74.0	80.0	80.9
ARGenSeg	65.6	68.6	79.1

D Additional Ablation Studies

Table 10: Ablation study of MLLM backbones and image generation strategies. The segmentation performance is measured in cIoU.

Method	Backbone	Generation Strategy	RefCOCO	RefCOCO+	RefCOCOg
HiMTok [57]	InternVL-2.5	Single-scale VQ	81.1	77.1	75.8
ARGenSeg-LLaVA	LLaVA–1.5	Multi-scale VQ	72.7	68.3	69.1
ARGenSeg-InternVL	InternVL-2.5	Multi-scale VQ	**82.2**	**77.9**	**78.4**
ARGenSeg-DiT	InternVL-2.5	Diffusion Head	59.0	62.7	64.1

D.1 Ablation on MLLM Backbone

Our approach, which integrates a VQVAE codebook into the MLLM's token space, is designed to be model-agnostic. To demonstrate this portability, we replaced the default InternVL-2.5 backbone with LLaVA-1.5 [33], a LLaMA-2-based MLLM. As shown in Tab. 10, our pipeline successfully imparts segmentation capabilities to LLaVA-1.5.

As established in Sec. 4.6, referring segmentation performance is highly correlated with the MLLM's underlying understanding ability. Consequently, given LLaVA-1.5's weaker understanding capabilities compared to InternVL-2.5, the resulting segmentation performance is expectedly lower. Nevertheless, Tab. 10 shows that with the same powerful InternVL-2.5 backbone, our method outperforms HiMTok. This confirms that our performance gains are inherent to our approach and not merely a byproduct of a stronger backbone.

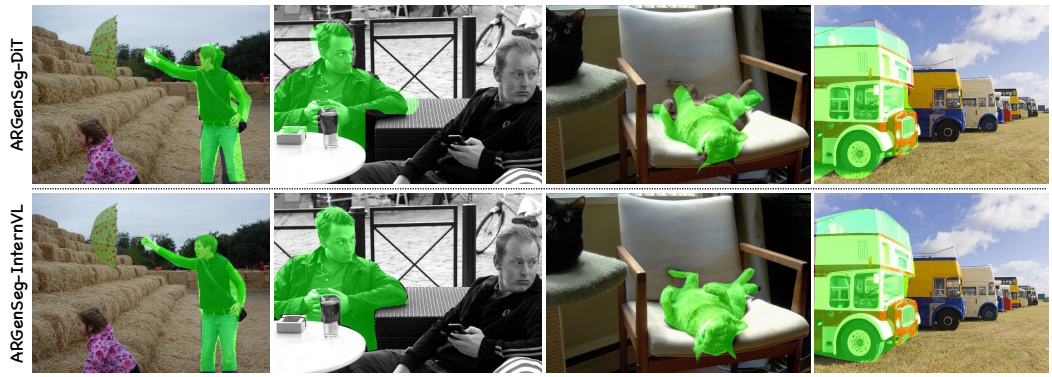


Figure 7: Comparison between direct visual token generation and DiT-based generation. The DiT-based approach, which uses semantic embeddings from the MLLM, struggles with pixel-level accuracy, leading to artifacts like spatial shifts and imprecise boundaries.

D.2 Ablation on Image Generation Strategy

To further validate our choice of generation strategy, we explore an alternative approach where the MLLM outputs semantic embeddings to a separate diffusion head (DiT) for segmentation, inspired by MetaQuery [41]. Specifically, we configure the MLLM to generate learnable queries, which are then mapped to the feature space of the pre-trained SANA-1.5 1.6B [66] via a connector module.

This alternative strategy, labeled as ARGenSeg-DiT in Tab. 10, led to a severe performance degradation. As show in Fig. 7, while the model could roughly localize the target region, the generated masks suffered from significant artifacts, such as spatial shifts and inflation, indicating poor pixel-level accuracy. This experiment underscores the importance of the MLLM directly generating discrete image tokens to maintain the high pixel-level precision crucial for segmentation tasks.

E Limitations

This paper proposes a novel image segmentation paradigm based on autoregressive image generation, integrating multimodal understanding, generation, and image segmentation into a unified framework. Our model demonstrates strong performance across a range of segmentation tasks, and further shows the potential to extend to more complex scenarios, such as interactive segmentation and text-to-image generation. The unified framework also shows promise for expanding to broader tasks, such as image editing and depth estimation. However, due to resource constraints, exploring these extensions is beyond the scope of this work, and we consider them as promising directions for future research.

F Broader Impacts

This work contributes to the development of unified multimodal frameworks by integrating dense image segmentation into the unified multimodal understanding and generation models. The proposed framework may inspire future research toward more generalizable, modular, and efficient visual-language models that require fewer task-specific components. Potential applications include human-robot interaction, assistive vision systems, and real-world visual understanding under low supervision. However, like most large-scale models, ARGenSeg may inherit biases from pre-trained components or datasets. Care should be taken to evaluate fairness and robustness when deploying it in real-world scenarios, especially in sensitive domains such as healthcare or surveillance.