DENSE SEGMENTATION AS FIRST-ORDER OUTPUT OF VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive vision-language models (VLMs) exhibit fine-grained spatial reasoning when describing images in words. However, producing the corresponding segmentation masks usually requires external segmentation models or learned mask decoders. This work introduces FOMO (*first-order mask output*) a geometry language that makes dense segmentation a native autoregressive output. FOMO's deterministic encoder uses adaptive quadtrees and pre-defined geometry stencils to describe binary masks as token sequences. Homogeneous regions are compactly encoded with only a few tokens, while additional tokens are allocated to complex boundary regions. FOMO sequences generated by VLMs deterministically form a mask without neural decoders. Theoretical proofs and empirical experiments demonstrate the codec's near-exact reconstruction, a Spearman correlation of $\rho = 0.997$ between token budget and mask quality, and graceful degradation under most grammar-preserving token corruptions. Moreover, with the FOMO vocabulary extension and fine-tuning via low-rank adaptation, Qwen3.5-9B learns to generate masks for referring-expression segmentation queries without requiring any changes to its vision encoder or added external modules. Our method achieves 76.0%, 71.7%, and 71.7% cumulative IoU for RefCOCO val, RefCOCO+ val, and RefCOCOg val, using fixed geometric reconstruction without an external segmentation model or learned mask decoder.

1 INTRODUCTION

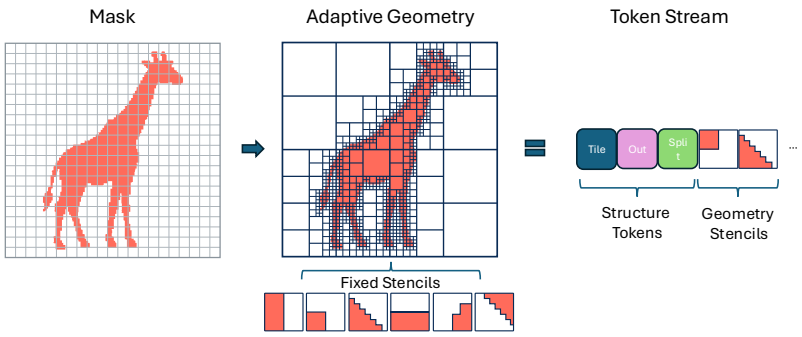


Figure 1: Our proposed geometry language represents binary masks via adaptive quadtrees that describe the mask's boundary at various scales using a pre-defined set of geometry tokens (stencils).

Recent work suggests that autoregressive vision-language models (VLMs) encode fine-grained spatial information in their internal image representations (Ranasinghe et al., 2024; Kang et al., 2025). Yet, VLMs struggle to express this information via their discrete output token stream. Modern VLMs support grounding through bounding boxes or points (Qwen Team, 2026; GLM-5V-Team et al., 2026). For a request such as "segment the taller giraffe", these outputs do not specify a pixel-level mask. Returning a more accurate pixel-level mask would answer the query more faithfully and enable advanced downstream tasks, such as image editing, compositing, or feature extraction.

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.

We seek an efficient interface or "geometry language" for VLMs to represent pixel-level segmentation masks. Such a language would serialize dense segmentation masks as a stream of tokens while fulfilling three core desiderata:

- **Compactness.** Compared to natural images, binary segmentation masks commonly contain large homogeneous regions with low information entropy. Sequence length should scale with the mask's geometric complexity rather than image resolution. Large homogeneous regions should be encoded via a few tokens, while additional tokens should only be allocated to define complex boundaries.
- **Robustness.** Autoregressive generation is intrinsically noisy, and the resulting token stream may contain erroneous tokens. Incorrect tokens should have a limited, localized effect rather than corrupting the entire encoded mask, and the decoder should continue to produce a valid mask whose quality degrades gracefully even as generation errors accumulate.
- **Native generation.** Mask tokens should be emitted directly by the VLM as a first-order autoregressive output. These tokens should then be decoded into a fully functional mask using a fixed parameter-free procedure without relying on a learned decoder. This preserves the standard next-token prediction objective and ensures that mask generation is compatible with the same conversational conditioning, instruction-tuning, and post-training machinery used for text.

Existing strategies to extract dense segmentation masks from VLMs fail to simultaneously fulfill all three requirements. Several prior approaches serialize the mask directly as an ordered sequence of boundary coordinates (Castrejon et al., 2017; Zhu et al., 2022; Liu et al., 2023b; Pramanick et al., 2024; Chen et al., 2022), but such sequences are neither robust nor compact. Most approaches delegate mask generation to a separate module, relying on promptable external segmentation models (Chen et al., 2024; Ren et al., 2026; Lan et al., 2025; 2026), trainable pixel decoders operating on latent representations (Lai et al., 2024; Ren et al., 2024; Rasheed et al., 2024; Wu et al., 2025; Zhang et al., 2024c;b;a), or learned codecs that translate a tokenized representation back into a mask (Wang et al., 2025b; Deng et al., 2026; Wang et al., 2025a;c; Zheng et al., 2025; Zhou et al., 2026). In each case, the mask is not a native output of the VLM itself, and its quality and computational cost strongly depend on the added module. In this work, we show that it is possible to do with deterministic encoding and reconstruction, without a neural mask encoder or decoder. By carefully choosing how geometry is translated into tokens, the referring expression task can be learned in its simplest form, as a rule-based language with a grammar, fully interpretable like a natural language.

We propose a novel geometry language, implemented as a deterministic codec that converts masks into tokens and decodes tokens back into masks. Our codec combines established concepts from image compression, while avoiding the use of external modules, such as segmentation models or learned tokenizers. At its core, our method represents the mask's boundary using a quadtree that recursively allocates additional tokens to describe complex parts of the boundary, while only requiring few tokens for simple regions (Fig. 1). The resulting language – called *FOMO* for first-order mask output – can compactly describe complex geometries, is robust against localized token errors, and can be generated by VLMs as a first-order output. Specifically, we make the following contributions:

- **A geometry language for masks.** Our FOMO language encodes binary masks as discrete tokens in the same representation space as text and decodes them back to full-resolution masks without requiring any external segmentation tools or learned modules.
- **Theoretical and empirical codec validation.** We formally prove universal expressivity, perimeter-bounded token counts, and conditional bounds on reconstruction errors. In empirical experiments, we demonstrate near-exact reconstruction with a Spearman rank correlation of $\rho = 0.997$ between token budget and mask quality, and graceful degradation under most grammar-preserving token corruptions.
- **Native segmentation with a pretrained VLM.** With a vocabulary extension and LoRA fine-tuning, Qwen3.5-9B learns our geometry language using 129M trainable parameters (1.4% of the base model). We compare FOMO with 18 baseline configurations across RefCOCO, RefCOCO+, and RefCOCOg. Its reported scores exceed those of several baselines under their respective protocols, while trailing the strongest methods.

2 Related Work

Methods that produce segmentation masks from VLMs differ in how the target geometry is represented and at which stage the pixel-level computation occurs.

Masks as sequences of coordinates. Polygon-RNN, SeqTR, PolyFormer, VistaLLM, SimpleSeg and Pix2Seq (Castrejon et al., 2017; Zhu et al., 2022; Liu et al., 2023b; Pramanick et al., 2024; Song et al., 2026; Chen et al., 2022) serialize masks as vertices or boundary points and generate this sequence auto-regressively. FOMO shares this sequence-generation formulation, but replaces vertex lists with hierarchical local boundary descriptions designed for compactness and localized effects of token errors.

VLM + external segmentation tool. Several approaches use the VLM only for reasoning and grounding, and delegate mask generation to an off-the-shelf segmentation tool, such as SAM (Kirillov et al., 2023). SAM4MLLM or GETok (Ren et al., 2026; Chen et al., 2024) iteratively generate segmentation prompts based on points, bounding boxes, and basic masks that are forwarded to SAM together with the input image. LISA (Lai et al., 2024) uses special segmentation tokens to prompt a separate mask predictor. The reliance on external tools results in the generated mask no longer being a first-order output of the VLM. This makes end-to-end training significantly more complex, and interleaving segmentation with conversational output requires re-tokenization. Crucially, any spatial reasoning may originate from the external module processing the input image, instead of being aligned with the VLM's internal reasoning.

Mask prediction from VLM features. Several methods predict masks from continuous VLM features using learned segmentation modules (Lai et al., 2024; Ren et al., 2024; Rasheed et al., 2024; Wu et al., 2025; Zhang et al., 2024c;a; 2026b;a). Our method instead emits explicit geometry tokens whose reconstruction uses fixed rules and requires no access to latent image features. Similar to methods that use external segmentation tools, the mask is no longer a native first-order autoregressive output. Its quality and the required compute are dictated by the added pixel decoder and its training signal.

Masks as tokenized sequence. A more recent line of work expresses masks as discrete entries to a pre-defined codebook. VLMs can subsequently be trained via the standard next-token prediction objective to process and output such "mask languages." Text4Seg (Lan et al., 2025) encodes coarse mask grids using row-wise RLE and optionally refines them with SAM. Text4Seg++ (Lan et al., 2026) uses box-wise run-length tokens on a finer grid, supporting direct mask output. Both quantize boundaries on uniform grids. FOMO instead refines its geometric representation adaptively near complex boundaries.

Learned visual tokenizers and codebooks provide an alternative approach. UViM, All in Tokens, and PaLI-3 (Kolesnikov et al., 2022; Ning et al., 2023; Chen et al., 2023) established that short discrete codes can be expanded into dense visual outputs by a trained network. Recent mask-specific systems include HiMTok, LlamaSeg, ALTo, ARGenSeg, and Seg-VAR (Wang et al., 2025b; Deng et al., 2026; Wang et al., 2025a;c; Zheng et al., 2025). These approaches vary in sequence length and generation order, but all rely on a learned dense reconstruction mapping. ALTo and SAMTok (Wang et al., 2025a; Zhou et al., 2026) additionally condition this mapping on the image. FOMO shares their discrete output interface, but gives tokens fixed geometric meanings and reconstructs masks without a learned de-tokenizer.

3 Geometry Language Design and Properties

The codec deterministically encodes binary masks and decodes complete grammatical streams, independently of how the tokens are predicted. Given a fixed bank of geometric shapes, which we call stencils, the encoder maps a binary mask to a compact token stream of stencil tokens and structural tokens that describe where the stencils are placed (see Fig. 2). A VLM can learn to output the language with ordinary next-token training, so that the decoder can assemble the token stream back into a mask.

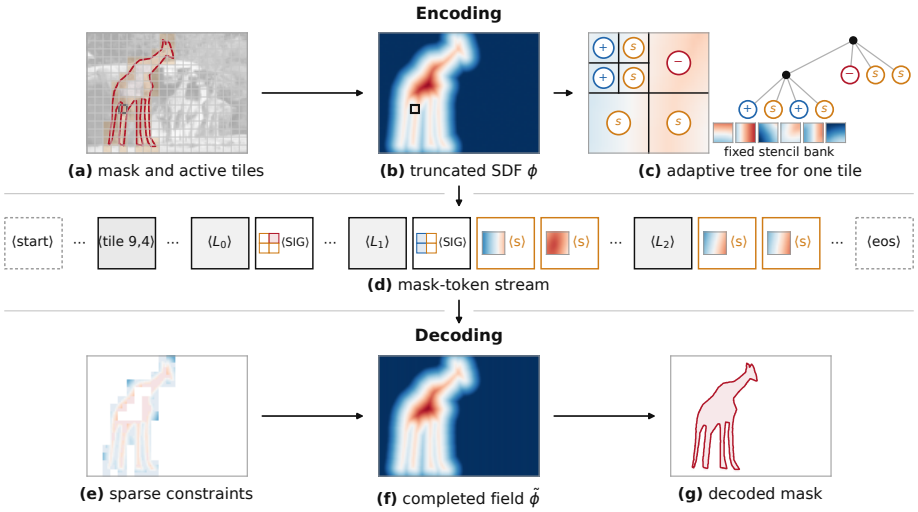


Figure 2: Boundary-aware mask codec. Encoding (a–d): the mask is resized and divided into tiles (a), and converted to a TSDF (b). A quadtree describes each boundary tile with sign decisions or fixed-bank stencils (c), and all trees are serialized breadth-first (d). Decoding (d–g): the tokens yield sparse stencil and sign constraints (e), before Laplacian inpainting produces a dense TSDF (f), which is finally converted into a mask (g).

3.1 Encoding: Binary Mask to Tokens

Representing masks as truncated signed-distance fields (Fig. 2a,b). At its core, our method aims to split a binary mask into local tiles and describe them via a set of predefined geometric shapes called stencils. Thus, it requires a similarity measure to compare tiles and stencils. For binary masks, naively calculating their pixel-wise difference does not directly measure how far the corresponding mask boundaries are displaced. Taking heed from the field of shape analysis (Delfour & Zolésio, 2005), we therefore convert the binary mask m into a truncated signed-distance field (TSDF) Φ (Curless & Levoy, 1996). Comparing two TSDF via their ℓ_2 distance can be an efficient approximation for their boundary displacement.

Using $d(p, A) = \min_{a \in A} \|p - a\|_2$ to denote the closest distance between a given point p and a region A, which can be either the foreground $A_f = \{p \mid m(p) = 1\}$ or the background $A_b = \{p \mid m(p) = 0\}$, the TSDF is defined as $\phi(p) = \text{clip}(d(p, A_f) - d(p, A_b), [-T, T])$. The resulting field's magnitude denotes the distance to the nearest pixel with the opposite label, capped at T pixels.

The TSDF field is partitioned into $S \times S$ tiles that align with the VLM's image tokenization grid. For a given tile, let ϕ be its local TSDF, and $\phi_{\min}$ and $\phi_{\max}$ be the TSDF's minimum and maximum values. A tile is deemed active exactly when $\phi_{\min} \leq B_{\text{tile}}$ and $\phi_{\max} \geq -B_{\text{tile}}$ with a scalar threshold B_{tile}. Active tiles describe the regions in the neighborhood of the mask's boundary that possess the most information entropy. In the following, only active tiles are encoded and decoded, while the remaining inactive tiles are implicitly represented.

Describing the mask via a forest of quadtrees (Fig. 2c). Each active tile forms the root of a quadtree. Subsequently, each node of the quadtree is assigned one of four types (Algorithm 1). If the tile lies inside the mask, determined via $\phi_{\max} < -B_{\text{leaf}}$, the node becomes an `IN` leaf. Conversely, if the tile lies outside the mask, determined via $\phi_{\min} > B_{\text{leaf}}$, the node becomes an `OUT` leaf.

If the tile contains any point close to the boundary, controlled by a scalar threshold B_{leaf}, the encoder attempts to represent it using stencils from a predefined stencil bank $\mathcal{B}$. Each stencil describes a local TSDF instead of an abstract latent code. An acceptance test with tolerance δ tests if the stencil matches the tile's boundary with sufficient accuracy. If the acceptance test passes, the node becomes the corresponding `STENCIL#` leaf. Stencil bank generation, stencil matching, and the acceptance test are described in more detail below.

Algorithm 1 Decision making at each quadtree node

Require: Local TSDF ϕ of side length s, depth d, stencil bank $\mathcal{B} = \{b_k\}_{k=0}^{M-1} \subset [-1,1]^{R\times R}$, and parameters $D, T, B_{\text{leaf}}, B_{\text{accept}}, q, \delta$
$\mathcal{J}_s$ keeps the smallest index among stencils whose rendered $s \times s$ fields coincide

1: **if** $\min \phi > B_{\text{leaf}}$ **then return** `OUT`
2: **if** $\max \phi < -B_{\text{leaf}}$ **then return** `IN`
3: $\phi' \leftarrow \text{resize}_{R\times R}\big(\text{clip}(\phi/s, [-1,1])\big)$ ▷ normalize to $R \times R$
4: $j \leftarrow \arg\min_{k\in\mathcal{J}_s} \|\phi' - b_k\|_2$
5: **if** $d = D$ **or** $\text{accept}(\phi', b_j, s)$ **then return** $\texttt{STENCIL}_j$
6: **return** SPLIT into four quadrant patches at depth $d{+}1$

Otherwise, the node becomes a `SPLIT` node. The corresponding tile is partitioned into four quadrants, which become the node's children. This process is applied recursively up to a maximum quadtree depth D, at which point the node becomes the best matching `STENCIL#` without conducting an acceptance test.

Stencil bank generation. We create our stencil bank by mining canonical boundary patches from training masks, clustering them into 512 prototypes with mini-batch k-means, and applying four rotations and an optional horizontal flip. The resulting bank of 4096 stencils is frozen before VLM training. Unlike a latent codebook, however, each entry directly supplies an interpretable numeric TSDF patch, obviating the need for any learned tokenizer.

Stencil matching across quadtree depths. Any quadtree will describe tiles of different sizes at its various depths. In order to compare these tiles against a single stencil bank, we convert their local TSDF ϕ to a common $R \times R$ representation. For a node of side length s, we compute $\phi' = \text{resize}_{R\times R}\big(\text{clip}(\phi/s, [-1,1])\big)$, using bilinear interpolation. Stencil matching is conducted by selecting the best matching stencil b^* via $j = \arg\min_{k\in\mathcal{J}_s} \|\phi' - b_k\|_2, \qquad b^* = b_j.$

Acceptance test. After identifying the best matching stencil b^*, an acceptance test checks whether it approximates the target boundary with sufficient accuracy. We compare the tile's TSDF and the best matching stencil within a narrow band i around the target boundary that satisfies $|\phi_i| \le B_{\text{accept}}$. At each selected location, we compute the error in pixels as $e_i = s|\phi'_i - b^*_i|$. The factor s converts the normalized differences to pixel units, so the same tolerance δ applies at every depth. We accept the stencil if the 95th percentile of the checked errors is at most δ. The percentile-based error calculation prevents single outliers from forcing excessive splitting.

Coarse-to-fine serialization (Fig. 2d). To obtain a sequence for next-token prediction, we serialize the quadtree forest breadth-first across all active tiles. This orders the mask description from coarse structure to finer detail. The stream begins with `START`, `TILEMAP`, and a strictly increasing row-major list of active-tile identifiers $\texttt{TILE}_i$. Starting with the active-tile roots, each depth is introduced by a $\texttt{DEPTH}_d$ marker followed by one token per queued node. Each queued node emits either a stencil token or one of $3^4 = 81$ *child-signature* tokens. A child signature records the four quadrant states in top-left, top-right, bottom-left, bottom-right order. Each child is classified as outside, inside, or unresolved. The outside and inside states represent the `OUT` and `IN` leaves of Algorithm 1 and require no separate tokens. Unresolved children enter the next depth queue, where each receives its own stencil or child-signature token. As a result, the early parts of a sequence already sketch a coarse segmentation map that is further refined as the sequence grows with increasing depth level.

3.2 Decoding: tokens to binary mask

From tokens to TSDF constraints (Fig. 2e). Decoding generates three maps, which are subsequently used to decode the TSDF $\hat{\Phi}$, by parsing the breadth-first serialized sequence. $\texttt{TILE}_i$ tokens localize and initialize the quadtree roots. Each signature token inserts its inside and outside quadrants into a sign map $\sigma \in \{-1, 0, 1\}^{H\times W}$ with -1 denoting inside quadrants, $+1$ denoting outside quadrants, and 0 denoting pixels without either constraint. Next, the sign map is used to initialize a first estimate of the TSDF $\hat{\Phi}^{(0)} \in [-T, T]^{H\times W}$, which is further refined by inserting the local TSDF of the individual stencils. Simultaneously, we obtain a binary indicator map $\chi \in \{0, 1\}^{H\times W}$

that is 1 in regions that have been defined by either a stencil or inside/outside quadrants, and is 0 in undefined regions.

Completing the mask (Fig. 2f,g). We initialize each four-connected component of unknown pixels $\chi(p) = 0$ from its adjacent observed values. If these values are all strictly negative or positive, the component is initialized to $-T$ or $+T$.Otherwise, if at least 30% of them are negative and at least 30% nonnegative, each pixel takes $-T$ if the nearest of these values is negative and $+T$ otherwise. Else we use normalized multiscale bilinear interpolation. With no observations, we use zero. Clipping the resulting field to $[-T, T]$ gives $\hat{\Phi}^{(0)}$. We then refine it with n synchronous four-neighbor updates, keeping χ fixed. $\hat{\Phi}^{(k+1)}(p) = \frac{\chi(p)\hat{\Phi}^{(0)}(p)+0.2\sum_{p'\in\mathcal{N}_4(p)}\hat{\Phi}^{(k)}(p')}{\chi(p)+0.8}$ Here, $\mathcal{N}_4(p)$ denotes the four neighboring pixels. After each update, we clip $\hat{\Phi}$ to $[-T, -10^{-3}]$ where $\sigma = -1$, to $[10^{-3}, T]$ where $\sigma = +1$, and to $[-T, T]$. This constrains the Laplacian inpainting to adhere to the initial sign tokens. The final mask is $\hat{m}(p) = 1$ if $\hat{\Phi}^{(n)}(p) < 0$, and 0 otherwise.

3.3 Theoretical properties

We analyze the geometry language and codec independently of VLM training, distinguishing representational capacity, encoder token counts, and decoder guarantees. Let L count horizontally or vertically adjacent pixel pairs with different target labels within the image, counting each pair once. In the following we assume an *idealized instantiation* of the codec. Formal statements and proofs are given in Sec. C.

(1) Universal expressivity. On a fixed pixel grid, every binary mask has an exact representation when the language permits $D \geq \log_2 S - 1$ and sufficient sequence length, for a power-of-two tile side S. A legal stream can declare every pixel's inside or outside label through child signatures. Initialization and projected updates then preserve the mask exactly (Prop. C.1). Our $S = 32, D = 3$, the same construction assigns each 2×2 block its majority label, changing at most $2L$ pixels.

(2) Perimeter-bounded token count. The complete stream uses $\mathcal{O}(1+L(D+1))$ tokens, provided $B_{\text{tile}} \geq 1$, $B_{\text{leaf}} \geq 0$, both bands are at most half the smallest node side, and T exceeds both bands (Prop. C.2). Under these conditions, each depth queues at most a constant times L nodes for a nonconstant mask while constant masks use three or six tokens. Our configuration satisfies these conditions. At fixed depth, the bound is linear in perimeter.

(3) Certified errors before averaging. For each stencil leaf, define its certificate ρ as the largest target distance $|\Phi(p)|$ at a wrongly labeled pixel, or zero if none exists. Choose a radius $0 \leq r < T$. Suppose every leaf has $\rho \leq r$, and stencil pixels adjacent to unobserved regions have the correct labels. With $B_{\text{tile}} \geq 1$, $B_{\text{leaf}} \geq 0$, and no averaging, errors then lie only where $|\Phi| \leq r$ (Prop. C.3). There are at most $(2\lfloor r \rfloor + 1)^2 L$ such errors, and none if $r < 1$. In practice, these conditions hold with $r = 2$ on 77.5% of 1,000 validation targets.

(4) Locality of averaging. At $T = 32$, a pixel keeps its initial sign if every pixel within four horizontal or vertical steps is observed and has that sign with magnitude at least one. This holds for any number of updates (Prop. C.4). It describes locality relative to the initialized observations, not necessarily the target boundary.

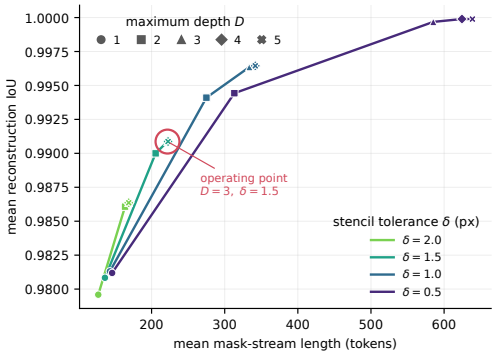
4 Experiments and Results

First, we analyze the codec in isolation. We discuss reconstruction fidelity as a function of token budget, and robustness of the deterministic decoder to token corruption (Sec. 4.1). Afterwards, we evaluate VLMs that have been trained to use our geometry token language for referring expression segmentation (Sec. 4.2).

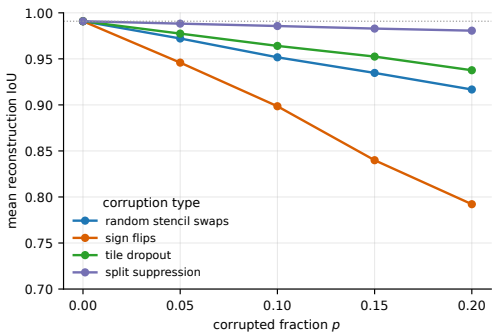
4.1 Codec properties

Token budget vs. reconstruction fidelity. We encode and reconstruct 1000 ground-truth masks from RefCOCO+ (unc, val) (Yu et al., 2016), reporting reconstruction IoU against mask-stream length. We sweep the codec's two budget parameters, maximum depth $D \in \{1, 2, 3, 4, 5\}$ and tolerance $\delta \in \{0.5, 1.0, 1.5, 2.0\}$ px (Fig. 3a). Mean stream length ranges from 127 to 639 tokens

and mean IoU from 0.9796 to 0.9999, with a near-perfect rank correlation between budget and fidelity (Spearman $\rho = 0.997$). In the following experiments, we use the mid-budget point $D = 3$, $\delta = 1.5$ px, which reconstructs at 0.991 mean IoU and 0.993 cIoU with 222 tokens per mask on average. Re-encoding the ground truth thus costs less than one IoU point, so almost all of the gap in prediction score Tab. 1 is attributable to the VLM and its ability to process our geometry language rather than the codec.



(a) Codec reconstruction quality vs. mask-stream length on RefCOCO+ val, for 5×4 combinations of maximum depth D and stencil tolerance δ. Each curve fixes a tolerance δ and sweeps D from 1 to 5. Reconstruction quality improves almost monotonically with the token budget ($\rho = 0.997$).



(b) Reconstruction quality when a fraction p of the stencils, inside/outside states, active tiles, or splits is corrupted. Sign flips are most harmful whereas split suppression has almost no effect.

Robustness to token stream corruption. Next, we test the ability of our codec to reconstruct token streams that contain erroneous tokens as this may occur during inference. To this end, streams are corrupted with grammar-preserving operators by replacing a fraction p of stencils with random other stencils, flipping declared signs inside signatures, dropping entire tiles, and suppressing splits by forcing the best stencil instead. We report the reconstruction quality as function of corruption rates $p \in \{0.05, 0.1, 0.15, 0.20\}$ (Fig. 3b). Split suppression have a minor impact lowering mean IoU by only 0.010, as coarser stencils absorb the missing detail, and tile dropout by 0.053. Comparatively, random stencil swaps lower it by 0.074, and sign flips, which relabel entire quadrants, by 0.199, respectively. These corruptions can be viewed as a worst-case scenario. In reality, a trained VLM will most likely select plausible stencils instead of random ones. On the same targets, 95% of accepted stencil positions (70% at depth 0, over 99% at depths 2 and 3) admit at least one other stencil that also passes the encoder's acceptance test, and replacing every stencil with its second-best match, or with a random one of its second- to fifth-best matches, lowers mean IoU by only 0.003 and 0.006.

4.2 Referring Expression Segmentation

We extend the tokenizer of Qwen3.5-9B (Qwen Team, 2026) with our proposed geometry vocabulary by fine-tuning the model via LoRA adapters (Hu et al., 2022) (r=32, α=64), the new token embeddings, and a geometry output head (129M trainable parameters in total), while keeping all pretrained weights, including the vision encoder, frozen. We train on 810,855 segmentation instructions drawn from RefCOCO/RefCOCO+ (Yu et al., 2016), RefCOCOg Mao et al. (2016), gRefCOCO (Liu et al., 2023a), and LVIS (Gupta et al., 2019), for three epochs using supervised fine-tuning with cross-entropy-based next-token prediction. During inference, grammar-constrained greedy decoding is used. Full training details are in the supplementary Sec. A.

We evaluate on RefCOCO and RefCOCO+ (unc splits) and RefCOCOg (umd splits) and report cumulative IoU (Tab. 1). Every generated stream decodes to a mask, as grammar-constrained generation ensures completion within the 2,048-token limit (Sec. A.4). We compare FOMO with 18 baseline configurations, grouped by how their predicted representations are converted into segmentation masks. Our supervised model reaches 76.0, 71.7, and 71.7 cIoU on the respective validation splits. Its reported scores are comparable to LISA and F-LMM on RefCOCO and exceed both on RefCOCO+ and RefCOCOg. FOMO trails the strongest decoder- and tokenizer-based systems. Cru-

Table 1: Comparison of our method with 18 baseline configurations on referring expression segmentation (cIoU, %). We organize methods by how their predicted representation is converted into a pixel-level mask. Baseline scores are taken from the respective papers and follow their own training and evaluation protocols.

	RefCOCO			RefCOCO+			RefCOCOg	
Method	val	testA	testB	val	testA	testB	val	test
Masks as coordinate sequences								
PolyFormer-L Liu et al. (2023b)	76.0	78.3	73.3	69.3	74.6	61.9	69.2	70.2
SimpleSeg-7B Song et al. (2026)	80.9	77.8	75.2	72.4	77.3	66.1	73.3	74.1
VLM + external segmentation tool								
GROUNDHOG-7B Zhang et al. (2024b)	78.5	79.9	75.7	70.5	75.0	64.9	74.1	74.6
Text4Seg (13B + SAM-H) Lan et al. (2025)	80.2	82.7	77.3	73.7	78.6	67.6	74.0	75.1
SAM4MLLM-7B Chen et al. (2024)	77.1	80.9	72.5	71.5	76.8	64.7	74.5	75.2
Mask prediction from VLM features								
LISA-7B (ft) Lai et al. (2024)	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
PixelLM-7B Ren et al. (2024)	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5
SELF1E-SEG-8B Zhang et al. (2026a)	84.7	86.2	83.4	80.2	84.2	77.0	82.1	82.8
PaDT Pro-7B Su et al. (2026)	86.0	86.1	86.4	82.5	84.1	80.7	83.5	83.3
FlowSeg (Qwen3-4B) Zhang et al. (2026b)	85.8	87.4	84.8	80.2	84.5	76.9	86.5	86.1
GLaMM-7B Rasheed et al. (2024)	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9
F-LMM (LLaVA-1.6-7B) Wu et al. (2025)	75.8	79.5	72.4	65.8	75.2	58.5	70.1	71.7
Masks as token sequences								
Text4Seg-8B Lan et al. (2025)	74.7	77.4	71.6	68.5	73.6	62.9	70.7	71.6
LlamaSeg-L(1.5B) Deng et al. (2026)	56.5	60.5	52.6	41.6	46.2	36.2	51.0	50.4
HiMTok-8B(ft, no SAM) Wang et al. (2025b)	85.0	85.2	83.5	79.7	82.7	76.0	80.0	80.6
ALToLLM-8B Wang et al. (2025a)	85.8	86.6	84.7	81.3	83.8	77.0	80.6	81.4
ARGenSeg-8B(ft) Wang et al. (2025c)	86.3	87.5	82.7	82.3	85.8	77.0	81.7	83.5
SAMTok (Qwen3-VL-4B) Zhou et al. (2026)	83.4	85.0	82.1	80.2	83.4	76.6	80.7	81.0
Mask generated natively by the VLM								
FOMO (Qwen3.5-9B, ours)	76.0	78.8	73.0	71.7	76.2	66.6	71.7	72.7

cially, unlike the best performing baselines, our method does not rely on any external segmentation tools and reconstructs masks with fixed rules.

Furthermore, this evaluation is not training-scale matched with several of the baselines using large datasets to pre-train and fine-tune model components. For example, HiMTok and ALToLLM first train their mask (de-)tokenizers on large class-agnostic mask corpora, then train the VLM on more than 7M mixed samples that include general captioning and visual question-answering data Wang et al. (2025b;a). By contrast, we merely update 129M trainable parameters on 811k referring instructions, with no mask-corpus stage, and no general-data mix.

Further failure analysis indicates that our method is limited primarily by the VLM's predictions rather than decoding errors. Unlike learned codebook indices, whose spatial meaning is supplied by a de-tokenizer, a geometry-language prediction is an explicit, parsable description of subdivision, sign, and stencil tokens. We analyze the SFT checkpoint on the validation splits of RefCOCO, RefCOCO+, and RefCOCOg. Although 6.7% of expressions obtain less than 0.1 IoU, all corresponding streams parse and decode, and for 76.1% of them the predicted mask matches another annotated COCO object in the image (IoU (≥ 0.5)). Thus, they typically describe a clean mask of the wrong object rather than a broken mask of the correct one. When the generated active tiles overlap the target's with a Jaccard index below 0.5 (13.0% of expressions), mean IoU is 0.22, versus 0.83 otherwise. Together, these results indicate that the dominant failures occur upstream of fine boundary refinement, in referent selection.



Figure 4: Qualitative gRefCOCO results from our final SFT checkpoint. Top: prediction against ground truth (yellow: overlap, green: ground-truth only, orange: prediction only). Bottom: quadtree cells decoded from the generated stream.

5 Discussion

This work introduced FOMO, a geometry language that makes dense segmentation a first-order autoregressive output of VLMs. Its deterministic codec uses adaptive quadtrees and pre-defined geometry stencils to encode homogeneous regions with few tokens while allocating more to complex boundaries. A fixed, non-learned procedure decodes the token stream. Under the stated assumptions, we prove exact representability at sufficient depth and sequence length, perimeter-bounded token counts, and conditional reconstruction-error bounds before averaging. Empirically, the codec reconstructs masks at 0.991 mean IoU with only 222 tokens on average, shows a near-perfect rank correlation between token budget and mask quality, and degrades gracefully under most grammar-preserving token corruptions.

With only a small vocabulary extension and LoRA fine-tuning, Qwen3.5-9B learns to express segmentation masks natively in our geometry language, without changes to its vision encoder or added external modules. FOMO trails the strongest decoder- and tokenizer-based systems on the three RefCOCO benchmarks. However, compared to the best-performing baselines, our method does not rely on any external segmentation models or learned mask decoders. These additional modules dictate the mask's quality and compute cost and prevent it from being a first-order output of the VLM.

Limitations and future work. A limitation of FOMO is the sequential nature of its geometry language. While sequence length scales with geometric complexity rather than image resolution, complex boundaries and multi-object masks still require long token sequences, increasing training cost and inference latency. Instance identifiers and compositing rules could extend FOMO to multi-instance and panoptic segmentation, but would further lengthen sequences. Alternatives to strict next-token prediction may reduce the resulting latency. Since our breadth-first serialization already sketches a coarse segmentation map before refining it, the token stream lends itself to masked or diffusion-based iterative refinement (Chang et al., 2022; Li et al., 2022), parallel decoding over weakly dependent tokens (Wang et al., 2025d; Pang et al., 2025; Chen et al., 2025), or next-scale prediction across quadtree depths (Tian et al., 2024). On the input side, the frozen vision encoder was not primarily optimized to capture the fine details required for dense spatial tasks. Finally, stencil bank curation and the balance between vocabulary size, stencil tolerance δ, and maximum quadtree depth D could be further tuned. These choices determine the compactness and fidelity of our geometry language and may affect how easily VLMs learn to generate it.

Conclusion. FOMO demonstrates that dense segmentation can be a first-order output of VLMs, expressed in a deterministic and fully interpretable geometry language rather than produced by external segmentation models or learned mask decoders. With only a small vocabulary extension and LoRA fine-tuning, a pretrained VLM learns to generate this language natively, bringing us closer to models that speak geometry as fluently as natural language.

AI USE STATEMENT

We used generative AI tools to assist with method implementation, code development and debugging, data preparation, and experiment execution, including model training. We also used these tools to help develop and check mathematical proofs. Additionally, we used generative AI tools to polish author-written manuscript text for clarity and readability.

The implementation and data preparation pipeline were checked using automated tests and end-to-end validation.

We have reviewed and take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Sections 3.1 and 3.2 define the codec. Section A describes the training data, including the split policy, the model adaptation, the optimization and decoding hyperparameters, and the evaluation protocol. Source code, the stencil bank and training configuration will be released upon publication.

REFERENCES

Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4485–4493. Ieee, 2017.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315. IEEE, 2022.

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

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023.

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, 2024.

Zigeng Chen, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Collaborative decoding makes visual auto-regressive modeling efficient. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23334–23344. IEEE, 2025.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, 1996.

Michel Delfour and Jean-Paul Zolésio. Shape identification via metrics constructed from the oriented distance function. *Control and Cybernetics*, 34(1):137–164, 2005.

Jiru Deng, Tengjin Weng, Tianyu Yang, Wenhan Luo, Zhiheng Li, and Wenhao Jiang. Llamaseg: image segmentation via autoregressive mask generation. In *European Conference on Computer Vision*, pp. 523–541. Springer, 2026.

GLM-5V-Team, Wenyi Hong, Xiaotao Gu, Ziyang Pan, Zhen Yang, Yuting Wang, Yue Wang, Yuanchang Yue, Yu Wang, Yanling Wang, Yan Wang, Xijun Liu, Wenmeng Yu, Weihan Wang, Wei Li, Shuaiqi Duan, Sheng Yang, Ruiliang Lv, Mingdao Liu, Lihang Pan, Ke Ning, Junhui Ji, Jinjiang Wang, Jing Chen, Jiazheng Xu, Jiale Zhu, Jiale Cheng, Ji Qi, Guobing Gan, Guo Wang, Cong Yao, Zijun Dou, Zihao Zhou, Zihan Wang, Zhiqi Ge, Zhijie Li, Zhenyu Hou, Zhao Xue, Zehui Wang, Zehan Qi, Zehai He, Yutao Zhang, Yusen Liu, Yukuo Cen, Yuchen

Li, Yuan Wang, Yu Yang, Yongbin Liu, Yijian Lu, Yifan Xu, Yanzi Wang, Yanxiao Zhao, Yanfeng Wang, Yadong Xue, Yabo Xu, Xinyu Zhang, Xinyu Liu, Xiao Liu, Wenyi Zhao, Wenkai Li, Tianyu Tong, Tianshu Zhang, Shudan Zhang, Shengdong Yan, Qinkai Zheng, Mingde Xu, Licheng Bao, lat Long long, Jiaxing Xu, Jiaxin Fan, Jiawen Qian, Jiali Chen, Jiahui Lin, Jiadai Sun, Haozhi Zheng, Haoran Wang, Haochen Li, Hanyu Lai, Han Xu, Fan Yang, Dan Zhang, Da Yin, Chuangxin Zhao, Chengcheng Wu, Boyan Shi, Bowen Lv, Bowei Jia, Bo Li, Bin Chen, Baoxu Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-5v-turbo: Toward a native foundation model for multimodal agents, 2026. URL https://arxiv.org/abs/2604.26752.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5351–5359. IEEE, 2019.

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.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9339–9350. IEEE, 2025.

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.

Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *Advances in Neural Information Processing Systems*, 35:26295–26308, 2022.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589. IEEE, 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 *International Conference on Learning Representations*, volume 2025, pp. 1634–1661, 2025.

Mengcheng Lan, Chaofeng Chen, Jiaxing Xu, Zongrui Li, Yiping Ke, Xudong Jiang, Yingchen Yu, Yunqing Zhao, and Song Bai. Text4seg++: Advancing image segmentation via generative language modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

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 *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

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.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18653–18663. IEEE, 2023b.

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.

Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 792–807, Cham, 2016. Springer International Publishing.

Jia Ning, Chen Li, Zheng Zhang, Chunyu Wang, Zigang Geng, Qi Dai, Kun He, and Han Hu. All in tokens: Unifying output space of visual tasks via soft token. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 19843–19853. IEEE, 2023.

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T Freeman, and Yu-Xiong Wang. Randar: Decoder-only autoregressive visual generation in random orders. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 45–55. IEEE, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14076–14088. IEEE, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S Ryoo, and Tsung-Yu Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12977–12987. IEEE, 2024.

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.

Xiangxuan Ren, Zhongdao Wang, Liping Hou, Pin Tang, Guoqing Wang, and Chao Ma. Grounding everything in tokens for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41171–41181, 2026.

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.

Tianhui Song, Haoyu Lu, Hao Yang, Lin Sui, Haoning Wu, Zaida Zhou, Zhiqi Huang, Yiping Bao, Y Charles, Xinyu Zhou, et al. Towards pixel-level vlm perception via simple points prediction. *arXiv preprint arXiv:2601.19228*, 2026.

Yongyi Su, Haojie Zhang, Shijie Li, Nanqing Liu, Jingyi Liao, Junyi Pan, Yuan Liu, Xiaofen Xing, Chong Sun, Chen Li, et al. Patch-as-decodable-token: Towards unified multi-modal vision tasks in mllms. In *International Conference on Learning Representations*, volume 2026, pp. 147893–147919, 2026.

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.

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. *arXiv preprint arXiv:2505.16495*, 2025a.

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

Xiaolong Wang, Lixiang Ru, Ziyuan Huang, Kaixiang Ji, Dandan Zheng, Jingdong Chen, and Jun Zhou. Argenseg: Image segmentation with autoregressive image generation model. *Advances in Neural Information Processing Systems*, 38:28312–28336, 2025c.

Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12955–12965. IEEE, 2025d.

Size Wu, Sheng Jin, Wenwei Zhang, Lumin Xu, Wentao Liu, Wei Li, and Chen Change Loy. F-lmm: Grounding frozen large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24710–24721. IEEE, 2025.

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.

Anqi Zhang, Xiaokang Ji, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. Self1e: Rethinking mllm itself as a segmenter with a single segmentation token. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a.

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.

Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog grounding large language models to holistic segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14227–14238. IEEE, 2024b.

Zekang Zhang, Guangyu Gao, Youyun Tang, ChengJing Wu, Xiaochao Qu, Chi Harold Liu, Jianbo Jiao, Yunchao Wei, Luoqi Liu, and Ting Liu. Flowseg: Dynamic semantic flow for llm-conditioned segmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026b.

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, 2024c.

Rongkun Zheng, Lu Qi, Xi Chen, Yi Wang, Kun Wang, and Hengshuang Zhao. Seg-var: Image segmentation with visual autoregressive modeling. *Advances in Neural Information Processing Systems*, 38:165921–165942, 2025.

Yikang Zhou, Tao Zhang, Dengxian Gong, Yuanzheng Wu, Ye Tian, Haochen Wang, Haobo Yuan, Jiacong Wang, Lu Qi, Hao Fei, et al. Samtok: Representing any mask with two words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37852–37863, 2026.

Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *European Conference on Computer Vision*, pp. 598–615. Springer, 2022.