Grounding Everything in Tokens for Multimodal Large Language Models

Supplementary Material

We provide supplementary material for further study and analysis related to the main paper, arranged as follows:

- Additional experimental results extending the main findings (Sec. A)
- Real-world driving dataset curation (Sec. B)
- Additional implementation details, including training setup, offset-aware dataset construction, and reward design (Sec. C)
- Additional qualitative results and visual analysis (Sec. D)

A. Additional Experimental Results

A.1. More Benchmarks

Referring Captioning evaluates region understanding given referring inputs (e.g., bounding boxes, masks). We evaluate region-based caption generation on refCOCOg [16] and Visual Genome [7]. As shown in Tab. A, GETok achieves competitive or superior performance relative to models that rely on specialized region feature extractors (✓), highlighting the effectiveness of GETok for region-aware comprehension. GETok is particularly effective in scenarios with overlapping objects, where traditional bounding boxes often fail to precisely capture targeted regions.

Table A. **Region-Level Captioning** results on the refCOCOg and visual genome datasets.

Methods	Region Feat. Extractor	refCOCOg		Visual Genome	
		METEOR	CIDEr	METEOR	CIDEr
GRIT [22]	✔	15.2	71.6	17.1	142.0
SLR [28]	✔	15.9	66.2	-	-
GPT4RoI [29]	✔	-	-	17.4	145.2
GLaMM [19]	✔	16.2	106.0	19.7	**180.5**
Groma [15]	✔	16.8	107.3	19.0	158.4
Kosmos-2 [17]	✘	14.1	62.3	-	-
Shikra-7B [3]	✘	15.2	72.7	-	-
GETok-SFT	✘	**16.9**	**110.5**	**19.0**	165.9

Generalized RES validates multi-instance grounding through grid token sequences, demonstrating simultaneous referencing capability for multiple objects within a single spatial representation. GETok naturally supports multi-instance expressions. We evaluate GETok on the gRefCOCO dataset [10] for multi-instance segmentation. As shown in Tab. B, GETok achieves competitive performance relative to specialized methods while maintaining architectural simplicity.

Object Pointing evaluates precise point-level localization. Instead of restricting the target to a single predefined point,

Table B. **Generalized Referring Expression Segmentation** results (cIoU) on the RefCOCO (+/g) datasets.

Methods	Training M-Dec.	Validation	Test-A	Test-B	Average
LAVT [26]	✔	58.4	65.9	55.8	60.0
ReLA [10]	✔	63.6	70.0	61.0	64.9
LISA [9]	✔	63.5	68.2	61.8	64.5
GSVA [23]	✔	68.0	71.8	63.8	67.9
GETok-SFT	✘	66.9	72.3	64.1	67.8
GETok-RL	✘	**67.4**	**74.1**	**65.6**	**69.0**

GETok supports flexible point annotations by marking representative object positions, making the representation more adaptable to diverse object types and scene layouts. As shown in Tab. C, GETok achieves competitive performance compared to methods trained with substantially more data. The advantage is particularly pronounced in dense object scenarios, where grid tokens reduce coordinate representation from multiple sequential tokens (e.g., `['(', '124', ',', '143', ')']`) to *a single* spatial token (e.g., <grid$_{12,14}$>), eliminating the formatting errors that accumulate with longer text-based coordinate sequences.

Table C. **Object Pointing** results on HumanRef and RefCOCOg datasets.

Methods	HumanRef	refCOCOg val	refCOCOg test
OVIS2.5-9B [14]	62.3	**85.0**	**84.5**
Molmo-7B-D [5]	70.0	83.7	83.6
Qwen2.5-VL-7B [1]	65.1	78.9	79.4
GETok-SFT	**70.7**	84.1	82.9

A.2. More Discussions

How Should Points be Represented? We analyze three representation formats that operate purely through *vocabulary-level modifications*: text coordinates, bin tokens, and grid tokens, all of which require no architectural changes. Among them, bin tokens and text coordinates share the same 1D numerical nature, with bin tokens merely quantizing coordinates into discrete indices, and empirical evidence shows that bin-based methods can even underperform text coordinates [3]. The key difference, therefore, lies between these 1D schemes and the 2D spatial encoding of grid tokens, which addresses three fundamental limitations:

1) 1D-2D Representation Gap: A single 1D token cannot directly represent a 2D location; instead, multiple tokens must be combined to denote a coordinate. This composition hinders the implicit semantic features of the 2D space from being effectively mapped into the token embeddings.

2) Format Brittleness: Syntactic elements introduce ex-

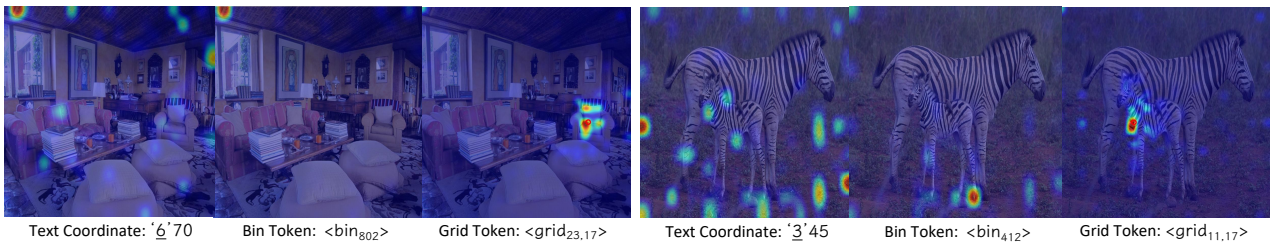


Figure A. **Visualization of spatial responses for different localization vocabularies.** We aggregate attention maps between location tokens and image patches to obtain heatmaps for text coordinates, 1D bin tokens, and grid tokens. Grid tokens produce smooth, topology-aware activations that align with object extents.

ponential failure rates that are particularly problematic in multi-object scenarios. For example, with 98% per-token accuracy, a 12-token box sequence has a 78% validity probability, dropping to 48% for three boxes (36 tokens).

3) Metric–Objective Mismatch: Token-level cross-entropy on digit sequences correlates poorly with geometric error. Small changes in token indices can correspond to large jumps in image space.

Using Qwen2.5-VL-7B with identical RefCOCO/+/g instruction-tuning data, we compare text, bin, and grid formats in Tab. D, and observe a clear advantage for grid tokens. Furthermore, as shown in Fig. A, grid tokens produce smooth, locally coherent activations that closely follow object extents because each token is tied to a fixed 2D region in the image plane. In contrast, text and bin tokens yield fragmented, geometry-agnostic responses without a stable 2D correspondence.

Table D. Ablation on **point representation formats** for REC on the RefCOCO/+/g datasets.

Methods	refCOCO Test-A	refCOCO+ Test-A	refCOCOg Test
Text Coordinates	92.9	89.9	87.4
Bin tokens	92.3	89.9	87.1
Grid tokens	**93.0**	**90.6**	**87.6**

Why GRPO Works with GETok? GETok's structured representation creates an ideal action space for GRPO optimization. As shown in Fig. B, GETok achieves accelerated convergence at equivalent training steps compared to text coordinates, validating its structured action space advantage for GRPO optimization. We attribute this advantage to two key factors: (1) The 2D grid structure provides a stable foundation for policy learning, unlike text coordinates, where minor token changes yield discontinuous spatial shifts. (2) The finite $n \times n$ token format is easier to learn than text coordinates. This compact set allows the model to focus on spatial layout rather than complex text patterns, leading to faster convergence.

How to Represent Masks with Sparse Geometry? We analyze existing sparse geometric representations, such as

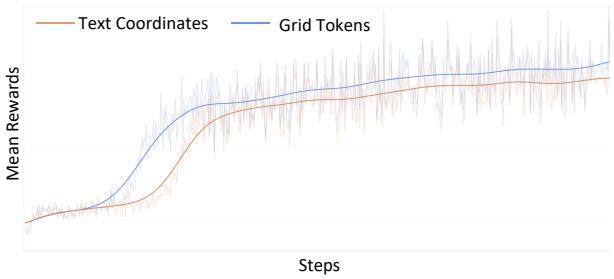


Figure B. **Reward curve comparison between grid tokens and text coordinates.** GETok achieves faster convergence and higher rewards than text coordinates.

single points, bounding boxes, fixed sets of one or two points, or randomly sampled points, all of which suffer from redundancy and an inability to unambiguously capture complex mask semantics as shown in Fig. C. We introduce a novel greedy algorithm that automatically extracts an appropriate set of such tokens from a target mask. Compared to methods that require training a dedicated mask decoder [9, 19, 23], this design offers several advantages:

1) At training time, our method avoids any mask-specific loss, decoder, or supervision, offering a simpler alternative compared to methods that rely on task-specific decoders.

2) At inference time, our method offers strong flexibility as our decoder is purely plug-and-play and can be seamlessly updated without retraining the referring VLM. For example, replacing SAM [6] with advanced SAM2 [20], our method achieves a performance gain of 0.8% cIoU on refCOCO val at no cost. In contrast, LISA has to retrain the full model for this replacement, which is particularly costly.

A.3. Ablation Studies

Image Preprocessing. We investigate the impact of different image preprocessing strategies on localization performance as shown in Tab. E. Padding gives the worst results, because the added gray borders effectively downscale the informative region and distract the model from relevant content. Center cropping risks semantic distortion by removing

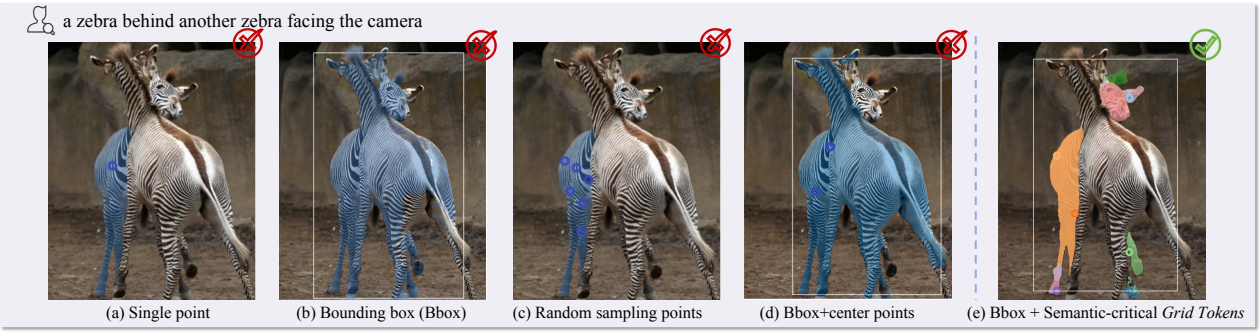


Figure C. **Comparison of mask representation strategies.** We convert continuous masks into discrete, segment-critical grid tokens to achieve precise region referencing.

peripheral image areas. For example, in a referring expression such as "the person on the far left," cropping may exclude the target entirely, leading to ground-truth mismatch. In contrast, resizing and dynamic resolution achieve comparable performance in our experiments. We therefore adopt simple resizing as our default preprocessing strategy.

Table E. Ablation on **image preprocessing** strategies for REC on RefCOCOg.

Methods	RefCOCOg
Padding	85.9
Center Crop	86.2
Dynamic	87.1
Resize	**87.4**

Reward Function. For grid token generation, removing the semantic-critical points reward causes the model either to collapse to one or two high-confidence points or to overpopulate a small region with redundant points, as shown in Tab. F. Removing the box reward yields the largest drop, and visual inspection shows that points become scattered in the absence of a stable coarse prior. By contrast, the mask reward mainly provides fine-grained geometric supervision, especially for thin structures and concave regions that are not well constrained by box and point-level signals alone.

For offset token refinement, we focus on whether offsets perform genuine geometric corrections. The mask IoU gain and box refinement rewards provide instance-level guidance that promotes updates with improved mask and box IoU. The point refinement reward further stabilizes behavior by reducing large mask changes caused by a few erroneous point adjustments.

Reasoning vs. No Reasoning for Offset Refinement. The `<think>` process has been shown to be beneficial for multimodal understanding, especially in cases that require complex semantic reasoning [12, 13, 21]. We further examine its role in the refine stage. Empirically, the performance gap between using and omitting `<think>` during refinement is negligible (0.1% gIoU), suggesting that offset refinement does not substantially benefit from additional verbal reasoning. We observe that the model rarely produces meaningful explanations for point-level updates and instead repeats almost the same `<think>` content as in the propose step, so we do not enforce `<think>` generation in this stage.

Table F. Ablation on **reward design** for grid-token generation and offset-token refinement.

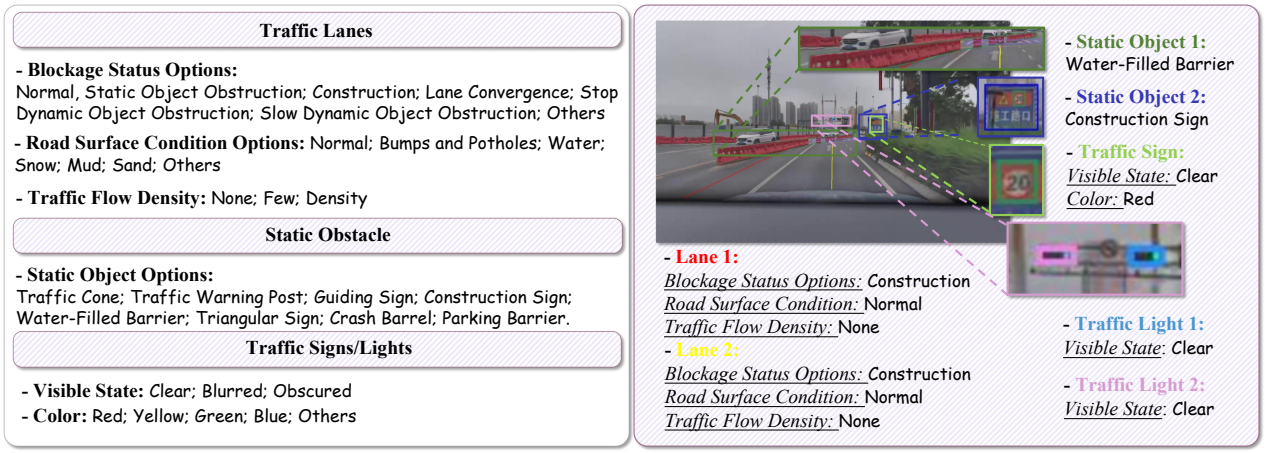
Reward for Grid Token Generation				
Variant	Mask	Box	Sem. points	ReasonSeg
w/o Sem. points	✔	✔		58.6
w/o Mask reward		✔	✔	59.1
w/o Box reward	✔		✔	57.2
Full (ours)	✔	✔	✔	**60.1**
Reward for Offset Token Refinement				
Variant	Point gain	Box gain	Mask IoU gain	ReasonSeg
w/o Mask IoU gain	✔	✔		61.8
w/o Box ref.	✔		✔	61.2
w/o Point ref.		✔	✔	60.5
Full (ours)	✔	✔	✔	**62.8**

B. Real-World Driving Dataset

We construct a proprietary autonomous driving dataset to evaluate GETok in complex real-world scenarios and to support comparison with strong baselines. The dataset contains 1,988 training samples with 29,825 annotations and 980 test samples with 14,524 annotations, covering diverse urban scenes including intersections, highways, and pedestrian zones.

As illustrated in Fig. D(a), the dataset categorizes driving targets into three classes: Traffic Lanes, Static Obstacles, and Traffic Signs with hierarchical annotations for multi-granular reasoning. Based on these annotations, we design a series of classification tasks to evaluate the model's ability to understand and refer to specific regions in driving scenes.

Fig. D(b) shows an example from the dataset. Each sample is annotated using category labels selected from the tax-



(a) Referring Classification Options

(b) Annotation Example

Figure D. **Overview of the driving dataset annotations.** (a) Summarizes the taxonomy of annotated driving targets (lanes, static obstacles, and traffic signs/lights) with hierarchical labels. (b) Illustrates an example scene annotated with points, polygons, lane polylines, bounding boxes, and masks for referring and safety-related queries.

onomy summarized in Fig. D(a). Overall, driving scenes provide a realistic setting that demands understanding and referring to regions in multiple formats, including points, polygons, polylines, bounding boxes, and masks, highlighting the application potential of a unified and robust localization framework.

C. More Implementation Details

C.1. Training Setup

Supervised Fine-Tuning. The model is fine-tuned on the mixed-task corpus summarized in Tab. G. All location-related annotations (points, boxes, masks) are converted into GETok's grid tokens. The offset-aware dataset is constructed on top of RefCOCO/+/g, and a more systematic description of the construction pipeline is provided in Sec. C.2. We use a per-device batch size of 2 with 8 gradient accumulation steps, yielding an effective batch size of 16 per device. All input images are resized to 840×840, and training is conducted with bfloat16 mixed precision.

Reinforcement Learning. We first perform a cold-start stage to adapt the model to the newly introduced tokens while mixing in CoT-style instruction data, thereby preserving its original multimodal capabilities. Building on this checkpoint, we further optimize the policy with GRPO on both grid-token placement and offset-token refinement. Each update is regularized by a KL-divergence penalty to the SFT policy with coefficient 1×10^{-2}. For each prompt, we sample 8 candidate responses to estimate group-wise advantages. For offset tokens, we empirically find that about 200 steps are sufficient to obtain satisfactory refinement performance.

Table G. Summary of training data composition.

Stage	Datasets	Task
SFT	LLaVA-665K [11]	Image reasoning
	RefCOCO/+/g [16, 27]	Referring grounding
	COCO-Stuff [2]; ADE20K [30]	Segmentation (seg.)
	Visual Genome [7]	Image captioning
	PACO-LVIS [18]; PASCAL-Part [4]	Part-level seg.
	gRefCOCO [10]	Multi-instance seg.
	Pixmo-point [5]	Object pointing
	GETok-Offset	Referring refinement
Cold Start	RefCOCO/+/g [16, 27]	Referring seg.
	LLaVA-CoT-100K [24]	Instruction tuning
	GETok-Offset	Offset training
GRPO	RefCOCOg [16] subset (3.0K)	Single-target seg.
	LISA++ [25] (2.0K); gRefCOCO [10] (4.0K)	Multi-instance seg.

C.2. Offset-Aware Dataset Curation Details

Region Definitions. Let $\mathbf{M}_{\text{gt}} \in \{0,1\}^{H\times W}$ be the binary foreground mask. We place an $n \times n$ grid and denote the pixel center of cell (i,j) by $\mathbf{c}_{i,j} = (x_{i,j}, y_{i,j})^\top$.To construct pools of candidate grid tokens, we employ morphology-based bands scaled according to the offset step size. Let $\mathcal{K}_k \in \{0,1\}^{k\times k}$ represent a square structuring element with a side length of k pixels. We define:

$$\begin{aligned} k_e &= \lfloor s_y \rfloor + 1, \quad & \mathbf{E} &= \mathbf{M}_{\text{gt}} \ominus \mathcal{K}_{k_e}, \\ k_d &= 2\lfloor s_y \rfloor + 1, \quad & \mathbf{D} &= \mathbf{M}_{\text{gt}} \oplus \mathcal{K}_{k_d}, \end{aligned} \tag{1}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, while $\ominus$ and $\oplus$ represent morphological erosion and dilation respectively. A thin boundary band is additionally defined as:

$$\mathbf{B} = \text{dilate}(\text{grad}(\mathbf{M}_{\text{gt}}), \mathcal{K}_b), \tag{2}$$

where $\mathrm{grad}(\mathbf{M}_{\mathrm{gt}})$ is the morphological gradient and b is a small width parameter. By construction, $\mathbf{E} \subset \mathbf{M}_{\mathrm{gt}} \subset \mathbf{D}$: $\mathbf{E}$ forms a step-sized interior buffer, $\mathbf{D}$ creates a step-sized exterior halo, and $\mathbf{B}$ captures edge uncertainty as a narrow boundary ribbon.

Grid Point Categorization and Sampling. We define a one-step hit test to determine reachability:

$$\mathrm{Hit}(i,j) \triangleq \exists\, \boldsymbol{\delta} \in \{-1,0,1\}^2 : \mathbf{M}_{\mathrm{gt}}(\mathbf{c}_{i,j} + \mathbf{S}\boldsymbol{\delta}) = 1. \quad (3)$$

Each grid center is assigned to exactly one category via the hierarchical decision rule:

$$\mathrm{pool}(i,j) = \begin{cases} \text{Hard-Delete}, & \mathbf{B}(y_{i,j}, x_{i,j}) = 1 \\ & \wedge\ \boldsymbol{M}_{\mathrm{gt}}(y_{i,j}, x_{i,j}) = 0 \\ & \wedge\ \neg\,\mathrm{Hit}(i,j), \\ \text{Inside}, & \mathbf{E}(y_{i,j}, x_{i,j}) = 1, \\ \text{Ring}, & \mathbf{D}(y_{i,j}, x_{i,j}) = 1 \\ & \wedge\ \boldsymbol{M}_{\mathrm{gt}}(y_{i,j}, x_{i,j}) = 0, \\ \text{Far}, & \text{otherwise.} \end{cases} \quad (4)$$

Following pool formation $\mathcal{P}_{\mathrm{hard}} \to \mathcal{P}_{\mathrm{inside}} \to \mathcal{P}_{\mathrm{ring}} \to \mathcal{P}_{\mathrm{far}}$, we sample $K \sim \pi_K$ grids per image with preferential selection from $\mathcal{P}_{\mathrm{inside}}$ and $\mathcal{P}_{\mathrm{ring}}$, while maintaining representation from all categories for robustness. Then, the complete construction process, detailed in Algorithm 1, processes each image-mask-query triple to automatically produce conversational data containing grid tokens and their corresponding offset targets.

Algorithm 1: Offset-Supervised Data Construction

Input: Referring dataset $\mathcal{D}$; grid size n; offset granularity m; IoU threshold τ
Output: JSONL conversations containing grids and offset targets

foreach $(I, \mathbf{M}_{gt}, q) \in \mathcal{D}$ **do**
- Resize $I, \mathbf{M}_{\mathrm{gt}}$ to $H \times W$; compute $s_x = W/m$, $s_y = H/m$, $\mathbf{S} = \mathrm{diag}(s_x, s_y)$;
- `// grid pools via morphology (cf. (1)--(2))`
- Compute $\mathbf{E}, \mathbf{D}, \mathbf{B}$; assign each grid cell (i,j) to one of INSIDE/RING/FAR/HARD-DELETE by rule (4);
- `// Segmentation grids and offsets`
- Sample K grids $\{(i_k, j_k)\}_{k=1}^{K}$ from the pools;
- **for** $k = 1$ **to** K **do**
  - Set $\mathbf{c}_k \leftarrow \mathbf{c}_{i_k, j_k}$;
  - **if** $\mathbf{M}_{gt}(y_{i_k}, x_{i_k}) = 1$ **then**
    - emit `[OFF_0_0]`
  - **else if** $\mathrm{Hit}_{3\times3}(i_k, j_k)$ **then**
    - pick $(\delta_u, \delta_v) \in \{-1,0,1\}^2$ with $\mathbf{M}_{\mathrm{gt}}(\mathbf{c}_k + \mathbf{S}\boldsymbol{\delta}) = 1$, and emit `[OFF_`$\delta_u$`_`$\delta_v$`]`
  - **else**
    - emit `<DELETE>`
- `// Bounding-box corner offsets`
- Let $B^\star \leftarrow \mathrm{BBox}(\mathbf{M}_{\mathrm{gt}})$; jitter its TL/BR to grid corners $(i_{\mathrm{tl}}, j_{\mathrm{tl}}), (i_{\mathrm{br}}, j_{\mathrm{br}})$;
- Evaluate all offset pairs for the two corners (apply $\mathbf{S}$-scaled displacements), obtain $\mathrm{IoU}_{\max}$;
- **if** $\mathrm{IoU}_{\max} \geq \tau$ **then**
  - emit the two corner offsets
- **else**
  - emit `<DELETE>` for both corners
- `// Serialization`
- Write a JSONL sample with image tag, user prompt q and grids (user turn), and the offsets (assistant turn);

C.3. Reward Details

Multi-object Matching. From each line in `<answer>`, we extract a predicted instance consisting of an optional box $\hat{\boldsymbol{b}}_p \in \mathbb{R}^4$ and a point set $\mathcal{P}_p = \{\boldsymbol{q}\} \subset \mathbb{R}^2$. Let there be P predictions and G ground-truth instances with binary masks $\{\boldsymbol{M}_{\mathrm{gt}}\}_{g=1}^{G}$ and tight boxes $\{\boldsymbol{b}_g\}_{g=1}^{G}$. We define pairwise similarities between prediction p and ground truth g:
i) Box IoU:

$$\mathrm{IoU}_{p,g} \in [0,1]. \quad (5)$$

ii) Point-hit ratio: the fraction of predicted points that land inside $\mathbf{M}_{\mathrm{gt}}$,

$$H_{p,g} = \frac{1}{\max(1, |\mathcal{P}_p|)} \sum_{\mathbf{q} \in \mathcal{P}_p} \mathbb{1}\{\mathbf{q} \in \mathbf{M}_{\mathrm{gt}}\} \in [0,1]. \quad (6)$$

iii) Normalized L_1 box score:

$$S_{p,g}^{\ell_1} = \mathrm{clip}\left(1 - \frac{\|\hat{\mathbf{b}}_p - \mathbf{b}_g\|_1/4}{\tau_{\ell_1}},\, 0,\, 1\right). \quad (7)$$

These are combined into a similarity used only for the assignment:

$$\mathrm{Sim}_{p,g} = \mathrm{IoU}_{p,g} + H_{p,g} + S_{p,g}^{\ell_1}, \quad (8)$$

We solve a Hungarian assignment [8] with costs $C_{p,g} = 3 - \mathrm{Sim}_{p,g}$, yielding matched pairs $\mathcal{M} \subseteq \{1..P\} \times \{1..G\}$. We use τ_{ℓ_1}=18 px.

Semantic-Critical Points Reward. For each $(p,g) \in \mathcal{M}$, we compute a key points quality:

$$F_{p,g} \triangleq S(m_p)\Big(w_H\, H_{p,g} + w_{\mathrm{spr}}\, \mathrm{Spread}_{p,g}\Big) - \lambda_m\, m_p. \quad (9)$$

where $H_{p,g}$ is the hit ratio, and $\mathrm{Spread}_{p,g}$ rewards larger nearest-neighbor spacing normalized by object scale:

$$\bar{d}_p = \frac{1}{m_p} \sum_{i=1}^{m_p} \min_{j \neq i} \|\boldsymbol{q}_i - \boldsymbol{q}_j\|_2, \qquad \mathrm{Spread}_{p,g} = \mathrm{clip}\big(\bar{d}_p/(\rho_s r_g),\, 0,\, 1\big). \quad (10)$$

The multiplicative saturation $S(m) = 1 - \exp(-m/m_0)$ discourages degenerate few-point outputs, and the linear

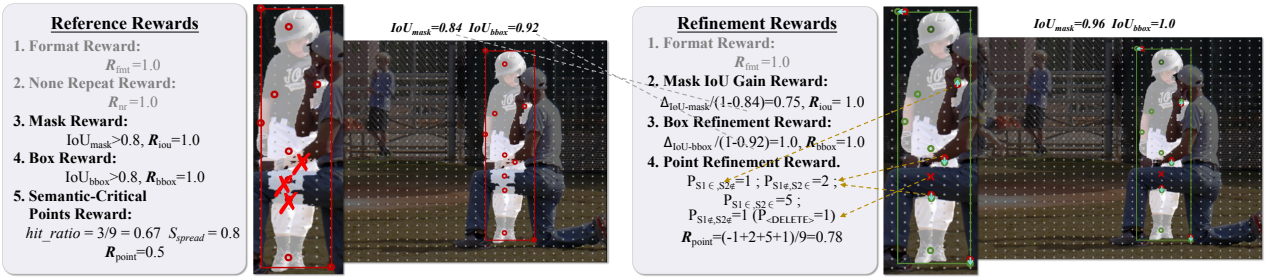


Figure E. **Illustration of reward computation for grid token generation and refinement.** The diagram demonstrates how different reward components are calculated based on predicted outputs and ground-truth annotations.

term $\lambda_m m_p$ penalizes overly long point lists. We aggregate across matches with point-count weighting:

$$T = \mathrm{clip}\left(\frac{\sum_{(p,g)\in\mathcal{M}} m_p\, F_{p,g}}{\sum_{p=1}^{P} \max(1, m_p)},\ 0,\ 1\right). \tag{11}$$

We set w_H=0.6, w_{spr}=0.4, λ_m=0.02, ρ_s=0.30.

Point Refinement Reward. Let $\mathbf{M}_{\mathrm{gt}}^{(k)} \subset \mathbb{Z}^2$ be the ground-truth mask of the k-th instance. The coarse point set is $\mathcal{C}_k = \{\mathbf{c}_{k,p}\}_{p=1}^{P_k}$ and the refined set is $\mathcal{C}_k^{\mathrm{off}} = \{\mathbf{c}_{k,p}^{\mathrm{off}}\}_{p=1}^{P_k}$, with a one-to-one correspondence over p (if a point is deleted, we keep its index p and mark a delete flag). Define the inclusion indicators $I_{k,p} = \mathbb{I}\big[\mathbf{c}_{k,p} \in \mathbf{M}_{\mathrm{gt}}^{(k)}\big]$, $I_{k,p}^{\mathrm{off}} = \mathbb{I}\big[\mathbf{c}_{k,p}^{\mathrm{off}} \in \mathbf{M}_{\mathrm{gt}}^{(k)}\big]$. The point-wise reward $s_{k,p} \in \{-1, 0, 1\}$ is

$$\begin{cases} -1, & I_{k,p} = 1 \wedge I_{k,p}^{\mathrm{off}} = 0 \\ +1, & I_{k,p} = 0 \wedge I_{k,p}^{\mathrm{off}} = 1 \\ +1, & I_{k,p} = 1 \wedge I_{k,p}^{\mathrm{off}} = 1 \\ +1, & I_{k,p} = 0 \wedge \texttt{<DELETE>} \wedge \mathcal{N}_{3\times3}(\mathbf{c}_{k,p}) \cap \mathbf{M}_{\mathrm{gt}} = \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

where $\mathcal{N}_{3\times3}(\mathbf{c}_{k,p})$ is the 3×3 neighborhood centered at $\mathbf{c}_{k,p}$. The instance-level reward is obtained by averaging over all points of that instance. Fig. E provides a concrete example illustrating the reward computation process for better understanding.

D. Additional Visualization Results

Grid Tokens for Mask Representation. Fig. F presents additional qualitative results comparing predicted grid tokens, output masks, and GT annotations. The results are organized from top to bottom, ranging from predictions that are more precise than the GT mask to some failure cases. These visualizations highlight the following key observations:

(1) High-Quality Predictions: The model can generate highly accurate grid tokens, which align well with the GT masks. These results demonstrate the effectiveness of grid tokens in precisely localizing and referring to objects in complex scenes.

(2) *Failure Cases:* In some cases, accurate grid-token predictions still yield imperfect masks due to discrepancies in SAM's mask decoding. Nonetheless, as discussed in Sec. A.2, this training-free decoding remains advantageous compared to training task-specific mask decoders. Introducing offset tokens further mitigates these errors by refining point locations and aligning the decoded masks more closely with object boundaries.

The qualitative results underscore the robustness of grid tokens as a referring representation, even in cases where segmentation performance is suboptimal.

SFT Benchmarks Qualitative Results. Fig. G demonstrates the unified representation capability of GETok across diverse vision-language tasks. Our approach establishes a cohesive framework that processes various query types through a consistent token vocabulary, spanning image-, point-, box-, and mask-level formats while eliminating the need for task-specific output heads.

Self-Improving Mechanism. Fig. H presents additional qualitative examples demonstrating the propose-and-refine workflow of GETok for fine-grained mask prediction. The left panel shows that for interior points unambiguously inside the mask, the model correctly maintains their positions without unnecessary adjustments, focusing refinement efforts exclusively on boundary regions. The right panel illustrates a failure case primarily caused by erroneous refinement decisions resulting from initial tokens placed near misleading edge features. These examples collectively highlight the method's capacity to maintain accurate localization through coordinated grid and offset token operations, even in challenging scenarios.

Figure F. **More qualitative results of the segmentation task.** From top to bottom, the predictions are ordered by decreasing Intersection-over-Union (IoU) scores relative to the ground truth masks.

Figure G. **Unified GETok representations across diverse vision-language tasks.** GETok provides a unified representation framework that handles diverse visual concepts without task-specific architectural modifications.

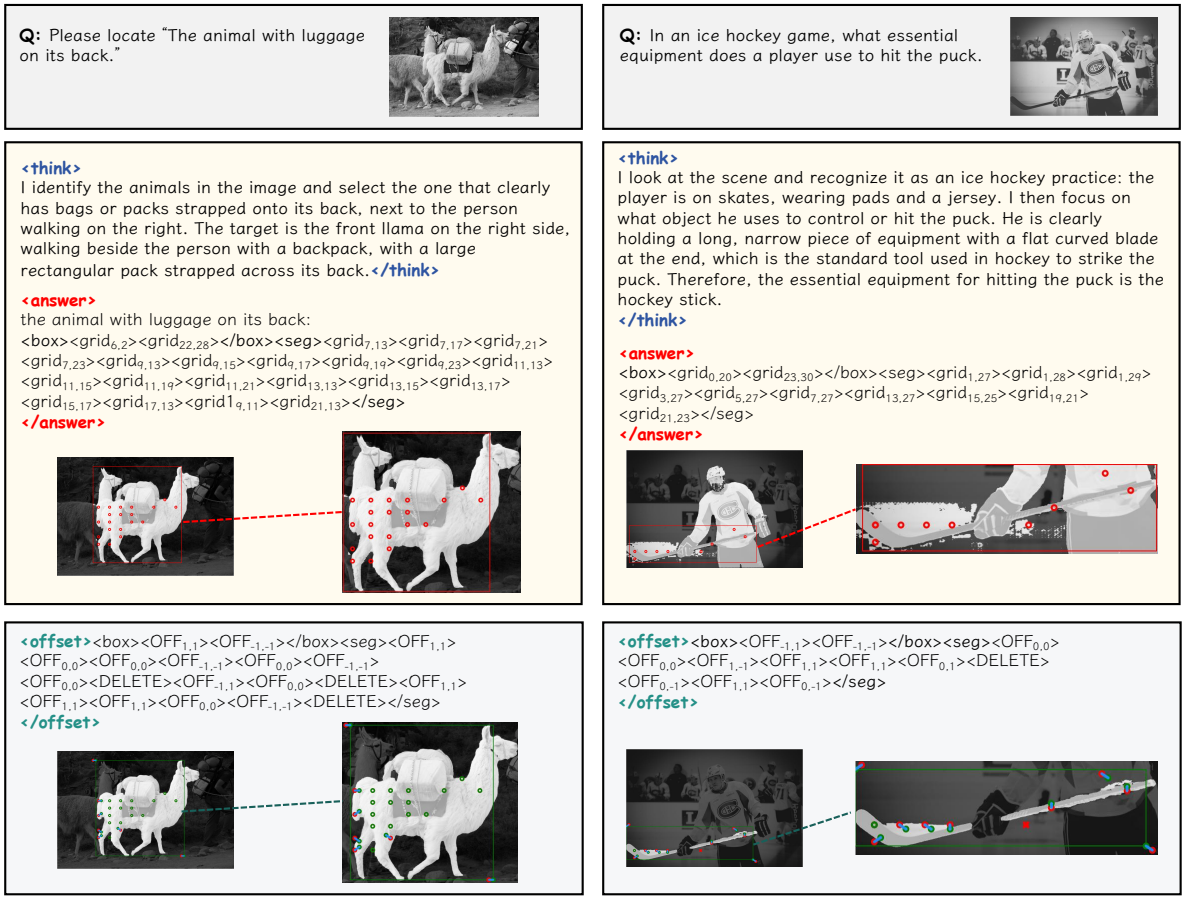


Figure H. **More qualitative results of the self-improving mechanism.** Additional examples demonstrate how GETok establishes initial spatial proposals through grid tokens (red dots) and enables fine-grained adjustments via offset tokens (blue arrows), showing effective handling of objects across scales with enhanced precision on small targets.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 4

[3] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 1

[4] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 4

[5] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, pages 91–104, 2025. 1, 4

[6] 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 *ICCV*, pages 4015–4026, 2023. 2

[7] 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. *IJCV*, 123:32–73, 2017. 1, 4

[8] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 5

[9] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, pages 9579–9589, 2024. 1, 2

[10] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Gen-

eralized referring expression segmentation. In *CVPR*, pages 23592–23601, 2023. 1, 4

[11] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 4

[12] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025. 3

[13] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *ICCV*, pages 2034–2044, 2025. 3

[14] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*, 2024. 1

[15] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *ECCV*, pages 417–435. Springer, 2024. 1

[16] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016. 1, 4

[17] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 1

[18] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *CVPR*, pages 7141–7151, 2023. 4

[19] 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 *CVPR*, pages 13009–13018, 2024. 1, 2

[20] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[21] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 3

[22] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding. In *ECCV*, pages 207–224, 2024. 1

[23] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *CVPR*, pages 3858–3869, 2024. 1, 2

[24] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *ICCV*, pages 2087–2098, 2025. 4

[25] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. Lisa++: An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023. 4

[26] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *CVPR*, pages 18155–18165, 2022. 1

[27] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 4

[28] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, pages 7282–7290, 2017. 1

[29] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. arxiv 2023. *arXiv preprint arXiv:2307.03601*, 2023. 1

[30] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 4