Region-aware Anchoring Mechanism for Efficient Referring Visual Grounding

Supplementary Material

	Stage 1	Stage 2	Stage 3	Stage 4
Using Swin-B as Vision Backbone				
Q_i	2	2	6	2
visual output size	$\frac{H}{4} \times \frac{W}{4}$	$\frac{H}{8} \times \frac{W}{8}$	$\frac{H}{16} \times \frac{W}{16}$	$\frac{H}{32} \times \frac{W}{32}$
channel	96	192	384	768
Using ViT-B as Vision Backbone				
Q_i	3	3	3	3
visual output size	$\frac{H}{16} \times \frac{W}{16}$	$\frac{H}{16} \times \frac{W}{16}$	$\frac{H}{16} \times \frac{W}{16}$	$\frac{H}{16} \times \frac{W}{16}$
channel	768	768	768	768

Table 7. Detailed settings of different stages in our RaAM-RVG.

6. Additional Implementation Details

Tab. 7 presents the implementation details for using different vision backbones, including Swin Transformer-Base (Swin-B) [27] and Vision Transformer-Base (ViT-B) [11]. Here, Q_i denotes the number of vision model layers in the i-th stage. When ViT-B is used as the vision backbone, the visual feature dimensions in Sec. 3 are represented as $\mathbb{R}^{C_{v_i} \times (H_i W_i)}$, with the corresponding $\mathrm{Flatten}(\cdot)$ operation omitted. If Swin-B is used, the spatial dimensions of $[F_{scale}]$ from different stages in the Region-aware Prediction operation of Sec. 3.5 need to be unified to $H_2 \times W_2$. Accordingly, the operation in Eq. (10) is modified as:

$$R_n = \mathrm{Concat}[P_n^2 \odot X_n^2, \ldots, P_n^K \odot X_n^K].$$

7. Hyperparameter Experiments

Number N of Region-aware Anchor Tokens. To evaluate the impact of the number of tokens (N) used in region-aware anchors, we conducted experiments on the gRefCOCO Val set with varying N values, as shown in Tab. 8(a). The results indicate that increasing the number of anchor tokens N improves GRES performance. However, the performance gains diminish for $N > 16$, while the computational cost increases significantly. Consequently, we select $N = 16$ as the optimal trade-off, ensuring robust performance while maintaining computational efficiency.

Coefficient Selection for Loss Functions. The loss functions incorporate multiple weighted components, each modulated by coefficients λ_1 to λ_5. Additionally, $\mathcal{L}_p^{seg}$ includes a coefficient β to control the weight of object boundaries. To identify the optimal settings for maximizing model accuracy in object detection and segmentation while ensuring stable convergence, we systematically varied these coefficients. The results for different tasks are shown in Tab. 8(b) and Tab. 8(c). Following extensive tuning, we selected: $\lambda_1 = 0.1$, $\lambda_2 = 0.1$, $\lambda_3 = 1$, $\lambda_4 = 0.1$, $\lambda_5 = 1$, and $\beta = 1.2$. This configuration achieves an optimal balance between accuracy and model stability.

(a) Number N of anchor tokens

N	**cIoU**	**gIoU**
9	65.24	67.97
16	67.35	70.02
25	67.51	70.36
36	67.82	70.94

(b) Hyperparameters for REC and RES

β	λ_1	λ_2	λ_3	**P@0.5**	**oIoU**
1.2	0.05	0.1	1	90.94	78.83
1.2	0.1	0.1	1	**91.45**	**79.35**
1.2	0.2	0.1	1	90.07	78.42
1.2	0.1	0.2	1	89.65	79.44
1.2	0.1	0.1	1.5	89.76	79.51

(c) Hyperparameters for GRES

λ_4	λ_5	**cIoU**	**gIoU**
0.05	0.5	66.41	68.86
0.1	0.5	66.52	68.89
0.1	1.0	**67.35**	**70.02**
0.2	1.0	64.35	76.74
0.1	1.5	64.79	77.06

Table 8. Hyperparameter experiments conducted on the RefCOCO validation set for REC and RES, and on the gRefCOCO validation set for GRES.

Method	LAVT [54]	PolyFormer [25]	**RaAM (Ours)**
Runtime (ms) ↓	297.56	332.69	**266.56**
FPS ↑	67.46	60.24	**78.87**

Table 9. Speed comparison results. "↓" means lower is better, "↑" refers to upper is better, and "FPS" denotes frames per second.

8. Complexity Comparison

8.1. Speed Comparison

We validate the efficiency of our proposed RaAM by conducting a speed comparison with SOTA methods. For comparison, we selected advanced RES method LAVT [54] and

Figure 5. Visualizations of predicted results and ground truth on REC and RES. Green lines enclose successful cases, while red lines enclose failed cases.

Method	Param.(M)	RefCOCO	RefCOCO+	G-Ref
RaPM-RVG	211.69	**79.35**	**69.54**	**71.30**
LAVT [54]	217.19	72.73	62.14	61.24
w/ RaAM	217.45	**74.32**	**64.87**	**65.73**
PolyFormer [25]	336.26	75.96	69.33	69.20
w/ RaAM	322.53	**78.35**	**71.87**	**70.45**
P-RIS [39]	774.38	76.36	67.06	64.79
w/ RaAM	745.77	**78.44**	**68.90**	**67.32**

Table 10. Plug-and-play validation (w/ parameter comparisons).

	Methods	N-acc.	T-acc.
RES Methods	MattNet[57]	41.15	96.13
	VLT[10]	47.17	95.72
	LAVT[54]	49.32	96.18
	CGFormer[34]	51.01	96.23
GRES Methods	ReLA[23]	57.51	96.97
	RaAM-RVG	**65.76**	**97.85**

Table 11. No-object results on the gRefCOCO Val set.

multi-task visual grounding method PolyFormer [25]. The results are presented in Tab. 9. All models in the comparison use the same setup, incorporating Swin Transformer-base and BERT-base as the vision and language backbones, respectively, with a language token length of 20. The batch

	Methods	cIoU	gIoU
GRES Methods	ReLA[23]	56.08	57.67
	RaAM-RVG	**61.48**	**62.88**

Table 12. Multi-object results on the gRefCOCO Val set.

size is 16. To ensure a fair comparison, we exclude the point generation time cost from PolyFormer [25]. Compared to methods employing *Direct Interaction* strategies, our method achieves reduced runtime and demonstrates higher FPS. Parameter comparison is in the Appendix.

8.2. Parameter Comparison

To compare the space complexity of RaAM with existing methods, we conducted plug-and-play validation on expert models for RVG , including LAVT [54], PolyFormer [25], and P-RIS [39], and compared their performance and model parameter counts. The results on RES, as shown in Tab. 10, demonstrate that RaAM-RVG achieves superior performance with fewer parameters, highlighting its advantage in terms of spatial complexity.

9. Additional Experimental Results for GRES

To further validate the effectiveness of RaAM-RVG on the GRES task, we analyzed its performance on no-object and multi-object samples, which are unique to GRES. The performance evaluation on no-object samples is presented in

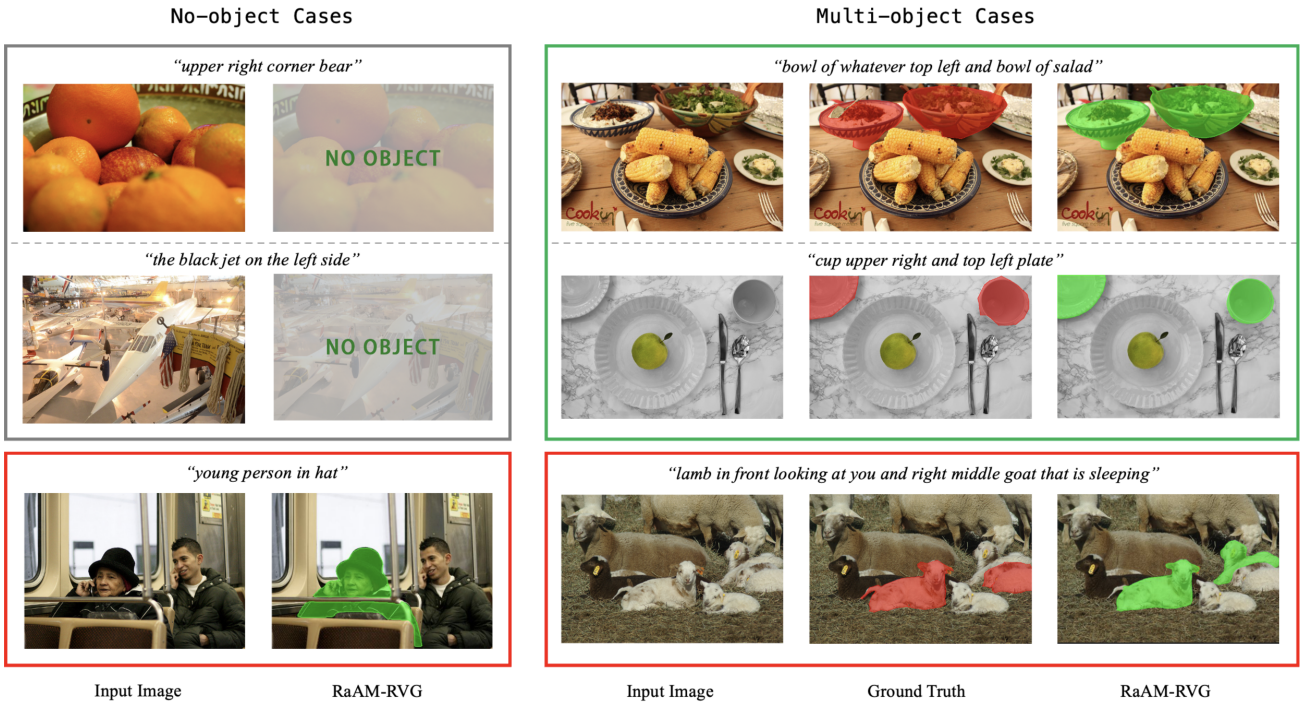

Figure 6. Visualization of GRES prediction results. The left side illustrates no-object cases, while the right side shows multi-object cases. Gray boxes indicate correctly predicted "No Object" scenarios, green boxes represent successful multi-object segmentations, and red boxes highlight failed cases.

Tab. 11. We report the No-object accuracy (N-acc.) and Target accuracy (T-acc.) metrics, which assess the model's ability to correctly identify instances where no target object is present. Our method demonstrates a significant advantage over ReLA in no-object cases, particularly achieving an 8.25% improvement in N-acc. Additionally, the results in Tab. 12 indicate that RaAM-RVG significantly outperforms ReLA in multi-object scenarios. These results highlight the overall effectiveness of RaAM in both no-object and multi-object scenarios, underscoring its superior capability in complex localization.

10. Additional Qualitative Results

Successes and Failures in REC and RES. We present the visualizations of REC and RES tasks using our RaAM-RVG, including both successful and failed cases. In Fig. 5, green lines enclose successful cases, while red lines enclose failed cases. Examples Fig. 5(a), (b), and (c) show results that match or even exceed the ground truth in accuracy. In Fig. 5(d), failures are due to inaccurate object annotations in the ground truth, while Fig. 5(e) fails due to ambiguity in the provided expression. In Fig. 5(f), the boundaries of the pizza slice in the image are challenging to distinguish, indicating that further model optimization is needed for improved recognition.

No-object and Multi-object Cases in GRES. We present visualizations of GRES-specific scenarios using RaAM-RVG, focusing on no-object and multi-object examples, as shown in Fig. 6. Examples within the gray box demonstrate RaAM-RVG's capability to effectively identify no-object cases, while examples within the green box illustrate the model's ability to segment multiple targets simultaneously. In the failed no-object case, the model incorrectly segmented a "person in hat," overlooking the condition "young." In the failed multi-object case, the model mislocalized the second object, segmenting the goat that did not meet the requirement of "sleeping." These failures indicate that the model lacks full discriminatory capability for objects that partially satisfy the language conditions. Enhancing the preservation of linguistic details during vision-language interaction and reflecting them in object recognition results remains a critical direction for future research.