WHAT ATTENTION AGGREGATION HIDES: ATTENTION ENTROPY ROUTING FOR REFERRING SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-guided segmentation requires a model to choose visual context that separates a referred region from nearby distractors. Existing methods mainly process Transformer features formed by aggregation weighted by attention, whose outputs do not explicitly record how concentrated the underlying attention distributions are. We introduce **Attention Entropy Routing (AER)**, which treats normalized multiscale attention entropy as a complementary conditioning signal: language generates a bank of channel modulation vectors, local visual features and aligned entropy route among them, and pooled entropy conditions multiscale fusion. An analysis of single-head attention aggregation with fixed value vectors motivates this design, while heads trained on frozen visual features and controlled routing and spatial permutation studies test its effects at the task and spatial levels. Controlled comparisons show gains over visual routing from local entropy and further gains from global conditioning, while evaluations on RefCOCO, RefCOCO+, RefCOCOg, and remote sensing benchmarks with synthetic nonuniform haze cover natural and degraded visual settings.

1 INTRODUCTION

Vision–language learning has made natural language a practical interface for visual recognition (Radford et al., 2021). Language-guided segmentation brings this interface to individual pixels: given an expression, the model must identify the described object and delineate its extent (Liu et al., 2017; Wang et al., 2022). This requires more than matching words to appearance. The model must decide which visual context separates the referred region from nearby distractors, and the useful context can vary across positions and expressions. The same challenge arises in natural images and remote sensing imagery, where object appearance, scale, and surrounding layout differ substantially (Yuan et al., 2024; Liu et al., 2024).

Existing methods address this challenge through language-aware visual encoding (Yang et al., 2022), language-guided queries (Ding et al., 2021), and contrastive image–text alignment (Wang et al., 2022). These approaches primarily condition the processing of visual features. For a Transformer backbone, a visual token is formed by an aggregation weighted by attention (Liu et al., 2021b). The resulting feature records the aggregated value content, but it does not explicitly describe how concentrated the underlying attention distribution is. Two positions can therefore have similar aggregated features while one relies on a few dominant keys and the other combines information from a broad support. Existing segmentation pipelines do not use this difference to select how processing conditioned on language should be applied.

This observation leads to a concrete question: can attention concentration provide a useful condition in addition to visual content and language? A single-head aggregation with fixed value vectors can produce the same output from distinct attention distributions with different entropies, an instance of attention nonidentifiability (Brunner et al., 2020). The example motivates attention entropy as a compact descriptor of concentration, not as a substitute for the attended content or a direct measure of prediction uncertainty. We therefore test whether normalized, spatially aligned entropy supplies complementary information for referring segmentation when the visual representation is held fixed.

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 propose **Attention Entropy Routing (AER)** to implement this idea. The expression generates a bank of channel modulation vectors. At each image position, the visual feature and aligned entropy maps from four encoder stages determine how these responses generated from language are mixed, allowing local processing to vary with the concentration of visual evidence. Pooled entropy then conditions multiscale fusion at the image level. We evaluate the resulting hypothesis with matched heads trained on frozen visual features, comparisons between local and image-mean routing, global fusion ablations, and spatial permutation tests. We further evaluate AER on RefCOCO, RefCOCO+, and RefCOCOg, and use remote sensing benchmarks with synthetic nonuniform haze as a controlled evaluation under degraded visual evidence.

Our contributions are threefold:

- We introduce AER, which uses local attention entropy with visual content to route channel modulation vectors generated from language and uses pooled entropy for multiscale fusion.
- We characterize, for a single-head aggregation with fixed value vectors, when identical aggregation outputs can correspond to different entropy values, and test whether this potentially complementary signal improves segmentation with frozen visual features and aligned spatial inputs.
- We evaluate AER on RefCOCO, RefCOCO+, and RefCOCOg, and on RefSegRS and RRSIS-D under synthetic nonuniform haze. The experiments separate the effects of local routing, global fusion, and spatial correspondence while measuring performance across natural and degraded visual settings.

2 Related Work

2.1 Language-Guided Segmentation

Early referring segmentation methods combined convolutional image features with recurrent language representations, using repeated multimodal interaction or mask refinement to connect an expression to image regions (Liu et al., 2017; Li et al., 2018). Cross-modal attention then made the interaction depend on spatial and linguistic context (Ye et al., 2019; Hui et al., 2020). Transformer methods develop this idea through query generation guided by language (Ding et al., 2021) and language-aware visual encoding (Yang et al., 2022). Contrastive image–text representations provide another route to dense alignment (Wang et al., 2022). In remote sensing, multiscale interaction and orientation modeling address variation in object size and layout (Yuan et al., 2024; Liu et al., 2024), while foundation model approaches combine pretrained visual, linguistic, and segmentation representations (Chen et al., 2025). AER augments interaction between feature representations across modalities with attention concentration. It uses the statistic as an input to local channel routing and scale fusion at the image level.

2.2 Attention Statistics and Conditional Processing

Attention selects the context used to form a feature, and its weights expose statistics of that selection. Attention entropy has been used to study Transformer training stability and prevent entropy collapse (Zhai et al., 2023), and to select informative observations in active visual exploration (Pardyl et al., 2023). Conditional transformations provide a way to use such information during feature processing. FiLM generates channel modulation from language (Perez et al., 2018), and learned routing assigns weights that depend on the input to candidate transformations (Shazeer et al., 2017). AER combines channel modulation generated from language with an explicit attention statistic: visual content and local entropy determine the mixture at each position, while pooled entropy contributes to scale scores.

3 Task and Motivation

3.1 Language-Guided Segmentation

Given an image $x \in \mathbb{R}^{H_0 \times W_0 \times 3}$ and a referring expression $s = (w_1, \ldots, w_L)$, the task is to predict the region described by s. The target is a binary mask $y \in \{0, 1\}^{H_0 \times W_0}$. Our model produces

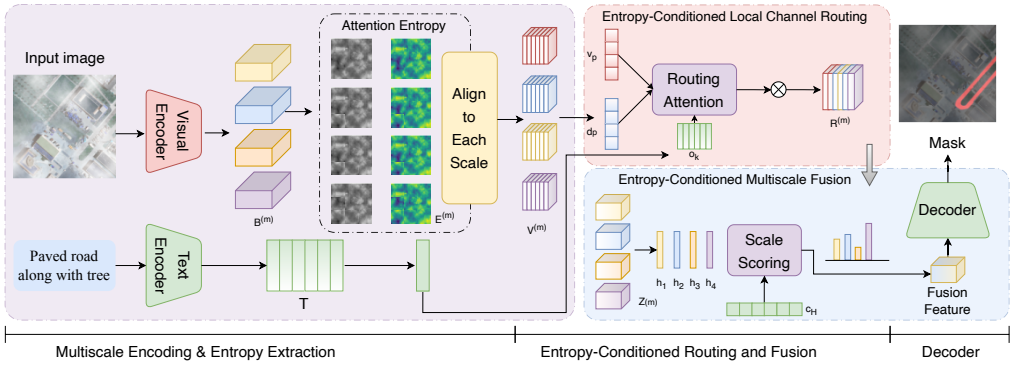


Figure 1: Overview of Attention Entropy Routing (AER). The visual encoder produces multiscale features and attention entropy maps; aligned entropy conditions local channel routing, while pooled entropy conditions multiscale fusion before the decoder predicts the mask.

foreground probabilities $\hat{y} = f_\theta(x, s) \in [0,1]^{H_0 \times W_0}$; the predicted binary mask is $\hat{m}_p = \mathbf{1}[\hat{y}_p \geq 0.5]$. Training pairs each image and expression with its target mask.

3.2 Attention Aggregation and Its Ambiguity

For a single attention head with fixed value vectors $u_i \in \mathbb{R}^D$, let a_i denote the attention weight on key i. Define the aggregated output and Shannon entropy as

$$z(a) = \sum_{i=1}^{N} a_i u_i, \qquad H(a) = -\sum_{i=1}^{N} a_i \log a_i, \tag{1}$$

with $a_i > 0$ and $\sum_i a_i = 1$. The map from weights to output is linear. Its null space determines whether distinct weight distributions can share the same output.

Proposition 1 (Aggregation can conceal concentration) *Let the augmented value matrix be*

$$M = \begin{bmatrix} u_1 & \cdots & u_N \\ 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{(D+1)\times N}. \tag{2}$$

If $\operatorname{rank}(M) < N$*, there exist strictly positive probability vectors* a *and* b *such that* $z(a) = z(b)$ *and* $H(a) \neq H(b)$.

The proof is given in Appendix A. For a concrete example, take scalar values $(u_1, u_2, u_3) = (0, 1, 2)$ and

$$a = (0.1, 0.8, 0.1), \qquad b = (0.4, 0.2, 0.4). \tag{3}$$

Both yield $z = 1$, whereas $H(b) - H(a) = 0.6 \log 2$. This is an instance of attention nonidentifiability (Brunner et al., 2020), with entropy distinguishing distributions that produce the same output from a single head for fixed values. The diagnostic with frozen features in Section 5.3 evaluates its predictive value alongside the full visual representation.

4 Attention Entropy Routing

AER has two paths conditioned on entropy. The visual encoder first produces four feature maps and the attention probabilities of their final blocks. After the encoder completes all stages, we align the four entropy maps to each routing grid. At each position, the aligned entropies and the visual feature route among channel modulation vectors generated from language. The routed feature maps are then resized to a common grid, where pooled entropy conditions fusion at the image level. A decoder modulated by language predicts the final mask.

Representations. The visual encoder $\mathcal{E}_v$ is a hierarchical Swin Transformer (Liu et al., 2021b), with outputs $B^{(m)} \in \mathbb{R}^{H_m \times W_m \times C_m}$ for stages $m = 1, \ldots, 4$. The language encoder $\mathcal{E}_t$ produces token features $T = \mathcal{E}_t(s) \in \mathbb{R}^{L \times C_t}$ and a pooled sentence representation $\bar{T} \in \mathbb{R}^{C_t}$. A projection for each stage puts each visual map into a common channel space:

$$V^{(m)} = P_m(B^{(m)}) \in \mathbb{R}^{H_m \times W_m \times C}. \tag{4}$$

We write $v_p^{(m)} \in \mathbb{R}^C$ for the feature at position p. The visual backbone completes all four stages before the routing and fusion operations.

4.1 Multiscale Entropy Extraction and Alignment

Attention concentration. For stage m, we read attention probabilities from its final self-attention block, after softmax and before dropout. Let $\widetilde{A}_{p,q}^{(m,h)}$ denote the weights for head h, renormalized over the valid keys $\mathcal{V}_m(p)$, and let $N_{m,p} = |\mathcal{V}_m(p)|$. The normalized query entropy is

$$E_p^{(m)} = -\frac{1}{n_m \log N_{m,p}} \sum_{h=1}^{n_m} \sum_{q \in \mathcal{V}_m(p)} \widetilde{A}_{p,q}^{(m,h)} \log \widetilde{A}_{p,q}^{(m,h)}, \tag{5}$$

where n_m is the number of heads. We use $0 \log 0 = 0$ and set $E_p^{(m)} = 0$ for $N_{m,p} = 1$. For larger supports, dividing by $\log N_{m,p}$ places entropy on a common $[0, 1]$ scale, with higher values indicating more diffuse attention.

We restore each entropy map to image coordinates by reversing window partitioning and cyclic shifts, then remove padding. Bilinear resizing aligns its query locations with routing stage m:

$$d_p^{(m)} = \mathrm{Concat}_{j=1}^{4} \left[\mathrm{Resize}_{j \to m}(E^{(j)})_p \right] \in [0, 1]^4. \tag{6}$$

Each entry describes concentration at one representation scale. Their spatial averages form the global descriptor

$$c_H = \mathrm{Concat}_{j=1}^{4} \left[\mathrm{GAP}_{\mathrm{valid}}(E^{(j)}) \right] \in \mathbb{R}^4, \tag{7}$$

where pooling excludes padded queries. The local vector retains spatial variation for channel routing, while c_H summarizes the image for scale fusion.

4.2 Local Channel Routing Conditioned on Entropy

Modulation vectors generated from language. The sentence embedding generates K channel modulation vectors at each stage:

$$O^{(m)} = \mathrm{Reshape}(W_o^{(m)} \bar{T}) \in \mathbb{R}^{K \times C}. \tag{8}$$

Each row $o_k^{(m)}$ is a C-dimensional candidate modulation vector. We omit m below for readability; routing projections are learned separately at each stage. The visual feature and entropy form the routing query, and the modulation vectors supply the keys:

$$q_p = W_q(v_p + W_d d_p), \qquad k_k = W_k o_k, \tag{9}$$

$$\alpha_{p,k} = \mathrm{softmax}_k \left(\frac{q_p^\top k_k}{\sqrt{C}} \right). \tag{10}$$

Here $W_d \in \mathbb{R}^{C \times 4}$ and $W_q, W_k \in \mathbb{R}^{C \times C}$. The weights $\alpha_{p,k}$ are positive and sum to one. They define a local mixture of the responses generated from language:

$$g_p = \sum_{k=1}^{K} \alpha_{p,k} \phi_o(o_k), \qquad \tilde{u}_p = \phi_v(v_p) \odot g_p, \tag{11}$$

where ϕ_o and ϕ_v are learned linear projections. The mixture g_p modulates the visual feature across channels. The routed map preserves the original features through a residual connection:

$$R^{(m)} = V^{(m)} + \mathrm{MLP}_m \left(\mathrm{Reshape}\{\tilde{u}_p^{(m)}\}_p \right). \tag{12}$$

The response to entropy depends on the expression through the modulation keys. At fixed visual content, changing entropy can alter the relative preference between channel responses. Appendix B derives this dependence and gives its finite change and differential forms.

4.3 Multiscale Fusion Conditioned on Entropy

Scoring scales conditioned on language. The routed feature $R^{(m)}(x, s)$ already depends on the expression through its modulation vectors and routing weights. We project and resize the four maps to a common grid, then pool them:

$$Z^{(m)} = \mathrm{Resize}\big(\psi_m(R^{(m)})\big), \qquad h_m = \mathrm{GAP}(Z^{(m)}), \tag{13}$$

where $h_m \in \mathbb{R}^C$ is a scale descriptor. The fusion score is

$$e_m = w_s^\top h_m + \frac{(Uh_m)^\top (V_H c_H)}{\sqrt{r}}, \qquad \beta_m = \mathrm{softmax}_m(e_m), \tag{14}$$

with $U \in \mathbb{R}^{r\times C}$, $V_H \in \mathbb{R}^{r\times 4}$, and $w_s \in \mathbb{R}^C$ shared across stages. Text guides these weights through $R^{(m)}$ and h_m. Global entropy interacts with each scale descriptor, allowing the same c_H to make different contributions to the four scores.

The fused feature is

$$F_{\mathrm{fuse}} = \sum_{m=1}^{4} \beta_m Z^{(m)}. \tag{15}$$

Each β_m is shared across the image. Local routing supplies spatially varying feature selection, while fusion selects the relative contribution of each scale for the image–expression pair.

4.4 Network Integration and Training Objective

The network first computes visual features, sentence features, and attention entropy. Local routing yields $R^{(1)}, \ldots, R^{(4)}$, which are fused into F_{fuse}. A residual convolution refines this map, and projections for each scale supply the decoder at successive resolutions. At each decoder level, the sentence representation generates an affine channel modulation, followed by upsampling and convolutional refinement. A final 1×1 projection and sigmoid produce $\hat{y}$ at the input resolution. Appendix C specifies these decoder operations.

The segmentation objective combines binary cross-entropy and soft Dice loss:

$$\mathcal{L}_{\mathrm{seg}} = \lambda_b \mathcal{L}_{\mathrm{BCE}} + (1 - \lambda_b)\mathcal{L}_{\mathrm{Dice}}, \tag{16}$$

with $\lambda_b = 0.9$ and Dice smoothing constant $\epsilon = 1$. Appendix C.4 gives the pixelwise definitions. A routing auxiliary loss encourages distinct channel responses in the modulation bank. Let $G_{ij}^{(m)}$ be the cosine similarity between $o_i^{(m)}$ and $o_j^{(m)}$, computed with stabilized norms. We use

$$\mathcal{L}_{\mathrm{op}} = \frac{1}{4K^2} \sum_{m=1}^{4} \|G^{(m)} - I_K\|_F^2, \qquad \mathcal{L} = \mathcal{L}_{\mathrm{seg}} + \lambda_o \mathcal{L}_{\mathrm{op}}. \tag{17}$$

Losses are averaged over training samples, with $K = 4$ and $\lambda_o = 0.1$. During end-to-end training, gradients pass through the local mixtures, scale weights, and entropy computation to the encoders. Target masks provide the sole supervision.

5 Experiments

We evaluate segmentation performance on natural images and on remote sensing images with fixed synthetic nonuniform haze, then examine local entropy routing, scale fusion, predictive complementarity with frozen features, and spatial correspondence.

5.1 Experimental Setup

Datasets and evaluation roles. RefCOCO and RefCOCO+ (Yu et al., 2016) pair MS COCO objects with short referring expressions. Both use the UNC partition, with validation, testA (people), and testB (other objects) subsets. RefCOCO+ excludes absolute location words, placing greater emphasis on appearance. RefCOCOg (Mao et al., 2016) contains longer descriptions; we use its UMD

partition (Nagaraja et al., 2016). These three datasets provide the general referring segmentation evaluation.

For remote sensing images, RefSegRS (Yuan et al., 2024) contains 4,420 image, expression, and mask triplets across 14 categories, with 2,172 training, 431 validation, and 1,817 test samples. RRSIS-D (Liu et al., 2024) contains 17,402 triplets across 20 categories, with 12,181 training, 1,740 validation, and 3,481 test samples. We extend the evaluation by applying fixed synthetic nonuniform haze to these remote sensing datasets. We generate haze with multiscale Perlin noise and the atmospheric scattering model (Appendix E.1). All compared methods share the same hazy observations generated separately for each split, expressions, masks, and target presence labels.

Comparison methods. The natural image comparison includes VLT (Ding et al., 2021), CRIS (Wang et al., 2022), LAVT (Yang et al., 2022), DMMI (Hu et al., 2023), CARIS (Liu et al., 2023), and ReMamber (Yang et al., 2024). Their scores are taken from the respective publications under the original training and evaluation protocols. The haze comparison includes general RIS models RRN (Li et al., 2018), CMSA (Ye et al., 2019), LSCM (Hui et al., 2020), CMPC+ (Liu et al., 2021a), LAVT (Yang et al., 2022), CrossVLT (Cho et al., 2023), CARIS (Liu et al., 2023), DMMI (Hu et al., 2023), DeRIS (Dai et al., 2025), and RefSegformer (Wu et al., 2024); and remote sensing models LGCE (Yuan et al., 2024), RMSIN (Liu et al., 2024), FIANet (Lei et al., 2025), MCDNet (Zhang et al., 2026), and RSRefSeg2 (Chen et al., 2025). We use the official public repository for every baseline, retrain it on the corresponding synthetic haze dataset, and evaluate it with the shared split, haze images, expressions, masks, and target presence labels. For every retrained method, the checkpoint with the best validation result is selected. The tables list each method's encoders. RSRefSeg2's cIoU and gIoU correspond to oIoU and mIoU here.

Evaluation metrics. We report pooled foreground IoU (oIoU) for the natural image comparison. On the haze benchmarks, we also report IoU averaged over samples (mIoU) and the percentage of samples whose IoU exceeds X ($P@X$), for $X \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. The ablations and diagnostic with frozen features use test mIoU on hazy RefSegRS and validation oIoU on the original RefCOCO images. IoU and $P@X$ are reported as percentages, and IoU differences are in percentage points. The relative modulation change Δ_g in Section 5.3 is dimensionless. Architectural and optimization settings are provided in Appendix C.

5.2 Main Results

General referring segmentation. Table 1 reports overall IoU on the three natural image datasets. AER has the highest listed score on all eight splits, improving over CARIS by 2.67–6.79 percentage points. The largest gains occur on the testB splits of RefCOCO and RefCOCO+, at 6.79 and 6.54 points, respectively. The gains on RefCOCOg extend the comparison to longer referring expressions.

Table 1: Overall IoU on three natural image referring segmentation datasets. The best and second best scores in each column are shown in red and blue, respectively.

Method	Vision	Language	RefCOCO			RefCOCO+			RefCOCOg	
			Val	TestA	TestB	Val	TestA	TestB	Val(U)	Test(U)
VLT (Ding et al., 2021)	Darknet-53	Bi-GRU	65.65	68.29	62.73	55.50	59.20	49.36	52.99	56.65
CRIS (Wang et al., 2022)	CLIP R-101	CLIP text	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36
LAVT (Yang et al., 2022)	Swin-B	BERT	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09
DMMI (Hu et al., 2023)	Swin-B	BERT	74.13	77.13	70.16	63.98	69.73	57.03	63.46	64.19
CARIS (Liu et al., 2023)	Swin-B	BERT	75.14	78.10	71.75	66.63	72.04	58.51	64.31	65.82
ReMamber (Yang et al., 2024)	Mamba-B	CLIP text	74.54	76.74	70.89	65.00	70.78	57.53	63.90	64.00
AER	Swin-B	BERT	**79.83**	**81.57**	**78.54**	**69.30**	**77.61**	**65.05**	**68.92**	**70.26**

Extension to synthetic nonuniform haze. Tables 2 and 3 give the complete validation and test results under synthetic nonuniform haze. On RefSegRS, AER reaches test mIoU of 68.82, exceeding FIANet's 62.07 by 6.75 percentage points; its test oIoU of 77.56 exceeds DMMI's 74.51 by 3.05 points. AER also has the highest listed test $P@X$ at all five thresholds. On RRSIS-D, AER improves test mIoU over RSRefSeg2 from 64.28 to 65.91 and test oIoU from 76.85 to 80.52, gains of 1.63 and

Table 2: Complete validation and test results on RefSegRS under synthetic nonuniform haze. All methods use the same hazy observations for training and evaluation. The best and second best scores in each column are shown in red and blue, respectively.

Method	Vision	Language	P@0.5		P@0.6		P@0.7		P@0.8		P@0.9		oIoU		mIoU	
			Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
RRN	R-101	LSTM	9.05	5.83	5.10	3.19	3.71	1.16	0.23	0.17	0.00	0.00	34.80	31.00	12.46	8.61
CMSA	R-101	–	12.53	9.36	6.50	4.79	2.55	1.38	0.23	0.11	0.00	0.00	45.57	43.40	18.50	13.45
LSCM	R-101	LSTM	20.88	12.22	12.30	6.88	6.26	2.70	1.62	0.66	0.00	0.11	48.72	43.39	35.28	26.35
CMPC+	R-101	LSTM	22.51	15.74	17.63	11.72	11.83	6.93	5.10	2.15	0.70	0.44	57.50	52.84	26.77	19.47
LAVT	Swin-B	BERT	43.39	20.20	25.06	12.71	14.15	6.38	6.26	2.15	1.16	0.22	67.10	56.73	45.74	30.20
CrossVLT	Swin-B	BERT	85.85	61.92	74.94	45.07	48.96	23.83	21.58	10.68	6.73	2.92	78.86	73.26	65.62	52.75
CARIS	Swin-B	BERT	79.35	58.78	67.75	40.56	36.19	20.80	18.33	11.61	5.10	3.14	78.78	73.26	62.45	51.77
DMMI	Swin-B	BERT	90.95	70.56	85.15	56.47	65.20	30.60	25.52	13.32	7.66	2.86	82.15	74.51	70.63	57.03
LGCE	Swin-B	BERT	87.01	62.52	76.80	42.93	46.64	21.85	18.33	9.91	6.50	2.20	79.60	71.64	66.12	52.37
RMSIN	Swin-B	BERT	82.37	56.14	67.75	35.50	38.28	18.00	15.31	6.77	3.94	1.38	74.18	68.00	63.56	49.28
RefSegformer	Swin-B	BERT	87.24	61.36	70.07	40.40	38.75	21.24	21.58	11.83	6.26	2.70	79.74	72.77	65.13	51.97
DeRIS	Swin-B	BEiT3	85.15	59.49	70.77	39.52	37.59	17.83	16.71	7.54	3.25	0.88	72.96	65.79	63.96	50.65
FIANet	Swin-B	BERT	94.20	78.04	88.86	65.38	76.10	40.67	32.95	17.94	8.12	2.59	79.63	73.48	74.00	62.07
MCDNet	Swin-B	BERT	93.27	77.05	89.56	64.34	77.73	42.16	40.14	16.84	6.96	2.70	79.35	72.87	74.78	61.56
RSRefSeg2	SAM2.1	SigLIP2	86.54	71.71	83.76	62.08	78.19	49.92	65.43	28.95	**22.97**	7.43	78.52	68.89	75.72	60.69
AER	Swin-B	BERT	**95.36**	**83.16**	**92.11**	**76.17**	**86.54**	**59.00**	**69.14**	**30.99**	14.15	**8.09**	**85.31**	**77.56**	**79.72**	**68.82**

Table 3: Complete validation and test results on RRSIS-D under synthetic nonuniform haze. All methods use the same hazy observations for training and evaluation. The best and second best scores in each column are shown in red and blue, respectively.

Method	Vision	Language	P@0.5		P@0.6		P@0.7		P@0.8		P@0.9		oIoU		mIoU	
			Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val	Test
RRN	R-101	LSTM	33.16	31.17	24.60	23.13	15.52	15.37	7.07	7.41	1.09	1.26	49.97	48.88	32.82	31.13
CMSA	R-101	–	41.72	40.33	33.68	32.46	24.43	24.07	14.54	14.59	3.16	3.30	61.50	61.35	38.88	37.71
LSCM	R-101	LSTM	44.43	42.29	37.53	35.74	29.31	28.07	19.66	18.93	6.90	7.35	61.41	59.72	39.78	38.57
CMPC+	R-101	LSTM	49.60	47.40	40.17	38.32	28.79	28.30	17.53	15.80	3.45	3.02	65.29	63.93	45.31	44.07
LAVT	Swin-B	BERT	58.22	56.28	51.72	49.38	42.64	40.56	32.18	30.57	17.64	17.01	70.63	69.26	52.11	50.25
CrossVLT	Swin-B	BERT	71.67	69.09	63.97	62.74	53.79	53.17	42.70	40.91	25.17	23.99	77.11	73.88	62.28	60.63
CARIS	Swin-B	BERT	73.74	73.28	65.23	63.03	53.91	52.28	40.11	38.67	22.70	20.68	77.00	73.13	61.92	60.43
DMMI	Swin-B	BERT	69.54	69.52	63.10	62.34	53.51	51.80	40.86	39.39	24.02	22.95	77.68	76.56	61.19	60.42
LGCE	Swin-B	BERT	52.76	52.69	45.75	45.68	38.74	38.70	28.28	28.27	13.56	13.53	69.56	70.69	48.31	48.37
RMSIN	Swin-B	BERT	73.28	73.11	65.92	65.64	55.23	54.78	42.76	41.08	23.45	22.90	77.37	76.74	63.97	63.34
RefSegformer	Swin-B	BERT	65.00	65.07	58.91	58.32	49.94	49.61	40.52	38.58	23.68	23.07	75.97	76.16	58.39	57.75
DeRIS	Swin-B	BEiT3	66.49	66.22	59.83	58.69	50.23	47.69	37.76	36.25	19.43	19.94	74.13	73.62	58.59	57.79
FIANet	Swin-B	BERT	73.16	72.36	66.26	65.21	54.94	53.15	42.13	39.87	23.10	22.15	76.66	76.25	62.81	62.78
MCDNet	Swin-B	BERT	71.09	71.07	63.45	63.09	52.99	51.88	39.66	39.13	21.44	21.60	76.34	75.36	62.26	61.43
RSRefSeg2	SAM2.1	SigLIP2	76.67	76.10	**71.67**	**70.84**	**61.90**	**59.72**	**49.43**	**46.45**	**29.08**	27.46	77.57	76.85	65.66	64.28
AER	Swin-B	BERT	**79.60**	**78.25**	67.41	65.64	57.13	56.77	46.90	44.24	28.97	**27.69**	**82.83**	**80.52**	**66.73**	**65.91**

3.67 points. RSRefSeg2 retains higher test P@0.6, P@0.7, and P@0.8: 70.84, 59.72, and 46.45, compared with AER's 65.64, 56.77, and 44.24.

5.3 Why Does Entropy Help?

We first test whether entropy complements fixed visual features, then separate local routing from global fusion and intervene on spatial correspondence. These analyses use hazy RefSegRS test mIoU and RefCOCO validation oIoU.

Conditional complementarity. The diagnostic with frozen features in Table 4 tests whether entropy improves prediction from a fixed visual representation. Both conditions use the same Swin-B visual encoder as AER and the same BERT encoder; these encoders are initialized with ImageNet-22K classification weights and official BERT weights, then remain frozen while the prediction heads are trained. We use the prediction-head topology implemented by FIANet: the four backbone inputs have 1024, 512, 256, and 128 channels; three top-down fusion stages use 512 hidden channels, and

Table 4: Entropy conditioning with frozen visual features and matched prediction heads. Δ: gain over visual projection in percentage points.

Head conditioning	RefSegRS		RefCOCO	
	mIoU	Δ	oIoU	Δ
Visual projection	66.68	0	76.95	0
Aligned entropy	67.27	+0.59	78.52	+1.57

Table 5: Entropy ablation and spatial permutation. Scores: hazy RefSegRS test mIoU and RefCOCO validation oIoU. Δ: permuted minus aligned; Δ_g: relative modulation change.

(a) Local routing and scale fusion.

Configuration	RefSegRS	RefCOCO
Visual routing	66.38	77.21
Image mean entropy	66.97	77.69
Local entropy	67.53	78.76
AER	68.82	79.83

(b) Spatial intervention.

Dataset	Input	IoU	Δ	Δ_g
RefSegRS	Aligned	67.53	0	0
	Permuted	64.65	−2.88	0.43
RefCOCO	Aligned	78.76	0	0
	Permuted	75.46	−3.30	0.37

each stage contains two 3×3 convolutions with BatchNorm2d and ReLU. Bilinear interpolation uses `align_corners=True`, and a final 1×1 convolution produces two logits. This head has 20,651,010 trainable parameters, and both diagnostic conditions use this same head.

Entropy improves RefSegRS mIoU from 66.68 to 67.27 and RefCOCO oIoU from 76.95 to 78.52, gains of 0.59 and 1.57 percentage points, respectively. With the visual representation fixed, these gains support entropy as a complementary input for the matched prediction heads.

Local routing and scale fusion. Table 5(a) separates local entropy routing from global entropy in scale fusion. Within each dataset, the four configurations share the backbone, modulation bank, decoder, and training schedule. Visual routing uses a fixed four-dimensional projection $h_p = P_v v_p$ and routes from $q_p = W_q(v_p + W_d h_p)$, matching the routing parameter count of the entropy variants. Image mean entropy replaces h_p with the spatial mean of d_p, broadcast to every position; local entropy uses d_p itself. These three configurations use the visual scale score $e_m = w_s^\top h_m$. AER adds the global entropy interaction.

Local entropy improves RefSegRS mIoU from 66.38 to 67.53 and RefCOCO oIoU from 77.21 to 78.76 relative to visual routing, gains of 1.15 and 1.55 percentage points. It exceeds image mean entropy by 0.56 and 1.07 points, respectively, supporting the use of spatially varying entropy inputs on both datasets. Adding global entropy fusion yields a further 1.29 and 1.07 points, bringing AER to 68.82 mIoU and 79.83 oIoU.

Spatial correspondence. The spatial intervention changes the correspondence between entropy and image positions while retaining the entropy values. We use the local entropy configuration, cache its visual features, entropy maps, and modulation vectors generated from the sentence, then permute blocks of equal size from the local entropy input. The same permutation applies to all four entropy channels, with a common normalized block layout across routing stages. Only blocks with matching valid support are exchanged. This preserves the values and their pairing across channels.

For each image, the relative change in effective channel modulation is

$$\Delta_g = \frac{1}{4} \sum_{m=1}^{4} \frac{1}{S_m} \sum_{p=1}^{S_m} \frac{\| g_p^{(m),\mathrm{perm}} - g_p^{(m),\mathrm{orig}} \|_2}{\| g_p^{(m),\mathrm{orig}} \|_2 + \epsilon_g}, \tag{18}$$

where S_m counts valid positions and $\epsilon_g > 0$ stabilizes the denominator. In Table 5(b), block permutation reduces RefSegRS mIoU from 67.53 to 64.65 and RefCOCO oIoU from 78.76 to 75.46, decreases of 2.88 and 3.30 percentage points. The corresponding modulation changes are $\Delta_g = 0.43$ and 0.37. These paired changes indicate that the trained local router uses the correspondence between each entropy value and its image position on both datasets.

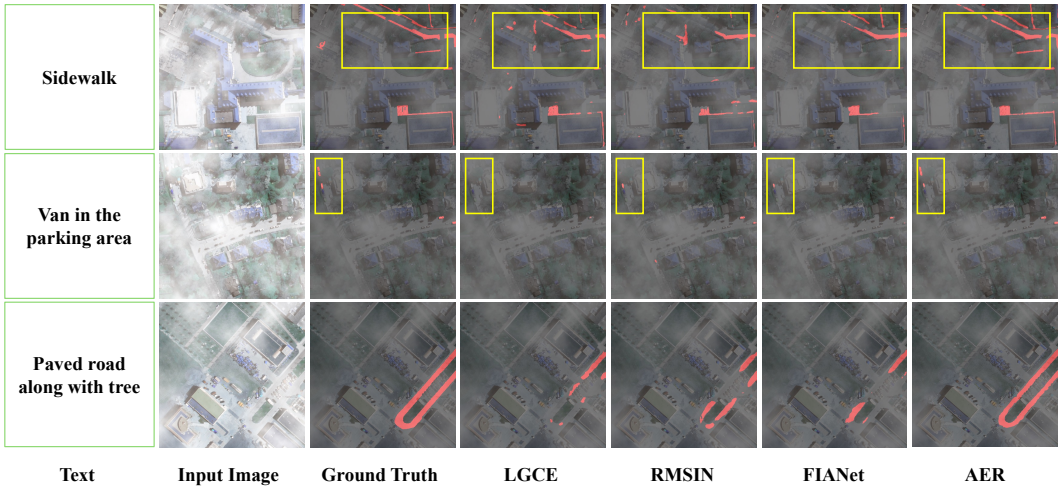


Figure 2: Qualitative comparison on RefSegRS under synthetic nonuniform haze.

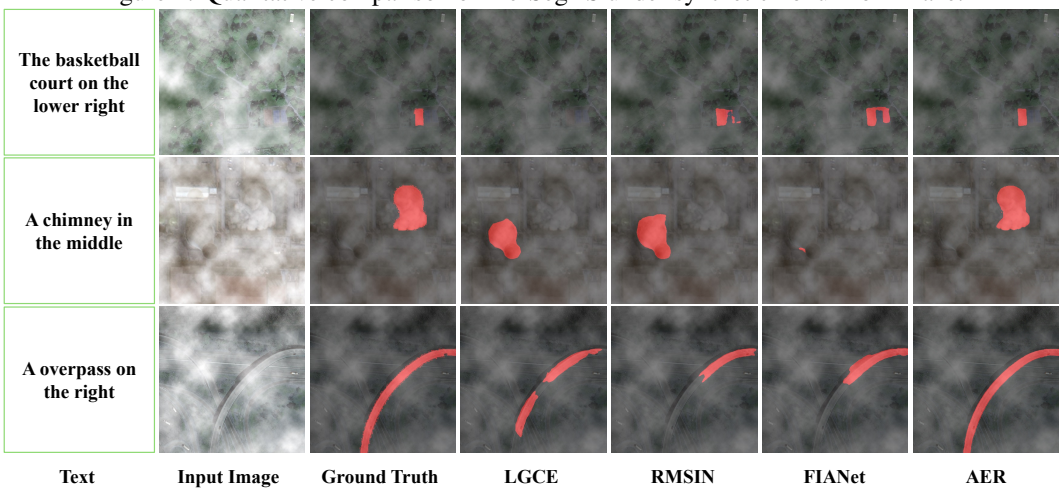


Figure 3: Qualitative comparison on RRSIS-D under synthetic nonuniform haze.

Qualitative analysis. Figures 2 and 3 present comparisons with LGCE, RMSIN, and FIANet on RefSegRS and RRSIS-D, respectively. The examples illustrate target selection and mask continuity under synthetic nonuniform haze.

6 Conclusion

We introduced Attention Entropy Routing (AER) for language-guided segmentation. AER uses aligned local entropy to route language-generated channel modulation and pooled entropy to condition multiscale fusion. Across the evaluated settings, matched-head comparisons, local/global controls, and spatial permutation support entropy as a useful conditioning signal whose effect depends on image alignment. The single-head analysis motivates this design by showing that distinct attention distributions can share an aggregated output while having different entropies; it does not provide a guarantee for the full network.

Limitations. Attention entropy is a lossy scalar: it summarizes concentration but discards the identity, arrangement, and semantic content of attended keys. The study uses one Swin-B/BERT configuration and synthetic haze, so transfer to other backbones and real degradations remains untested.

AI USE STATEMENT

We used generative AI tools solely for language polishing to improve the grammar, clarity, and readability of the manuscript. No other content, including research ideas, methodology, theoretical analysis, code, data, figures, or experimental results, was generated using these tools. We take full responsibility for the accuracy and integrity of the final manuscript.

REFERENCES

Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020.

Keyan Chen, Chenyang Liu, Bowen Chen, et al. Rsrefseg 2: Decoupling referring remote sensing image segmentation with foundation models. *IEEE Transactions on Geoscience and Remote Sensing*, 64:1–20, 2025.

Yubin Cho, Hyunwoo Yu, and Suk-Ju Kang. Cross-aware early fusion with stage-divided vision and language transformer encoders for referring image segmentation. *IEEE Transactions on Multimedia*, 26:5823–5833, 2023.

Ming Dai, Wenxuan Cheng, Jiang-jiang Liu, et al. Deris: Decoupling perception and cognition for enhanced referring image segmentation through loopback synergy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19936–19946, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16321–16330, 2021.

Yutao Hu, Qixiong Wang, Wenqi Shao, et al. Beyond one-to-one: Rethinking the referring image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4067–4077, 2023.

Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *European conference on computer vision*, pp. 59–75. Springer, 2020.

Sen Lei, Xinyu Xiao, Tianlin Zhang, Heng-Chao Li, Zhenwei Shi, and Qing Zhu. Exploring fine-grained image-text alignment for referring remote sensing image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–11, 2025.

Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753, 2018.

Chenxi Liu, Zhe Lin, Xiaohui Shen, et al. Recurrent multimodal interaction for referring image segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 1271–1280, 2017.

Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4761–4775, 2021a.

Sihan Liu, Yiwei Ma, Xiaoqing Zhang, Haowei Wang, Jiayi Ji, Xiaoshuai Sun, and Rongrong Ji. Rotated multi-scale interaction network for referring remote sensing image segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26648–26658. IEEE, 2024.

Sun-Ao Liu, Yiheng Zhang, Zhaofan Qiu, Hongtao Xie, Yongdong Zhang, and Ting Yao. Caris: Context-aware referring image segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 779–788, 2023.

Ze Liu, Yutong Lin, Yue Cao, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021b.

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 *European Conference on Computer Vision*, pp. 792–807. Springer, 2016.

Adam Pardyl, Grzegorz Rypeść, Grzegorz Kurzejamski, Bartosz Zieliński, and Tomasz Trzciński. Active visual exploration based on attention-map entropy. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 1303–1311, 2023.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11686–11695, 2022.

Jianzong Wu, Xiangtai Li, Xia Li, Henghui Ding, Yunhai Tong, and Dacheng Tao. Toward robust referring image segmentation. *IEEE Transactions on Image Processing*, 33:1782–1794, 2024.

Yuhuan Yang, Chaofan Ma, Jiangchao Yao, et al. Remamber: Referring image segmentation with mamba twister. In *European conference on computer vision*, pp. 108–126. Springer, 2024.

Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18155–18165, 2022.

Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10502–10511, 2019.

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.

Zhenghang Yuan, Lichao Mou, Yuansheng Hua, and Xiao Xiang Zhu. Rrsis: Referring remote sensing image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–12, 2024.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 40770–40803, 2023.

Jingwen Zhang, Lingling Li, Licheng Jiao, Xu Liu, Fang Liu, Wenping Ma, and Shuyuan Yang. A multiscale vision-text collaborative dual-encoder for referring rs image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 64:1–15, 2026.

A When Can the Same Output Have Different Entropies?

Proposition 1 concerns the augmented value matrix M in Eq. (2). We give its complete proof below.

Proof. The rank condition gives a nonzero vector $\delta \in \mathbb{R}^N$ with $M\delta = 0$. Its entries satisfy $\sum_i \delta_i = 0$ and $\sum_i \delta_i u_i = 0$. Start with the uniform distribution $a_i = 1/N$ and choose a nonzero t with

$$|t| < \frac{1}{N\|\delta\|_\infty}. \tag{19}$$

Then $b_i = 1/N + t\delta_i$ is strictly positive and sums to one. The weighted outputs satisfy

$$z(b) - z(a) = t \sum_i \delta_i u_i = 0. \tag{20}$$

Since b is nonuniform, $H(b) < \log N = H(a)$. Both distributions can be expressed by finite softmax scores $\log a_i$ and $\log b_i$. This proves the claim.

Checking the introductory example. For the distributions in Eq. (3), the outputs are both 1. Their entropies are $H(a) = \log 5 - 1.4 \log 2$ and $H(b) = \log 5 - 0.8 \log 2$. The difference is therefore $0.6 \log 2$. Normalizing both entropies by $\log 3$ preserves their inequality.

B How Language Changes the Response to Entropy

The routing equations provide a direct link between the entropy input and the expression. Holding the visual feature and expression fixed, the log ratio between two routing weights is

$$\log \frac{\alpha_{p,k}}{\alpha_{p,j}} = \frac{(v_p + W_d d_p)^\top W_q^\top (k_k - k_j)}{\sqrt{C}}. \tag{21}$$

Subtracting the log ratios for two entropy inputs d and d' cancels the visual term and gives

$$\log \frac{\alpha_k(d')}{\alpha_j(d')} - \log \frac{\alpha_k(d)}{\alpha_j(d)} = \frac{(d' - d)^\top W_d^\top W_q^\top (k_k - k_j)}{\sqrt{C}}. \tag{22}$$

Here the position index is omitted, and both the visual feature and expression are held fixed. The gradient with respect to the four entropy inputs is

$$\nabla_{d_p} \log \frac{\alpha_{p,k}}{\alpha_{p,j}} = \frac{W_d^\top W_q^\top (k_k - k_j)}{\sqrt{C}}. \tag{23}$$

The expression determines $k_k - k_j$ and therefore the direction and magnitude of the response to entropy.

C Implementation Details

C.1 Backbone and Optimization

AER uses a Swin-B (Liu et al., 2021b) visual encoder initialized with ImageNet-22K classification weights and the base BERT encoder (Devlin et al., 2019) initialized with its official pretrained weights. The BERT encoder has 12 layers and hidden size 768, and each visual stage is projected to $C = 512$ channels; the remaining AER parameters are randomly initialized. For the natural image datasets, we follow the LAVT (Yang et al., 2022) training configuration: images are resized to 480×480, no data augmentation is applied, AdamW uses weight decay 0.01 and an initial learning rate of 5×10^{-5} with polynomial decay, and training lasts 40 epochs with batch size 32. Each epoch visits every training object once and samples one referring expression for that object. We train a separate model on each dataset and select the checkpoint on its validation split. For the remote sensing datasets, images are resized to 480×480, AdamW uses weight decay 0.01, and the batch size is 8. RefSegRS is trained for 60 epochs with an initial learning rate of 5×10^{-5}, whereas RRSIS-D is trained for 40 epochs with an initial learning rate of 3×10^{-5}. The learning rate follows the

Table 6: Optimization settings for AER. Natural image datasets use a separate model for each dataset. The remote sensing columns refer to the synthetic haze experiments.

Setting	Natural images	RefSegRS	RRSIS-D
Input size	480×480	480×480	480×480
Optimizer	AdamW	AdamW	AdamW
Initial learning rate	5×10^{-5}	5×10^{-5}	3×10^{-5}
Weight decay	0.01	0.01	0.01
Batch size	32	8	8
Training epochs	40	60	40

repository's LambdaLR factor $(1 - x/(N_{\text{iter}}\,\text{epochs}))^{0.9}$. We use the fixed seed 2401, evaluate the validation split after every epoch, and retain the checkpoint with the highest validation overall IoU.

Table 6 summarizes the optimization settings for each dataset. The natural image configuration is used separately for RefCOCO, RefCOCO+, and RefCOCOg.

C.2 Entropy Extraction

For a single attention head, let q_p^{att} and k_i^{att} be the query and key vectors, with key dimension d_k. Over the valid key set $\mathcal{V}(p)$, attention assigns weights

$$a_{p,i} = \frac{\exp(\ell_{p,i})}{\sum_{j \in \mathcal{V}(p)} \exp(\ell_{p,j})}, \qquad \ell_{p,i} = \frac{(q_p^{\text{att}})^\top k_i^{\text{att}}}{\sqrt{d_k}} + b_{p,i}, \tag{24}$$

where $b_{p,i}$ is the positional bias. Masked and padded keys are excluded.

Let $A_{p,q}^{(m,h)}$ denote attention after relative position bias, the window mask, and softmax, before attention dropout. Attention over valid keys is normalized as

$$\widetilde{A}_{p,q}^{(m,h)} = \frac{A_{p,q}^{(m,h)}}{\sum_{q' \in \mathcal{V}_m(p)} A_{p,q'}^{(m,h)}}, \qquad q \in \mathcal{V}_m(p). \tag{25}$$

We compute $-\sum_q \widetilde{A}_{p,q}^{(m,h)} \log \widetilde{A}_{p,q}^{(m,h)} / \log N_{m,p}$ for each head and average over heads to obtain Eq. (5), using zero for singleton supports. For shifted windows, the query entropy map is assembled using the inverse window partition and inverse cyclic shift, followed by removal of padding. Bilinear interpolation aligns the entropy maps with each routing grid. Global means exclude padded query locations. Entropy is evaluated with a numerically stable logarithm. During end-to-end training, gradients propagate through attention probabilities to the backbone.

C.3 Feature Fusion and Decoding

Each ψ_m in Eq. (13) applies a 1×1 convolution, normalization, and activation. After scale fusion, the feature is refined and redistributed:

$$\begin{aligned} F_{\text{ref}} &= F_{\text{fuse}} + \text{Conv}_{3\times3}\left(\rho\left(\text{Conv}_{1\times1}(F_{\text{fuse}})\right)\right), \\ F_o^{(m)} &= \eta_m(F_{\text{ref}}), \end{aligned} \tag{26}$$

where ρ is the activation and η_m projects and resizes features to the corresponding decoder level. For decoder feature D_l, the sentence representation generates channel parameters:

$$[\gamma_l, \delta_l] = \mathcal{M}_d^{(l)}(\bar{T}), \qquad \widetilde{D}_l = \gamma_l \odot D_l + \delta_l. \tag{27}$$

The parameters are broadcast across spatial positions. Following the FIANet decoder implementation, the three top-down refinement stages use 512 channels. Their concatenated inputs are $1024 + 512$, $512 + 256$, and $512 + 128$ channels, respectively; each stage applies two 3×3 convolutions, BatchNorm2d, and ReLU. Bilinear upsampling uses `align_corners=True`, and a final 1×1 projection maps 512 channels to the two segmentation logits before the sigmoid produces $\hat{y}$ at the image resolution.

Table 7: Differences between the existing end-to-end ablations. Values are percentage point differences computed from Table 5a, using hazy RefSegRS test mIoU and RefCOCO validation oIoU.

Comparison	RefSegRS	RefCOCO
Image mean entropy minus visual routing	+0.59	+0.48
Local entropy minus image mean entropy	+0.56	+1.07
Local entropy minus visual routing	+1.15	+1.55
AER minus local entropy	+1.29	+1.07
AER minus visual routing	+2.44	+2.62

C.4 Segmentation Loss

Let Ω denote the valid output pixels. The segmentation objective combines binary cross-entropy and soft Dice loss:

$$\begin{aligned} \mathcal{L}_{\mathrm{BCE}} &= -\frac{1}{|\Omega|}\sum_{p\in\Omega}\big[y_p \log \hat{y}_p + (1-y_p)\log(1-\hat{y}_p)\big], \\ \mathcal{L}_{\mathrm{Dice}} &= 1 - \frac{2\sum_{p\in\Omega}\hat{y}_p y_p + \epsilon}{\sum_{p\in\Omega}\hat{y}_p + \sum_{p\in\Omega} y_p + \epsilon}, \\ \mathcal{L}_{\mathrm{seg}} &= \lambda_b \mathcal{L}_{\mathrm{BCE}} + (1-\lambda_b)\mathcal{L}_{\mathrm{Dice}}, \end{aligned} \tag{28}$$

where $\epsilon = 1$ is the Dice smoothing constant and $\lambda_b = 0.9$.

C.5 Modulation Bank Regularization

The auxiliary loss in Eq. (17) is computed from the modulation vectors generated from the sentence before the projection ϕ_o. For each sample and stage, we normalize the K modulation vectors with numerically stabilized norms and form their cosine similarity matrix. The squared deviation from I_K is averaged over stages, vector pairs, and samples. This is the modulation bank regularizer used throughout the end-to-end ablations.

D Interpreting the Component Comparisons

Table 7 gives the differences between the configurations in Table 5a. The comparisons for local routing share the visual scale score $e_m = w_s^\top h_m$. The comparison between local entropy and AER adds the complete global entropy interaction in Eq. (14), including its learned projections U and V_H.

Local conditioning. The visual routing control uses a fixed projection P_v and the query $q_p = W_q(v_p + W_d P_v v_p)$. Equivalently,

$$q_p = W_q(I + W_d P_v)v_p. \tag{29}$$

The local entropy variant instead supplies the aligned concentration statistics from all four stages. Its comparison with image mean entropy tests the contribution of spatial variation in these statistics.

Global fusion. The scale score in Eq. (14) can be written as

$$e_m = \left(w_s + \frac{U^\top V_H c_H}{\sqrt{r}}\right)^\top h_m. \tag{30}$$

Global entropy thus conditions the scoring vector applied to each scale descriptor. The difference between AER and local entropy measures the effect of this complete fusion branch, combining its conditioning input and learned interaction.

System and component comparisons. The benchmark comparisons in Tables 1–3 evaluate complete segmentation systems. The comparisons within the architecture in Table 7 evaluate changes to the conditioning paths while retaining the modulation bank and decoder. For example, on RefCOCO validation, visual routing reaches 77.21 oIoU and AER reaches 79.83, giving a difference of 2.62 points within the architecture.

E Experimental Protocols and Data Provenance

E.1 Synthetic Nonuniform Haze

We combine multiscale Perlin noise with the atmospheric scattering model (ASM). For each clean RGB image, we generate a two-dimensional multiscale Perlin field and normalize it to $\tilde{P}(x) \in [-1, 1]$. The base frequency is 8, the number of octaves is 4, and persistence is 0.6. We define the spatially varying optical thickness and shared transmission map as

$$\tau(x) = \beta\big(1 + \gamma\tilde{P}(x)\big), \qquad t(x) = \exp\big(-\tau(x)\big), \tag{31}$$

where $\beta \sim U(1.5, 2.5)$ controls overall haze density and $\gamma \sim U(0.30, 0.50)$ controls spatial variation. The hazy RGB channels are then generated as

$$I_{\mathrm{haz}}^{(r)}(x) = I_{\mathrm{clr}}^{(r)}(x)t(x) + A^{(r)}\big(1 - t(x)\big), \tag{32}$$

where each atmospheric light channel is sampled independently as $A^{(r)} \sim U(0.75, 0.95)$.

All hazy images are generated offline. For each distinct original image, the Perlin field and the corresponding β, γ, and atmospheric light parameters are sampled once using random seeds specific to each split, producing one fixed hazy counterpart. All referring expressions associated with the same original image share that counterpart. Training, validation, and test use these fixed images without online resampling. Every compared method uses the same data split, hazy images, referring expressions, segmentation masks, and target presence labels.

E.2 Baseline Sources and Evaluation Settings

The natural image baseline scores are taken from the respective publications. For RefSegRS and RRSIS-D, all compared methods are retrained and evaluated using their official publicly released implementations and the fixed synthetic haze images described above. The same splits, images, expressions, masks, and target presence labels are used for every method in these two datasets. The official FIANet repository uses Python 3.7 and PyTorch 1.12.1, a Swin-B patch-4/window-12 encoder initialized from the ImageNet-22K 384-pixel weights, BERT-base-uncased, input size 480×480, batch size 8, AdamW with weight decay 0.01, and a power law LambdaLR schedule $(1 - x/(N_{\mathrm{iter}}\,\mathrm{epochs}))^{0.9}$. Its RefSegRS command uses 60 epochs, learning rate 5×10^{-5}, and `num_tmem=1`; its RRSIS-D command uses 40 epochs, learning rate 3×10^{-5}, and `num_tmem=3`. The training script fixes the seed at 2401, evaluates the validation split after each epoch, and writes the checkpoint with the highest validation overall IoU as `model_best`.