UNISEEK: UNIFIED REFERRING EXPRESSION LOCALIZATION WITH COUNT-GUIDED INSTANCE SLOTS AND LIGHTWEIGHT ADD-ONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Referring expression localization identifies a target set from natural language and represents its spatial extent using bounding boxes or masks. Unifying these tasks within a multimodal large language model (MLLM) requires handling variable target counts and heterogeneous geometric outputs while preserving fine spatial information. We propose UniSeek, a single model for referring expression comprehension (REC), referring expression segmentation (RES), and their generalized counterparts (GREC/GRES). UniSeek connects the MLLM backbone to downstream lightweight decoders through instance slots trained with explicit counting supervision. A Sparse Localization Decoder and a Dense Localization Decoder predict bounding boxes and masks, respectively. A Lightweight Image Local Detail Injector provides local detail features without an external pretrained image encoder. Experiments on both RefCOCO/+/g and gRefCOCO show that UniSeek achieves leading box-level localization results and competitive segmentation performance. For example, UniSeek-8B achieves a mean REC Acc@IoU=0.5 of 93.96% across eight splits, outperforming all compared baselines on every split. These evaluations clearly demonstrate that the proposed UniSeek re-calibrates the modern state-of-the-art performance under most settings of the interested tasks here.

1 INTRODUCTION

Multimodal large language models (MLLMs) increasingly support visual understanding through natural language, but many vision-language interactions require spatially precise outputs beyond text (Wu et al., 2024a). Visual grounding associates linguistic descriptions with image regions (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016). Referring expression comprehension (REC) localizes a described target with a bounding box, whereas referring expression segmentation (RES) delineates it with a pixel-level mask (Hu et al., 2016; Li & Sigal, 2021). Their generalized counterparts, generalized referring expression comprehension (GREC) and generalized referring expression segmentation (GRES), accommodate no-target, single-target, and multi-target expressions (Liu et al., 2023a; He et al., 2023; Hu et al., 2023; Ding et al., 2026). These tasks vary along two dimensions, output granularity and instance count, while sharing expression understanding and target identification (Yu et al., 2018; Luo et al., 2020; Kamath et al., 2021). We collectively refer to REC/RES/GREC/GRES as *referring expression localization* and seek to address them within a unified model.

Prior work has advanced unified geometric outputs (Zhu et al., 2022; Wang et al., 2025a; Zhou et al., 2026), explicit target counting (Luo et al., 2025; Wang et al., 2025c), and native visual feature reuse (Ren et al., 2024; Zhang et al., 2026; Yao et al., 2026). These advances motivate combining all four tasks within one MLLM with explicit counting supervision and no external pretrained image encoder. Achieving this combination requires addressing two challenges that link target cardinality, output geometry, and spatial detail.

The first challenge is to handle variable instance counts and heterogeneous geometric outputs within a common generation interface (Lu et al., 2023; Wang et al., 2023b; Wu et al., 2024a). Sparse boxes and dense masks require different predictions, while generalized expressions yield variable target sets with a long-tailed instance-count distribution. We introduce **UniSeek** with two newly designed lightweight decoders, a Sparse Localization Decoder and a Dense Localization Decoder. Count-guided instance

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.

slots connect both decoders to a shared MLLM Backbone. Explicit group-level counting supervision trains the backbone to predict how many instances to localize. Predicted counts govern the number of slots, whose hidden states condition the task-specific decoders on the referring expression.

The second challenge is to supply multi-level semantics and local spatial detail at a modest additional cost. Multi-level and high-resolution representations support precise localization (Lin et al., 2017; Ranftl et al., 2021; Wang et al., 2021), while detail refinement improves mask boundaries (Kirillov et al., 2020; Zhang et al., 2021). We develop a Lightweight Image Local Detail Injector comprising a Detailed Patch Embedder, a Shared Semantic Feature Pyramid Network, and a Detailed Embedding Injector. This module combines native multi-layer semantics with gated sub-patch detail injection (Bachlechner et al., 2021), supplying both decoders without an external pretrained image encoder.

In summary, our main contributions are as follows:

1. We propose UniSeek, a unified model for REC/RES/GREC/GRES with explicit counting supervision and newly designed lightweight task-specific decoders. Count-guided instance slots connect a shared MLLM Backbone to two task-specific decoders with approximately 23M parameters in total, providing a common interface for variable instance counts and heterogeneous geometric outputs.
2. We develop a Lightweight Image Local Detail Injector that combines native multi-level semantics with local spatial detail, supporting both decoders without an external pretrained image encoder. The injector introduces approximately 5M additional parameters.
3. UniSeek-8B achieves 93.96% mean REC Acc@IoU=0.5 across eight RefCOCO/+/g splits, outperforming all compared baselines on each split. On gRefCOCO, it ranks first in five of six GREC split–metric pairs, while UniSeek maintains competitive RES/GRES performance.

2 Related Work

Unified Box and Mask Prediction. Related foundations include box localization (Yu et al., 2018; Deng et al., 2021; Li et al., 2022a; Kamath et al., 2021; Zhang et al., 2023; Liu et al., 2024), language-conditioned segmentation (Hu et al., 2016; Ding et al., 2021; Yang et al., 2022; Wang et al., 2022b), and joint box–mask prediction (He et al., 2017; Luo et al., 2020; Li & Sigal, 2021). Geometry interfaces now include coordinate generation (Chen et al., 2023a; Bai et al., 2024; Peng et al., 2024; Jiang et al., 2026), atomic box decoding (Wang et al., 2026), and sequence-based task interfaces (Chen et al., 2022a; Wang et al., 2022a). SeqTR, Pix2Seq v2, and Florence-2 serialize boxes and mask contours (Zhu et al., 2022; Chen et al., 2022b; Xiao et al., 2024a), with contour prediction (Liu et al., 2023b) trading representation accuracy against sequence length. Other approaches represent masks through textual labels (Lan et al., 2025) or discrete tokens (Lu et al., 2023; Wang et al., 2025a; Zhou et al., 2026). An alternative uses MLLM hidden states to condition task-specific decoders (Wu et al., 2024a; Lai et al., 2024; Rasheed et al., 2024; Zhang et al., 2024d), with segmentation approaches also drawing on promptable foundation models (Kirillov et al., 2023; Ravi et al., 2025). UniSeek follows the decoder-based formulation, using instance slots as a common interface for continuous box prediction and dense mask prediction.

Target Cardinality Modeling. Single-target REC/RES (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016) extend to no-target and multi-target settings in GREC/GRES (Liu et al., 2023a; He et al., 2023; Hu et al., 2023; Jiang et al., 2025). ReLA represents the target set through a union mask, while GSVA (Xia et al., 2024) organizes expression-conditioned masks with segmentation tokens and handles absent targets with a rejection token; neither explicitly predicts the instance count per expression. Ding et al. (2026) introduce explicit counting in GREx, which jointly predicts boxes, masks, and counts. CoHD (Luo et al., 2025) adds counting supervision to hierarchical decoding, and HieA2G (Wang et al., 2025c) classifies five count categories. GREx unifies GREC/GRES, and HieA2G evaluates all four tasks with task-specific downstream fine-tuning, both without MLLM backbones. UniSeek couples explicit group-level counting supervision with MLLM instance-slot generation: constrained decoding makes each requested task section's slot count match the predicted exact count, connecting cardinality prediction to box and mask output organization.

Native Visual Feature Reuse and Local Detail Supplementation. Patch-based encoders such as ViT (Dosovitskiy et al., 2021) and ViT-based CLIP (Radford et al., 2021), together with hierarchical

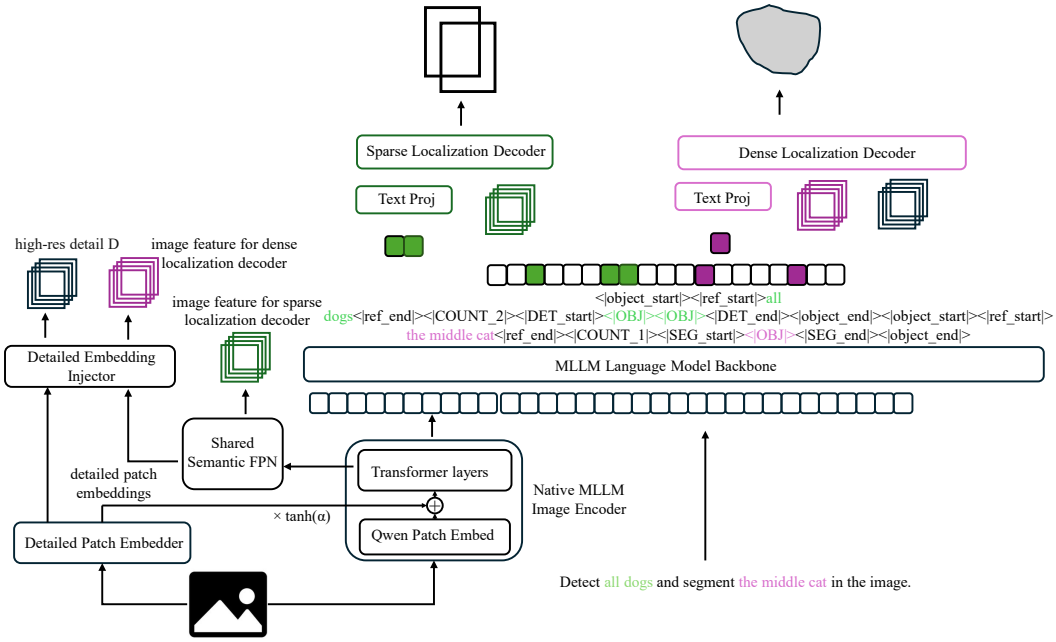


Figure 1: Overview of the UniSeek architecture.

spatial merging in Swin (Liu et al., 2021), have advanced image understanding, but spatial compression can limit the fine-grained boundary information available for dense prediction. To supply spatial detail, LISA (Lai et al., 2024) and GSVA (Xia et al., 2024) use an additional SAM image encoder. Chen et al. (2023b) and Li et al. (2022b) explore adapting general-purpose visual backbones for dense prediction through ViT-Adapter and ViTDet, respectively. Within MLLMs, PixelLM (Ren et al., 2024) reuses native multi-layer features with a lightweight pixel decoder, while OMG-LLaVA (Zhang et al., 2024b) shares a segmentation-capable encoder. SELF1E (Zhang et al., 2026) retains pre-compression spatial features and incorporates language-model semantic residuals for segmentation-token matching. Qwen3-VL-Seg (Yao et al., 2026) combines native multi-layer features and shallow CNN pixel features through box-guided gating. UniSeek extracts sub-patch detail from preprocessed patch pixels and injects it into the native visual encoding pathway. Both localization decoders share the resulting multi-level semantic features, while the dense branch additionally receives high-resolution detail for mask decoding and query-conditioned refinement, without an external pretrained image encoder.

3 Method

3.1 Overview

Given an image and a text instruction, UniSeek predicts all referred instances as bounding boxes or masks. Each referring expression receives an explicit count and corresponding instance slots; a zero count yields an empty output (Appendix C).

As illustrated in Figure 1, a Qwen3-VL (Bai et al., 2025) MLLM Backbone generates these count-guided instance slots, whose hidden states condition the Sparse and Dense Localization Decoders. Both decoders reuse native image features through a Shared Semantic Feature Pyramid Network. A Detailed Patch Embedder injects local information into native visual tokens, while a Detailed Embedding Injector and query-conditioned refinement provide additional detail for mask prediction. The enhanced visual tokens enter the language backbone through native spatial merging and projection.

3.2 Count-Guided Instance Slots with Explicit Counting Supervision

Structured Output and Constrained Decoding. Each reference group is serialized as an *object block* containing the referring expression, its count token, and the requested task sections. Each <|OBJ|> token and its hidden state constitute an *instance slot*, conditioning the corresponding

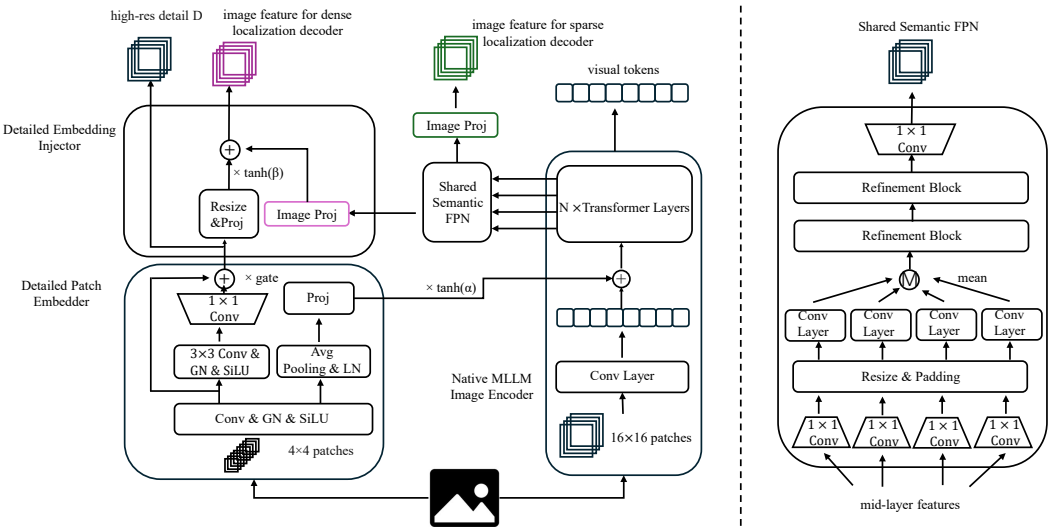


Figure 2: Lightweight Image Local Detail Injector. Left: the Detailed Patch Embedder and the Detailed Embedding Injector. Right: the Shared Semantic Feature Pyramid Network. Each Refinement Block uses a 3×3 convolution, Group Normalization with 32 groups, and a SiLU activation.

decoder to produce one box or mask. Without explicit counts, the model must learn cardinality implicitly by deciding when to stop generating slots. Termination supervision occurs only at the end of each task section, making the stopping signal sparse relative to slot-generation supervision for multi-target expressions. This sparsity makes the stopping decision harder to learn, motivating explicit count prediction as direct cardinality supervision. During inference, constrained decoding (Scholak et al., 2021) enforces the predicted exact count as the slot quota in each requested task section. A zero count closes the section without slots; counts beyond the exact-count range use an overflow mechanism. Appendix D specifies the vocabulary, serialization, and decoding rules.

Counting Supervision. The autoregressive loss $\mathcal{L}_{\mathrm{AR}}$ supervises all valid answer positions, including count tokens and instance slots. We additionally apply weighted cross-entropy to the vocabulary logits restricted to exact-count classes. For sample b, let $\mathcal{G}_b^{\mathrm{exact}}$ contain its valid exact-count groups and $\mathcal{B}_{\mathrm{grp}}$ contain samples with at least one such group. With count logits $\mathbf{z}_{b,g}^{\mathrm{cnt}}$ and ground-truth count $n_{b,g}$, the auxiliary loss is

$$\mathcal{L}_{\mathrm{cnt}}^{\mathrm{grp}} = \frac{1}{|\mathcal{B}_{\mathrm{grp}}|} \sum_{b \in \mathcal{B}_{\mathrm{grp}}} \frac{1}{|\mathcal{G}_b^{\mathrm{exact}}|} \sum_{g \in \mathcal{G}_b^{\mathrm{exact}}} w_{n_{b,g}}^{\mathrm{grp}} \, \mathrm{CE}(\mathbf{z}_{b,g}^{\mathrm{cnt}}, n_{b,g}). \tag{1}$$

We average within each sample before averaging across samples, preventing examples with more reference groups from dominating supervision. The weights $w_{n_{b,g}}^{\mathrm{grp}}$ rebalance the long-tailed count distribution using shared weights within count intervals. Appendix G.5 gives the intervals, weights, and auxiliary-loss warmup.

3.3 Lightweight Image Local Detail Injector

Detailed Patch Embedder. As illustrated in Figure 2, we extract sub-patch features $\mathbf{D}_j = \phi_{\mathrm{detail}}(\mathbf{P}_j)$ from each patch's pixels $\mathbf{P}_j$. These features follow two paths. First, pooled detail is projected by W_{v} and injected into the native patch embedding $\mathbf{e}_j$ through a learnable scalar gate α:

$$\widetilde{\mathbf{e}}_j = \mathbf{e}_j + \tanh(\alpha)\, W_{\mathrm{v}}\, \mathrm{LN}(\mathrm{AvgPool}(\mathbf{D}_j))\,. \tag{2}$$

Injection precedes positional encoding and the visual Transformer, preserving the visual sequence length and ordering. Second, the unpooled features are assembled into a high-resolution detail map $\mathbf{D}$, with a gated residual convolutional module mixing information across patch boundaries.

Shared Semantic Feature Pyramid Network. Following multi-layer feature aggregation (Ranftl et al., 2021; Ke et al., 2023), we extract features from three intermediate layers and the final layer of the native image encoder before spatial merging. The features are restored to spatial maps,

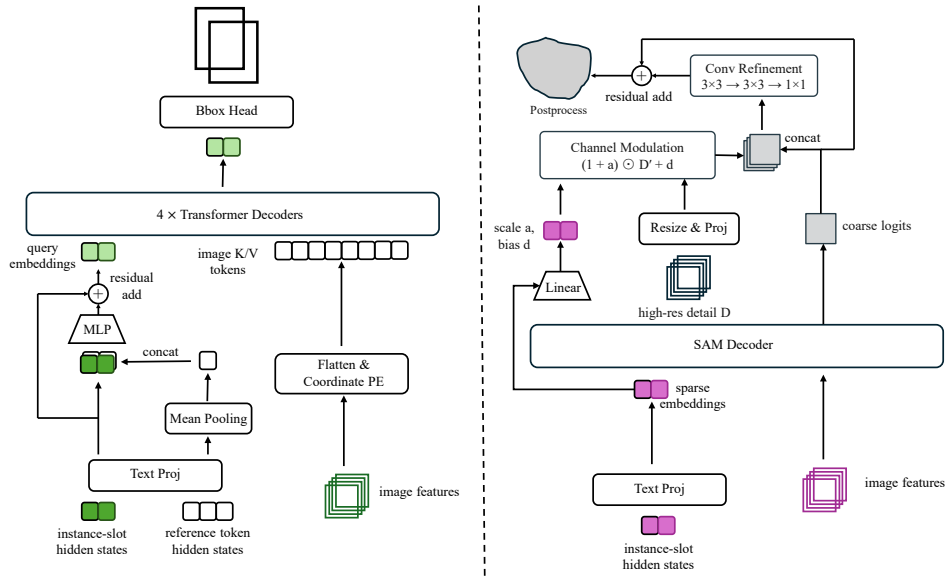


Figure 3: Task-specific decoders. Left: the Sparse Localization Decoder. Right: the Dense Localization Decoder.

projected and aligned to a common grid, then averaged and refined into $\mathbf{F}_{\text{sem}}$. This shared semantic representation supplies both localization decoders through their respective Image Projection modules.

Detailed Embedding Injector. The dense branch additionally receives unpooled detail through gated residual fusion:

$$\mathbf{F}_{\text{seg}} = \mathbf{E}_{\text{seg}} + \tanh(\beta)\,\Pi_{\text{d}}(\text{Resize}(\mathbf{D}))\,, \qquad \mathbf{E}_{\text{seg}} = \Pi_{\text{seg}}(\mathbf{F}_{\text{sem}}). \tag{3}$$

Here, Π_{seg} projects semantic features, Π_{d} projects spatially aligned detail, and β is a learnable scalar gate. The dense decoder uses $\mathbf{F}_{\text{seg}}$ for initial mask prediction and retains $\mathbf{D}$ for subsequent refinement. The sparse decoder directly projects $\mathbf{F}_{\text{sem}}$, sharing early detail enhancement without decoder-side detail fusion. Appendix G.2 provides the operator configurations and initialization.

3.4 Task-Specific Decoders

The decoders in Figure 3 predict geometry for the instance slots supplied by each task section.

Sparse Localization Decoder. For REC/GREC, we fuse each instance-slot representation with its referring-expression context to condition the localization query on the expression. Let $\mathbf{h}_{g,i}$ denote the hidden state of instance slot i in group g. Let $\{\mathbf{h}^{\text{ref}}_{g,k}\}_{k=1}^{K_g}$ denote the reference-token hidden states for that group. A shared Text Projection W_{t} yields the slot embedding $\mathbf{s}_{g,i} = W_{\text{t}}\mathbf{h}_{g,i}$ and mean-pooled reference context $\mathbf{c}_g = \frac{1}{K_g}\sum_{k=1}^{K_g} W_{\text{t}}\mathbf{h}^{\text{ref}}_{g,k}$. A residual Multilayer Perceptron ϕ_{q} then forms the reference-conditioned query:

$$\mathbf{u}_{g,i} = \mathbf{s}_{g,i} + \phi_{\text{q}}([\mathbf{s}_{g,i}; \mathbf{c}_g])\,. \tag{4}$$

Here, $[\,\cdot\,;\,\cdot\,]$ denotes channel-wise concatenation. Following query-based detection (Carion et al., 2020; Zhu et al., 2021), Transformer decoder layers (Vaswani et al., 2017) read projected $\mathbf{F}_{\text{sem}}$ with coordinate positional embeddings, and a box head predicts normalized coordinates.

Box regression constrains query–region correspondence only indirectly, so we also align fused queries with target regions before Transformer decoding. Each candidate j is a ground-truth box provided for the query's image during training, with $\mathbf{v}_j$ averaged over its position-encoded visual-grid features. For each query i, same-image regions form positive and negative sets $\mathcal{P}_i$ and $\mathcal{N}_i$ according to overlap with its target box. With $\mathcal{A}_{\text{RR}}$ denoting queries with nonempty positive and negative sets, we minimize

$$\mathcal{L}_{\text{RR}} = -\frac{1}{|\mathcal{A}_{\text{RR}}|}\sum_{i\in\mathcal{A}_{\text{RR}}} \log \frac{\sum_{j\in\mathcal{P}_i}\exp(\text{sim}(\mathbf{u}_i,\mathbf{v}_j)/T)}{\sum_{j\in\mathcal{P}_i\cup\mathcal{N}_i}\exp(\text{sim}(\mathbf{u}_i,\mathbf{v}_j)/T)}, \tag{5}$$

where sim is cosine similarity and T is the temperature; we set $\mathcal{L}_{\mathrm{RR}} = 0$ if $\mathcal{A}_{\mathrm{RR}} = \emptyset$. The sparse loss combines this alignment term with L1 box regression and generalized IoU (Rezatofighi et al., 2019).

Dense Localization Decoder. For RES/GRES, each projected slot acts as a text embedding prompt for SAM's prompt encoding and mask decoding architecture (Kirillov et al., 2023). We randomly initialize these modules and use $\mathbf{F}_{\mathrm{seg}}$ as their image embedding, without an external image encoder. The decoder produces initial mask logits $\hat{\mathbf{m}}^{c}_{g,i}$, which are refined using instance-specific local detail. We align $\mathbf{D}$ to the mask resolution as $\mathbf{D}'$ and predict channel scales $\mathbf{a}_{g,i}$ and shifts $\mathbf{d}_{g,i}$ from the projected query (Perez et al., 2018):

$$\hat{\mathbf{m}}_{g,i} = \hat{\mathbf{m}}^{c}_{g,i} + \phi_{\mathrm{m}}\big(\big[(1 + \mathbf{a}_{g,i}) \odot \mathbf{D}' + \mathbf{d}_{g,i}; \hat{\mathbf{m}}^{c}_{g,i}\big]\big)\,. \tag{6}$$

Here, ϕ_{m} predicts a convolutional residual, and $\odot$ applies channel-wise scales broadcast across spatial positions. Both initial and refined masks receive binary cross-entropy and Dice supervision (Milletari et al., 2016). Appendix G.3 details both decoders and output processing.

Joint Training Objective. The training objective combines autoregressive, counting, and localization losses:

$$\mathcal{L}_{\mathrm{tot}} = \mathcal{L}_{\mathrm{AR}} + \rho(t)\lambda_{\mathrm{grp}}\mathcal{L}^{\mathrm{grp}}_{\mathrm{cnt}} + \lambda_{\mathrm{det}}\mathcal{L}_{\mathrm{det}} + \lambda_{\mathrm{seg}}\mathcal{L}_{\mathrm{seg}}. \tag{7}$$

The factor $\rho(t)$ gradually activates auxiliary counting supervision. Appendix G.5 specifies the loss components, coefficients, and handling of groups without targets.

4 Experiments

4.1 Datasets

We trained UniSeek on 700,000 sample entries, equally divided between bounding box localization and pixel-level segmentation. The collection combines referring-expression data with generic detection and segmentation data. We used official dataset splits and excluded from the merged training set any image appearing in a validation or test split of any source dataset. R-RefCOCO, R-RefCOCO+, and R-RefCOCOg (Wu et al., 2024b) provide challenging no-target expressions for GREC/GRES and are included in these totals. Appendix E details the training sources (Table 10)and task and instance-count distributions (Figure 9).

4.2 Implementation Details

We initialized the MLLM Backbone from Qwen3-VL-Instruct (Bai et al., 2025) and used supervised fine-tuning with autoregressive, localization, and counting objectives. At checkpoint save time, only the VLM language component and output head were trainable; visual modules and localization decoders were frozen (Appendix I). We used AdamW (Loshchilov & Hutter, 2019) with learning-rate warmup and cosine decay, together with mixed precision, gradient checkpointing, and DeepSpeed ZeRO (Rajbhandari et al., 2020). The group-level counting loss used a separate warmup and instance-count interval reweighting. Complete model configurations, hyperparameters, and training resources are provided in Appendix G; the validation protocol and checkpoint-selection score are detailed in Appendix G.6.

4.3 Main Results

We evaluated UniSeek on REC, RES, GREC, and GRES to assess referring expression localization across different target cardinalities and geometric output granularities. The results show leading bounding box localization performance and competitive pixel-level segmentation within a unified architecture. Detailed definitions of the evaluation metrics are provided in Appendix F.

Referring Expression Comprehension Our model achieved broad gains in single-target bounding box localization (Table 1). UniSeek-8B achieved the highest accuracy on all eight evaluation splits, tying with UniSeek-2B on RefCOCOg test. On RefCOCO+ testB, it exceeded the strongest baseline by 3.99 percentage points. UniSeek-2B outperformed WeDetect-Ref, which uses the same Qwen3-VL-2B backbone, on seven of the eight splits.

Table 1: REC on RefCOCO/+/g. Results are $\mathrm{Acc@IoU} = 0.5$. All results are percentages; higher is better. Bold and underlined values denote the highest and second-highest distinct scores per column, respectively; ties receive the same formatting. * marks models fine-tuned on the corresponding dataset. These units and conventions apply to all subsequent main-result tables.

Method	MLLM backbone size	RefCOCO			RefCOCO+			RefCOCOg	
		Val.	Test A	Test B	Val.	Test A	Test B	Val.	Test
Models without MLLMs									
EEVG (Chen et al., 2024a)	/	90.47	92.73	87.72	81.79	87.80	74.94	85.19	84.72
Grounding DINO (Liu et al., 2024)	/	84.61	88.65	80.51	73.67	82.19	64.10	78.33	78.11
LoA-Trans* (Huang & Satoh, 2024)	/	87.75	90.60	84.81	79.56	84.95	71.75	80.80	80.18
OneRef* (Xiao et al., 2024b)	/	91.89	94.31	88.58	86.38	90.38	79.47	86.82	87.32
RECANTFormer* (Hemanthage et al., 2024)	/	–	83.08	76.51	–	70.43	58.08	65.40	–
SimVG* (Dai et al., 2024)	/	92.93	94.70	90.28	87.28	91.64	82.41	87.99	89.15
PropVG* (Dai et al., 2025b)	/	92.70	95.07	89.58	87.27	90.87	81.26	88.15	88.30
HieA2G (Wang et al., 2025c)	/	87.80	90.30	84.00	80.70	85.60	72.90	83.70	83.80
InstanceVG (Dai et al., 2025a)	/	94.42	96.04	92.39	90.12	92.89	85.94	89.58	90.62
MLLMs Directly Generating Bounding Boxes									
CogVLM (Wang et al., 2024)	17B	92.76	94.75	88.99	88.68	92.91	83.39	89.75	90.79
Ferret (You et al., 2024)	13B	89.48	92.41	84.36	82.81	88.14	75.17	85.83	86.34
Kosmos-2 (Peng et al., 2024)	1.6B	52.32	57.42	47.26	45.48	50.73	42.24	60.57	61.65
ARGUS (Man et al., 2025)	8B	89.80	92.90	85.40	84.70	90.10	77.10	86.70	85.20
HiMTok (Wang et al., 2025a)	8B	92.90	94.70	89.30	87.60	91.50	81.50	88.50	89.00
CogVLM-Grounding + SPR (Qiu et al., 2025)	17B	92.95	94.87	89.15	88.83	92.95	83.84	90.01	90.96
Group Revision* (Liu et al., 2026)	7B	90.72	92.36	86.67	84.49	89.28	74.26	87.64	87.97
Rex-Omni (Jiang et al., 2026)	2B	86.60	89.50	82.80	79.60	84.80	71.40	85.30	86.20
MLLMs with Bounding Box Decoders									
LLaVA-Grounding (Zhang et al., 2024a)	7B	89.16	–	–	81.68	–	–	84.82	–
TCRT* (Chen et al., 2025)	8B	91.07	92.85	86.24	85.75	90.76	77.59	86.81	87.13
WeDetect-Ref* (Fu et al., 2026)	2B	94.30	95.60	92.60	88.10	92.60	83.10	92.00	92.20
UniSeek	2B	94.41	95.51	92.89	90.11	93.19	85.40	92.39	**93.59**
UniSeek	8B	**95.73**	**96.52**	**94.42**	**92.45**	**94.86**	**89.93**	**94.17**	**93.59**

Table 2: RES on RefCOCO/+/g. Results are cumulative IoU (cIoU), also called overall IoU (oIoU), computed as the total intersection divided by the total union across expressions. † marks models fine-tuned on the corresponding dataset whose source papers use the equivalent oIoU label. All entries are included in the per-column rankings.

Method	MLLM backbone size	RefCOCO			RefCOCO+			RefCOCOg	
		Val.	Test A	Test B	Val.	Test A	Test B	Val.	Test
Task-specific Vision-Language Models									
OneRef† (Xiao et al., 2024b)	/	79.83	81.86	76.99	74.68	77.90	69.58	74.06	74.92
ReMamber† (Yang et al., 2024)	/	74.54	76.74	70.89	65.00	70.78	57.53	63.90	64.00
PropVG† (Dai et al., 2025b)	/	81.80	83.74	79.33	74.81	78.72	69.15	75.54	77.40
MLLMs Directly Generating Mask Boundaries									
VistaLLM (Pramanick et al., 2024)	7B	74.50	76.00	72.70	69.10	73.70	64.00	69.00	70.90
MLLMs Generating Mask Tokens									
HiMTok (Wang et al., 2025a)	8B	81.10	81.20	79.20	77.10	78.80	71.50	75.80	76.70
SAMTok (Zhou et al., 2026)	3B	82.40	83.90	79.90	78.40	81.80	74.80	79.00	79.10
MLLMs with a Single Segmentation Token									
SELF1E (Zhang et al., 2026)	2B	80.20	82.10	77.60	74.60	79.10	69.20	77.00	77.80
MLLMs with Mask Decoders									
GLaMM (Rasheed et al., 2024)	7B	79.50	83.20	76.90	72.60	78.70	64.60	74.20	74.90
GroundHog-7B (Zhang et al., 2024c)	7B	78.50	79.90	75.70	70.50	75.00	64.90	74.10	74.60
GSVA-7B (Xia et al., 2024)	7B	76.40	77.40	72.80	64.50	67.70	58.60	71.10	72.00
LLaVA-Grounding (Zhang et al., 2024a)	7B	77.13	–	–	68.79	–	–	71.54	–
PerceptionGPT (Pi et al., 2024)	7B	75.10	78.60	71.70	68.50	73.90	61.30	70.30	71.70
PixelLM (Ren et al., 2024)	7B	73.00	76.50	68.20	66.30	71.70	58.30	69.30	70.50
SAM4MLLM (Chen et al., 2024b)	8B	79.80	82.70	74.70	74.60	80.00	67.20	75.50	76.40
u-LLaVA (Xu et al., 2023)	7B	83.00	**85.10**	80.50	77.10	81.70	70.60	77.10	78.00
MVP-LM (Xiang et al., 2025)	1.3B	**83.60**	**85.10**	**82.50**	73.90	76.80	65.40	75.10	75.60
RAS-13B (Cao et al., 2025)	13B	79.40	82.60	75.90	72.20	77.30	64.70	73.20	74.50
SegLLM-7B (Wang et al., 2025b)	7B	80.20	81.50	75.40	70.30	73.00	62.50	72.60	73.60
UniSeek	2B	81.73	83.10	80.00	77.44	81.09	73.04	78.74	**81.47**
UniSeek	8B	82.64	83.90	81.61	**79.45**	**82.67**	**76.44**	**79.31**	80.40

Referring Expression Segmentation Our model achieved competitive single-target segmentation without introducing an external pretrained image encoder (Table 2). UniSeek-8B achieved the highest cIoU on all three RefCOCO+ splits and RefCOCOg validation. UniSeek-2B achieved the highest cIoU on RefCOCOg test, exceeding the strongest baseline by 2.37 percentage points.

Generalized Referring Expression Comprehension Our model performed strongly in localizing complete target sets with variable instance counts (Table 3). UniSeek-8B achieved the highest strict set-matching accuracy on both test splits, exceeding the strongest baseline on testB by 13.10

Table 3: GREC on val/testA/testB. Precision is $\mathrm{Pr@}(\mathrm{F1}=1, \mathrm{IoU} \geq 0.5)$; N-acc is no-target accuracy.

Method	MLLM backbone size	Val.		Test A		Test B	
		Precision	N-acc	Precision	N-acc	Precision	N-acc
Models without MLLMs							
UNINEXT* (Yan et al., 2023)	/	58.20	50.60	46.40	49.30	42.90	48.20
Grounding DINO (Liu et al., 2024)	/	46.18	81.44	38.60	76.06	35.87	<u>80.58</u>
RECANTFormer* (Hemanthage et al., 2024)	/	57.73	52.70	57.82	53.38	49.49	54.53
SimVG* (Dai et al., 2024)	/	62.10	54.70	64.60	57.20	54.80	57.20
HieA2G* (Wang et al., 2025c)	/	61.30	56.10	66.00	60.10	56.50	56.00
PropVG* (Dai et al., 2025b)	/	67.80	60.30	68.80	69.90	59.00	65.00
InstanceVG (Dai et al., 2025a)	/	72.20	72.80	<u>70.20</u>	71.10	60.80	65.20
MLLMs Directly Generating Bounding Boxes							
Ferret* (You et al., 2024)	7B	54.80	48.90	49.50	45.20	43.50	43.80
MLLMs with Task-Specific Decoders							
ROD-MLLM (Yin et al., 2025)	7B	53.20	51.80	55.40	63.70	51.80	57.90
UniSeek	2B	**77.21**	<u>87.39</u>	69.79	<u>77.05</u>	<u>67.58</u>	71.77
UniSeek	8B	<u>76.46</u>	**90.01**	**77.28**	**90.70**	**73.90**	**80.88**

Table 4: GRES on val/testA/testB.

Method	MLLM backbone size	Val.			Test A			Test B		
		gIoU	cIoU	N-acc	gIoU	cIoU	N-acc	gIoU	cIoU	N-acc
Models without MLLMs										
InstanceVG (Dai et al., 2025a)	/	73.36	<u>69.22</u>	72.84	<u>75.21</u>	**74.51**	71.09	66.74	65.67	65.18
MLLMs Generating Mask Tokens										
SAMTok (Zhou et al., 2026)	3B	70.50	68.20	60.50	73.80	73.00	58.80	65.90	65.40	55.30
SAMTok (Zhou et al., 2026)	7B	74.40	**71.40**	67.50	**77.80**	72.90	77.40	68.50	68.20	59.80
MLLMs with Mask Decoders										
GSVA (Xia et al., 2024)	13B	63.20	62.38	54.51	69.52	69.86	57.84	62.06	60.77	58.30
SAM4MLLM (Chen et al., 2024b)	7B	68.96	66.33	62.96	70.54	70.13	65.82	63.98	63.21	61.61
Ground-V (Zong et al., 2025)	7B	46.70	48.40	36.40	63.20	63.70	39.90	51.30	53.90	44.10
RAS (Cao et al., 2025)	13B	68.86	64.44	57.19	74.83	<u>74.40</u>	54.86	66.74	64.51	54.13
UniSeek	2B	**77.21**	68.37	**84.33**	74.65	72.69	<u>85.30</u>	<u>74.20</u>	<u>74.39</u>	<u>72.97</u>
UniSeek	8B	<u>76.55</u>	68.25	<u>82.95</u>	74.36	73.57	**93.19**	**75.53**	**74.99**	**82.82**

percentage points. UniSeek-2B also surpassed all listed baselines in testB strict set-matching accuracy and achieved the highest validation accuracy at 77.21%. For no-target recognition, UniSeek-8B ranked first across the validation and both test splits.

Generalized Referring Expression Segmentation UniSeek-8B achieved the highest gIoU and cIoU on testB and the highest no-target accuracy on both test splits (Table 4). On testB, it exceeded the strongest baselines by 7.03 percentage points in gIoU and 6.79 percentage points in cIoU. On testA, it ranked fifth in gIoU and third in cIoU among all listed models. On validation, UniSeek-2B achieved the highest gIoU and no-target accuracy.

Additional evaluations. Appendix A examines localization robustness through F1 at stricter IoU thresholds on RefCOCO/+/g validation, compares UniSeek with SAM2-base segmentation pipelines prompted by Qwen- or UniSeek-generated boxes on the same validation splits, and assesses exact counting accuracy across target cardinalities.

4.4 Ablation Studies

All ablations used Qwen3-VL-2B-Instruct with a reduced training set and budget; Appendix G.7 details the differences from the main experiments. All ablation results are percentages; higher is better, and bold denotes the best result in each column.

Visual features and local detail. The Shared Semantic Feature Pyramid Network (FPN) improved REC but did not improve RES on its own (Table 5). This may reflect its emphasis on semantic aggregation, which does not directly supply the fine boundary information needed for segmentation. The

Table 5: Visual ablations on RefCOCO.

Config.	Val.		Test A		Test B	
	Acc	cIoU	Acc	cIoU	Acc	cIoU
w/o VC	90.86	73.09	93.00	75.10	87.72	70.08
FPN only	91.70	73.07	93.55	74.95	89.99	68.85
w/o DEI	92.01	73.38	**93.72**	75.63	90.17	70.12
Full	**92.03**	**74.69**	93.68	**77.26**	**90.22**	**71.80**

Table 6: Counting ablations on gRefCOCO.

Config.	Val.		Test A		Test B	
	N-acc	T-acc	N-acc	T-acc	N-acc	T-acc
w/o CT, CL	66.15	97.61	69.47	96.38	66.45	97.27
CT only	65.70	99.04	72.88	83.83	67.90	88.81
Full	**69.83**	**99.72**	**78.67**	**97.47**	**70.83**	**97.83**

Detailed Embedding Injector (DEI), designed specifically for the segmentation branch, consistently improved RES while leaving REC largely unchanged. In Table 5, VC denotes the added visual components; Acc denotes REC localization accuracy at an IoU threshold of 0.5 (Acc@IoU=0.5), and cIoU denotes RES cumulative IoU.

Explicit counting supervision. Count tokens (CT) alone were insufficient for reliable target-presence recognition (Table 6): under autoregressive supervision alone, improved no-target accuracy came at the expense of target-present accuracy on the test splits. Adding the auxiliary counting loss (CL) improved both metrics across all splits, supporting explicit supervision beyond the count-token output format. Results are aggregated over GREC/GRES, with N-acc measuring empty predictions on no-target expressions and T-acc measuring nonempty predictions on target-present expressions.

Mask refinement. Query-conditioned mask refinement (MR) consistently improved segmentation (Table 7). In Tables 7 and 8, ✗ and ✓ indicate that the corresponding component is disabled and enabled, respectively. Both variants retain DEI, so this gain supports supplementing decoder-side detail injection with refinement of the predicted masks.

Reference–region alignment. The reference–region alignment loss (RR) yielded modest, consistent gains in bounding box localization across all splits (Table 8). With box regression losses retained in both variants, this gain supports directly aligning reference-conditioned queries with target regions.

Table 7: Mask refinement on RefCOCO RES (cIoU).

MR	Val.	Test A	Test B
✗	74.57	76.32	70.72
✓	**74.69**	**77.26**	**71.80**

Table 8: Reference–region alignment on RefCOCO REC (Acc@IoU=0.5).

RR	Val.	Test A	Test B
✗	91.86	93.34	89.74
✓	**92.03**	**93.68**	**90.22**

4.5 Qualitative Analysis

Appendix B presents qualitative results on the validation and test splits. REC/RES examples (Figure 6) illustrate localization with multiple distractors, part-level targets, cluttered backgrounds, misspelled referring expressions, blurred targets, and occlusion. For GREC/GRES (Figure 7), UniSeek returns empty predictions for no-target expressions with visually similar distractors and demonstrates robustness to varying target cardinalities, including two, three, and four instances.

5 Conclusion

We presented UniSeek, a unified MLLM for REC, RES, GREC, and GRES that handles variable target cardinalities and both bounding box and mask outputs. Count-guided instance slots connect explicit group-level counting supervision to lightweight Sparse and Dense Localization Decoders, supporting no-target, single-target, and multi-target expressions through a common interface. The Lightweight Image Local Detail Injector combines native multi-level visual semantics with local spatial detail, enabling precise localization without an external pretrained image encoder. Experiments on RefCOCO/+/g and gRefCOCO demonstrate leading bounding box localization and competitive segmentation performance. UniSeek-8B achieves 93.96% mean REC accuracy across eight splits and ranks first in five of six GREC split–metric pairs. UniSeek-2B achieves the highest GREC validation strict set-matching accuracy, completing the model family's leading results across all six pairs. Together, these findings demonstrate the effectiveness of combining explicit cardinality modeling, shared visual representations, and task-specific geometric decoding for unified referring expression localization.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing, identify related work, and locate evaluation metrics reported in that work. We did not use these tools to draft sections of the paper or generate synthetic datasets. The use of generative AI to prove mathematical claims is not applicable to this work. We have reviewed all AI-assisted work, verified the cited references, and checked the reported baseline results against their source papers. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

All datasets used in this work are publicly available and have been cited. This paper does not release any new datasets and raises no issues concerning the Code of Ethics.

REPRODUCIBILITY STATEMENT

The experimental code and accompanying instructions for reproducing our results will be released publicly at a later date. Evaluation metrics, model configurations, and training details are provided in Appendices F and G.

REFERENCES

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 2021.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. In *International Conference on Learning Representations (ICLR)*, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

Shengcao Cao, Zijun Wei, Jason Kuen, Kangning Liu, Lingzhi Zhang, Jiuxiang Gu, HyunJoon Jung, Liang-Yan Gui, and Yu-Xiong Wang. Refer to Any Segmentation Mask Group With Vision-Language Prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21853–21863, 2025.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, 2020.

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*, 2023a.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A Language Modeling Framework for Object Detection. In *International Conference on Learning Representations (ICLR)*, 2022a.

Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David Fleet, and Geoffrey E. Hinton. A Unified Sequence Interface for Vision Tasks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31333–31346, 2022b.

Wei Chen, Long Chen, and Yu Wu. An Efficient and Effective Transformer Decoder-Based Framework for Multi-Task Visual Grounding. In *Computer Vision – ECCV*, 2024a.

Wenbo Chen, Zhen Xu, Ruotao Xu, Si Wu, and Hau-San Wong. Task-aware Cross-modal Feature Refinement Transformer with Large Language Models for Visual Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3931–3941, 2025.

Yi-Chia Chen, Wei-Hua Li, Cheng Sun, Yu-Chiang Frank Wang, and Chu-Song Chen. SAM4MLLM: Enhance Multi-Modal Large Language Model for Referring Expression Segmentation. In *Computer Vision – ECCV 2024*, 2024b.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions. In *International Conference on Learning Representations (ICLR)*, 2023b.

Zhenfang Chen, Peng Wang, Lin Ma, Kwan-Yee K. Wong, and Qi Wu. Cops-Ref: A New Dataset and Task on Compositional Referring Expression Comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10086–10095, 2020.

Floriana Ciaglia, Francesco Saverio Zuppichini, Paul Guerrie, Mark McQuade, and Jacob Solawetz. Roboflow 100: A Rich, Multi-Domain Object Detection Benchmark. *arXiv preprint arXiv:2211.13523*, 2022.

Ming Dai, Lingfeng Yang, Yihao Xu, Zhenhua Feng, and Wankou Yang. SimVG: A Simple Framework for Visual Grounding with Decoupled Multi-modal Fusion. In *Advances in Neural Information Processing Systems*, 2024.

Ming Dai, Wenxuan Cheng, Jiang-Jiang Liu, Lingfeng Yang, Zhenhua Feng, Wankou Yang, and Jingdong Wang. Improving Generalized Visual Grounding with Instance-aware Joint Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025a. doi: 10.1109/TPAMI.2025.3607387.

Ming Dai, Wenxuan Cheng, Jiedong Zhuang, Jiang-jiang Liu, Hongshen Zhao, Zhenhua Feng, and Wankou Yang. PropVG: End-to-End Proposal-Driven Visual Grounding with Multi-Granularity Discrimination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7058–7068, 2025b.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. TransVG: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1769–1779, 2021.

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 (ICCV)*, 2021.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Yu-Gang Jiang. GREx: Generalized referring expression segmentation, comprehension, and generation. *International Journal of Computer Vision*, 134:79, 2026. doi: 10.1007/s11263-026-02729-y.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Shenghao Fu, Yukun Su, Fengyun Rao, Jing LYU, Xiaohua Xie, and Wei-Shi Zheng. WeDetect: Fast Open-Vocabulary Object Detection as Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20377–20387, 2026.

Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Shuting He, Henghui Ding, Chang Liu, and Xudong Jiang. GREC: Generalized Referring Expression Comprehension. *arXiv preprint arXiv:2308.16182*, 2023.

Bhathiya Hemanthage, Hakan Bilen, Phil Bartie, Christian Dondrup, and Oliver Lemon. RECANT-Former: Referring Expression Comprehension with Varying Numbers of Targets. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21784–21798, 2024. doi: 10.18653/v1/2024.emnlp-main.1214.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *Computer Vision – ECCV 2016*, pp. 108–124. Springer, 2016. doi: 10.1007/978-3-319-46448-0_7.

Yutao Hu, Qixiong Wang, Wenqi Shao, Enze Xie, Zhenguo Li, Jungong Han, and Ping Luo. Beyond One-to-One: Rethinking the Referring Image Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4067–4077, 2023.

Ziling Huang and Shin'ichi Satoh. LoA-Trans: Enhancing Visual Grounding by Location-Aware Transformers. In *Computer Vision – ECCV 2024*, 2024.

Qing Jiang, Lin Wu, Zhaoyang Zeng, Tianhe Ren, Yuda Xiong, Yihao Chen, Liu Qin, and Lei Zhang. Referring to Any Person. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21667–21678, 2025.

Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25472–25483, 2026.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR – Modulated Detection for End-to-End Multi-Modal Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1086.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 2017.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 2020. doi: 10.1007/s11263-020-01316-z.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589, 2024.

Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wei Zhang. Text4Seg: Reimagining Image Segmentation as Text Generation. In *International Conference on Learning Representations (ICLR)*, pp. 1634–1661, 2025.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Muchen Li and Leonid Sigal. Referring Transformer: A One-step Approach to Multi-task Visual Grounding. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19652–19664, 2021.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring Plain Vision Transformer Backbones for Object Detection. In *Computer Vision – ECCV 2022*, 2022b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv preprint arXiv:1405.0312*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23592–23601, 2023a.

Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R. Manmatha. PolyFormer: Referring image segmentation as sequential polygon generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18653–18663, 2023b.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *Computer Vision – ECCV 2024*, 2024.

Yuyuan Liu, Yiping Ji, Anjie Le, Jiayuan Zhu, Jiazhen Pan, Can Peng, Jiajun Deng, Fengbei Liu, and Junde Wu. From Failure to Feedback: Group Revision Unlocks Hard Cases in Object-Level Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4816–4828, 2026.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-IO: A Unified Model for Vision, Language, and Multi-Modal Tasks. In *International Conference on Learning Representations (ICLR)*, 2023.

Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-Task Collaborative Network for Joint Referring Expression Comprehension and Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10034–10043, 2020.

Zhuoyan Luo, Yinghao Wu, Tianheng Cheng, Yong Liu, Yicheng Xiao, Hongfa Wang, Xiao-Ping Zhang, and Yujiu Yang. CoHD: A counting-aware hierarchical decoding framework for generalized referring expression segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22685–22694, 2025.

Yunze Man, De-An Huang, Guilin Liu, Shiwei Sheng, Shilong Liu, Liang-Yan Gui, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. Argus: Vision-Centric Reasoning with Grounded Chain-of-Thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14268–14280, 2025.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and Comprehension of Unambiguous Object Descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proceedings of the Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016. doi: 10.1109/3DV.2016.79.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 891–898, 2014.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding Multimodal Large Language Models to the World. In *International Conference on Learning Representations (ICLR)*, pp. 51575–51598, 2024.

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*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11671.

Renjie Pi, Lewei Yao, Jiahui Gao, Jipeng Zhang, and Tong Zhang. PerceptionGPT: Effectively Fusing Visual Perception into LLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27124–27133, 2024.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

Shraman Pramanick, Guangxing Han, Rui Hou, Sayan Nag, Ser-Nam Lim, Nicolas Ballas, Qifan Wang, Rama Chellappa, and Amjad Almahairi. Jack of All Tasks Master of Many: Designing General-Purpose Coarse-to-Fine Vision-Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14076–14088, 2024.

Han Qiu, Peng Gao, Lewei Lu, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Spatial Preference Rewarding for MLLMs Spatial Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 720–730, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2020.

Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. PACO: Parts and Attributes of Common Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7141–7151, 2023.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12179–12188, 2021.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S. Khan. GLaMM: Pixel Grounding Large Multimodal Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

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

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, 2019.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Shweta Singh, Aayan Yadav, Jitesh Jain, Humphrey Shi, Justin Johnson, and Karan Desai. Benchmarking Object Detectors with COCO: A New Path Forward. In *Computer Vision – ECCV*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3Det: Vast Vocabulary Visual Detection Dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023a.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022a.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. *arXiv preprint arXiv:2605.27365*, 2026. doi: 10.48550/arXiv.2605.27365.

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

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. CogVLM: Visual Expert for Pretrained Language Models. In *Advances in Neural Information Processing Systems*, 2024.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

XuDong Wang, Shaolun Zhang, Shufan Li, Konstantinos Kallidromitis, Kehan Li, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. SegLLM: Multi-round Reasoning Segmentation. In *International Conference on Learning Representations (ICLR)*, 2025b.

Yaxian Wang, Henghui Ding, Shuting He, Xudong Jiang, Bifan Wei, and Jun Liu. Hierarchical alignment-enhanced adaptive grounding network for generalized referring expression comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(8):8042–8050, 2025c. doi: 10.1609/aaai.v39i8.32867.

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 (CVPR)*, 2022b.

Jiannan Wu, Muyan Zhong, Sen Xing, Zeqiang Lai, Zhaoyang Liu, Zhe Chen, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, Ping Luo, Yu Qiao, and Jifeng Dai. VisionLLM v2: An end-to-end generalist multimodal large language model for hundreds of vision-language tasks. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

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, 2024b. doi: 10.1109/TIP.2024.3371348.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

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

Wentao Xiang, Haoxian Tan, Yujie Zhong, Cong Wei, Dengjie Li, and Yujiu Yang. Advancing Visual Large Language Model for Multi-granular Versatile Perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22153–22164, 2025.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4818–4829, 2024a.

Linhui Xiao, Xiaoshan Yang, Fang Peng, Yaowei Wang, and Changsheng Xu. OneRef: Unified One-tower Expression Grounding and Segmentation with Mask Referring Modeling. In *Advances in Neural Information Processing Systems*, 2024b.

Jinjin Xu, Liwu Xu, Yuzhe Yang, Xiang Li, Fanyi Wang, Yanchun Xie, Yi-Jie Huang, and Yaqian Li. u-LLaVA: Unifying Multi-Modal Tasks via Large Language Model. *arXiv preprint arXiv:2311.05348*, 2023.

Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal Instance Perception As Object Discovery and Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15325–15336, 2023.

Yuhuan Yang, Chaofan Ma, Jiangchao Yao, Zhun Zhong, Ya Zhang, and Yanfeng Wang. ReMamber: Referring Image Segmentation with Mamba Twister. In *Computer Vision – ECCV 2024*, 2024.

Yuxiang Yang, Yingqi Deng, Yufei Xu, and Jing Zhang. APTv2: Benchmarking Animal Pose Estimation and Tracking with a Large-scale Dataset and Beyond. *arXiv preprint arXiv:2312.15612*, 2023.

Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H. S. Torr. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yuan Yao, Qiushi Yang, Humen Zhong, Jiangning Wei, Yifang Men, Shuai Bai, Miaomiao Cui, and Zhibo Yang. Qwen3-VL-Seg: Unlocking open-world referring segmentation with vision-language grounding. *arXiv preprint arXiv:2605.07141*, 2026. doi: 10.48550/arXiv.2605.07141.

Heng Yin, Yuqiang Ren, Ke Yan, Shouhong Ding, and Yongtao Hao. ROD-MLLM: Towards More Reliable Object Detection in Multimodal Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14358–14368, 2025.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. In *International Conference on Learning Representations (ICLR)*, 2024.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2636–2645, 2020.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling Context in Referring Expressions. In *Computer Vision – ECCV 2016*, 2016.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. MAttNet: Modular Attention Network for Referring Expression Comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Anqi Zhang, Xiaokang Ji, Guangyu Gao, Jianbo Jiao, Chi Harold Liu, and Yunchao Wei. Rethinking MLLM itself as a segmenter with a single segmentation token. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19196–19207, 2026.

Gang Zhang, Xin Lu, Jingru Tan, Jianmin Li, Zhaoxiang Zhang, Quanquan Li, and Xiaolin Hu. RefineMask: Towards high-quality instance segmentation with fine-grained features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *International Conference on Learning Representations (ICLR)*, 2023.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, and Jianwei Yang. LLaVA-Grounding: Grounded Visual Chat with Large Multimodal Models. In *Computer Vision – ECCV 2024*, 2024a.

Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. OMG-LLaVA: Bridging Image-level, Object-level, Pixel-level Reasoning and Understanding. In *Advances in Neural Information Processing Systems*, 2024b.

Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. GROUNDHOG: Grounding Large Language Models to Holistic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14227–14238, 2024c.

Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. PSALM: Pixelwise SegmentAtion with Large Multi-Modal Model. In *Computer Vision – ECCV 2024*, 2024d.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing Through ADE20K Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 633–641, 2017.

Yikang Zhou, Tao Zhang, Dengxian Gong, Yuanzheng Wu, Ye Tian, Haochen Wang, Haobo Yuan, Jiacong Wang, Lu Qi, Hao Fei, Shunping Ji, Anran Wang, Zhuochen Wang, Yujing Wang, Cheng Chen, and Xiangtai Li. SAMTok: Representing any mask with two words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37852–37863, 2026.

Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. SeqTR: A Simple Yet Universal Network for Visual Grounding. In *Computer Vision – ECCV 2022*, 2022.

Chenchen Zhu, Fanyi Xiao, Andres Alvarado, Yasmine Babaei, Jiabo Hu, Hichem El-Mohri, Sean Chang Culatana, Roshan Sumbaly, and Zhicheng Yan. EgoObjects: A Large-Scale Egocentric Dataset for Fine-Grained Object Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

Yongshuo Zong, Qin Zhang, Dongsheng An, Zhihua Li, Xiang Xu, Linghan Xu, Zhuowen Tu, Yifan Xing, and Onkar Dabeer. Ground-V: Teaching VLMs to Ground Complex Instructions in Pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24635–24645, 2025.

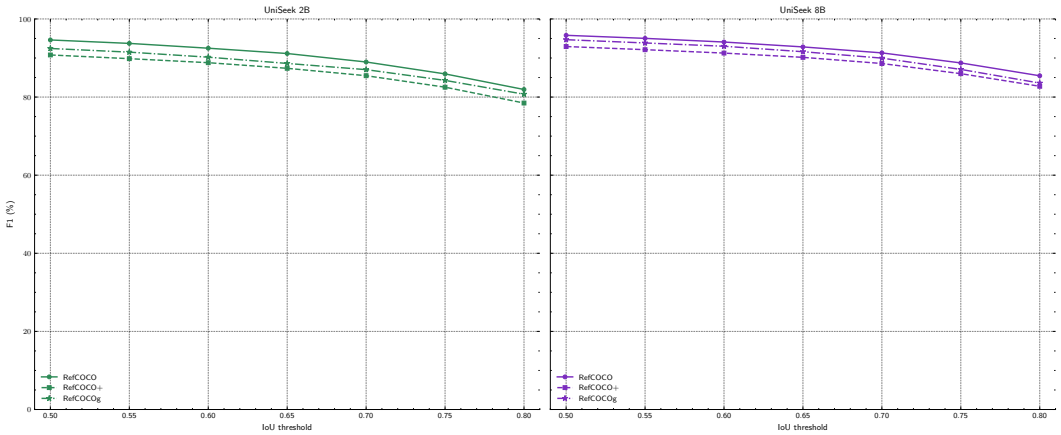


Figure 4: F1 at IoU thresholds from 0.50 to 0.80 on the validation splits of RefCOCO, RefCOCO+, and RefCOCOg. Left: UniSeek-2B. Right: UniSeek-8B. Both panels use the same vertical scale. Higher thresholds require more precise spatial overlap between predictions and ground-truth instances.

A Additional Experiments

We supplement the main results with three evaluations addressing localization under stricter overlap requirements, comparison with a text-guided SAM-based segmentation pipeline, and exact counting across target cardinalities.

A.1 Robustness to Stricter IoU Thresholds

We evaluated UniSeek-2B and UniSeek-8B on the validation splits of RefCOCO, RefCOCO+, and RefCOCOg at IoU thresholds from 0.50 to 0.80 in increments of 0.05. This evaluation examines whether the models retain accurate localization when predictions must satisfy increasingly strict spatial overlap requirements.

F1@IoU=t is a box-level micro F1 score. Within each expression, valid predicted boxes and ground-truth boxes are greedily matched one-to-one by repeatedly selecting the remaining pair with the highest IoU. Matches with IoU $\geq t$ count as true positives (TP); all remaining predictions and ground-truth boxes count as false positives (FP) and false negatives (FN), respectively. We sum TP, FP, and FN across all expressions and report

$$\mathrm{F1}_{\mathrm{IoU}\geq t}\ (\%) = 100 \times \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{8}$$

with a score of zero when the denominator is zero.

As shown in Figure 4, F1 decreased progressively as the IoU threshold increased for both model sizes. At IoU 0.80, UniSeek-2B achieved F1 scores of 81.94%, 78.47%, and 80.73% on RefCOCO, RefCOCO+, and RefCOCOg, respectively; UniSeek-8B achieved 85.45%, 82.76%, and 83.58%. Relative to their scores at IoU 0.50, the two models retained 86.44–87.33% and 88.27–89.19% of their F1, respectively. These results indicate that UniSeek retains substantial localization performance under stricter overlap requirements.

UniSeek-8B outperformed UniSeek-2B at every evaluated threshold on all three datasets. Its F1 decreased by 10.36, 10.18, and 11.11 percentage points from IoU 0.50 to 0.80, compared with 12.68, 12.31, and 11.71 percentage points for UniSeek-2B. Thus, the larger model achieved both higher F1 and smaller absolute declines across the evaluated threshold range.

A.2 Comparison with a Qwen-to-SAM2 Segmentation Pipeline

We compared direct localization and segmentation on the validation splits of RefCOCO, RefCOCO+, and RefCOCOg at the 2B and 8B model scales. Throughout this comparison, Qwen denotes Qwen3-VL (Bai et al., 2025), and SAM2-base denotes SAM2 (Ravi et al., 2025) loaded from

Table 9: Direct localization and segmentation on RefCOCO/+/g validation. Acc denotes bounding box Acc@IoU=0.5.

Method	Size	RefCOCO		RefCOCO+		RefCOCOg	
		Acc	cIoU	Acc	cIoU	Acc	cIoU
Qwen + SAM2-base	2B	55.17	29.67	48.82	27.95	42.69	9.71
Qwen + SAM2-base	8B	24.05	16.91	22.26	15.73	24.14	15.71
UniSeek box + SAM2-base	2B	–	47.38	–	44.77	–	42.68
UniSeek box + SAM2-base	8B	–	46.84	–	45.47	–	45.34
UniSeek	2B	94.41	81.73	90.11	77.44	92.39	78.74
UniSeek	8B	**95.73**	**82.64**	**92.45**	**79.45**	**94.17**	**79.31**

`sam2_hiera_base_plus.pt`. Both Qwen3-VL and SAM2 were kept frozen throughout these evaluations.

In Qwen + SAM2-base, Qwen directly generated bounding boxes, which served as spatial prompts for SAM2-base to produce masks. In UniSeek box + SAM2-base, the corresponding UniSeek model supplied the predicted boxes to the same SAM2-base model. UniSeek itself used its Sparse Localization Decoder for bounding boxes and its Dense Localization Decoder for masks.

The input protocol was identical across experimental settings except for an additional JSON-format instruction appended when Qwen directly generated bounding boxes. The appended instruction was:

> Required response format for this evaluation: return only a valid JSON array of bounding boxes for all objects matching the referring expression above. Each element must be an array of four numbers `[x0, y0, x1, y1]`, with `x0` $<$ `x1` and `y0` $<$ `y1`. Normalize coordinates to the range 0 to 1000 relative to the full displayed image; (0, 0) is the top-left and (1000, 1000) is the bottom-right. Return one box per matching object, and include all matching objects. If no object matches, return `[]`. For a single object, still use a nested array. Use this bounding-box format even when the task above asks for a segmentation mask. Do not include labels, dictionary keys, Markdown fences, explanations, or any text outside the JSON array.

Table 9 reports bounding box Acc@IoU=0.5 and mask cIoU using the definitions in Appendix F. For Qwen + SAM2-base, Acc evaluates the directly generated Qwen boxes, whereas cIoU evaluates the resulting SAM2-base masks. UniSeek box + SAM2-base uses exactly the same boxes as UniSeek at the corresponding model scale, so its identical Acc values need not be repeated.

UniSeek achieved higher bounding box accuracy and mask cIoU than Qwen + SAM2-base across all three datasets at both model scales. Relative to direct Qwen box generation, UniSeek improved Acc by 39.24–49.70 percentage points at 2B and 70.03–71.68 points at 8B. Qwen-8B had lower box accuracy than Qwen-2B on all three datasets under this input protocol; increasing model size alone did not improve localization here.

Replacing Qwen boxes with UniSeek boxes increased SAM2-base cIoU across all three datasets at both model scales. However, the resulting cIoU remained below that of UniSeek's own Dense Localization Decoder. UniSeek-2B exceeded UniSeek-2B box + SAM2-base by 34.35, 32.67, and 36.06 percentage points on RefCOCO, RefCOCO+, and RefCOCOg, respectively. The corresponding gains for UniSeek-8B were 35.80, 33.98, and 33.97 points. Thus, the segmentation advantage persisted when both approaches used UniSeek as the referring-expression localization model.

A.3 Exact Counting across Target Cardinalities

We evaluated the exact counting accuracy of UniSeek-2B and UniSeek-8B on gRefCOCO testA and testB, comprising 22,925 and 19,091 expressions, respectively. Expressions were grouped by their ground-truth target count into six bins: 0, 1, 2, 3–4, 5–10, and 10+, where 10+ denotes more than ten targets. This evaluation extends the target-presence analysis in Table 6 to exact cardinality prediction.

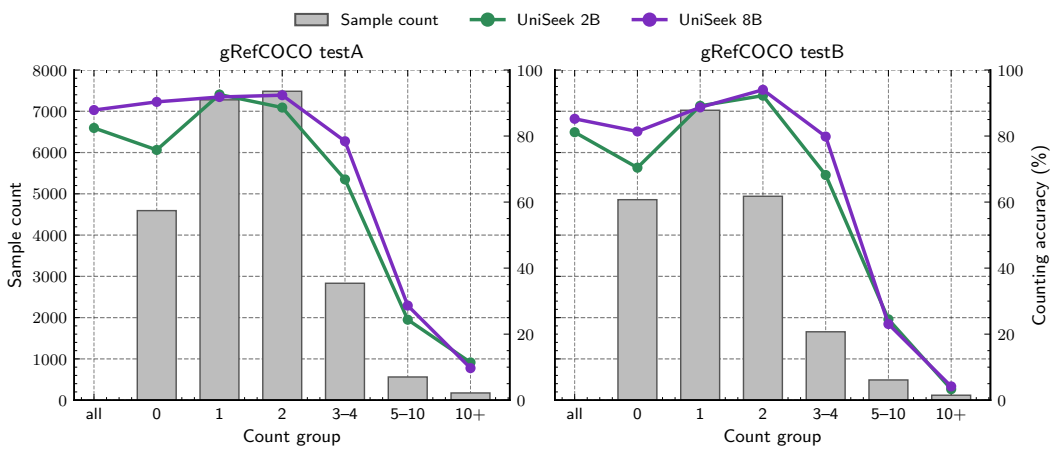


Figure 5: Exact counting accuracy across target cardinalities on gRefCOCO testA (left) and testB (right). Gray bars show the number of expressions in each ground-truth count bin (left axis); colored lines show counting accuracy for UniSeek-2B and UniSeek-8B (right axis). The "all" point reports overall accuracy. The 10+ bin denotes more than ten targets.

Let n_i and $\hat{n}_i$ denote the ground-truth and predicted numbers of referred instances for expression i. For a cardinality bin $\mathcal{B}$, we define

$$\text{count_acc}(\mathcal{B}) = \frac{100}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{1}[\hat{n}_i = n_i]. \tag{9}$$

The intervals group expressions only for reporting: predictions must match the exact ground-truth count, including within the 3–4, 5–10, and 10+ bins. Counting accuracy is evaluated independently of box or mask IoU. The "all" group reports accuracy over all expressions in each split.

Figure 5 shows that UniSeek-2B achieved overall counting accuracies of 82.45% and 81.16% on testA and testB, respectively, while UniSeek-8B achieved 87.87% and 85.22%. The results support effective counting beyond the dominant single-target training regime, particularly for UniSeek-8B. Its no-target accuracies were 90.35% and 81.39%, compared with 91.82% and 88.68% for single-target expressions; its two-target accuracies were slightly higher, at 92.37% and 94.01%. Accuracy remained relatively high for three to four targets. Together with the counting ablation in Section 4.4, these results support the effectiveness of explicit counting supervision for the more common long-tail cardinalities, while showing that robustness across these groups is stronger in the 8B model.

Performance nevertheless decreased sharply for the extreme tail: across both models and splits, accuracy was 23.06–28.65% for 5–10 targets and 3.33–11.43% for more than ten targets. This decline is consistent with the scarcity of high-cardinality training groups shown in Figure 9. Such groups provide substantially less supervision than single-target groups. Accurate counting in these extreme long-tail regimes remains an important limitation.

Unlike generic object detection and instance segmentation, which may require enumerating many objects in an image, GREC/GRES count only the instances referred to by an expression and typically involve small target sets. In the evaluated gRefCOCO splits, the 5–10 and 10+ bins together contained only 737 of 22,925 expressions on testA (3.21%) and 610 of 19,091 on testB (3.20%). These proportions include expressions with exactly five targets, so the share referring to more than five is smaller still. Thus, the severe high-cardinality failure cases are rare in these referring-expression benchmarks.

B Qualitative Results

Red boxes and green overlays denote UniSeek's predicted bounding boxes and segmentation masks, respectively. Each panel includes its referring expression and predicted instance counts.



Figure 6: Qualitative REC/RES results on challenging single-target examples.

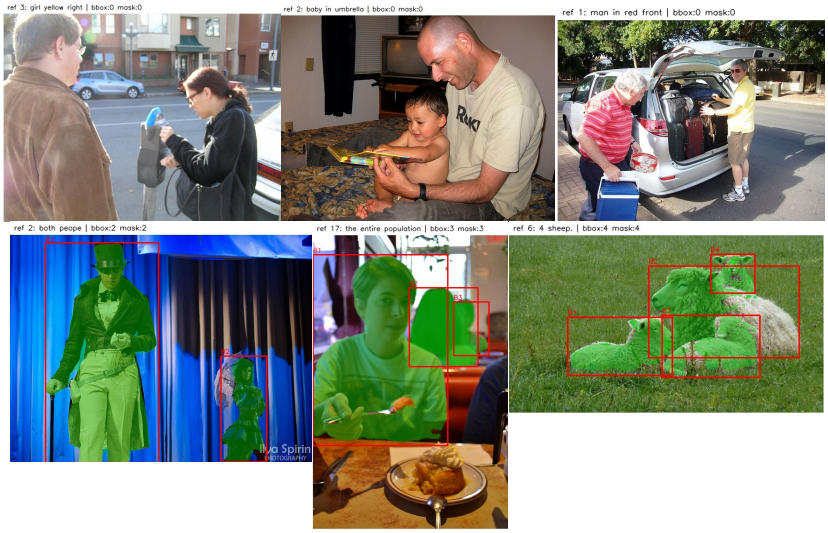


Figure 7: Qualitative GREC/GRES results. The top row shows no-target expressions; the bottom row shows expressions referring to two, three, and four instances, from left to right.

C Problem Formulation

Given an image I and a text instruction q, UniSeek predicts the referred instances and their requested geometry. We describe q using G reference groups, $\mathcal{R} = \{(r_g, \mathcal{T}_g)\}_{g=1}^{G}$, where r_g is a referring expression and $\mathcal{T}_g \subseteq \{\mathrm{det}, \mathrm{seg}\}$ specifies bounding box prediction or mask prediction.

For an expression referring to n_g instances, UniSeek predicts

$$f_\Theta(I, q) = \left\{ \left(\hat{n}_g, \left\{ \hat{\mathcal{Y}}_g^{\tau} \right\}_{\tau \in \mathcal{T}_g} \right) \right\}_{g=1}^{G}. \tag{10}$$

For an exact predicted count $\hat{n}_g$, the requested sets are $\hat{\mathcal{Y}}_g^{\mathrm{det}} = \{\hat{\mathbf{b}}_{g,i}\}_{i=1}^{\hat{n}_g}$ and $\hat{\mathcal{Y}}_g^{\mathrm{seg}} = \{\hat{\mathbf{m}}_{g,i}\}_{i=1}^{\hat{n}_g}$. A zero count yields empty sets. REC and RES use the single-target setting, whereas GREC and GRES additionally accommodate no-target and multi-target expressions. Counts beyond the exact-count range use the overflow mechanism in Appendix D.

```
<|object_start|>
  <|ref_start|>all dogs<|ref_end|><|COUNT_2|>
  <|DET_start|><|OBJ|><|OBJ|><|DET_end|>
<|object_end|>
<|object_start|>
  <|ref_start|>the red cat<|ref_end|><|COUNT_0|>
  <|SEG_start|><|SEG_end|>
<|object_end|>
```

Figure 8: Illustrative output for an image with two dogs and no red cat. The first reference group requests bounding boxes, with two instance slots in its detection section. The second requests a mask for a no-target expression and contains no instance slot.

D Detailed Answer Format and Constrained Decoding

Object-block serialization. We serialize each reference group as an *object block*, delimited by <|object_start|> and <|object_end|>. A block contains the referring expression, its instance count, and the requested task sections. The referring expression is enclosed by <|ref_start|> and <|ref_end|>, followed immediately by a count token. Detection and segmentation sections use <|DET_start|>/<|DET_end|> and <|SEG_start|>/<|SEG_end|>, respectively. Each <|OBJ|> token and its hidden state constitute an instance slot, which conditions the corresponding decoder to produce one bounding box or mask. Within each group, the ith <|OBJ|> slot is supervised by the ith target, without IoU/Dice-based or Hungarian reassignment. Detection-led samples sort clipped, valid, deduplicated boxes lexicographically by (x_0, y_0, x_1, y_1), whereas segmentation-led samples preserve input mask order; paired boxes and masks share that order. Groups follow their serialized order, and a 2D mask constitutes one target.

Group-level counts. The vocabulary contains 10 exact-count tokens, <|COUNT_0|> through <|COUNT_9|>, and an overflow token, <|COUNT_OVERFLOW|>, for larger counts. A no-target expression uses <|COUNT_0|> and retains the task-section delimiters with no instance slots between them.

Figure 8 illustrates the object-block serialization for multi-target and no-target expressions.

Constrained decoding. During autoregressive inference, constrained decoding masks inadmissible tokens, following the general principle of grammar-constrained generation (Scholak et al., 2021). After the referring expression ends, only an exact-count token or the overflow token is permitted. For an exact predicted group count $\hat{n}_g$, each requested task section must generate <|OBJ|> until it reaches $\hat{n}_g$ slots. After this quota is reached, only the section's closing delimiter is permitted. A zero-count task section closes immediately. Consequently, each normally completed, non-overflow task section contains exactly the predicted number of slots. Overflow counts impose no exact slot quota.

E Dataset Details

The core training collection contains 700,000 sample entries, comprising 350,000 bounding box localization entries and 350,000 pixel-level segmentation entries. A sample entry is not necessarily a unique image, as different tasks or annotations may reuse an image. The reported totals include the challenging no-target samples from R-RefCOCO, R-RefCOCO+, and R-RefCOCOg (Wu et al., 2024b). Table 10 and Figure 9 summarize the dataset sources and distribution; their task-specific contributions are described below.

Object detection. This category contains 165,274 samples (23.61% of the core collection), drawn from COCO 2017 (Lin et al., 2014), LVIS (Gupta et al., 2019), Open Images (Kuznetsova et al., 2020), V3Det (Wang et al., 2023a), APTv2 (Yang et al., 2023), BDD100K (Yu et al., 2020), DeepFashion (Liu et al., 2016), EgoObjects (Zhu et al., 2023), nuImages (Caesar et al., 2019), PACO (Ramanathan et al., 2023), and Roboflow100 (Ciaglia et al., 2022). These datasets cover generic objects, animals, autonomous driving, clothing, egocentric scenes, and object parts. They provide category-conditioned, multi-instance bounding box supervision for object recognition and spatial localization across different scenes.

Table 10: Training datasets by task. * denotes partial sampling of the configured training annotations. OD and IS denote object detection and instance segmentation, respectively.

Main task	Subtask	Datasets
Detection	REC	RefCOCO, RefCOCO+ (Kazemzadeh et al., 2014; Yu et al., 2016), RefCOCOg (Mao et al., 2016), RefCLEF* (Kazemzadeh et al., 2014), Flickr30k Entities* (Plummer et al., 2015), Visual Genome* (Krishna et al., 2017), Cops-Ref* (Chen et al., 2020)
	GREC	gRefCOCO (Liu et al., 2023a), R-RefCOCO, R-RefCOCO+, R-RefCOCOg (Wu et al., 2024b), Ref-ZOM* (Hu et al., 2023)
	OD	Open Images* (Kuznetsova et al., 2020), APTv2* (Yang et al., 2023), BDD100K* (Yu et al., 2020), DeepFashion* (Liu et al., 2016), EgoObjects* (Zhu et al., 2023), LVIS* (Gupta et al., 2019), nuImages* (Caesar et al., 2019), PACO* (Ramanathan et al., 2023), Roboflow100* (Ciaglia et al., 2022), V3Det* (Wang et al., 2023a), COCO 2017* (Lin et al., 2014)
Segmentation	RES	RefCOCO, RefCOCO+ (Kazemzadeh et al., 2014; Yu et al., 2016), RefCOCOg (Mao et al., 2016), RefCLEF (Kazemzadeh et al., 2014; Hu et al., 2016)
	GRES	gRefCOCO (Liu et al., 2023a), R-RefCOCO, R-RefCOCO+, R-RefCOCOg (Wu et al., 2024b), Ref-ZOM (Hu et al., 2023)
	IS	ADE20K (Zhou et al., 2017), COCO-Panoptic* (Kirillov et al., 2019), COCO-ReM* (Singh et al., 2024), LVIS* (Gupta et al., 2019), LVIS-PACO* (Ramanathan et al., 2023), PASCAL Context (Mottaghi et al., 2014)

Referring expression comprehension (REC). This category contains 139,585 samples (19.94%), drawn from RefCOCO, RefCOCO+ (Kazemzadeh et al., 2014; Yu et al., 2016), RefCOCOg (Mao et al., 2016), RefCLEF (Kazemzadeh et al., 2014), and Cops-Ref (Chen et al., 2020), together with Flickr30k Entities (Plummer et al., 2015) and Visual Genome (Krishna et al., 2017), which are included for phrase and region grounding. These data train the model to associate natural-language descriptions with image regions.

Generalized referring expression comprehension (GREC). This category contains 45,141 samples (6.45% of the core collection) from gRefCOCO (Liu et al., 2023a; He et al., 2023; Ding et al., 2026), Ref-ZOM (Hu et al., 2023), and R-RefCOCO/+/g (Wu et al., 2024b). R-RefCOCO/+/g contributes challenging no-target expressions. Together, these sources cover no-target and multi-target referring expressions for bounding box localization, supervising target-set identification and cardinality prediction.

Referring expression segmentation (RES). This category contains 71,677 samples (10.24%), using mask annotations from RefCOCO, RefCOCO+ (Kazemzadeh et al., 2014; Yu et al., 2016), RefCOCOg (Mao et al., 2016), and RefCLEF (Kazemzadeh et al., 2014; Hu et al., 2016). They provide pixel-level supervision for predicting target regions and boundaries conditioned on natural language.

Generalized referring expression segmentation (GRES). This category contains 57,866 samples (8.27%) from gRefCOCO (Liu et al., 2023a), Ref-ZOM (Hu et al., 2023), and R-RefCOCO/+/g (Wu et al., 2024b). R-RefCOCO/+/g contributes challenging no-target expressions, complementing the single-target and multi-target cases with supervision for predicting empty masks when no instance matches.

Instance segmentation. This category aggregates 220,457 generic mask-training samples (31.49%) from ADE20K (Zhou et al., 2017), COCO-Panoptic (Kirillov et al., 2019; Lin et al., 2014), COCO-ReM (Singh et al., 2024), LVIS (Gupta et al., 2019), LVIS-PACO (Ramanathan et al., 2023), and PASCAL Context (Mottaghi et al., 2014), supplementing dense spatial supervision across categories and scenes.

We retain the original referring expressions for natural-language referring data and construct localization instructions from category names for category-annotated data. Appendix H lists the localization

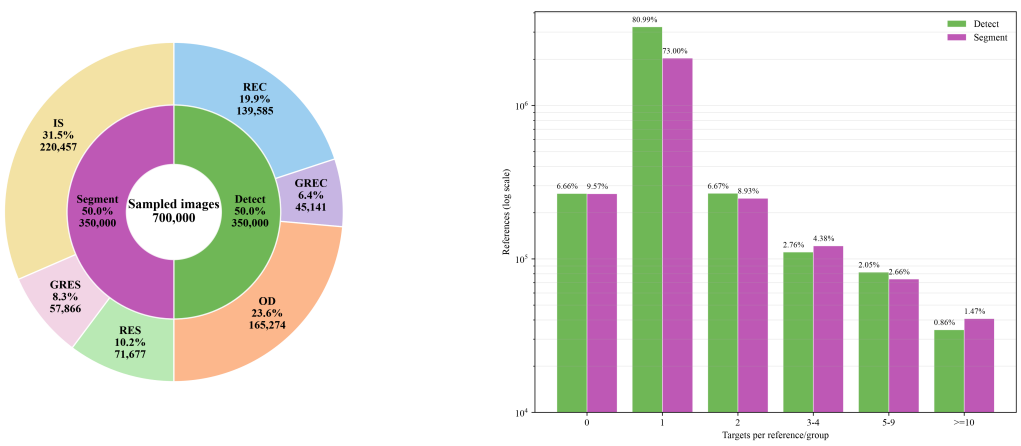


Figure 9: Training data distribution before per-sample selection and truncation. Left: sample entries by subtask; the label "Sampled images" refers to sample entries, not deduplicated images. Right: the long-tailed instance-count distribution of reference groups, shown on a logarithmic scale. OD and IS denote object detection and instance segmentation, respectively.

instructions. Targets associated with the same expression or category form a reference group. Groups are randomly ordered during training and selected as complete target sets.

F Evaluation Metrics

Each evaluation unit is an image–expression pair. Predictions and ground-truth instances are grouped by expression before computing the metrics. All results are reported as percentages, with higher values indicating better performance. Let N denote the number of evaluated expressions and $\mathbf{1}[\cdot]$ the indicator function.

REC accuracy. For a predicted box $\hat{b}_i$ and its ground-truth box b_i, intersection over union is $\mathrm{IoU}(\hat{b}_i, b_i) = |\hat{b}_i \cap b_i|/|\hat{b}_i \cup b_i|$. REC accuracy is

$$\mathrm{Acc@IoU} = 0.5 = \frac{100}{N}\sum_{i=1}^{N} \mathbf{1}[\mathrm{IoU}(\hat{b}_i, b_i) \geq 0.5]. \tag{11}$$

A missing prediction is incorrect. If multiple boxes are returned, the evaluator uses the first box when scores are tied.

GREC strict set-matching accuracy. For each expression, predicted and ground-truth boxes are matched one-to-one by greedily selecting the remaining pair with the highest IoU. Matching stops when no remaining pair reaches IoU 0.5. Let TP_i be the number of matched pairs, and let FP_i and FN_i count unmatched predicted and ground-truth boxes. We compute

$$F_{1,i} = \frac{2\mathrm{TP}_i}{2\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i}, \qquad \Pr@(\mathrm{F1} = 1, \mathrm{IoU} \geq 0.5) = \frac{100}{N}\sum_{i=1}^{N} \mathbf{1}[F_{1,i} = 1]. \tag{12}$$

For a no-target expression, $F_{1,i} = 1$ if the predicted box set is empty and 0 otherwise. Thus, this metric requires the entire target set to be correct, with no missing or extra instances. The column labeled Precision in Table 3 reports this expression-level accuracy rather than conventional instance-level precision.

RES and GRES cumulative IoU (cIoU). Let P_i and G_i be the unions of the predicted and ground-truth foreground masks for expression i. Multi-instance masks are merged within each expression before evaluation. Define $I_i = |P_i \cap G_i|$ and $U_i = |P_i \cup G_i|$. Cumulative IoU aggregates intersection and union pixel counts over the evaluation split:

$$\mathrm{cIoU} = \mathrm{oIoU} = 100\,\frac{\sum_{i=1}^{N} I_i}{\sum_{i=1}^{N} U_i}. \tag{13}$$

Correct empty predictions for no-target expressions contribute zero to both sums. False-positive foreground pixels increase the denominator and reduce cIoU. Overall IoU (oIoU) and cumulative IoU (cIoU) name the same metric: both aggregate intersection and union pixel counts before taking their ratio. This differs from averaging the IoU of individual expressions. We use cIoU as the common label in Table 2 and include source-reported oIoU results in the same rankings. These baseline values are taken from their source papers rather than recomputed with our evaluator.

GRES generalized IoU (gIoU). This metric averages expression-level mask IoU, including no-target expressions:

$$\mathrm{gIoU} = \frac{100}{N}\sum_{i=1}^{N} s_i, \qquad s_i = \begin{cases} I_i/U_i, & G_i \neq \emptyset, \\ 1, & G_i = \emptyset \text{ and } P_i = \emptyset, \\ 0, & G_i = \emptyset \text{ and } P_i \neq \emptyset. \end{cases} \tag{14}$$

Unlike cIoU, gIoU assigns equal weight to each expression. Here, gIoU denotes the generalized referring segmentation metric, not the generalized bounding box IoU used in the training loss.

No-target accuracy (N-acc). Let $\mathcal{E}_0$ contain the expressions with no ground-truth target. No-target accuracy is

$$\text{N-acc} = \frac{100}{|\mathcal{E}_0|}\sum_{i\in\mathcal{E}_0} \mathbf{1}[\text{prediction for expression } i \text{ is empty}]. \tag{15}$$

For GREC, an empty prediction contains no bounding boxes. For GRES, its union foreground mask is empty. N-acc is evaluated only on no-target expressions, whereas GREC Precision and GRES gIoU include both target-present and no-target expressions.

G Implementation Details

G.1 Model Configuration and Input Processing

The two UniSeek variants initialize the MLLM Backbone from Qwen3-VL-2B-Instruct and Qwen3-VL-8B-Instruct (Bai et al., 2025), respectively. At checkpoint save time, the native image encoder, vision-to-language merger, Shared Semantic Feature Pyramid Network, local detail modules, and localization decoders were frozen; trainable parameters were confined to the VLM language component and output head. Appendix I reports the corresponding parameter counts. Both variants use the same additional-module configuration, detailed below.

Input images use dynamic resolution, with total pixel counts in $[16 \times 28^2, 2560 \times 28^2]$, or 12,544–2,007,040 pixels. This range constrains image area, rather than fixing the width or height. Segmentation images and supervision masks are resized with their aspect ratios preserved and padded to 512×512. The maximum sequence length for text and visual inputs is 4096.

G.2 Visual Feature and Local Detail Implementation

Table 11 summarizes the dimensions and operator configurations of UniSeek's additional modules.

Patch and detail pathways. For pooled injection in Equation 2, average pooling and LN precede projection to the native patch embedding dimension. The scalar gate α is initialized to zero, initially preserving the native embeddings. The unpooled detail map $\mathbf{D}$ has 64 channels and valid extent $4H_\mathrm{p} \times 4W_\mathrm{p}$ for a native patch grid of $H_\mathrm{p} \times W_\mathrm{p}$.

Semantic feature aggregation. The four native visual features are restored to maps $\{\mathbf{F}^{(\ell)}\}_{\ell=1}^{4}$. Each is independently projected, resized with its aspect ratio preserved, and padded to the common grid. The shared representation is

$$\mathbf{F}_\mathrm{sem} = \Phi_\mathrm{out}\left(\frac{1}{4}\sum_{\ell=1}^{4}\Phi_\ell\Big(\mathcal{A}_\mathrm{sp}\Big(W_\ell * \mathbf{F}^{(\ell)}\Big)\Big)\right). \tag{16}$$

Here, $*$ denotes convolution, W_ℓ is the layer-specific channel projection, and $\mathcal{A}_\mathrm{sp}$ performs resizing and padding. The module Φ_ℓ applies spatial processing, while Φ_out contains two Refinement Blocks and the output projection. The resulting $\mathbf{F}_\mathrm{sem}$ has shape $1024 \times 64 \times 64$.

Table 11: Configuration of the visual modules and task-specific decoders. GN denotes Group Normalization (Wu & He, 2018); LN denotes Layer Normalization.

Module	Configuration
Detailed Patch Embedder	16×16 input patch; convolution with kernel and stride 4×4, GN, SiLU; output $64 \times 4 \times 4$ per patch.
Detail-map mixing	Gated residual path: 3×3 depthwise convolution, GN, SiLU, 1×1 pointwise convolution; 64 channels.
Semantic feature pyramid	Four layer-specific 1×1 projections to 256 channels; aligned 64×64 grid; 3×3 convolution, GN, SiLU per layer; average fusion, two Refinement Blocks, 1×1 projection to 1024 channels.
Refinement Block	3×3 convolution, GN with 32 groups, SiLU; unchanged channels and spatial size.
Detailed Embedding Injector	Semantic projection to $256 \times 64 \times 64$; detail projection by 1×1 convolution, GN, SiLU.
Sparse Localization Decoder	256-dimensional image features and queries; four Transformer decoder layers, each with eight attention heads, feed-forward dimension 1024, dropout 0.1.
Reference-query fusion	LN, linear layer, GELU, final linear layer; maps the concatenated 512-dimensional slot and reference context to 256 dimensions.
Dense Localization Decoder	Embedding dimension 256; mask refinement uses 64 hidden channels.

Dense-branch detail injection. We extract the valid region of $\mathbf{D}$ and bilinearly resize it to the 64×64 semantic grid before applying Π_{d}. The gate β in Equation 3 is initialized to zero. The resulting $\mathbf{F}_{\mathrm{seg}}$ has 256 channels, while the sparse branch separately projects $\mathbf{F}_{\mathrm{sem}}$ to 256 channels.

G.3 Localization Decoder Implementation

Sparse queries and geometric outputs. The final linear layer of ϕ_{q} is zero-initialized, so the initial fused query equals the projected slot representation. Normalized cell-center coordinates on each image's valid feature grid pass through a linear layer and LN to produce coordinate positional embeddings. These are added to the projected image features before flattening into key/value tokens. After Transformer decoding, a Multilayer Perceptron and sigmoid predict normalized box-center coordinates, width, and height. These predictions are converted to corner coordinates and clipped to the valid range.

Reference–region alignment. Target-region features are averaged on the visual grid after positional embeddings are added. The candidate count varies with the available same-image ground-truth boxes; only $|\mathcal{P}_i| + |\mathcal{N}_i|$ regions enter the denominator. For each query i, candidate regions with an IoU of at least 0.999 with the query's target box enter $\mathcal{P}_i$. Regions with an IoU of at most 0.3 enter $\mathcal{N}_i$. Intermediate overlaps are excluded. Equation 5 uses temperature $T = 0.07$ and excludes queries lacking either positives or negatives. The alignment loss is zero when no valid query remains. If only the target region is available, the query contributes no alignment loss but still receives L1/GIoU supervision.

Dense queries and mask refinement. Text Projection maps each slot hidden state to 256 dimensions. SAM's prompt encoding and mask decoding modules are randomly initialized, and its image encoder is omitted. Resize and Projection align $\mathbf{D}$ to the initial mask resolution as a 64-channel map $\mathbf{D}'$. A linear map of the projected query predicts channel scales and shifts, broadcast across spatial positions in Equation 6. The residual prediction layer is zero-initialized, preserving the initial mask at initialization. The quantities $\hat{\mathbf{m}}^{\,c}$ and $\hat{\mathbf{m}}$ denote logits; inference restores their spatial coordinates to the original image to obtain the instance masks in Equation 10.

G.4 Optimization and Training Resources

We used AdamW (Loshchilov & Hutter, 2019) with a peak learning rate of 2×10^{-5} for trainable parameters. Weight decay was 0.01, and the gradient-norm clipping threshold was 1.0. The learning rate increased linearly for the first 500 optimization steps and then followed cosine decay, for a total of 16,000 optimization steps. Training used BF16 mixed precision, gradient checkpointing, and DeepSpeed ZeRO-2 (Rajbhandari et al., 2020). Table 12 summarizes batch sizes and computational

Table 12: Training resources and batch configurations.

Setting	UniSeek-2B	UniSeek-8B
Training devices	16 A100 GPUs	8 H200 GPUs
Per-device microbatch size	8	8
Gradient accumulation steps	8	16
Effective global batch size	1024	1024
Optimization steps	16,000	16,000
Training time	Approximately 8 days	Approximately 12 days
Per-device validation batch size	6	12

resources. The effective global batch size is the product of the number of devices, the per-device microbatch size, and the number of gradient accumulation steps. The two variants therefore used the same number of samples per parameter update.

G.5 Joint Loss and Counting Reweighting

Language and counting losses. The base autoregressive loss supervises valid answer positions, including structural delimiters, count tokens, and instance slots. Auxiliary counting logits are selected at positions predicting ground-truth exact-count tokens, respecting the causal next-token shift, and restricted to the 10 exact-count classes. Equation 1 contributes zero when its valid sample set is empty. The language objective is

$$\begin{aligned} \mathcal{L}_{\text{lang}} &= \mathcal{L}_{\text{AR}} + \rho(t)\lambda_{\text{grp}}\mathcal{L}_{\text{cnt}}^{\text{grp}}, \\ \rho(t) &= \min\left(1, \frac{t+1}{T_{\text{w}}}\right). \end{aligned} \tag{17}$$

Here, t is the zero-indexed optimization step, $\lambda_{\text{grp}} = 2.4$, and $T_{\text{w}} = 1000$. This auxiliary-loss warmup is independent of learning-rate warmup.

Sparse localization loss. The bounding box branch uses

$$\mathcal{L}_{\text{det}} = 5\mathcal{L}_1^{\text{bbox}} + 2\mathcal{L}_{\text{GIoU}} + \mathcal{L}_{\text{RR}}. \tag{18}$$

The coordinate loss sums absolute normalized coordinate errors per instance and averages over valid instances. The GIoU loss averages $1 - \text{GIoU}$ over corresponding box pairs. Predictions and annotations follow the output instance-slot order; no-target groups contribute no instance regression term.

Dense localization loss. Refined masks receive BCE and Dice supervision, with an auxiliary term for initial masks:

$$\begin{aligned} \mathcal{L}_{\text{seg}} = {} & 2\mathcal{L}_{\text{BCE}}(\hat{\mathbf{m}}, \mathbf{m}) + 0.5\mathcal{L}_{\text{Dice}}(\hat{\mathbf{m}}, \mathbf{m}) \\ & + 0.25\left[2\mathcal{L}_{\text{BCE}}(\hat{\mathbf{m}}^{\,c}, \mathbf{m}) + 0.5\mathcal{L}_{\text{Dice}}(\hat{\mathbf{m}}^{\,c}, \mathbf{m})\right], \end{aligned} \tag{19}$$

Here, $\mathbf{m}$ is the ground-truth instance mask, and Dice loss uses sigmoid-transformed logits. In Equation 7, the autoregressive loss coefficient is 1 and $\lambda_{\text{det}} = \lambda_{\text{seg}} = 0.8$.

Counting supervision uses exact-count classes from 0 to 9, partitioned into four intervals according to the ground-truth count. Classes in the same interval share a weight. Table 13 lists the weights for group-level counting. Overflow tokens contribute to base autoregressive supervision but not to the auxiliary counting classification loss.

This weighting scheme combines inverse-frequency weighting with adjustments for the practical requirements of GREC/GRES. We increase the weight of no-target samples to improve performance on expressions that refer to no instance. We also temper the emphasis on the extreme long tail, including $10+$ samples, to prevent rare boundary cases from degrading performance on common samples. Consistent with this rationale, overflow counts receive only base autoregressive supervision, limiting their influence on the counting objective.

G.6 Validation Protocol and Checkpoint Selection

Validation and checkpoint saving took place every 250 optimization steps. At the end of training, we selected the checkpoint with the highest final selection score. Validation used a fixed subset stratified by enabled task.

Table 13: Weights for the auxiliary group-level counting loss by ground-truth instance-count interval.

Ground-truth count	Group-level counting
0	2.00
1	0.50
2–4	0.50
5–9	1.00

Checkpoint selection first aggregates weighted metrics within each decoding task and then normalizes task weights over tasks with valid validation samples. The bounding box task score is

$$S_{\mathrm{det}} = 0.45\,\mathrm{Pr@}(\mathrm{F1}=1, \mathrm{IoU} \geq 0.5) + 0.35\,(\mathrm{Acc@IoU}=0.5) + 0.20\,\mathrm{N\text{-}acc}. \tag{20}$$

The mask task score is

$$S_{\mathrm{seg}} = 0.40\,\mathrm{cIoU} + 0.40\,\mathrm{gIoU} + 0.20\,\mathrm{N\text{-}acc}. \tag{21}$$

The configured weights of the bounding box and mask tasks are both 0.50. The composite decoder score is

$$S_{\mathrm{dec}} = \frac{S_{\mathrm{det}} + S_{\mathrm{seg}}}{2}. \tag{22}$$

The group-level counting score combines the accuracy of the actual number of generated instances, exact-count token accuracy, count consistency, and macro-averaged accuracy across instance-count intervals, with coefficients of 0.35, 0.20, 0.15, and 0.30, respectively. Group-level count consistency measures agreement between the predicted count and the actual number of generated instances. Interval macro-averaging uses exact-count token classification accuracy, includes only intervals with at least 20 valid samples, and assigns equal weight to the included intervals. Missing counting metrics are omitted and the remaining weights are renormalized. Denoting the group-level score by S_{grp}, we obtain

$$S_{\mathrm{cnt}} = S_{\mathrm{grp}}, \qquad S_{\mathrm{raw}} = 0.80 S_{\mathrm{dec}} + 0.20 S_{\mathrm{cnt}}. \tag{23}$$

All metrics above are converted to the range $[0, 1]$. Let e_{grp} be the format error rate for group-level counting. The format-adjusted score is

$$S_{\mathrm{fmt}} = S_{\mathrm{raw}}\,[1 - e_{\mathrm{grp}}]\,. \tag{24}$$

An eligible checkpoint must be evaluated with full autoregressive generation and have a group-level counting format error rate of at most 0.03. The final ranking score is

$$S_{\mathrm{select}} = \begin{cases} S_{\mathrm{fmt}}, & \text{full autoregressive evaluation; format threshold satisfied,} \\ S_{\mathrm{fmt}} - 1, & \text{full autoregressive evaluation; format threshold not satisfied,} \\ S_{\mathrm{fmt}} - 3, & \text{otherwise, including missing generation-evaluation evidence.} \end{cases} \tag{25}$$

This rule prioritizes generation-based validation results that satisfy the format requirement over those that do not. It also prioritizes generation-based validation over warmup-stage validation using given queries. If no checkpoint satisfies the format threshold, the same score selects the best available checkpoint.

G.7 Ablation Settings

The ablations in Section 4.4 used Qwen3-VL-2B-Instruct and restricted joint training to the complete RefCOCO REC/RES and gRefCOCO GREC/GRES training sets used in the main experiments, excluding the other training sources. We reduced the training budget from 16,000 to 1,000 optimization steps. Learning-rate warmup was shortened from 500 to 31 steps, and auxiliary counting-loss warmup from 1,000 to 63 steps. The training and data-order seeds were both 42.

All configurations started from the same original pretrained backbone, with matched random initialization for shared modules. Data order, effective batch size, input processing, optimizer, and learning-rate settings were held fixed across variants. We evaluated the final step-1,000 checkpoint on the corresponding val/testA/testB splits after freezing the configurations. Unlike the main experiments, these runs did not perform validation during training or apply the count-dependent checkpoint-selection rule in Appendix G.6. All localization metrics came from actual model-generated outputs without teacher-forcing.

H LOCALIZATION INSTRUCTION TEMPLATES

The following localization instruction templates use `[OBJ]` as a placeholder for the referring expression or category name. Capitalization variants are retained. For REC and RES inference, we prepend the image and a newline to the respective task-specific templates, such as `Detect [OBJ].` and `Segment [OBJ].`. Single-target supervision uses `<|COUNT_1|>` and one `<|OBJ|>` slot, but inference follows the predicted count rather than fixing it to one (Appendix D).

```
Detect [OBJ].

detect [OBJ].

Please detect [OBJ] in this image.

Please detect [OBJ] in this image.  Output the bounding box coordinates
in [x0, y0, x1, y1] format.

Find [OBJ] in the image.  Output the bounding box coordinates in [x0, y0,
x1, y1] format.

Detect [OBJ] in the image.  Output the bounding box coordinates in [x0,
y0, x1, y1] format.

Locate [OBJ] in the image.  Output the bounding box coordinates in [x0,
y0, x1, y1] format.

Identify [OBJ] in the image.  Output the bounding box coordinates in [x0,
y0, x1, y1] format.

Please locate [OBJ]. Output the bounding box coordinates in [x0, y0, x1,
y1] format.

What is the location of [OBJ]? Return its bounding box as [x0, y0, x1,
y1].

Mark the region where [OBJ] appears using [x0, y0, x1, y1] format.

Can you find [OBJ] in this picture?  Give the coordinates as [x0, y0, x1,
y1].

Highlight [OBJ] in the image and output its bounding box in [x0, y0, x1,
y1].

Indicate where [OBJ] is located with bounding box coordinates [x0, y0, x1,
y1].

Show me the bounding box for [OBJ] in [x0, y0, x1, y1] format.

Return the bounding box coordinates for [OBJ] in the image.

Give the coordinates of the box around [OBJ] using [x0, y0, x1, y1]
format.

Determine the bounding box of [OBJ] and return it as [x0, y0, x1, y1].

Identify the bounding box location of [OBJ] using the format [x0, y0, x1,
y1].

please detect [OBJ] in this image.

detect [OBJ]. Output the bounding box coordinates in [x0, y0, x1, y1]
format.

please detect [OBJ] in this image.  Output the bounding box coordinates
in [x0, y0, x1, y1] format.

find [OBJ] in the image.  Output the bounding box coordinates in [x0, y0,
x1, y1] format.
```

```
detect [OBJ] in the image.  Output the bounding box coordinates in [x0,
y0, x1, y1] format.

locate [OBJ] in the image.  Output the bounding box coordinates in [x0,
y0, x1, y1] format.

identify [OBJ] in the image.  Output the bounding box coordinates in [x0,
y0, x1, y1] format.

please locate [OBJ]. Output the bounding box coordinates in [x0, y0, x1,
y1] format.

what is the location of [OBJ]? Return its bounding box as [x0, y0, x1,
y1].

mark the region where [OBJ] appears using [x0, y0, x1, y1] format.

can you find [OBJ] in this picture?  Give the coordinates as [x0, y0, x1,
y1].

highlight [OBJ] in the image and output its bounding box in [x0, y0, x1,
y1].

indicate where [OBJ] is located with bounding box coordinates [x0, y0, x1,
y1].

show me the bounding box for [OBJ] in [x0, y0, x1, y1] format.

return the bounding box coordinates for [OBJ] in the image.

give the coordinates of the box around [OBJ] using [x0, y0, x1, y1]
format.

determine the bounding box of [OBJ] and return it as [x0, y0, x1, y1].

identify the bounding box location of [OBJ] using the format [x0, y0, x1,
y1].


Segment [OBJ].

segment [OBJ].

Segment [OBJ] in this image.

What is [OBJ] in this image?  Please respond with a segmentation mask.

What is [OBJ] in this image?  Please output a segmentation mask.
```

I Architectural Complexity Analysis

Parameter structure and feature reuse. UniSeek augments its pretrained MLLM backbone with the Shared Semantic Feature Pyramid Network, local detail modules, task-specific projections, and localization decoders. Both localization branches reuse features from the native image encoder. In particular, the dense branch uses SAM's prompt encoding and mask decoding architecture without its image encoder. This avoids the parameters and forward computation of an additional image encoder, while introducing the feature aggregation and localization modules described in Appendix G.2.

We counted parameters from the DeepSpeed `param_shapes` and `frozen_param_shapes` metadata of the two saved checkpoints, counting shared parameters once and excluding buffers. UniSeek-2B contains 2,150,407,802 total parameters (2.150B), of which 1,720,321,024 (1.720B) were trainable at checkpoint save time. UniSeek-8B contains 9,413,279,594 total parameters (9.413B), of which 8,190,227,456 (8.190B) were trainable. Total counts describe the modules stored in each checkpoint, whereas trainable counts describe the parameters unfrozen in the saved training configuration. The 2B and 8B names refer to the backbone variants, not the complete checkpoint sizes.

For UniSeek-2B, the output head shares its weights with the language-model embeddings and contributes no additional unique parameters. For UniSeek-8B, the trainable count comprises 7,567,897,600 language-model parameters and 622,329,856 parameters in the separate output head. Its checkpoint additionally retains a frozen `vlm.lm_head` with 621,821,952 parameters, which is included in the total count. Both checkpoints also retain a frozen pointing decoder, with 5,398,018 and 5,922,306 parameters for the 2B and 8B variants, respectively. Decoder counts include their internal text and image projections and refinement modules. Consequently, checkpoint totals should not be interpreted as counts of parameters necessarily used by a particular inference task.

At save time, all visual modules and localization decoders were frozen; all trainable parameters belonged to the VLM language component and output head. This accounting describes the saved state and does not establish the freezing schedule at earlier training steps. Reloading a checkpoint with a different freezing configuration can change its trainable parameter count without changing its total parameter count.

Spatial processing. Let N denote the number of native image patches and S the number of spatial positions in the shared semantic grid ($S = 64^2$ in our implementation). The Detailed Patch Embedder preserves the visual sequence length, so it does not increase the backbone attention cost through additional visual tokens. With fixed patch size, channel widths, and convolution kernels, patch-detail extraction and detail-map mixing scale as $\mathcal{O}(N)$. Projection and aggregation of a fixed number of native feature maps onto the semantic grid scale as $\mathcal{O}(N + S)$. These operations nevertheless introduce additional computation and intermediate feature storage.

Instance-dependent computation. Let K denote the number of instance slots supplied to a localization decoder. For a standard sparse Transformer decoder layer with width d, self-attention, cross-attention, and projections contribute $\mathcal{O}(K^2d + KSd + (K + S)d^2)$ operations, assuming the feed-forward width is proportional to d. For the dense branch, instance-wise mask decoding and refinement scale linearly with K when spatial resolutions and decoder dimensions are fixed. The shared image representation does not require re-encoding the image for each instance. However, generating additional slots also incurs autoregressive backbone computation; the decoder scaling above therefore does not characterize the complete inference cost.