See to Believe: Aligning Multimodal Chain-of-Thought Reasoning with Visual Evidence

Changsong Wen[1,*], Yanjie Wang[1,*], Zelin Peng[1], Yu Huang[1], Yaoming Wang[2], Xiaokang Yang[1], Wei Shen[1,✉]

[1]MoE Key Lab of Artificial Intelligence, School of Computer Science
Shanghai Jiao Tong University [2]Meituan
{changsong, wei.shen}@sjtu.edu.cn

Abstract

Identifying objects from complex, implicit textual queries requires multimodal reasoning that connects language with visual inputs. Existing methods either rely on a multimodal large language model (MLLM) to reason implicitly within hidden representations, or produce explicit but purely textual chain-of-thought that terminates in a coarse bounding box. In both cases, the reasoning process is not aligned with visual evidence at each step, leaving intermediate errors unverifiable, which can propagate to the final prediction. We propose Reasoning with Visual Anchors (ReVA), a framework where each chain-of-thought step is explicitly aligned to a specific image region. ReVA introduces anchor tokens into intermediate reasoning steps, each directly aligned with region-level features in the image encoder's latent space. Such supervision makes each anchor attend to its corresponding image region, grounding each step in concrete visual evidence. To supervise the full reasoning chain, we further require each anchor to correctly transition to the next along the spatial relation described in the text. This helps the model learn to identify the target and predict its segmentation mask. To support training, we construct a large-scale Chain-of-Anchor dataset of over 245K quality-filtered samples from the RefCOCO series, each annotated with a spatially grounded chain-of-thought containing explicit anchor regions. The dataset is constructed by leveraging a strong MLLM to generate reasoning traces, followed by a three-stage verification pipeline ensuring spatial accuracy and reasoning quality. Experiments on reasoning and referring segmentation benchmarks demonstrate that ReVA achieves state-of-the-art performance (e.g., +5.4 gIoU and +6.7 cIoU on the ReasonSeg test set).

1 Introduction

Understanding visual queries often requires reasoning beyond object recognition. For example, a query like "the object that the person is about to sit on" requires understanding actions, spatial relationships, and commonsense knowledge before identifying the target. Reasoning segmentation [18] requires localizing objects based on complex, implicit queries that go beyond simple noun phrases. Unlike referring segmentation [13], where the target is often described by explicit visual attributes, reasoning segmentation demands complex inference to bridge the gap between an implicit linguistic description and a concrete image mask.

Most existing approaches [18, 33, 29] combine a multimodal large language model (MLLM) with a segmentation decoder, where a special token (e.g., `[SEG]`) is appended to the output sequence

[*]Equal contribution.
[✉]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

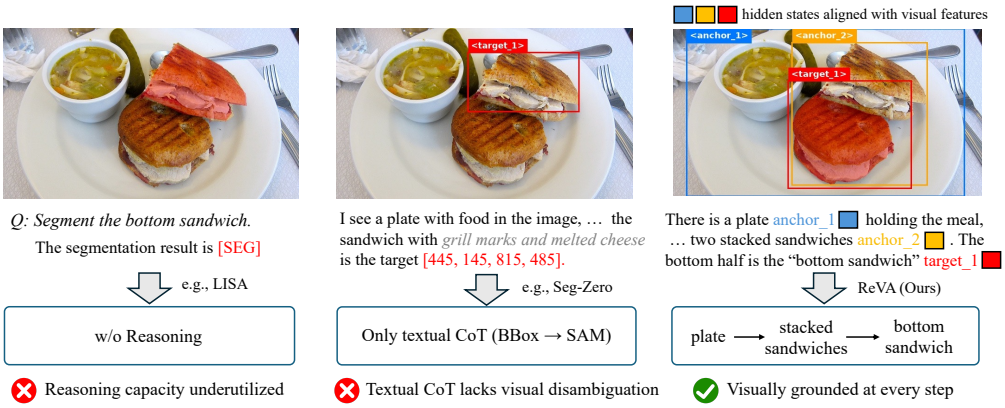

Figure 1: Comparison of the paradigms for reasoning segmentation. (a) Methods that directly output the result leave the model's reasoning capacity underutilized. (b) Textual CoT methods reason in natural language without visual grounding, making descriptions visually ambiguous when multiple similar objects are present. (c) ReVA (Ours) grounds each reasoning step to specific image regions via visual anchors during training, making intermediate steps visually verifiable.

and its hidden representation is projected into the decoder to produce the mask (Fig. 1 (a)). This paradigm leaves the model's reasoning capacity underutilized: no intermediate reasoning steps are generated, and the model must implicitly compress all reasoning into its hidden representations. More recently, reinforcement learning-based methods such as Seg-Zero [24] and VisionReasoner [25] train the MLLM to produce explicit Chain-of-Thought (CoT) reasoning and output a bounding box, which is then forwarded into SAM [17] as a box prompt for mask generation. While these methods make the reasoning trace explicit, the reasoning steps remain textual: purely linguistic descriptions can be visually ambiguous when multiple similar objects are present, leading to incorrect localization, as shown in Fig. 1 (b). Although methods such as Visual CoT [34] introduce bounding boxes into intermediate reasoning steps, the coordinates are merely decoded as text tokens without attending to the corresponding visual features, making intermediate reasoning unverifiable and allowing errors to propagate to the final prediction. These approaches either leave intermediate reasoning implicit or express it through textual CoT. In both cases, the alignment between intermediate reasoning representations and the corresponding image regions is not explicitly supervised.

To this end, we propose Reasoning with Visual Anchors (ReVA), a framework that grounds each CoT reasoning step in visual features during training, keeping the reasoning process connected to visual evidence. ReVA trains the MLLM with a Chain-of-Anchor (CoA) reasoning trace, where each anchor is a special token whose hidden representation is directly aligned to the features of a referent image region. The model attends to the corresponding image region through the anchor token, grounding in concrete visual content. To ensure that each grounded visual reasoning step can propagate to the final target, we further introduce a multi-hop visual transition supervision, which supervises the propagation from anchors to the target by requiring each hop to follow the spatial relation described in the intervening text.

Existing segmentation datasets provide target masks but lack reasoning traces with grounded anchors. Therefore, we construct the Chain-of-Anchor dataset, a large-scale collection of over 245K quality-filtered samples. The dataset is built on existing RefCOCO datasets [16, 28, 47] by augmenting each sample with a grounded reasoning trace generated by an MLLM, introducing no additional mask annotation beyond original targets. To ensure annotation quality, we apply rule-based and model-based quality filtering to check spatial accuracy and reasoning quality.

We conduct extensive experiments on both reasoning and referring segmentation benchmarks to comprehensively evaluate ReVA. On the ReasonSeg test set, ReVA surpasses the previous state of the art by 6.7 cIoU points, demonstrating its effectiveness in segmenting objects described by complex, implicit queries. ReVA also achieves consistently strong results across the RefCOCO series, highlighting the applicability of its anchor-based grounding strategy to standard referring segmentation tasks.

2 Related Work

2.1 Multimodal Large Language Models and Chain of Thought

Recent multimodal large language models (MLLMs) have become general-purpose vision-language assistants for open-ended interaction [10, 11, 12]. Proprietary models such as GPT-4 [15], Gemini [9], and Claude [1] demonstrate strong cross-modal understanding, while open-source models such as the LLaVA family [20, 19] and the Qwen-VL series [2, 35, 4, 3] explore visual instruction tuning, dynamic-resolution modeling, and multilingual multimodal interaction. Despite this progress, most MLLMs remain optimized for image understanding or coarse visual grounding, rather than pixel-level reasoning over image masks.

To handle complex queries, recent work incorporates explicit reasoning process into MLLMs, following paradigms such as Chain-of-Thought and Tree of Thoughts [43, 46]. In multimodal settings, this line includes multimodal thought chains [26] and visual CoT alignment [34]. These approaches improve multi-step deduction, but the reasoning process still unfolds primarily in the text token space, leaving intermediate steps not connected to concrete image regions. ReVA addresses this limitation by grounding intermediate reasoning in region-level visual features, making each step more directly verifiable against the image.

2.2 Reasoning Segmentation

LISA [18] introduces reasoning segmentation and the embedding-as-mask paradigm, in which a special `[SEG]` token's MLLM hidden state is projected to a mask decoder, making implicit textual queries compatible with pixel-level prediction. This design established a widely adopted paradigm that subsequent methods extend with multi-target segmentation [33, 38], language-based grounding [32, 48], and generalization to video and multi-task settings [41, 42, 23, 49]. Other methods inject useful signals into the reasoning process, READ [30] extracts spatial prompts from token-patch similarity maps to enhance segmentation performance. Despite their diversity, all these methods only provide learning objectives for the final mask, leaving the intermediate reasoning steps from query to target region entirely unexplored.

Recent reinforcement-learning and agentic methods encourage explicit reasoning with textual CoT. Seg-Zero [24], VisionReasoner [25], and PixelThink [36] decouple reasoning from segmentation: the MLLM produces explicit CoT and a bounding box, which an external segmentation model converts to a mask. LENS [52] introduces context queries into the above framework to enable end-to-end training. Recent works also explore agent-based paradigms that leverage MLLMs to interactively prompt SAM for reasoning segmentation. SAM-Veteran [8] and SAM3-Agent [7] introduce a framework that emulates human-like SAM usage through iterative bounding box generation, point-based mask refinement, and adaptive termination, achieving strong performance.

While these works improve segmentation accuracy, the intermediate reasoning process is still unsupervised as a chain of concrete visual evidence. ReVA targets this by introducing visual anchors into the reasoning chain: each anchor is aligned with latent region-level evidence, while multi-hop transition supervision encourages the chain to progress toward the final target in a visually verifiable manner.

3 Methodology

Problem Definition Reasoning segmentation requires producing pixel-level masks for objects described by implicit queries. The input of reasoning segmentation consists of an image $I \in \mathbb{R}^{H\times W\times 3}$ and a text query Q, where the target object is not explicitly referred to as an object but must be inferred through reasoning. The model must output a binary mask $\hat{\mathbf{M}} \in \{0,1\}^{H\times W}$ that segments the target object inferred from Q.

Overview Fig. 2 illustrates the overall ReVA framework. During training, ReVA decomposes the reasoning process into a reasoning trace consisting of k anchor tokens $\{a_i\}_{i=1}^{k}$ and a final target token t. The hidden state $\mathbf{h}_{a_k}$ of each anchor token, rendered as *<anchor_k>*, is projected into the visual feature space and aligned with its corresponding grounded image region R_k, forming a visual evidence chain. A visual transition loss models the spatial transitions between successive anchors,

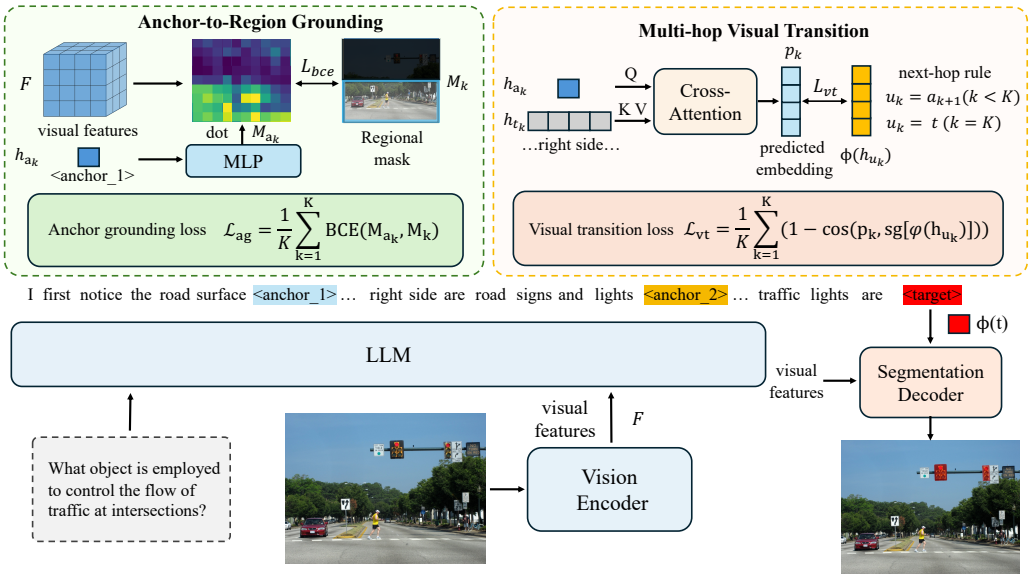

Figure 2: Overview of ReVA. The LLM is supervised by a reasoning trace interleaving text with anchor tokens and a final target token. Each anchor is grounded to a referent image region via an MLP projection supervised by $\mathcal{L}_{\text{ag}}$, and a cross-attention transition head predicts the next reasoning hop supervised by $\mathcal{L}_{\text{vt}}$, ensuring the reasoning chain progresses correctly toward the target. The target token hidden state, together with visual features, is decoded into the final segmentation mask.

ensuring the reasoning trace progressively leads to the target. The hidden state of the target token $\mathbf{h}_t$ is projected into the decoder to generate the segmentation mask $\hat{\mathbf{M}}$. During inference, the model generates the target token given only the image and query, and decodes it into the final segmentation mask.

3.1 Reasoning with Visual Anchors

Chain-of-Anchor Modeling. Given image I and query Q, the MLLM parameterized by θ is supervised with an annotated CoA reasoning trace $\mathcal{T}$, represented as a token sequence:

$$\mathcal{T} = (x_1, x_2, \ldots, x_T), \qquad p_\theta(\mathcal{T} \mid I, Q) = \prod_{i=1}^{T} p_\theta(x_i \mid x_{<i}, I, Q),$$

where p_θ denotes the conditional next-token distribution defined by the MLLM, and $\mathcal{T}$ contains ordinary text tokens, anchor tokens, and the final target token. Each anchor token follows an intermediate referent object or image region mentioned during reasoning (e.g., "the keyboard" or "the road surface"), while the target token marks the final answer. During training, the trace is supervised by the standard causal language modeling loss for the outputs:

$$\mathcal{L}_{\text{t}} = -\sum_{i=1}^{T} \log p_\theta(x_i \mid x_{<i}, I, Q).$$

Anchor-to-Region Grounding. To ensure that each reasoning step is grounded in concrete visual evidence rather than purely textual reasoning, we explicitly align each anchor token a_k with the features from a specific image region R_k. We project the anchor's hidden state $\mathbf{h}_{a_k} \in \mathbb{R}^D$ into the vision encoder's feature space via a two-layer MLP anchor projection head:

$$\mathbf{v}_{a_k} = \text{MLP}(\mathbf{h}_{a_k}).$$

Let $\mathbf{F} \in \mathbb{R}^{H' \times W' \times D_v}$ denote the visual features from the visual encoder. We compute a spatial logit map of the anchor via similarity with visual features by dot-product:

$$\mathbf{M}_{a_k} = \mathbf{F}_{h,w}^{\top} \mathbf{v}_{a_k}.$$

The anchor grounding loss is computed between this logit map and the ground-truth binary mask $\mathbf{M}_k \in \{0,1\}^{H' \times W'}$ of image region R_k:

$$\mathcal{L}_{\text{ag}} = \frac{1}{K} \sum_{k=1}^{K} \text{BCE}(\mathbf{M}_{a_k},\ \mathbf{M}_k)\,.$$

This ensures that each anchor token carries a spatially accurate representation of its referent image region, rather than merely encoding a textual description.

Multi-hop Visual Transition. Anchor-to-region grounding supervises each anchor in isolation, ensuring that each <*anchor_k*> refers to the correct image region. However, it does not imply that the reasoning chain follows the correct transitions between successive anchors toward the final target. Meanwhile, the language modeling loss $\mathcal{L}_{\text{t}}$ only supervises the text token sequence, but does not verify the spatial reasoning against visual representations. To bridge this gap, we introduce a multi-hop visual transition mechanism, implemented with a lightweight transition head that supervises whether the grounded reasoning chain correctly progresses from one anchor to the next in the visual representation space. Given the generated Chain-of-Anchor reasoning trace, for the k-th hop, let the current anchor be a_k, and the next hop be:

$$u_k = \begin{cases} a_{k+1}, & k < K, \\ t, & k = K, \end{cases}$$

where K is the number of anchors. The current anchor provides the source visual representation, while the intervening text provides the relational condition. The transition head applies text-conditioned cross-attention by taking the anchor hidden state as the query and the text hidden states as the key and value:

$$\mathbf{p}_k = \text{CrossAttn}(Q{=}\mathbf{h}_{a_k},\ K{,}V{=}\mathbf{h}_{u_k}),$$

We supervise this prediction using the projected representation of the true next destination:

$$\mathcal{L}_{\text{vt}} = \frac{1}{K} \sum_{k=1}^{K} \left(1 - \cos(\mathbf{p}_k,\ \text{sg}[\psi(\mathbf{h}_{u_k})])\right),$$

where $\psi(\cdot)$ is the projection head, and $\text{sg}[\cdot]$ denotes the stop-gradient operation. This objective directly supervises whether each grounded reasoning step leads to the correct next destination. If the current anchor is mislocalized, or if the intervening text span fails to encode the correct spatial relation, the predicted transition cannot match the representation of the next anchor or the final target, and the model is penalized accordingly. In this way, the intermediate anchors are no longer optimized as isolated auxiliary signals, but are explicitly tied to the quality of the reasoning chain toward the final target.

Target-to-Mask Decoding. Finally, the hidden state of the final target token $\mathbf{h}_t$ is projected via target projection head $\phi(\cdot)$ and passed into a segmentation decoder to produce the predicted mask $\hat{\mathbf{M}}$. The segmentation loss combines binary cross-entropy and Dice loss:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{BCE}}(\hat{\mathbf{M}},\ \mathbf{M}^*) + \mathcal{L}_{\text{Dice}}(\hat{\mathbf{M}},\ \mathbf{M}^*),$$

where $\mathbf{M}^*$ is the ground-truth binary mask.

3.2 Dataset Construction

Visual Evidence Chain Generation Training ReVA requires CoT reasoning traces with visual anchor annotations towards the final target. Standard referring segmentation datasets provide reliable target masks but lack the intermediate reasoning traces that connect the query to the target. We therefore construct a Chain-of-Anchor dataset from the RefCOCO series [16] (I, Q, M) triplets, as summarized in Fig. 3. For each triplet, the ground-truth mask M and its bounding box coordinates are provided as text to a strong MLLM (Qwen3-VL-235B-A22B). The model is then prompted to generate a concise reasoning trace containing visual anchor tokens, each with bounding box coordinates marking an intermediate image region that helps locate the target. Each trace ends with a <*target*> token denoting the object to be segmented.

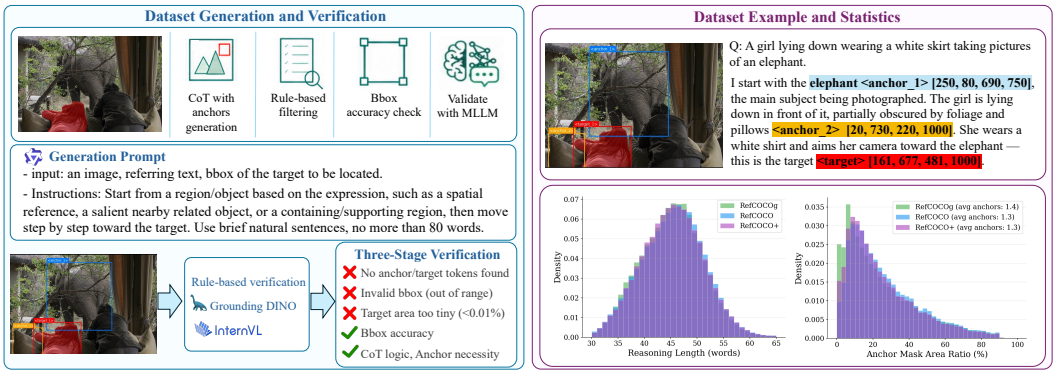

Figure 3: Chain-of-Anchor data generation and filtering pipeline. A strong MLLM generates spatially grounded reasoning traces, which are then filtered through rule-based filtering and two model-based visual verification.

Trace Verification and Dataset Statistics We apply a three-stage verification pipeline to ensure the quality of the generated traces, as shown in Fig. 3. The first stage applies rule-based checks, removing samples with missing grounding tokens, out-of-bound or degenerate boxes, duplicated anchors or targets too small for stable supervision. The second stage verifies spatial grounding accuracy: each anchor bounding box is validated against detections from Grounding DINO [22] using the anchor's noun phrase as the query. Samples are discarded if any anchor box fails to match a detection above the IoU threshold. The third stage employs a separate MLLM (InternVL3.5 [37]) as an independent judge, scoring each remaining trace on reasoning quality and anchor necessity. Using a different model family from the generator avoids bias. Traces scoring below a threshold are removed. After these stages, the final dataset contains over 245K quality-filtered training samples. Each retained sample provides supervision for all three objectives: the reasoning trace, the anchor-region grounding, and the final segmentation mask, enabling ReVA to learn both the target and the visual evidence chain that leads to it. Detailed statistics and MLLM prompts are provided in the *appendix*.

4 Experiments

Implementation Details ReVA is built on Qwen3-VL-8B [3] as the multimodal large language model backbone and SAM-H [17] as the segmentation backbone. The <*anchor_k*> and <*target*> are added to the tokenizer as special tokens, and their embeddings are learned during training. The trainable components include the Qwen3-VL backbone, the anchor/target projection head, the visual transition head, and the SAM mask decoder. The SAM image encoder is initialized from pretrained SAM-H weights and retained as the visual feature extractor following [18, 30].

All models are trained for 4200 steps on $8\times$ NVIDIA H200 GPUs, using a global batch size of 256. We use the AdamW optimizer with an initial learning rate of 4×10^{-5} and a weight decay of 0.1. The learning rate follows a cosine decay schedule with 3% warmup steps. We jointly optimize the reasoning trace, anchor-to-region grounding, and multi-hop visual transition prediction, all with the loss weight set to 1. For the final mask prediction, we combine BCE and Dice losses.

During inference, ReVA takes only the image and text query as input, and decodes the final mask from the generated target token. We report gIoU and cIoU on ReasonSeg, and cIoU on RefCOCO, RefCOCO+, and RefCOCOg following the standard evaluation protocols.

Datasets Following the standard setup in LISA [18], our training data consists of four groups: (1) semantic segmentation datasets (ADE20K [50], COCO-Stuff [6], LVIS-PACO [31]) for general dense prediction across diverse scenes; (2) referring segmentation datasets [16, 28, 47] (RefCOCO, RefCOCO+, RefCOCOg), for which we generate Chain-of-Anchor reasoning traces using Qwen3-VL-235B to provide anchor supervision; (3) the ReasonSeg training set for reasoning segmentation; and (4) the LLaVA instruction tuning dataset [19] to retain the model's general visual-language reasoning ability. The CoA traces are derived from the same RefCOCO segmentation annotations without introducing additional labeled masks.

Table 1: Reasoning segmentation results on ReasonSeg, compared with prior state-of-the-art methods at comparable model scales. The best and second-best results are highlighted in **bold** and underlined.

Method	ReasonSeg val		ReasonSeg test					
	Overall		Short query		Long query		Overall	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
LISA-7B [18]	52.9	54.0	40.6	40.6	49.4	51.0	47.3	48.4
LISA-7B-LLaVA1.5 [18]	61.3	62.9	48.3	46.3	57.9	59.7	55.6	56.9
GSVA-7B [44]	50.5	56.4	–	–	–	–	–	–
SegLLM-7B [38]	–	54.3	–	–	–	54.2	–	48.4
CoReS-7B [5]	59.4	–	44.2	–	55.0	–	52.4	–
RSVP-7B [27]	58.6	48.5	48.5	44.3	57.1	53.8	56.6	51.6
POPEN-7B [51]	60.2	64.5	–	–	–	–	–	–
UniPixel-7B [23]	60.5	58.7	–	–	–	–	–	–
Seg-Zero [24]	62.6	62.0	–	–	–	–	57.5	52.0
SAM-R1 [14]	64.0	55.8	–	–	–	–	60.2	54.3
READ-7B [30]	59.8	<u>67.6</u>	<u>52.6</u>	<u>49.5</u>	<u>60.4</u>	<u>61.0</u>	58.5	<u>58.6</u>
SAM-Veteran [8]	<u>68.2</u>	67.3	–	–	–	–	<u>62.6</u>	56.1
PixelThink [36]	63.8	62.7	–	–	–	–	60.2	55.8
ReVA-8B	**70.3**	**70.6**	**67.5**	**61.8**	**68.1**	**65.4**	**68.0**	**65.3**

4.1 Main Results

We first report results on the reasoning segmentation benchmark ReasonSeg. We compare ReVA with prior state-of-the-art reasoning segmentation methods at comparable model scales, including supervised fine-tuning based approaches such as LISA [18], SegLLM [38], and READ [30], as well as reinforcement learning based methods including Seg-Zero [24], SAM-R1 [14], and SAM-Veteran [8]. Following the standard protocols [18, 30], we report results on the validation split and the test split, with the test split further divided into short-query and long-query subsets.

As shown in Table 1, ReVA achieves state-of-the-art performance across all evaluation splits. On the test set, ReVA reaches 68.0 gIoU and 65.3 cIoU overall, outperforming READ-7B by 9.5 gIoU and the best RL-based method SAM-Veteran by 5.4 gIoU. On the validation set, ReVA obtains 70.3 gIoU and 70.6 cIoU, surpassing SAM-Veteran by 2.1 gIoU and 3.3 cIoU. Notably, ReVA shows strong performance on the long-query split (68.1 gIoU, 65.4 cIoU), which involves more implicit and complex reasoning queries, suggesting that our method captures the visual grounded spatial relations required for challenging descriptions. On the short-query split, ReVA also achieves results with 67.5 gIoU and 61.8 cIoU, demonstrating consistent gains across both types of reasoning queries.

We further evaluate ReVA on the RefCOCO series benchmarks, comparing with other reasoning segmentation methods. As shown in Table 2, ReVA performs competitively against prior methods. On RefCOCO, ReVA reaches 80.3 cIoU on val and 78.6 on testB, surpassing prior methods, including Baseline Qwen3-VL-8B (w/ [SEG]) and SegLLM. On RefCOCO+, ReVA obtains 73.8, 78.9, and 69.2 cIoU on val, testA, and testB, respectively, outperforming the previous best results by a clear margin on testA and testB. On RefCOCOg, ReVA achieves 76.8 and 77.5 cIoU on val and test, exceeding all compared methods. Although referring expression segmentation requires less complex reasoning than ReasonSeg, these results demonstrate that ReVA generalizes well beyond reasoning segmentation, maintaining strong performance on referring segmentation.

4.2 Ablation Study

We conduct ablation experiments on the ReasonSeg validation and test sets to analyze the contribution of each component and validate key design choices. Results are summarized in Table 3.

Learning Objectives (A1). We use Qwen3-VL-8B with `[SEG]` token as the baseline and progressively add each learning objective to ablate their contributions. Adding the CoA reasoning trace with language and box supervision improves performance, confirming that explicit reasoning steps benefit segmenting the target object. Introducing anchor-to-region grounding $\mathcal{L}_{\text{ag}}$ yields a further

Table 2: Referring expression segmentation results on RefCOCO, RefCOCO+, and RefCOCOg. The best and second-best results are highlighted in **bold** and underlined.

Method	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val(U)	test(U)
CRIS [39]	70.5	73.2	66.1	62.3	68.1	53.7	59.9	60.4
LAVT [45]	72.7	75.8	68.8	62.1	68.4	55.1	61.2	62.1
PolyFormer-B [21]	74.8	76.6	71.1	67.6	72.9	59.3	67.8	69.1
LISA-7B [18]	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
PixelLM-7B [33]	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5
LaSagnA-7B [40]	76.8	78.7	73.8	66.4	70.6	60.1	70.6	71.9
CoReS-7B [5]	76.0	78.6	72.5	65.1	70.0	58.6	69.0	70.7
GSVA-7B [44]	77.2	78.9	73.5	65.9	69.6	59.8	72.7	73.3
Qwen3-VL-8B [3]	76.5	78.2	74.5	69.2	75.6	65.7	74.0	74.4
SegLLM-7B [38]	80.2	81.5	75.4	70.3	73.0	62.5	72.6	73.6
READ-7B [30]	78.1	80.2	73.2	68.4	73.7	60.4	70.1	71.4
SAM3-Agent [7]	68.6	72.3	63.9	59.3	64.5	55.6	66.4	66.9
Seg-Zero [24]	–	80.3	–	–	76.2	–	–	72.6
PixelThink [36]	–	79.3	–	–	74.8	–	–	73.9
SAM-Veteran [8]	–	80.8	–	–	76.6	–	–	73.4
ReVA-8B	**80.3**	**81.8**	**78.6**	**73.8**	**78.9**	**69.2**	**76.8**	**77.5**

Table 3: Ablation studies on ReasonSeg. **A1**: Incremental ablations on the learning objectives. **A2**: Anchor effectiveness analysis. **A3**: Architecture design choices (ablated from full ReVA).

Setting	Val		Test	
	gIoU	cIoU	gIoU	cIoU
A1. Learning Objectives				
Baseline (Qwen3VL + `[SEG]` token)	61.4	60.9	59.8	57.5
+ CoA trace text loss $\mathcal{L}_{\text{t}}$ (w/ bbox)	63.4	62.9	61.8	60.1
+ Anchor grounding loss $\mathcal{L}_{\text{ag}}$	68.7	68.6	67.0	63.8
+ Transition loss $\mathcal{L}_{\text{vt}}$ (ReVA)	**70.3**	**70.6**	**68.0**	**65.3**
A2. Anchor Effectiveness (ablated from full ReVA)				
Only CoT as text output (no anchor)	62.8	61.7	61.0	59.4
Replacing anchor with random vectors	63.2	63.3	62.0	60.4
Random drop anchors ($p = 0.2$)	67.6	68.4	66.8	63.2
A3. Architecture Design (ablated from full ReVA)				
w/o stop-gradient in $\mathcal{L}_{\text{vt}}$	67.8	68.2	66.3	62.6
Pooled feature alignment in $\mathcal{L}_{\text{ag}}$	66.5	66.8	64.4	61.9
MLP as transition head	69.6	69.5	67.3	64.8
Anchor + target fusion for decoding	68.8	68.5	66.4	64.1

gain, demonstrating that grounding each anchor in concrete visual evidence improves segmentation performance. Finally, incorporating multi-hop visual transition supervision $\mathcal{L}_{\text{vt}}$ achieves the best results, validating that constraining the relational transitions between successive anchors helps the model capture the full reasoning chain toward the target.

Anchor Effectiveness (A2). We conduct ablation experiments to verify the effectiveness of anchors. Replacing anchor hidden states with random vectors leads to a significant drop, showing that the learned anchor representations carry meaningful visual semantics. Randomly dropping anchors (with probability 0.2) also degrades performance, indicating that ReVA relies on a complete visual evidence chain. Adding MLLM-generated text CoT alone yields only 61.7 and 59.4 cIoU, showing limited contribution of the distilled text reasoning without visual grounding.

Architecture Design (A3). We conduct ablation experiments on the key architectural designs from the full ReVA model. Removing the stop-gradient operation in $\mathcal{L}_{\text{vt}}$ degrades performance, as gradients

from the transition loss flow back through the target representation and destabilize training. Replacing our spatial map-based supervision with pooled feature cosine alignment for $\mathcal{L}_{\text{ag}}$ also reduces accuracy. We observe that the cosine loss collapses to zero early in training, as aligning a single pooled vector is too easy and fails to provide an effective gradient signal. Replacing the cross-attention transition head with an MLP over only the anchor hidden state leads to a drop, confirming that intervening text carries essential relational information for transition prediction. Fusing anchor representations into the mask decoder decreases performance. Intermediate anchors introduce noise to the decoder since they correspond to referent regions rather than the target itself.

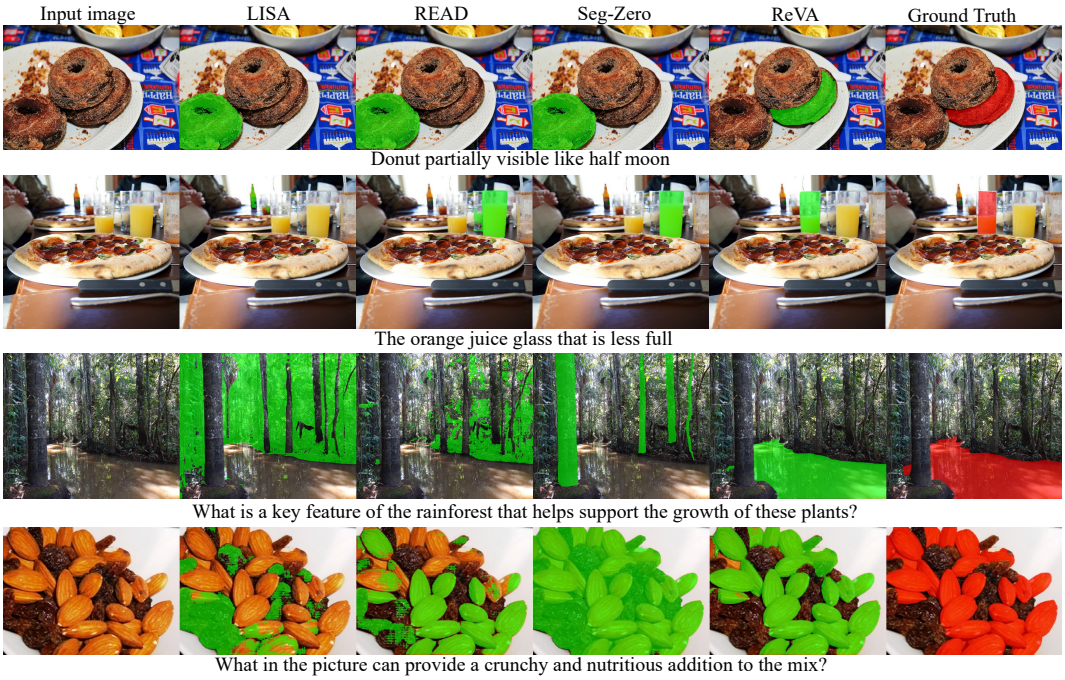

Figure 4: Qualitative comparison of ReVA against prior methods, including LISA [18], READ [30], and Seg-Zero [24]. The first two rows present examples from RefCOCO+, while the last two rows present examples from ReasonSeg.

4.3 Qualitative Results

As shown in Fig. 4, we compare ReVA with LISA [18], READ [30], and Seg-Zero [24] on both referring and reasoning segmentation. In the RefCOCO+ examples, prior methods often select incorrect instances or produce incomplete masks when multiple visually similar objects are present. ReVA better distinguishes the target and produces accurate segmentation mask. In the ReasonSeg examples, the queries require implicit reasoning to identify the target. Prior methods tend to drift toward salient but irrelevant regions, while ReVA correctly locates the intended object.

5 Conclusion

We propose ReVA, a framework for reasoning segmentation that grounds each step of chain-of-thought reasoning in concrete visual evidence. By introducing visual anchor tokens into the reasoning trace and supervising both their spatial alignment with image regions and their multi-hop transitions toward the target, ReVA turns textual reasoning traces into spatially grounded CoA training supervision. We also constructed a large-scale Chain-of-Anchor dataset of over 245K quality-filtered samples through a scalable generation-and-verification pipeline. Experiments on the ReasonSeg and RefCOCO benchmarks demonstrate that ReVA achieves state-of-the-art performance, outperforming prior methods.

Acknowledgment: This work was supported by the NSFC under Grant 62322604 and 62576207.

References

[1] Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024.

[2] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[3] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] S. Bai, K. Chen, X. Liu, J. Wang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] X. Bao, S. Sun, S. Ma, K. Zheng, Y. Guo, G. Zhao, Y. Zheng, and X. Wang. Cores: Orchestrating the dance of reasoning and segmentation. In *European Conference on Computer Vision*, pages 187–204. Springer, 2024.

[6] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[7] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[8] T. Du, H. Li, Z. Fan, J. Zhang, P. Pan, and Y. Zhang. SAM-veteran: An MLLM-based human-like SAM agent for reasoning segmentation. In *The Fourteenth International Conference on Learning Representations*, 2026.

[9] G. Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[10] T. Guan, Z. Wang, P. Fu, Z. Guo, W. Shen, K. Zhou, T. Yue, C. Duan, H. Sun, Q. Jiang, et al. A token-level text image foundation model for document understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23210–23220. IEEE, 2025.

[11] T. Guan, Z. Yang, J. Wan, M. Yang, Z. Guo, Z. Hu, R. Luo, R. Chen, S. Jiang, P. Wang, et al. Codepercept: code-grounded visual stem perception for mllms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 33542–33552, 2026.

[12] B. Hu and Y. Zhang. How ai and humans express comfort differently: A corpus-based appraisal analysis. *Corpus Pragmatics*, 10(1):16, 2026.

[13] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124, 2016.

[14] J. Huang, Z. Xu, J. Zhou, T. Liu, Y. Xiao, M. Ou, B. Ji, X. Li, and K. Yuan. SAM-r1: Leveraging SAM for reward feedback in multimodal segmentation via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[15] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[16] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[18] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.

[19] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[20] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023.

[21] J. Liu, H. Ding, Z. Cai, Y. Zhang, R. K. Satzoda, V. 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*, pages 18653–18663, 2023.

[22] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55, 2024.

[23] Y. Liu, Z. Ma, J. Pu, Z. Qi, Y. Wu, S. Ying, and C. W. Chen. Unipixel: Unified object referring and segmentation for pixel-level visual reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[24] Y. Liu, B. Peng, Z. Zhong, Z. Yue, F. Lu, B. Yu, and J. Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

[25] Y. Liu, T. Qu, Z. Zhong, B. Peng, S. Liu, B. Yu, and J. Jia. Visionreasoner: Unified reasoning-integrated visual perception via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[26] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[27] Y. Lu, J. Cao, Y. Wu, B. Li, L. Tang, Y. Ji, C. Wu, J. Wu, and W. Zhu. Rsvp: Reasoning segmentation via visual prompting and multi-modal chain-of-thought. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

[28] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *Proceedings of the European Conference on Computer Vision*, pages 792–807, 2016.

[29] Z. Peng, Z. Xu, Z. Zeng, C. Wen, Y. Huang, M. Yang, F. Tang, and W. Shen. Understanding fine-tuning clip for open-vocabulary semantic segmentation in hyperbolic space. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4562–4572, 2025.

[30] R. Qian, X. Yin, and D. Dou. Reasoning to attend: Try to understand how< seg> token works. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24722–24731, 2025.

[31] V. Ramanathan, A. Kalia, V. Petrovic, Y. Wen, B. Zheng, B. Guo, R. Wang, A. Marquez, R. Kovvuri, A. Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023.

[32] H. Rasheed, M. Maaz, S. Shaji, A. Shaker, S. Khan, H. Cholakkal, R. M. Anwer, E. Xing, M.-H. Yang, and F. S. Khan. Glamm: Pixel grounding large multimodal model. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[33] Z. Ren, Z. Huang, Y. Wei, Y. Zhao, D. Fu, J. Feng, and X. Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26374–26383, 2024.

[34] H. Shao, S. Qian, H. Xiao, G. Song, Z. Zong, L. Wang, Y. Liu, and H. Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[35] P. Wang, S. Bai, S. Tan, S. Wang, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[36] S. Wang, G. Fang, L. Kong, X. Li, J. Xu, S. Yang, Q. Li, J. Zhu, and X. Wang. Pixelthink: Towards efficient chain-of-pixel reasoning. *arXiv preprint arXiv:2505.23727*, 2025.

[37] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[38] X. Wang, S. Zhang, S. Li, K. Li, K. Kallidromitis, Y. Kato, K. Kozuka, and T. Darrell. SegLLM: Multi-round reasoning segmentation with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[39] Z. Wang, Y. Lu, Q. Li, X. Tao, Y. Guo, M. Gong, and T. Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11686–11695, 2022.

[40] C. Wei, H. Tan, Y. Zhong, Y. Yang, and L. Ma. Lasagna: Language-based segmentation assistant for complex queries. *arXiv preprint arXiv:2404.08506*, 2024.

[41] C. Wei, Y. Zhong, H. Tan, Y. Liu, Z. Zhao, J. Hu, and Y. Yang. Hyperseg: Towards universal visual segmentation with large language model. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2025.

[42] C. Wei, Y. Zhong, H. Tan, Y. Zeng, Y. Liu, Z. Zhao, and Y. Yang. Instructseg: Unifying instructed visual segmentation with multi-modal large language models. *arXiv preprint arXiv:2412.14006*, 2024.

[43] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[44] Z. Xia, D. Han, Y. Han, X. Pan, S. Song, and G. Huang. Gsva: Generalized segmentation via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3858–3869, 2024.

[45] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18155–18165, 2022.

[46] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc., 2023.

[47] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85, 2016.

[48] Y. Yuan, W. Li, J. Liu, D. Tang, X. Luo, C. Qin, L. Zhang, and J. Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28202–28211, 2024.

[49] Z. Zhang, Y. Ma, E. Zhang, and X. Bai. Psalm: Pixelwise segmentation with large multi-modal model. In *European Conference on Computer Vision*, pages 74–91. Springer, 2025.

[50] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[51] L. Zhu, T. Chen, Q. Xu, X. Liu, D. Ji, H. Wu, D. W. Soh, and J. Liu. Popen: Preference-based optimization and ensemble for lvlm-based reasoning segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30231–30240, 2025.

[52] L. Zhu, B. Ouyang, Y. Zhang, T. Cheng, R. Hu, H. Shen, L. Ran, X. Chen, L. Yu, W. Liu, and X. Wang. Lens: Learning to segment anything with unified reinforced reasoning, 2025.

A Chain-of-Anchor Dataset Details

A.1 Dataset Statistics

Table 4 summarizes the statistics of our Chain-of-Anchor (CoA) dataset, built from the RefCOCO series. We report the statistics before and after the three-stage verification pipeline.

Table 4: Chain-of-Anchor dataset statistics.

Source	Source	After Filtering		
	Samples	Samples	Avg. Anchors	Avg. Words
RefCOCO	120,624	90,429	1.30	45.3
RefCOCO+	120,191	86,340	1.32	45.3
RefCOCOg	80,512	69,199	1.41	44.7
Total	**321,327**	**245,968**	1.34	45.1

Table 5: Distribution of anchor counts per sample in the filtered CoA dataset.

Source	1 anchor	2 anchors	3 anchors
RefCOCO	63,348 (70.1%)	26,668 (29.5%)	413 (0.5%)
RefCOCO+	59,371 (68.8%)	26,627 (30.8%)	342 (0.4%)
RefCOCOg	41,602 (60.1%)	26,941 (38.9%)	656 (0.9%)

The overall retention rate of the filtering pipeline is 76.6%, indicating that the majority of generated traces are of high quality, while the filtering effectively removes the remaining problematic samples (missing anchors, spatially inaccurate grounding, logically inconsistent reasoning, or degenerate single-anchor traces where the anchor coincides with the target). Table 5 shows the distribution of anchor counts per sample in the final filtered dataset. The majority of samples contain 1–2 anchors, with a small portion containing 3 anchors for complex scenarios.

A.2 Chain-of-Anchor Generation Prompt

Figs. 5–7 show the prompt templates used to generate Chain-of-Anchor reasoning traces from Qwen3-VL-235B-A22B. The model is provided with the image, the referring expression, and the ground-truth target bounding box, and is asked to produce a concise reasoning trace with intermediate anchor tokens grounding relevant objects.

```
COA_SYSTEM_PROMPT = """\
You are a visual reasoning assistant for referring image segmentation. Given a referring expression and an image with the target, write a Chain-of-Thought (CoT) explaining which visual regions/objects (anchors) help locate the target.

**Rules:**
- Use 0-3 anchors. Only include anchors that necessarily help locate the target.
- If no reliable anchor can be found, output "The expression refers to <target_1> [x1, y1, x2, y2]".
- Embed tokens INLINE: `<anchor_N> [x1, y1, x2, y2]` right after the region/object name, `<target_1> [x1, y1, x2, y2]` at the end.
- Do NOT list all <anchor_N> tokens at the end — they must appear where first mentioned.
- Use brief natural sentences.

**Example** expression: "the cup to the right of the keyboard":
I locate the keyboard <anchor_1> [180, 420, 700, 610], the main reference. To its right sits a cluster of objects on the desk <anchor_2> [720, 380, 950, 680]. The cup with a handle directly right of the keyboard is the target <target_1> [730, 390, 860, 650]."""
```

Figure 5: System prompt of Chain-of-Anchor Generation.

```
COT_USER_TEMPLATE = """\
Referring expression: "{question}"
Target location: <target_1> [{nx1}, {ny1}, {nx2}, {ny2}] (image {width}x{height} px)

Write a concise Chain-of-Thought (0-3 anchors, all coords integers in [0, 1000]):
1. Start from a region/object based on the expression, such as a spatial reference, a salient nearby related object, or a
containing/supporting region, then move step by step toward the target.
2. Mark each anchor as `<anchor_N> [x1, y1, x2, y2]` inline.
3. End with `<target_1> [x1, y1, x2, y2]` matching the location above."""
```

Figure 6: User prompt of Chain-of-Anchor Generation.

```
QUALITY_SYSTEM_PROMPT = """\
You are a quality evaluator for visual grounding training data.
Given a referring expression, a Chain-of-Thought (CoT) reasoning with bounding boxes, and the image with those boxes
drawn, evaluate on these two dimensions (each 1-5):

- CoT logic: Is the reasoning coherent and does it logically lead from the expression to the target?
5: flawless logic, 3: some gaps but mostly makes sense, 1: incoherent or contradicts the image
- anchor_necessity: Is each anchor genuinely needed to locate the target? Are there redundant or irrelevant anchors?
5: every anchor serves a clear purpose, 3: one or two anchors are unnecessary, 1: most anchors are pointless

Respond in EXACTLY this JSON format (no other text):
{"cot_logic": <int>, "anchor_necessity": <int>, "reason": "<1-2 sentences>"}"""
```

Figure 7: System prompt of MLLM-based data filtering.

A.3 Verification Pipeline Details

Stage 1: Rule-based Filtering. We apply rule-based checks to remove wrong traces: (1) traces must contain at least one anchor token and exactly one target token; (2) all bounding box coordinates must be within [0, 1000] and have positive area (i.e., $x_2 > x_1$ and $y_2 > y_1$); (3) anchor bounding boxes must have a minimum area of 0.5% of the image to ensure stable supervision; (4) no duplicate anchor tokens are allowed within a single trace.

Stage 2: Grounding DINO Verification. For each anchor in the trace, we extract its noun phrase from the surrounding text and use Grounding DINO [22] with the noun phrase as the text query. We compute the IoU between the anchor's bounding box (from the generated trace) and the detection from Grounding DINO. Samples are retained only if all anchors achieve an IoU ≥ 0.5.

Stage 3: InternVL3.5 Judging. A separate MLLM (InternVL3.5-241B), from a different model family than the generator, independently evaluates each reasoning trace. The prompt used in quality filtering is shown in Fig 7. The judge scores each trace on: (1) whether the reasoning logically connects the expression to the target, (2) whether each anchor is necessary and contributes to the reasoning chain. Traces scoring below 3/5 are removed.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Reasoning segmentation still requires a large MLLM for reasoning, making inference resource-intensive.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the details can be find in our experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: All benchmarks (ReasonSeg, RefCOCO/+/g) are publicly available. We plan to release our code and the Qwen-generated traces.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details can be find in our experiment section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow standard practice in reasoning segmentation literature and directly report the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report compute resources in the experiment section. All models are trained on 8×NVIDIA H200 GPUs. Training takes approximately 16 hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on fundamental research in reasoning-based image segmentation. We do not foresee any direct negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our model performs segmentation and does not generate content. It poses no significant risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all datasets and models used in this work. All datasets are publicly available for research purposes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets at submission time.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Our method uses Qwen3-VL as the core reasoning backbone, which is fully described in the methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.