Reasoning-Guided Part-Level Visual Grounding via Reinforcement Learning

Kazi Sajeed Mehrab, Hani Alomari, Najibul Haque Sarker, Chia-Wei Tang, Zaber Ibn Abdul Hakim, Anuj Karpatne, and Chris Thomas

Department of Computer Science
Virginia Tech
{ksmehrab, hani, najibulhaque, cwtang, zaberhakim, karpatne, christhomas}@vt.edu

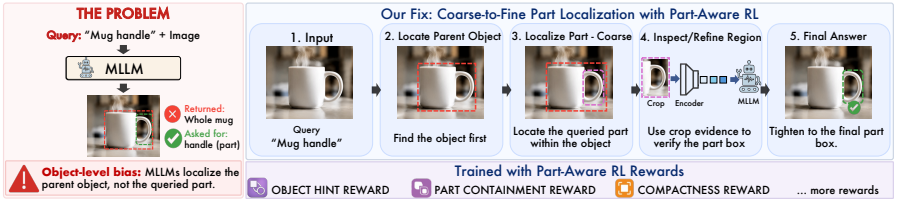

Fig. 1: Our method grounds parts coarse-to-fine: locate the object, then the part within it, then self-checks, re-encoding the predicted crop to refine.

Abstract. Multimodal large language models (MLLMs) ground whole objects well from free-form language queries, but they struggle when the query names a part rather than the object. We trace this to a missing object-part hierarchy, since parts are localized in the same single step used for objects. We propose Object-Part Hierarchical Reflective Grounding (OP-HRG), a coarse-to-fine reasoning-guided grounding strategy that first localizes the parent object and then the part within it. A self-check then reflects on the result, with an extension to re-encode the predicted crop to inspect the region it is correcting. We introduce a part-aware GRPO framework to train our pipeline with stage-wise rewards. A 4B model trained this way outperforms 7B grounding LLMs and SAM3 across PascalPart, PartImageNet, and InstructPart, and transfers to reasoning segmentation. [1]

1 Introduction

Localizing the parts of an object is a core requirement in many real-world settings. A robot told to "grasp the mug (object) by its handle (part)" must distinguish the handle from the body, and a clinician reading a scan must isolate specific anatomical structures (parts) rather than whole organs (objects). We use *part* in the sense established by part-segmentation benchmarks [8,14,49,51]: a constituent sub-region of a parent object – such as a car's wheel or a mug's

[1] Code and models will be available at https://github.com/sajeedmehrab/op-hrg.

handle – that is defined relative to the whole object and is often tied to a specific function or affordance. Localizing a part is harder than localizing the object that contains it: rather than finding a self-contained entity, a model must reason about spatial layout and containment relative to the parent object, much as a person first locates the mug and only then its handle.

Multimodal large language models (MLLMs) such as Qwen-VL [1,50] perform this localization through *visual grounding* – producing a box or mask for the image region named by a free-form query. The strongest publicly available, widely used MLLMs are now strong zero-shot object grounders, yet the same models remain weak when the query names a part rather than a whole object. When asked to ground a mug's handle, an MLLM is likely to return the entire mug instead (Figure 1). Two factors underlie this gap. First, vision-language pretraining is dominated by object-level descriptions, with part annotations comparatively rare and costly to obtain [14,49,51], biasing models toward coarse entities. Second, grounding MLLMs typically handle every query the same way. A part query goes through the same single-step localization as an object, with no mechanism to exploit the natural hierarchy between an object and its parts.

Prior part-grounding approaches span supervised segmenters, token-based MLLMs, and decoupled MLLM-to-SAM pipelines (Sec. 2). The closest to ours, Seg-Zero [30] and VisionReasoner [32], ground whole objects well, but like other grounding MLLMs they localize every query in a single step with no signal for object-part reasoning. We posit that MLLMs already have the capacity for hierarchical reasoning, but it stays dormant without a structured way to activate it, and reinforcement learning offers a direct way to reward and strengthen object-part reasoning.

We therefore propose Object-Part Hierarchical Reflective Grounding (OP-HRG), a prompting paradigm that guides MLLMs to *observe, reason, and localize in a structured coarse-to-fine manner*. Under OP-HRG, the model first decides whether the query refers to an object or a part. If it is a part, the model localizes the parent object as an anchor, then generates an initial part localization within that anchor. As a complementary step, the model reflects on its own answer, checking whether an adjustment is needed before producing the final result. This step-by-step structure mirrors how careful human observation works: locate the object, focus on the part, then verify. We pair this prompting paradigm with a dedicated part-aware Group Relative Policy Optimization (GRPO) framework. Unlike prior RL-based grounding methods that optimize a single localization reward, our reward provides separate, verifiable signals for each stage of the OP-HRG output: parent-object and part localization accuracy, part-in-object containment, self-reflection consistency, and improvement of the final answer over the initial one.

We evaluate our approach in the cross-dataset zero-shot setting on Pascal-Part [8,51] and PartImageNet [14] – benchmarks that part-specific segmenters typically train on, so we compare against MLLM-grounding methods that typically share our zero-shot setting. Using the Qwen3-VL-Instruct-4B model as our backbone, smaller than the 7B models used by competing methods, our method

outperforms these baselines on both object and part categories. Ablation studies confirm that both the OP-HRG prompting and the part-aware rewards are necessary. Our contributions are:

- We analyze the part-grounding gap in MLLMs and show that current pipelines, including RL-optimized ones, underperform on part queries.
- We propose OP-HRG, a structured prompting paradigm that guides MLLMs through object-first hierarchical localization, complemented by a self-reflective step and an active visual perception extension that re-encodes predicted regions as visual evidence for the critique. Our analysis shows the self-reflective step acts mainly as a train-time regularizer.
- We introduce a part-aware GRPO reward framework with stage-wise, verifiable rewards covering parent-object accuracy, part containment, critique consistency, and answer improvement.
- Across three benchmarks, our 4B-parameter model improves part grounding over strong MLLM-grounding baselines – cross-dataset zero-shot on PascalPart and PartImageNet, and in-domain on InstructPart, scales to an 8B backbone, shows only a modest trade-off in general object referring (RefCOCO/+/g), and transfers to reasoning segmentation (ReasonSeg).

2 Related Work

Part-Level Segmentation. Part segmentation has traditionally been approached through supervised methods trained on fixed label sets, evaluated on benchmarks such as PartImageNet [14], PascalPart [8], and InstructPart [49]. Open-vocabulary extensions [11,23,48] generalize to unseen parts through cross-modal correspondence learning or cost aggregation, while unified formulations such as Semantic-SAM [22] unify object-level and part-level predictions within a single framework. However, these approaches rely on part-specific segmentation supervision or learned visual–semantic correspondences, limiting their flexibility. Our work is different from this paradigm: rather than training a pixel-level part-segmentation model, we elicit part localization through structured MLLM reasoning and reinforcement learning with dedicated part-aware rewards. We target part grounding within MLLMs rather than part segmentation in general.

Promptable Segmentation Models. The Segment Anything Model (SAM) [19] established promptable segmentation at scale, producing high-quality masks from spatial prompts such as points and boxes. SAM2 [40] improves upon SAM-v1 for image segmentation from spatial prompts, and SAM3 [5] adds text-conditioned segmentation, predicting masks directly from short phrases. Because these models decode accurate masks from lightweight prompts, they serve as a common mask decoder in the MLLM-based grounding methods discussed below.

Visual Grounding with MLLMs. Recent MLLMs have enabled zero-shot visual grounding by localizing image regions from natural-language queries. Special-token methods such as LISA [21], GLaMM [39], Sa2VA [56], PixelLM [42], and UniPixel [29] embed segmentation tokens into the LLM output to condition a jointly trained mask decoder, the dominant MLLM-grounding paradigm. Refer-

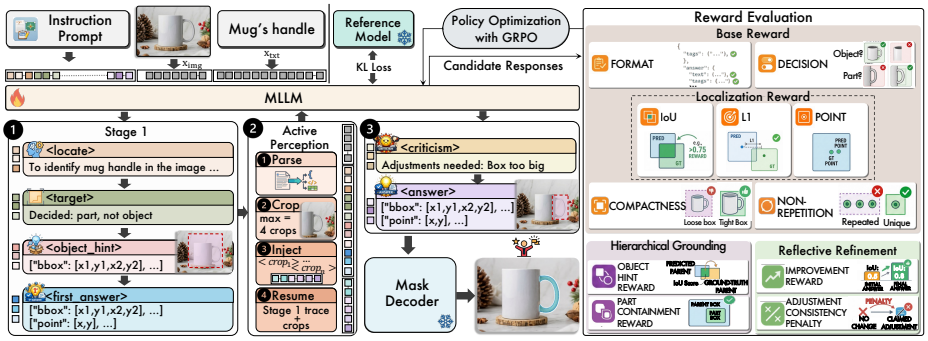

Fig. 2: Overview of the action steps and reward evaluation for OP-HRG.

ring frameworks [54,57] and open-set detectors such as Grounding DINO [28] and Grounded SAM [41] further advance visual grounding. More recently, reinforcement learning has emerged as an effective alignment strategy for grounding MLLMs. GRPO [43] removes the critic network to improve training efficiency and has been widely adopted in visual reasoning [9,33,37,46,53] and visual grounding [3,4,15,45,55,61,62], typically with IoU-oriented rewards. Most closely related to us are the decoupled MLLM-to-SAM pipelines Seg-Zero [30] and VisionReasoner [32], which pair GRPO-based optimization with a frozen mask decoder for reasoning-aware segmentation – the same paradigm we adopt. However, both treat all queries uniformly and optimize a single-stage localization reward, without modeling the object-part hierarchy or providing part-specific reward signals. Our framework extends this line of work by introducing object-part hierarchical reasoning and dedicated part-aware rewards into the GRPO alignment process.

Chain-of-Thought and Structured Reasoning for Visual Tasks. Multimodal chain-of-thought methods [13,38,59,60] improve reasoning through intermediate rationales, and spatially grounded approaches [6,25,36,44] align reasoning steps with image regions. Reflection-oriented paradigms [17,35] add self-critique to improve model's behavior and final answer quality. Our OP-HRG strategy builds on these directions: it enforces spatially grounded hierarchical reasoning through object-first localization and combines it with a self-reflective check for part-level grounding refinement and verification.

3 Method

In this section, we describe our model-agnostic reinforcement learning framework for grounding objects and object parts using the reasoning capabilities of multimodal LLMs (MLLM). Our approach combines a structured object-part prompting paradigm, which we term Object-Part Hierarchical Reflective Grounding (OP-HRG), with a part-aware reinforcement learning objective based on Group Relative Policy Optimization (GRPO). Figure 2 gives an overview of the action steps and reward evaluation.

Prompt Instructions

1. Decide whether the query is an object or part
2. If it is a part, locate the parent object that contains it
3. Predict tight boxes and a point enforcing point-inside and part-of constraints
4. Reflect on what the boxes enclose. State whether adjustment is needed
5. Refine the coordinates into final boxes and points for mask decoding

Output Structure

```
<locate> reasoning about what to locate </locate>
<target> object | part </target>
<object_hint> [{"bbox": [x1, y1, x2, y2]} </object_hint>
<first_answer> [{"bbox": [x1, y1, x2, y2], "point": [x, y]}] </first_answer>
<criticism> reflection on answer. ADJUSTMENT: YES | NO </criticism>
<answer> [{"bbox": [x1, y1, x2, y2], "point": [x, y]}] </answer>
```

Fig. 3: High-level structure of OP-HRG prompt and corresponding output format.

3.1 Problem Formulation

We consider the task of **visual grounding of objects and object parts**. Our system receives an RGB image $I \in \mathbb{R}^{H\times W\times 3}$ and a natural language query q. The query may denote a whole object category (e.g., "cat") or a semantic part of an object (e.g., "cat's tail"). During inference, the model must interpret the image and the query and return spatial localizations for the visual entity described by q, even when the object or part name has not been observed during training. Specifically, our end goal is to predict a binary segmentation mask that delineates q.

3.2 Decoupled Reasoning and Segmentation Architecture

Recent multimodal LLMs, like Qwen-VL [1,50], demonstrate strong reasoning and coordinate generation abilities, while large vision models such as the Segment Anything Model (SAM) [5,19,40] predict accurate, dense segmentation masks from geometric prompts. This complementarity motivates separating semantic reasoning from pixel segmentation, a decoupled design also adopted by recent grounding MLLMs [21,29,30,32,39,42,56]. We adopt this design.

In our pipeline, the MLLM reads (I, q) and generates bounding boxes and representative points $\mathcal{O}$. These localization primitives are then passed, together with the image, to a mask decoder, which outputs a segmentation mask $S \in \{0,1\}^{H\times W}$. Concretely, for each input (I, q), the MLLM produces a **set of localization primitives**

$$\mathcal{O} = \{(b_i, p_i)\}_{i=1}^{K}, \tag{1}$$

where K is the number of predicted regions matching the query. Each $b_i = (x_1^i, y_1^i, x_2^i, y_2^i)$ represents a tight 2D bounding box in image coordinates with $0 \le x_1^i < x_2^i \le W$ and $0 \le y_1^i < y_2^i \le H$. The primitive $p_i = (x^i, y^i)$ is a representative point required to lie inside the queried entity.

3.3 Object-Part Hierarchical Reflective Grounding

To address the challenges of grounding object parts, we introduce Object-Part Hierarchical Reflective Grounding (OP-HRG), a structured prompting paradigm that explicitly encodes the object-part hierarchy and incorporates self-reflective refinement. Figure 3 illustrates the high-level structure of the OP-HRG prompt and the corresponding expected output format. The full prompt and example outputs are in the Appendix C.

First, the model reasons about the query and determines whether it refers to a whole object or a part. This early decision governs the subsequent steps and enforces an explicit distinction between object-level and part-level grounding. *Second*, when the query refers to a part, the model localizes the parent object before the part itself, enforcing a coarse-to-fine process in which part localization is conditioned on object-level context, mirroring how humans search for object parts. *Third*, the model produces an initial localization as bounding boxes and representative points; for parts, these must respect the structural constraint that part boxes lie within the predicted object boxes. *Finally*, OP-HRG incorporates a self-reflective verification step in which the model critiques its own initial prediction, assesses whether the localization is tight and accurate, and decides whether adjustment is necessary, then either retains the initial prediction or produces a refined one. During training, this reflect-and-revise step provides a complementary corrective check on the initial localization.

As shown in Figure 3, the prompt enforces this reasoning as a fixed sequence of tagged outputs that expose intermediate reasoning states and localization decisions (reasoning trace, target decision, object hint, initial answer, critique, and refined answer). This structured format serves two purposes. First, it induces the desired hierarchical reasoning behavior during inference. Second, it enables fine-grained, verifiable reward signals for our reinforcement learning framework.

3.4 Active Visual Perception

In the OP-HRG formulation above, the self-reflective step reasons only over the textual bounding-box and point coordinates of the initial prediction. The reflective step therefore needs to resolve these coordinates against the whole image, rather than being able to inspect the enclosed image region of the model's first answer directly. We therefore extend OP-HRG with an *active visual perception* (AP) step that grounds the reflective step in fresh visual evidence. After the initial localization (`<first_answer>`), generation pauses and the predicted boxes are used to crop the corresponding regions from the input image. Each crop is re-encoded by the model's pretrained vision encoder and injected back as interleaved visual tokens, so the model performs its critique and reaches the final answer (`<criticism>` and `<answer>`) on this fresh visual evidence. This loop is applied both during training and at inference, so the policy learns to exploit the re-encoded crops when deciding if it needs to adjust its prediction.

3.5 Reinforcement Learning with Part-Aware Rewards

Although pretrained MLLMs possess substantial visual and semantic knowledge, they exhibit a bias toward coarse object-level grounding, in part due to the scarcity of part-centric supervision in large-scale MLLM training [14,27,49]. Reinforcement learning offers a potential mechanism for correcting this bias – it lets us reward tight, compact localizations and enforce structural constraints such as part–object containment and consistency between intermediate reasoning and final outputs. While prior work shows GRPO [43] is well-suited for aligning instruction-tuned MLLMs for visual grounding [30,32], these models remain weak at grounding object parts, motivating a dedicated framework for part-centric localization and object-part reasoning. We therefore train our multimodal LLM with GRPO under a composite, part-aware reward, using its reasoning to produce more accurate and compact object- and part-level bounding boxes and representative points. The reward is built from modular, verifiable components in three groups: **base rewards**, **hierarchical grounding rewards**, and **reflective refinement rewards**, described below. All components are normalized to $[0, 1]$ (or $[-1, 0]$ for penalties) and combined into a total reward normalized by the maximum achievable score (full normalization in Appendix D). We detail the GRPO objective in Section 3.6 and the detailed reward implementation in Appendix D.

Base Rewards. These rewards apply to every query regardless of whether it targets an object or a part.

The *format compliance reward* scores adherence to the OP-HRG tag sequence and the validity of JSON content within each tag, assigning individual credit for well-formed `<object_hint>`, `<first_answer>`, `<answer>`, and `<criticism>` blocks, normalized by the maximum achievable format score. The *target decision reward* is a binary signal that rewards correct classification of the query as *object* or *part*, which determines whether the model activates the object-part hierarchical reasoning path.

The *localization rewards* assess the geometric quality of predicted bounding boxes and representative points, computed identically for the initial and final predictions (`<first_answer>` and `<answer>` respectively) to provide signal at both stages. When multiple instances are present, we construct a cost matrix $C = \mathbf{1} - \mathrm{IoU}(\hat{\mathcal{B}}, \mathcal{B}^*)$ between the M predicted and N ground-truth boxes and solve the assignment via the Hungarian algorithm [20], scoring matched pairs and averaging over $\max(M, N)$ to penalize both missing and spurious detections. We reward three localization components:

- *IoU Reward*: the mean IoU over matched pairs.
- *L1 Distance Reward*: for each matched pair (i, j), the mean absolute coordinate difference $\ell_{1,ij}$ under an adaptive threshold $\tau_j^{(\ell_1)} = \alpha_{\ell_1} \cdot d_j$, where d_j is the diagonal of the j-th ground-truth box, α_{ℓ_1} a scaling factor, and the threshold is clamped to $\tau_{\max}^{(\ell_1)}$, giving the per-pair reward $\max(0,\, 1 - \ell_{1,ij}/\tau_j^{(\ell_1)})$.

– *Point Accuracy Reward*: a predicted point receives credit only if it lies inside its own predicted box *and* its distance to the matched ground-truth point falls within $\tau_j^{(p)} = \alpha_p \cdot d_j$, similarly clamped to $\tau_{\max}^{(p)}$.

Scaling the tolerance by the ground-truth box diagonal judges accuracy relative to region size, which matters for small part boxes that a fixed threshold would treat too permissively. We use a tighter scaling factor for box coordinates (α_{ℓ_1}) than for representative points (α_p), since a point may lie anywhere within the target region whereas box boundaries must align tightly with ground truth.

The *compactness reward* encourages tight spatial coverage by scoring each Hungarian-matched pair (i, j) with two complementary terms over the predicted box $\hat{b}_i$ and its matched ground-truth box b_j^*: a *precision* term $\rho_{ij} = |\hat{b}_i \cap b_j^*| \,/\, |\hat{b}_i|$, the fraction of the predicted box overlapping ground truth, and an *over-prediction penalty* $\omega_{ij} = -\min\big(1, \max(0, |\hat{b}_i|/|b_j^*| - 1)\big)$, saturating at -1 once the predicted box reaches twice the ground-truth area. The reward is the mean of $\rho_{ij} + \omega_{ij}$ across matched pairs, favoring tight boxes; this benefits object parts in particular, where even modest over-prediction can encompass neighboring parts or the parent object.

The *non-repetition reward* penalizes degenerate chain-of-thought outputs by checking for repeated sentences in the reasoning trace. The reward drops to zero if two or more exact sentence duplicates are detected.

Hierarchical Grounding Rewards. These components are active only for part queries and directly correspond to the object-part hierarchy introduced by OP-HRG.

The *object hint reward* evaluates the predicted parent-object boxes, which are matched to ground-truth object boxes via Hungarian matching on IoU, and the reward is the normalized average IoU over matched pairs.

The *part containment reward* enforces structural constraints on the final part predictions. For each predicted part box $\hat{b}^{\text{part}}$, we verify three conditions against the predicted and ground-truth object boxes: (i) $\hat{b}^{\text{part}}$ is spatially contained within at least one object box, (ii) it is not identical to that object box, and (iii) its area is strictly smaller, $|\hat{b}^{\text{part}}| < |\hat{b}^{\text{obj}}|$. Conditions (ii) and (iii) prevent the policy from trivially replicating the object box as the part prediction to satisfy containment without performing real part localization. The reward is the fraction of predicted parts satisfying all three conditions.

Reflective Refinement Rewards. These components incentivize the self-reflective loop in OP-HRG.

The *improvement reward* encourages meaningful refinement from the initial to the final prediction. For IoU, it is defined as

$$R_{\text{improv}}^{\text{IoU}} = \max\Big(0,\ R_{\text{IoU}}^{\text{final}} - \max\big(R_{\text{IoU}}^{\text{initial}},\ \lambda_{\text{IoU}} \cdot \text{IoU}_{\text{baseline}}\big)\Big), \tag{2}$$

where $\text{IoU}_{\text{baseline}}$ is the precomputed IoU of a strong external baseline, so the model earns reward only when it surpasses the stronger of its own first answer or

that baseline. This is critical for preventing reward exploitation: by rewarding the initial prediction independently and competing against the baseline, the policy has no incentive to produce deliberately poor first answers to inflate the improvement margin. WWe validate this design in Appendix D (Fig. 6), where removing the baseline reference causes the initial IoU to collapse toward zero. We similarly compute improvement rewards for the L1 and point rewards, and apply an explicit penalty $R_{\text{IoU}}^{\text{final}} - R_{\text{IoU}}^{\text{initial}}$ when the final IoU degrades relative to the initial prediction, discouraging harmful refinement.

The *adjustment consistency penalty* penalizes contradictions between the model's adjustment intent and its actual behavior: declaring `ADJUSTMENT: YES` with identical initial and final coordinates, or `ADJUSTMENT: NO` with changed coordinates, incurs a penalty of -1, while consistent behavior incurs none. This enforces honest self-reflection and prevents gaming of the critique mechanism.

3.6 Overall Training Objective

The pretrained multimodal LLM is adapted using GRPO with the composite reward above. Let π_{θ_0} denote the frozen reference policy and π_θ the updated policy. For each image–query pair (I, q), GRPO samples a group $\mathcal{G} = \{y_j\}_{j=1}^{|\mathcal{G}|}$ of structured outputs, scores each with the normalized reward $R(y_j) = \sum_k \lambda_k R_k(y_j)$, and computes advantages A_j as deviations from the group mean, eliminating the need for an explicit value function. The policy is optimized via a clipped surrogate objective with a KL-divergence regularizer:

$$L_{\text{GRPO}} = -\frac{1}{|\mathcal{G}|}\sum_{j=1}^{|\mathcal{G}|} \min(\rho_j A_j,\ \text{clip}(\rho_j, 1-\epsilon, 1+\epsilon)\, A_j) + \beta\, D_{\text{KL}}(\pi_\theta \| \pi_{\theta_0}), \quad (3)$$

where $\rho_j = \pi_\theta(y_j|I,q) \,/\, \pi_{\theta_0}(y_j|I,q)$ is the importance ratio, ϵ the clipping threshold, and β the KL strength. The clipped surrogate restricts update magnitudes for training stability, while the KL term keeps the policy from drifting from its pretrained behavior, so that improvements in part-centric localization preserve object-level performance rather than degrading the model's broader competence.

4 Experiments

4.1 Models

Reasoning Model. We use Qwen3-VL-Instruct as our reasoning component. We adopt the Instruct variant for two reasons. First, OP-HRG relies on explicit, step-by-step adherence to a structured prompt rather than free-form latent reasoning, and the instruction-tuned variant is well suited to following this fixed sequence of tagged steps. Second, on Qwen3-VL's own grounding evaluations [1], the Instruct variants attain stronger object detection performance than the Think variants at both the 4B and 8B scales, indicating they are the stronger starting point for spatial localization. We adopt the compact 4B model

as our main configuration to demonstrate that structured reasoning and targeted rewards, rather than scale alone, drive the gains, and additionally report an 8B configuration (Table 3) to show that the framework scales with backbone size. We do not perform a supervised fine-tuning cold-start, since Qwen3-VL is already instruction-tuned to emit bounding boxes and representative points. We apply the OP-HRG prompt structure and train from the pretrained weights with GRPO under our verifiable part-aware rewards, which directly reinforce and sharpen this existing capability rather than learning it from scratch.

Mask Decoder. For pixel-level segmentation, we employ the pretrained SAM2-Large model as our main frozen mask decoder. We use SAM2 for two reasons. First, prior baselines like [32] use the same decoder, and this ensures a fair comparison. Second, prior literature demonstrates that SAM-style models already produce high-quality masks from spatial prompts such as boxes and points. We therefore keep the SAM2 parameters frozen, isolating all improvements to the reasoning-guided localization abilities of the MLLM.

4.2 Datasets and Preprocessing

General Object Data. We use the 7k multi-object dataset of [32], which provides bounding boxes and representative points paired with free-form text queries. The purpose of this dataset is to train the model on visual grounding on a general, multi-object dataset that is not specific to object parts.

For part-level supervision, we employ the InstructPart train split [49] containing 1200 images. The dataset provides curated, part-focused images with object–part structure. The dataset provides part masks but no bounding boxes or representative points. Since we need bounding boxes and points as localization primitives to train our reasoning LLM, we derive these from the masks: each connected component is converted into a bounding box, and the deepest interior pixel is selected via the Euclidean distance transform as the representative point.

Baseline Signals. As described in our reward design, the improvement reward credits gains only when the final prediction surpasses the stronger of the model's own initial answer and an external baseline. For this external reference, we use SAM3 [5], which generates segmentation masks from short textual phrases. We pre-compute baseline IoU scores by passing all training queries to SAM3, measuring predicted box overlap with ground truth, and storing the results as an additional field in the training data.

4.3 Results

We evaluate on three benchmarks: InstructPart [49] (600 queries, 600 images), PascalPart [8] (specifically the PascalPart-116 split [51], which contains 1K object + 10K part queries across 851 images), and PartImageNet [14] (14K part queries across 4589 images), reporting gIoU (the mean IoU across all test queries). InstructPart is evaluated on the test split of the dataset used for part-level training, whereas PascalPart and PartImageNet are fully cross-dataset zero-shot – our model has seen neither images nor annotations from either benchmark.

Table 1: Results on part-grounding benchmarks (gIoU). Our 4B model surpasses larger 7B grounding LLMs and SAM3 on cross-dataset zero-shot part grounding (PascalPart, PartImageNet), while leading on objects. †InstructPart is evaluated on its test split; our RL training uses the InstructPart train split.

	Parts			Objects
Model	**InstructPart†**	**PascalPart**	**PartImgNet**	**Pascal-Obj**
Token-based grounding MLLMs				
LISA-7B [21]	43.26	13.82	29.91	83.50
Sa2VA-4B [56]	50.15	14.67	38.13	77.02
PixelLM-7B [42]	44.41	16.57	35.25	81.71
UniPixel-3B [29]	59.68	30.64	46.18	85.54
Decoupled MLLM-to-SAM				
Molmo + SAM3 (point) [5, 12]	51.02	8.87	17.30	57.57
Grounding DINO + SAM3 (box) [5, 28]	33.74	13.13	31.38	79.25
VisionReasoner-7B [32]	59.38	27.44	45.59	86.68
Text-promptable segmentation				
ClipSeg [34]	31.85	12.51	33.45	70.51
SAM3 (text) [5]	71.06	33.05	53.89	85.32
Qwen3-VL-Instruct-4B + SAM2				
zero-shot OP-HRG prompt	31.45	12.83	21.95	36.91
+ RL using InstructPart **(Ours)**	**75.56**	**38.59**	**56.87**	**87.50**
Δ vs. best baseline	*+4.50*	*+5.54*	*+2.98*	*+0.82*

Both feature diverse, naturally occurring scenes at far larger scale than the InstructPart training set, making them a challenging test of generalization. Therefore, the gains there show that training on just 1200 part-focused images with our method improves part grounding.

Table 1 compares our pipeline against baselines that span different paradigms for visual grounding. We first compare against a set of special-token grounding LLMs (LISA-7B [21], Sa2VA-4B [56], PixelLM-7B [42], and UniPixel-3B [29]), which embed segmentation tokens into the LLM output to condition a jointly trained mask decoder. All underperform our method on parts, with the strongest (UniPixel-3B) trailing by 15.88, 7.95, and 10.69 gIoU on InstructPart, PascalPart, and PartImageNet parts.

Next, we compare against decoupled MLLM-to-SAM pipelines, which, like ours, prompt a frozen mask decoder with MLLM predictions. Since this design relies on the MLLM producing bounding boxes and representative points as prompts, we first consider two baselines that isolate each modality. Molmo [12], a strong pointing model, is paired with SAM3 to generate masks from single-point predictions; Grounding DINO [28], a state-of-the-art open-set detector, is paired with SAM3 to generate masks from predicted bounding boxes. Both perform substantially worse than our method on parts, confirming that neither strong pointing nor strong detection alone suffices for part grounding. VisionReasoner [32] pairs a Qwen2.5-VL-7B model with SAM2 and uses RL alignment for grounding; with a smaller 4B model, our method improves on it by +16.18,

Table 2: Frozen mask-decoder swap with our trained MLLM fixed (gIoU).

	Parts			Obj
Decoder	**InstP†**	**PascP**	**PartIN**	**PascO**
SAM2 **(Ours)**	75.56	38.59	56.87	87.50
SAM3	75.77	39.05	56.23	87.73

Table 3: Active visual perception.

	Parts			Obj
Model	**InstP†**	**PascP**	**PartIN**	**PascO**
Ours (4B)	75.56	38.59	56.87	87.50
+ AP (4B)	76.34	39.04	55.98	88.09
AP (8B)	**77.00**	**40.28**	**58.19**	**89.62**

Table 4: Referring grounding (Acc@0.5) on RefCOCO/+/g.

Model	**RefC testA**	**RefC+ testA**	**RefCg test**	**Avg**
Specialist grounding models				
GLIP-T (ZS) [24]	54.3	52.8	66.9	58.0
MDETR [18]	89.6	84.1	80.9	84.9
GDINO-T [28]	91.9	87.4	84.9	88.1
Multimodal LLMs				
Shikra-7B [7]	90.6	87.4	82.2	86.7
InternVL2-8B [10]	91.1	87.9	82.7	87.2
Qwen2.5-VL-7B [2]	91.7	88.2	85.7	88.5
Qwen3-VL-Instruct-4B [1]	92.7	88.7	87.8	89.7
VisionReasoner-7B [32]	90.6	87.9	87.5	88.7
Ours-4B	89.3	83.3	86.9	86.5

+11.15, and +11.28 gIoU on InstructPart, PascalPart, and PartImageNet parts, demonstrating the effectiveness of structured hierarchical reasoning and part-aware rewards. SAM3 [5], the latest model in the Segment Anything family, is trained to segment visual concepts from textual phrases and is the strongest baseline. Nevertheless, our method outperforms SAM3 both on in-domain (InstructPart) and zero-shot parts evaluation. Finally, we compare against our base model: Qwen3-VL-Instruct-4B paired with SAM2 under the OP-HRG prompt, without any RL training. This vanilla configuration scores far below our trained model across every benchmark, on the identical architecture and prompt, isolating the impact of our reinforcement learning framework with part-aware rewards. Beyond parts, our method also achieves the strongest object-level performance (87.50 gIoU on Pascal-Obj) compared to all baselines in Table 1. This shows that part-centric training improves object-level grounding on these benchmarks.

Our decoupled design treats the mask decoder as a frozen, interchangeable module, raising the question of whether our gains depend on the specific decoder rather than the reasoning-guided localization. To test this, we hold our trained Qwen3-VL-Instruct-4B fixed and replace only the decoder, prompting SAM3 [5] with the same predicted boxes and points in place of SAM2. As Table 2 shows, the swap does not significantly shift the gIoU, confirming that the quality of our MLLM-generated prompts, not the decoder, drives performance; we retain SAM2 by default to match prior decoupled pipelines [32].

Table 3 reports our model with the active visual perception step introduced in Section 3.4. At 4B, conditioning the final answer on re-encoded crops improves performance on three of the four splits. The 8B configuration improves over its 4B counterpart, demonstrating that our framework scales to larger models.

We next verify that part-centric training does not compromise general object-level grounding. Table 4 reports referring grounding accuracy (Acc@0.5) on RefCOCO/+/g. Our 4B model attains 86.5 average accuracy, competitive with the 7B VisionReasoner [32] (88.7) and within roughly three points of its own Qwen3-VL-Instruct-4B base [1] (89.7). This trade-off is modest relative to the large part-grounding gains in Table 1, indicating that reinforcing fine-grained part localization largely preserves whole-object referring. Beyond explicit part

Table 5: Reasoning segmentation on ReasonSeg (gIoU). Our 4B model surpasses baselines, including the 7B VisionReasoner, showing that part-centric training also benefits reasoning-driven object segmentation.

Model	ReasonSeg
ReLA [26]	21.3
LISA-7B [21]	36.8
Qwen2.5-VL-7B [2]	52.1
SegZero-7B [31]	57.5
VisionReasoner-7B [32]	63.6
GenSeg-R1-4B [16]	68.4
Ours-4B	**69.6**

Table 6: Inference cost comparison

Model	Time (s)	Avg. tokens
VisionReasoner-7B [32]	1.86	272
ZS OP-HRG Prompt	3.31	607
Ours	2.87	564

Table 7: Ablation on the InstructPart test set.

Model	gIoU
(a) Is part-level training data sufficient?	
VisionReasoner (Qwen2.5-VL-7B + SAM2)	59.38
+ GRPO on InstructPart	62.33
(b) Are both OP-HRG components necessary?	
Ours w/o hierarchy & refinement	70.32
Ours w/o hierarchical grounding	73.77
Ours w/o reflective refinement	73.98
Ours	**75.56**

and object grounding, we assess whether our structured reasoning transfers to tasks requiring implicit multi-step inference. Table 5 reports gIoU on ReasonSeg, a reasoning-driven segmentation benchmark. Our 4B model reaches 69.6 gIoU, surpassing the baselines and indicating that the reasoning induced by OP-HRG benefits segmentation broadly, not only part localization.

Finally, we examine inference cost. Table 6 reports MLLM wall-clock time and average generated tokens on an NVIDIA L40 at batch size 32. Our method runs faster than the same base model under the OP-HRG prompt without RL, because GRPO training discourages unconstrained reasoning and yields more compact outputs (564 vs. 607 tokens). Relative to VisionReasoner, our hierarchical, self-reflective procedure adds about one second of overhead, a modest cost given the substantial part-grounding gains in Table 1. We provide qualitative examples illustrating our model's predictions across all benchmarks in Appendix H.

4.4 Ablation Studies

We conduct ablation experiments on the InstructPart test set [49], as shown in Table 7. We choose InstructPart for this analysis because it is drawn from the same domain as our part-level training data, allowing us to directly isolate the contribution of each architectural and reward component without conflating the effects of cross-dataset distribution shift.

Is part-level training data sufficient? VisionReasoner [32] achieves 59.38 gIoU on InstructPart using its original training and reward design. To test whether simply exposing a strong baseline to part data closes the gap, we retrain VisionReasoner on the InstructPart train split using GRPO with its original reward function, without OP-HRG prompting or our part-aware rewards. This yields a gain to 62.33 gIoU vs our 75.56, indicating that part data adaptation alone is insufficient.

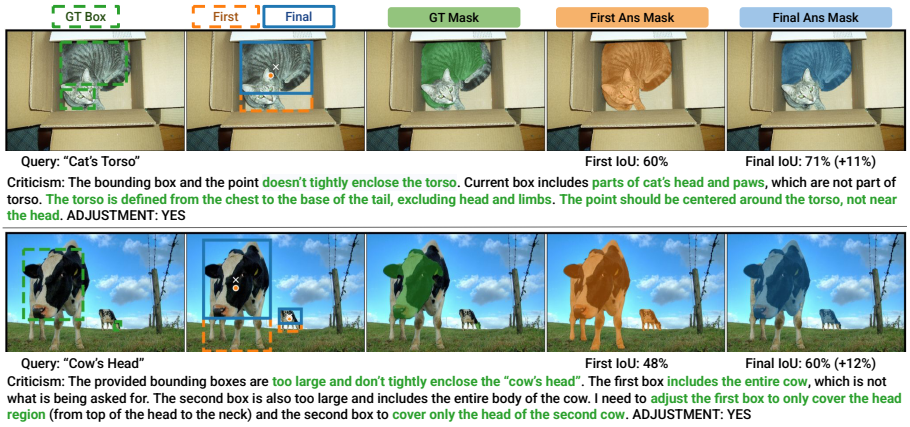

Fig. 4: Examples of the reflective step from an intermediate model checkpoint.

Are both object-part hierarchy and the reflective refinement components necessary? We ablate the two core mechanisms of OP-HRG independently. Training with only standard localization rewards (IoU, L1, point accuracy) and no OP-HRG structure reaches 70.32 gIoU, which confirms that GRPO alignment helps but trails our full method. Removing hierarchical grounding – the object-first localization stage, the part containment constraint, and the object hint reward – while keeping reflective refinement reduces performance to 73.77 gIoU; conversely, removing reflective refinement – the self-critique loop, the two-answer structure, and the improvement reward – while keeping hierarchical grounding gives 73.98 gIoU. Both underperform the full pipeline, showing that hierarchical grounding and reflective refinement are complementary and individually necessary.

4.5 Analyzing the Reflective Step

The ablation above shows that reflective refinement contributes a modest gain (+1.58 gIoU on InstructPart). We find that this gain acts primarily as a train-time regularizer rather than a test-time process: a model trained with the full objective but evaluated with a plain single-answer prompt retains nearly all of it (75.40 vs. 75.56), so the benefit is internalized into the weights rather than supplied by the prompt at test time.

Refinement is active early and progressively internalized. We checkpoint the model every 50 steps and trace its refinement behavior on InstructPart (Figure 5). Early in training the model revises often, with a net positive IoU change (top), confirming the reflective step does real corrective work. Figure 4 shows two such cases. The model tightens an over-broad box that had included the cat's head and paws down to just the torso (IoU 0.60 vs. 0.71), and narrows a whole-cow box to the queried cow's head alone (0.48 vs. 0.60). The bottom panel shows the initial- and final-answer gIoU: early on, the final sits above

the initial, but the two converge as training proceeds. Once the model achieves its best possible first answer, the reflective step increasingly acts as verification rather than correction, as further mentioned in Sec. 5.

5 Limitations

We observe that as training progresses over many steps, the reasoning model converges toward producing higher quality predictions at the initial stage, which in turn leads to the reflective mechanism consistently declining to adjust. This is a natural consequence of optimization: as the model's first-answer accuracy improves towards the best that it can achieve, there is less need for refinement, and our reward discourages unnecessary changes. However, this convergence effectively reduces the reflective refinement loop to a verification step rather than an active correction mechanism. Future work could explore the utility of the reflective stage even as the base localization quality improves. Our part containment reward is satisfied whenever a predicted part box falls within *any* parent object box. A part localized within the wrong object of the same category can still be scored as valid. As most of our InstructPart training data does not contain such multi-instance scenes, this has limited effect, and we leave this to future work.

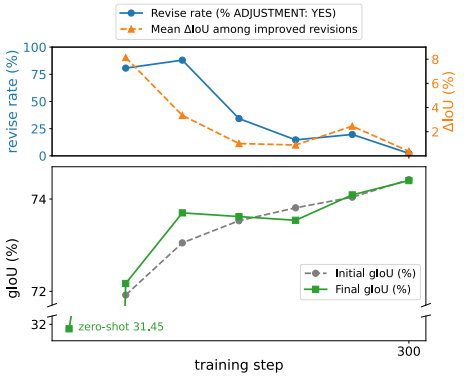

Fig. 5: Refinement behavior over training. Top: revision rate and mean IoU gain. Bottom: initial vs. final gIoU.

6 Conclusion

We presented Object-Part Hierarchical Reflective Grounding (OP-HRG), which addresses a basic weakness of current multimodal LLMs that ground whole objects well but fail on fine-grained parts. OP-HRG works coarse-to-fine. It first localizes the parent object, then the part within it, and refines and verifies the result with a self-check. We train it with a part-aware GRPO framework whose stage-wise rewards supervise each step. With a compact 4B-parameter model, our method outperforms larger baselines on PascalPart and PartImageNet (cross-dataset zero-shot) and InstructPart (in-domain) with only a modest trade-off in general object referring, and it transfers to reasoning segmentation. Ablations show that hierarchical grounding and reflective refinement contribute comparable, complementary gains, with the reflective step's benefit largely internalized during training. These results suggest that the capacity for fine-grained spatial reasoning already exists within pretrained MLLMs and can be drawn out through structured prompting and targeted reward design.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report. arXiv preprint arXiv:2502.13923 (2025). https://doi.org/10.48550/arXiv.2502.13923, https://arxiv.org/abs/2502.13923
3. Bai, S., Li, M., Liu, Y., Tang, J., Zhang, H., Sun, L., Chu, X., Tang, Y.: UniVG-R1: Reasoning guided universal visual grounding with reinforcement learning. arXiv preprint arXiv:2505.14231 (2025)
4. Cao, M., Zhao, H., Zhang, C., Chang, X., Reid, I., Liang, X.: Ground-R1: Incentivizing grounded visual reasoning via reinforcement learning. arXiv preprint arXiv:2505.20272 (2025)
5. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: SAM 3: Segment anything with concepts. In: International Conference on Learning Representations (2026)
6. Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F.: SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14455–14465 (June 2024)
7. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)
8. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1971–1978 (2014)
9. Chen, X., Li, W., Liu, C., Xie, C., Hu, X., Ma, C., Zhu, F., Zhao, R.: On the suitability of reinforcement fine-tuning to visual tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 3382–3386 (2025)
10. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24185–24198 (June 2024)
11. Choi, J., Lee, S., Lee, M., Lee, S., Shim, H.: Fine-grained image-text correspondence with cost aggregation for open-vocabulary part segmentation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 9782–9793 (2025)
12. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., et al.: Molmo and PixMo: Open weights and open data for state-of-the-art vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 91–104 (2025)
13. Fu, X., Liu, M., Yang, Z., Corring, J., Lu, Y., Yang, J., Roth, D., Florencio, D., Zhang, C.: ReFocus: Visual editing as a chain of thought for structured image understanding. In: International Conference on Machine Learning. pp. 17783–17805 (2025)

14. He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.N., Liu, S., Yang, C., Yu, Q., Yuille, A.: PartImageNet: A large, high-quality dataset of parts. In: European Conference on Computer Vision. pp. 128–145. Springer (2022)
15. He, Y., Chen, W., Jian, Z., Guo, T., Zhou, W., Li, M.: DR^{2}Seg: Decomposed two-stage rollouts for efficient reasoning segmentation in multimodal large language models. arXiv preprint arXiv:2601.09981 (2026)
16. Hegde, S., Chacko, J.S., Banerjee, D., Mahesh, U.: Genseg-r1: Rl-driven vision-language grounding for fine-grained referring segmentation. arXiv preprint arXiv:2602.09701 (2026)
17. Jian, P., Wu, J., Sun, W., Wang, C., Ren, S., Zhang, J.: Look again, think slowly: Enhancing visual reflection in vision-language models. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 9251–9270 (2025). https://doi.org/10.18653/v1/2025.emnlp-main.470
18. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1780–1790 (2021)
19. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
20. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
21. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: LISA: Reasoning segmentation via large language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9579–9589 (2024)
22. Li, F., Zhang, H., Sun, P., Zou, X., Liu, S., Li, C., Yang, J., Zhang, L., Gao, J.: Segment and recognize anything at any granularity. In: European Conference on Computer Vision. pp. 467–484. Springer (2024)
23. Li, J., Wu, J., Zhao, W., Bai, S., Bai, X.: PartGLEE: A foundation model for recognizing and parsing any objects. In: European Conference on Computer Vision. pp. 475–494. Springer (2024)
24. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10965–10975 (2022)
25. Li, Z., Luo, R., Zhang, J., Qiu, M., Huang, X., Wei, Z.: VoCoT: Unleashing visually grounded multi-step reasoning in large multi-modal models. In: Chiruzzo, L., Ritter, A., Wang, L. (eds.) Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 3769–3798. Association for Computational Linguistics, Albuquerque, New Mexico (Apr 2025), https://aclanthology.org/2025.naacl-long.192/
26. Liu, C., Ding, H., Jiang, X.: Gres: Generalized referring expression segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 23592–23601 (2023)
27. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21736–21746 (2023)
28. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding DINO: Marrying DINO with grounded pre-training for

open-set object detection. In: European conference on computer vision. pp. 38–55. Springer (2024)
29. Liu, Y., Ma, Z., Pu, J., Qi, Z., Wu, Y., Shan, Y., Wen, C.C.: Unipixel: Unified object referring and segmentation for pixel-level visual reasoning. Advances in Neural Information Processing Systems **38**, 126078–126108 (2025), https://papers.nips.cc/paper_files/paper/2025/file/b783c44ba9adbc30344473dc633b4869-Paper-Conference.pdf
30. Liu, Y., Peng, B., Zhong, Z., Yue, Z., Lu, F., Yu, B., Jia, J.: Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. arXiv preprint arXiv:2503.06520 (2025)
31. Liu, Y., Peng, B., Zhong, Z., Yue, Z., Lu, F., Yu, B., Jia, J.: Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. arXiv preprint arXiv:2503.06520 (2025)
32. Liu, Y., Qu, T., Zhong, Z., Peng, B., Liu, S., Yu, B., Jia, J.: VisionReasoner: Unified reasoning-integrated visual perception via reinforcement learning. In: International Conference on Learning Representations (2026)
33. Liu, Z., Sun, Z., Zang, Y., Dong, X., Cao, Y., Duan, H., Lin, D., Wang, J.: Visual-RFT: Visual reinforcement fine-tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2034–2044 (2025)
34. Lüddecke, T., Ecker, A.: Image segmentation using text and image prompts. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7086–7096 (2022)
35. Ma, X., Ding, Z., Luo, Z., Chen, C., Guo, Z., Wong, D.F., Feng, X., Sun, M.: DeepPerception: Advancing R1-like cognitive visual perception in MLLMs for knowledge-intensive visual grounding (2025), https://arxiv.org/abs/2503.12797
36. Man, Y., Huang, D.A., Liu, G., Sheng, S., Liu, S., Gui, L.Y., Kautz, J., Wang, Y.X., Yu, Z.: Argus: Vision-centric reasoning with grounded chain-of-thought. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
37. Meng, F., Du, L., Liu, Z., Zhou, Z., Lu, Q., Fu, D., Han, T., Shi, B., Wang, W., He, J., et al.: MM-Eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. arXiv preprint arXiv:2503.07365 (2025)
38. Mitra, C., Huang, B., Darrell, T., Herzig, R.: Compositional chain-of-thought prompting for large multimodal models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14420–14431 (June 2024)
39. Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R.M., Xing, E., Yang, M.H., Khan, F.S.: GLaMM: Pixel grounding large multimodal model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13009–13018 (2024)
40. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: SAM 2: Segment anything in images and videos. In: International Conference on Learning Representations (2025)
41. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al.: Grounded SAM: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024)
42. Ren, Z., Huang, Z., Wei, Y., Zhao, Y., Fu, D., Feng, J., Jin, X.: Pixellm: Pixel reasoning with large multimodal model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26374–26383 (2024)

43. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
44. Sharma, K., Vats, V.: Think to ground: Improving spatial reasoning in LLMs for better visual grounding. In: Workshop on Reasoning and Planning for Large Language Models (2025), https://openreview.net/forum?id=T2IHuIib74
45. Shen, C., Wei, W., Qu, X., Cheng, Y.: Satori-R1: Incentivizing multimodal reasoning with spatial grounding and verifiable rewards. arXiv preprint arXiv:2505.19094 (2025)
46. Shen, H., Liu, P., Li, J., Fang, C., Ma, Y., Liao, J., Shen, Q., Zhang, Z., Zhao, K., Zhang, Q., et al.: VLM-R1: A stable and generalizable R1-style large vision-language model. arXiv preprint arXiv:2504.07615 (2025)
47. Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., Wu, C.: Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv: 2409.19256 (2024)
48. Sun, P., Chen, S., Zhu, C., Xiao, F., Luo, P., Xie, S., Yan, Z.: Going denser with open-vocabulary part segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15453–15465 (2023)
49. Wan, Z., Xie, Y., Zhang, C., Lin, Z., Wang, Z., Stepputtis, S., Ramanan, D., Sycara, K.P.: InstructPart: Task-oriented part segmentation with instruction reasoning. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 24202–24227 (2025)
50. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
51. Wei, M., Yue, X., Zhang, W., Kong, S., Liu, X., Pang, J.: OV-PARTS: Towards open-vocabulary part segmentation. Advances in Neural Information Processing Systems **36**, 70094–70114 (2023)
52. Yang, S., Qu, T., Lai, X., Tian, Z., Peng, B., Liu, S., Jia, J.: Lisa++: An improved baseline for reasoning segmentation with large language model. arXiv preprint arXiv:2312.17240 (2023)
53. Yang, Y., He, X., Pan, H., Jiang, X., Deng, Y., Yang, X., Lu, H., Yin, D., Rao, F., Zhu, M., et al.: R1-Onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2376–2385 (2025)
54. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. In: International Conference on Learning Representations (2024)
55. You, Z., Wu, Z.: Seg-R1: Segmentation can be surprisingly simple with reinforcement learning. arXiv preprint arXiv:2506.22624 (2025)
56. Yuan, H., Li, X., Zhang, T., Sun, Y., Huang, Z., Xu, S., Ji, S., Tong, Y., Qi, L., Feng, J., et al.: Sa2VA: Marrying SAM2 with LLaVA for dense grounded understanding of images and videos. arXiv preprint arXiv:2501.04001 (2025)
57. Yuan, Y., Li, W., Liu, J., Tang, D., Luo, X., Qin, C., Zhang, L., Zhu, J.: Osprey: Pixel understanding with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28202–28211 (2024)
58. Zhan, G., Li, C., Liu, Z., Lu, Y., Wu, Y., Han, S., Zhu, L.: Scaling test-time inference for visual grounding. arXiv preprint arXiv:2601.13633 (2026)

59. Zhang, R., Zhang, B., Li, Y., Zhang, H., Sun, Z., Gan, Z., Yang, Y., Pang, R., Yang, Y.: Improve vision language model chain-of-thought reasoning. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1631–1662. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.acl-long.82, https://aclanthology.org/2025.acl-long.82/
60. Zhang, Z., Zhang, A., Li, M., Zhao, H., Karypis, G., Smola, A.: Multimodal chain-of-thought reasoning in language models. Transactions on Machine Learning Research (2023)
61. Zhou, D., He, M., Fang, Z., Yao, X., Liu, Y., Knoll, A., Cao, H.: AffordanceGrasp-R1: Leveraging reasoning-based affordance segmentation with reinforcement learning for robotic grasping. arXiv preprint arXiv:2602.03547 (2026)
62. Zhu, L., Ouyang, B., Zhang, Y., Cheng, T., Hu, R., Shen, H., Ran, L., Chen, X., Yu, L., Liu, W., Wang, X.: LENS: Learning to segment anything with unified reinforced reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 13952–13960 (2026). https://doi.org/10.1609/aaai.v40i16.38405