LATENTREFINE-SEGZERO: LATENT-SPACE REFINEMENT FOR REFERRING SEGMENTATION VIA REFLECTIVE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Significant progress has been made in referring segmentation by combining multimodal large language models (MLLMs) with the Segment Anything Model (SAM). However, balancing in-domain accuracy and out-of-domain generalization remains challenging. SFT-based methods achieve strong in-domain segmentation performance, but updating pretrained MLLM parameters with pixel-level losses may limit generalization to out-of-domain referring expressions. RL-based methods that avoid such updates typically control SAM through discrete box and point prompts, limiting fine-grained mask refinement. To mitigate this trade-off, we propose LatentRefine-SegZero, a latent-space refinement framework for referring segmentation based on reflective reinforcement learning. Our method comprises three components: (1) **Proposal assessment**, which uses reflective reasoning to assess proposal reliability and select a refinement pathway; (2) **Semantic alignment**, which converts proposal representations into supplementary latent prompts for mask refinement based on reliable proposals; and (3) **Semantic relocalization**, which converts reflective representations into new latent prompts for target relocalization. GRPO optimizes proposal generation and reliability assessment; pixel-level supervision trains lightweight adaptation modules without updating pretrained MLLM backbone weights. Experiments demonstrate that LatentRefine-SegZero achieves a favorable balance between in-domain segmentation accuracy and out-of-domain generalization. It leads the compared methods without pixel-loss updates to pretrained MLLM parameters by 1.4–3.9 cIoU points across all eight RefCOCO-series splits, while outperforming the compared methods with such updates on RefAdv and PR-Bench. It also improves over the Qwen2.5-VL+SAM2 baseline in both domains.

1 INTRODUCTION

Referring segmentation aims to identify a target through logical reasoning over a natural language query and generate its pixel-level binary mask (Yu et al., 2018; Yang et al., 2022; Wang et al., 2022; Liu et al., 2023b; Zou et al., 2023a;b; Yang et al., 2023; Lai et al., 2024; Ren et al., 2024a;b; Rasheed et al., 2024; Liu et al., 2025). Unlike closed-vocabulary and open-vocabulary semantic segmentation, referring segmentation handles more complex, context-dependent referring expressions and involves two interconnected dimensions of reasoning. The first is textual reasoning, which requires understanding the target description, attributes, and relationships expressed in the query. The second is visual reasoning, which involves searching for candidate objects and establishing fine-grained correspondences between textual descriptions and image regions. For example, resolving the query "the object held by the person in red who is closer to the car" requires joint reasoning about appearance attributes, spatial relationships, and object interactions. Such tasks reflect practical needs in intelligent assistants and robotics, requiring coordinated understanding of language and visual content. Multimodal large language models (MLLMs) (Li et al., 2023; Liu et al., 2023a; Wang et al., 2024; Liu et al., 2024; Yang et al., 2025; Bai et al., 2025b), with their integrated reasoning and multimodal perception capabilities, have therefore become a central component of recent reasoning segmentation methods.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

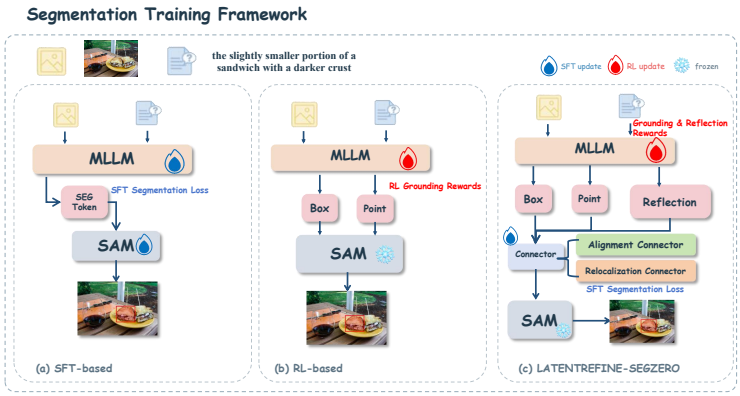

Figure 1: Comparison of SFT-based, RL-based, and LatentRefine-SegZero segmentation paradigms.

Recent studies on MLLM-based referring segmentation mainly follow two paradigms. SFT-based methods generate special segmentation tokens to control a learnable segmentation head or mask decoder, enabling strong adaptation to pixel-level segmentation tasks (Yan et al., 2024; Lai et al., 2024; Yan et al., 2025). However, directly updating the pretrained MLLM backbone with narrow pixel-level supervision may compromise its general reasoning capabilities and lead to overfitting to the training distribution, thereby limiting generalization to out-of-domain referring expressions (Chu et al., 2025). In contrast, RL-based methods optimize MLLMs with segmentation-quality rewards to generate bounding boxes or point prompts for SAM (Liu et al., 2025; Huang et al., 2025). Although this paradigm better preserves the original capabilities of MLLMs, some methods decouple prompt generation from SAM's actual mask outputs, while others rely on iterative explicit prompt refinement (Zhu et al., 2025; Du et al., 2026). Both remain constrained by the limited ability of MLLMs to model fine-grained spatial locations and mask-level errors, resulting in unstable refinement gains and noticeably lower in-domain performance than SFT-based methods.

Although SFT-based methods also consider OOD performance an important evaluation criterion, OOD benchmarks vary substantially in the degree of linguistic distribution shift they present. As shown in Fig. 2, GroundingSuite-Eval (Hu et al., 2025) and RefCLEF (Kazemzadeh et al., 2014) remain comparatively close to the RefCOCO (Yu et al., 2016; Mao et al., 2016) training distribution in the text embedding space. In contrast, Ref-Adv (Dong et al., 2026) forms a clearly separated cluster and has the largest centroid cosine distance. Thus, strong results on linguistically similar OOD benchmarks may not fully reflect generalization under larger distribution shifts or reveal the degradation of general referring ability after pixel-loss updates to pretrained MLLM parameters. These raise a key question: **how can we achieve the strong in-domain segmentation performance of SFT-based methods while maintaining generalization to out-of-domain referring expressions?** Preserving the generalization ability of RL-based methods calls for avoiding pixel-loss updates to pretrained MLLM parameters. At the same time, achieving the strong in-domain segmentation performance of SFT-based methods requires an effective intermediate bridge between the MLLM semantic space and the SAM prompt space.

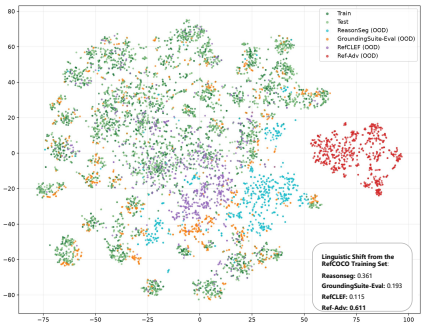

Figure 2: **Linguistic Shift from the RefCOCO Training Set.** Referring expressions are encoded using the frozen `all-roberta-large-v1` (Chen et al., 2026) encoder and visualized using PCA followed by t-SNE. Reported centroid cosine distances are computed in the original embedding space relative to the RefCOCO training-set centroid, with larger values indicating greater linguistic shift.

Based on this, we propose **LatentRefine-SegZero**, a latent-space refinement framework for referring segmentation based on reflective reinforcement learning. Given an image and a referring expression, the method first generates an initial target proposal consisting of a bounding box and points. It then reflects on this proposal, evaluates whether the current target understanding and localization are reliable, and accordingly selects the corresponding latent-space refinement pathway. Unlike methods that apply a unified explicit prompt correction strategy to all segmentation failures, LatentRefine-SegZero decouples the reliability assessment of the target proposal from the mask refinement process, thereby performing targeted corrections for different types of failures. During training, LatentRefine-SegZero combines reinforcement learning with pixel-level supervi-

sion. GRPO optimizes the MLLM's initial proposal generation, error reflection, and reliability-aware routing. Pixel-level losses computed from SAM's predicted masks and ground-truth masks train lightweight two-stage connectors that map MLLM representations into the SAM prompt space for mask refinement and target relocalization. This design prevents narrow pixel-level supervision from altering the pretrained MLLM backbone parameters, balancing in-domain segmentation performance and out-of-domain generalization. The main contributions of this work are as follows:

- We propose **LatentRefine-SegZero**, which evaluates the reliability of the initial target proposal through reflective reinforcement learning and dynamically selects between the semantic alignment and semantic relocalization refinement pathways, avoiding undifferentiated correction strategies for different types of segmentation errors.
- We design two lightweight and complementary connector modules. The **Alignment Connector** bridges the representation gap between the MLLM and SAM while preserving reliable geometric prompts. The **Relocalization Connector** uses reflective information to generate latent prompts and reinterprets the referring expression and localizes the target when the initial localization is unreliable.
- Extensive experiments demonstrate that LatentRefine-SegZero achieves the best in-domain results among the compared methods without pixel-loss updates to pretrained MLLM parameters, exceeding the strongest competing result by 1.4–3.9 cIoU points across all eight RefCOCO-series splits. On RefAdv and PR-Bench, it outperforms all compared methods with such updates. It also improves upon its Qwen2.5-VL+SAM2 baseline in both domains, demonstrating a favorable balance between in-domain accuracy and out-of-domain generalization.

2 Related Work

Referring Segmentation Benchmarks RefCOCO, RefCOCO+, and RefCOCOg are standard benchmarks for object grounding and segmentation guided by natural language (Yu et al., 2016; Mao et al., 2016). Recent benchmarks extend evaluation beyond explicit and relatively short expressions. ReasonSeg focuses on open visual and language instructions (Lai et al., 2024), while Ref-Adv and PR-Bench emphasize target disambiguation, distractor resistance, visual perception, and compositional reasoning (Dong et al., 2026; Gao et al., 2026). Together, they support the evaluation of segmentation accuracy and generalization to unseen distributions.

Supervised Learning from Representations to Masks Segmentation with MLLMs requires converting multimodal representations into spatial masks. The SAM family accepts points, boxes, and masks as prompts, making it a natural segmentation backend for MLLMs (Kirillov et al., 2023; Ravi et al., 2025). One line of work maps MLLM hidden states or special segmentation tokens to pixel decoders, supporting dense grounding, reasoning at the pixel level, prediction with absent or multiple targets, and query learning guided by language (Lai et al., 2024; Rasheed et al., 2024; Ren et al., 2024b; Zhang et al., 2024b; Xia et al., 2024; Zhang et al., 2024a; Qian et al., 2026). Another line converts multimodal representations into prompts compatible with SAM. EVF-SAM explores early fusion of visual and language features, SAM4MLLM learns point prompts generated by an MLLM, and LENS maps MLLM representations into segmentation prompts (Chen et al., 2024; Zhang et al., 2026; Zhu et al., 2026). Improvements to the SAM decoder can further enhance mask quality (Ke et al., 2023). Although supervised adaptation achieves strong performance on the training distribution, narrow pixel-level supervision may encourage overfitting, limiting generalization to out-of-domain referring expressions.

Reinforcement Learning and Refinement with Feedback Recent methods use reinforcement learning and model feedback to improve segmentation reasoning and spatial prompt generation. Seg-Zero uses GRPO to generate reasoning chains and structured box and point prompts, while VisionReasoner extends optimization driven by rewards to broader visual perception tasks (Liu et al., 2025; 2026). Seg-R1 and GenSeg-R1 generate spatial prompts compatible with SAM through reinforcement learning (You & Wu, 2025; Hegde et al., 2026). SAM-R1 incorporates segmentation rewards and feedback from SAM, whereas CoPRS converts chain of thought reasoning into a differentiable positional prior and combines reasoning

rewards with supervision at the pixel level (Huang et al., 2025; Lu et al., 2026). Other methods use feedback during inference. CoT-Seg and SegAgent inspect and correct segmentation results through multimodal reasoning, StAR increases reasoning computation during inference, and SAM-Veteran improves mask understanding and iterative interaction with SAM (Zhu et al., 2025; Kao et al., 2026; Yun et al., 2026; Du et al., 2026). However, these methods still rely largely on generating or correcting explicit boxes and points. Because MLLMs have limited ability to represent precise spatial locations and mask boundary errors, such refinement may produce unstable gains. Moreover, existing methods rarely distinguish an unreliable target proposal from a reliable proposal with an inaccurate mask, although the two cases require different refinement strategies (Zhu et al., 2025; Du et al., 2026).

Algorithm 1: Inference of LatentRefine-SegZero

Require: Image I and expression q
Ensure: Segmentation mask M
Construct u_{prop} containing q
$g^{(0)} \leftarrow \mathcal{G}_\theta(I, u_{\text{prop}})$
Retain input–response sequence $s^{(0)}$
$I^{(r)} \leftarrow \text{Render}(I, b^{(0)})$
Prepare reflective prompt u_{refl}
$(r, \rho) \leftarrow \mathcal{G}_\theta(I^{(r)}, u_{\text{refl}})$
Retain input–response sequence $s^{(r)}$
$F_I \leftarrow \mathcal{E}_{\text{img}}(I)$
$P_{\text{pt}} \leftarrow \mathcal{E}_{\text{prompt}}(p_1^{(0)}, p_2^{(0)})$
if $\rho =$ reliable **then**
    $h^{\text{align}} \leftarrow \mathcal{H}_\theta(s^{(0)}, Q_{\text{prop}})$
    $z_{\text{prop}}^{(0)} \leftarrow \mathcal{E}_{\text{prompt}}(g^{(0)})$
    $S_\rho \leftarrow [z_{\text{prop}}^{(0)}; \mathcal{C}_{\text{align}}(h^{\text{align}}, P_{\text{pt}})]$
else
    $h^{\text{reloc}} \leftarrow \mathcal{H}_{\theta, \Delta\theta_{\text{reloc}}}(s^{(r)}, Q_{\text{refl}})$
    $S_\rho \leftarrow \mathcal{C}_{\text{reloc}}(h^{\text{reloc}}, P_{\text{pt}})$
end if
$M \leftarrow \mathcal{D}_{\text{mask}}(F_I, S_\rho)$
return M

3 METHOD

To mitigate the trade-off between in-domain segmentation performance and out-of-domain generalization, we propose **LatentRefine-SegZero**, a refinement framework for referring segmentation. The framework uses reinforcement learning to optimize initial target proposal generation and reliability assessment, and then selects either semantic alignment or semantic relocalization according to proposal reliability. We next present the overall framework, the reflective reasoning mechanism, dual path latent space refinement, and the training and inference procedures.

3.1 PROBLEM FORMULATION AND FRAMEWORK OVERVIEW

LatentRefine-SegZero decomposes referring segmentation into proposal reasoning and latent prompt refinement. Given an image I and a referring expression q, the goal is to predict the segmentation mask M of the referred target. We denote the MLLM parameters by θ and use $\mathcal{G}_\theta(I, u)$ for response generation conditioned on an image I and a task prompt u, followed by extraction of the structured output.

The MLLM first generates an initial proposal:

$$g^{(0)} = \mathcal{G}_\theta(I, u_{\text{prop}}) = \left(b^{(0)}, p_1^{(0)}, p_2^{(0)}\right), \tag{1}$$

where u_{prop} is the proposal-generation prompt containing q, $b^{(0)}$ denotes the initial bounding box, and $p_1^{(0)}$ and $p_2^{(0)}$ denote two initial point prompts. The model then reflects on the proposal and predicts its reliability ρ. Based on this prediction, LatentRefine-SegZero dynamically selects one of two latent-space refinement pathways.

For a reliable proposal, the semantic alignment pathway preserves the initial prompts and complements them with a latent semantic prompt generated from the MLLM representation. For an unreliable proposal, the semantic relocalization pathway uses the reflection-enhanced MLLM representation to generate a new prompt embedding, without directly passing the initial prompt embedding to the final mask decoder.

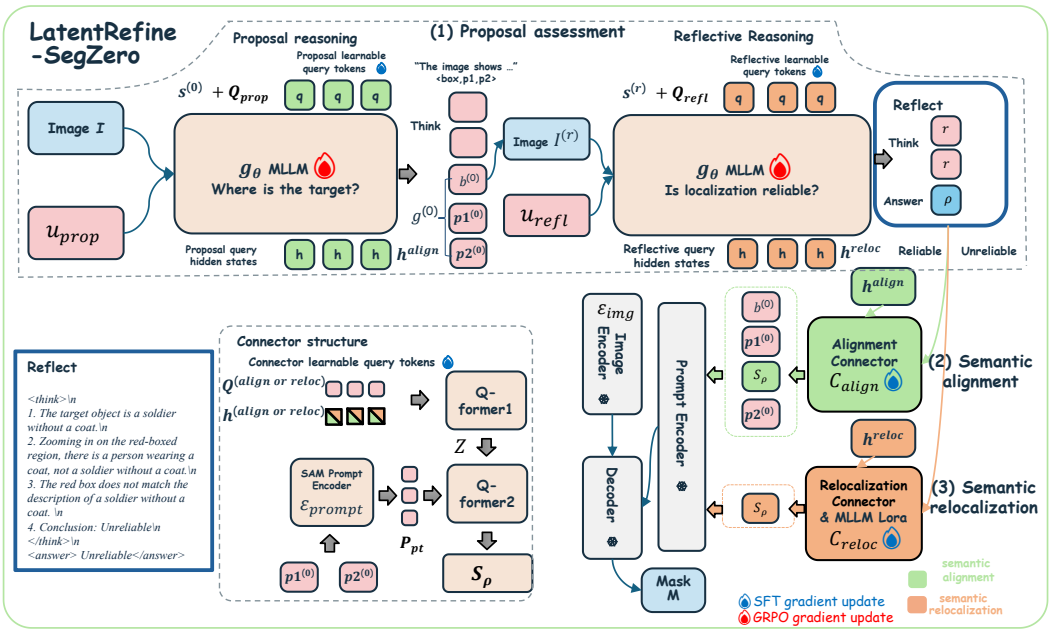

Figure 3: Overview of **LatentRefine-SegZero**. The framework comprises three components: (1) **Proposal assessment**, which uses reflective reasoning to assess proposal reliability and select a refinement pathway; (2) **Semantic alignment**, which converts proposal representations into supplementary latent prompts to refine masks for reliable proposals; and (3) **Semantic relocalization**, which converts reflective representations into new latent prompts to relocalize targets for unreliable proposals. Both pathways use lightweight two-stage connectors to map MLLM representations into the SAM prompt space. GRPO updates the MLLM for proposal generation and reflective reasoning, while pixel-level supervision trains the refinement pathways.

We denote SAM's image encoder, prompt encoder, and mask decoder by $\mathcal{E}_{\text{img}}$, $\mathcal{E}_{\text{prompt}}$, and $\mathcal{D}_{\text{mask}}$, respectively. The image features and initial proposal prompt embedding are

$$F_I = \mathcal{E}_{\text{img}}(I), \qquad z^{(0)}_{\text{prop}} = \mathcal{E}_{\text{prompt}}(g^{(0)}). \tag{2}$$

The final mask prediction is formulated as

$$M = \mathcal{D}_{\text{mask}}(F_I, S_\rho), \tag{3}$$

$$S_\rho = \begin{cases} \left[z^{(0)}_{\text{prop}}; \mathcal{C}_{\text{align}}(h^{\text{align}}, P_{\text{pt}})\right], & \rho = \text{reliable}, \\ \mathcal{C}_{\text{reloc}}(h^{\text{reloc}}, P_{\text{pt}}), & \rho = \text{unreliable}. \end{cases} \tag{4}$$

Here, $[\cdot;\cdot]$ denotes concatenation along the prompt token dimension. The representations h^{align} and h^{reloc} are MLLM query hidden states extracted after initial proposal generation and reflective reasoning, respectively. The semantic alignment connector $\mathcal{C}_{\text{align}}$ and semantic relocalization connector $\mathcal{C}_{\text{reloc}}$ map these representations into SAM-compatible prompt embeddings, using the initial point embeddings P_{pt} as spatial conditions. This design prevents narrow pixel-level supervision from altering the pretrained MLLM backbone parameters, balancing in-domain performance and out-of-domain generalization.

3.2 Reflective Reasoning

Referring segmentation errors can arise from incomplete mask decoding or incorrect referring comprehension. Existing methods often treat both with narrow pixel-level supervision or sparse rewards, without distinguishing their causes. We therefore introduce reflective reasoning to assess the initial proposal's reliability and select the appropriate latent-space refinement pathway.

Specifically, we render the initial bounding box $b^{(0)}$ on the original image I:

$$I^{(r)} = \mathrm{Render}(I, b^{(0)}). \tag{5}$$

The MLLM then evaluates the rendered proposal using a reflective prompt u_{refl}, generating the reflective context r and reliability prediction ρ:

$$(r, \rho) = \mathcal{G}_\theta(I^{(r)}, u_{\mathrm{refl}}), \qquad \rho \in \{\mathrm{reliable}, \mathrm{unreliable}\}. \tag{6}$$

When $\rho = \mathrm{reliable}$, the initial target interpretation and localization are considered sufficiently accurate, and the semantic alignment pathway is selected to preserve reliable spatial constraints. When $\rho = \mathrm{unreliable}$, the initial proposal is considered likely to deviate from the intended target, and the semantic relocalization pathway uses the reflective context to support target relocalization.

3.3 Latent-Space Refinement for Referring Segmentation

Let $s^{(0)}$ and $s^{(r)}$ denote the complete sequences for proposal generation and reflective reasoning, respectively, each comprising the input image, task prompt, and generated response. Following (Zhu et al., 2026), we append learnable query tokens to extract MLLM hidden states. Specifically, we use proposal and reflective queries for the two pathways:

$$h^{\mathrm{align}} = \mathcal{H}_\theta(s^{(0)}, Q_{\mathrm{prop}}), \tag{7}$$

$$h^{\mathrm{reloc}} = \mathcal{H}_{\theta, \Delta\theta_{\mathrm{reloc}}}(s^{(r)}, Q_{\mathrm{refl}}). \tag{8}$$

Here, $\mathcal{H}_\theta(s, Q)$ returns the query-position hidden states from an MLLM forward pass over s followed by Q. The relocalization pathway additionally uses LoRA (Hu et al., 2022) parameters $\Delta\theta_{\mathrm{reloc}}$.

Two-Stage Latent Prompt Connector. To map MLLM latent representations into SAM's spatially grounded prompt space, $\mathcal{C}_{\mathrm{align}}$ and $\mathcal{C}_{\mathrm{reloc}}$ adopt a two-stage connector built upon the QFormer architecture (Li et al., 2023), with the same architecture but independent parameters. We denote the QFormer at stage $k \in \{1, 2\}$ of pathway $\xi \in \{\mathrm{align}, \mathrm{reloc}\}$ by $\mathrm{QFormer}_k^\xi(Q, X)$, where Q denotes the queries and X the features attended to.

In the first stage, learnable connector queries Q_ξ aggregate h^ξ into latent semantic tokens relevant to the target:

$$Z_\xi = \mathrm{QFormer}_1^\xi(Q_\xi, h^\xi), \tag{9}$$

where h^ξ denotes the query hidden states of pathway ξ. The queries Q_ξ are distinct from the query tokens appended to the MLLM sequences.

In the second stage, the initial point prompts are encoded as spatial anchors:

$$P_{\mathrm{pt}} = \mathcal{E}_{\mathrm{prompt}}\left(p_1^{(0)}, p_2^{(0)}\right). \tag{10}$$

These embeddings query Z_ξ to extract semantic corrections for the corresponding locations and generate sparse prompt embeddings compatible with SAM:

$$\mathcal{C}_\xi(h^\xi, P_{\mathrm{pt}}) = \mathrm{QFormer}_2^\xi(P_{\mathrm{pt}}, Z_\xi). \tag{11}$$

Reliability-Aware Latent-Space Refinement. Proposal reliability ρ determines the representation and prompt composition used for refinement. For reliable proposals, $\mathcal{C}_{\mathrm{align}}$ uses h^{align} to generate a prompt embedding that complements $z_{\mathrm{prop}}^{(0)}$, preserving reliable spatial constraints while refining the mask. For unreliable proposals, $\mathcal{C}_{\mathrm{reloc}}$ uses h^{reloc} to generate a new prompt embedding, without directly passing $z_{\mathrm{prop}}^{(0)}$ to the mask decoder.

Both pathways use P_{pt} within their connectors, but only the alignment pathway also passes $z_{\mathrm{prop}}^{(0)}$ to the mask decoder. The final prediction follows Eq. 3 and Eq. 4.

3.4 Training and Inference

Reflective reasoning optimization. We optimize the MLLM parameters θ with GRPO for initial proposal generation and reliability assessment, using the following rewards:

$$\begin{aligned} R_{\text{prop}} &= R_{\text{format}} + R_{\text{boxIoU}} + R_{\text{boxL1}} + R_{\text{point}}, \\ R_{\text{refl}} &= R_{\text{format}} + R_{\text{cons}} + R_{\text{decision}}. \end{aligned} \tag{12}$$

These rewards encourage valid outputs, accurate proposals, and correct reliability predictions consistent with the reflective reasoning. Detailed reward definitions are provided in Appendix A.4.

Latent-space refinement optimization. With the MLLM backbone and SAM frozen, we train the refinement modules using pixel-level supervision:

$$\mathcal{L}_{\text{seg}}^{\xi} = \mathcal{L}_{\text{BCE}}(M_{\xi}, M^{\text{GT}}) + \mathcal{L}_{\text{Dice}}(M_{\xi}, M^{\text{GT}}), \quad \xi \in \{\text{align}, \text{reloc}\}, \tag{13}$$

where M_{ξ} is the prediction of pathway ξ and M^{GT} is the ground-truth mask. Pixel-level gradients propagate through the frozen SAM mask decoder to update the connectors, including their internal queries, and the proposal and reflective query tokens. The relocalization pathway additionally updates its LoRA parameters. Details of joint connector pre-alignment and pathway-specific training are provided in Appendix A.4.

Inference. Given an image I and a referring expression q, the MLLM generates an initial proposal and assesses its reliability through reflective reasoning. The selected refinement pathway constructs SAM-compatible prompt embeddings, which are decoded into the final mask. Algorithm 1 summarizes the complete procedure.

4 Experiments

4.1 Experimental Settings

Datasets and Evaluation Metrics. We train our model on the training splits of RefCOCO, RefCOCO+, and RefCOCOg, and evaluate it on their standard validation and test splits. To assess zero-shot generalization, we further evaluate on RefAdv and PR-Bench, with no samples from either benchmark used during training. Following prior work, we report cumulative IoU (cIoU) on the RefCOCO series, and mean bounding-box IoU together with Box Acc@0.5 on RefAdv and PR-Bench.

Implementation Details. We use Qwen2.5-VL-7B as the MLLM and frozen SAM2-Large as the segmentation model. For the MLLM, the longest side of each input image is resized to 840 while preserving its aspect ratio. Each two-stage connector contains 64 learnable queries of dimension 1,024 and produces 2 latent prompt tokens. We optimize the MLLM with GRPO using veRL (Sheng et al., 2024) for initial proposal generation and reliability assessment, with eight responses per prompt, a global batch size of 16, and a learning rate of 10^{-6}. The refinement modules are trained for three epochs with AdamW (Loshchilov & Hutter, 2019), a learning rate of 3×10^{-5}, weight decay 0.01, and an effective batch size of 64. Training is performed on 8 NVIDIA A800 GPUs using BF16 precision. Additional training details are provided in Appendix B.

4.2 In-Domain Performance and Out-of-Domain Generalization

Balancing in-domain performance and out-of-domain generalization. As shown in Table 1, LatentRefine-SegZero achieves a favorable balance between in-domain performance and out-of-domain generalization. Compared with methods that use pixel-level losses to update pretrained MLLM parameters, it achieves higher scores on both out-of-domain benchmarks: 0.47/0.50 on RefAdv and 0.56/0.61 on PR-Bench in BBox IoU and Box Acc@0.5, respectively. It also outperforms GLaMM on all RefCOCO+ and RefCOCOg splits. Among methods without pixel-loss updates to pretrained MLLM parameters, LatentRefine-SegZero achieves the highest cIoU across all eight RefCOCO-series splits, exceeding the strongest competing result on each split by 1.4–3.9 points. Its out-of-domain performance remains comparable to VisionReasoner.

Table 1: Comparison with state-of-the-art methods on the RefCOCO series, RefAdv, and PR-Bench. We report cIoU (%) on RefCOCO/+/g, and mean BBox IoU and Box Acc@0.5 on RefAdv and PR-Bench. † denotes dataset-specific training. The same LatentRefine-SegZero model appears in both method groups. The grouping criteria are detailed in Appendix B.2. Bold and underlined values indicate the best and second-best results within each group, including our method. The baseline is reported separately and excluded from ranking.

Method	In-Domain								Out-of-Domain			
	RefCOCO			RefCOCO+			RefCOCOg		RefAdv		PR-Bench	
	val	testA	testB	val	testA	testB	val	test	BBox IoU	Box Acc@0.5	BBox IoU	Box Acc@0.5
Baseline												
Qwen2.5-VL(7B) (Bai et al., 2025b) + SAM2 (Ravi et al., 2025)	–	76.1	–	–	71.4	–	–	64.8	0.38	0.39	0.50	0.54
Qwen3-VL(8B) (Bai et al., 2025a) + SAM3 (Carion et al., 2026)	–	78.6	–	–	75.5	–	–	72.1	0.51	0.56	0.62	0.69
SAM3 (Carion et al., 2026)	–	59.0	–	–	49.5	–	–	55.5	0.30	0.30	0.44	0.48
Pixel-loss updates to pretrained MLLM												
LISA (Lai et al., 2024)	74.1	76.5	71.1	62.4	67.4	56.5	66.4	68.5	0.31	0.26	0.39	0.37
VISA (Yan et al., 2024)	72.4	75.5	68.1	59.8	64.8	53.1	65.5	66.4	–	–	–	–
PixelLM (Ren et al., 2024b)	76.9	78.5	74.4	69.2	72.1	64.5	70.7	72.4	0.22	0.17	0.25	0.23
GLaMM (Rasheed et al., 2024)	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9	0.32	0.29	0.37	0.36
LENS† (Zhu et al., 2026)	**84.2**	**85.3**	**81.0**	**79.4**	**82.8**	**74.3**	**81.2**	**81.0**	0.27	0.19	0.28	0.22
LatentRefine-SegZero	80.0	82.8	76.4	74.8	79.8	68.3	76.1	76.9	**0.47**	**0.50**	**0.56**	**0.61**
No pixel-loss updates to pretrained MLLM												
SegAgent (Zhu et al., 2025)	78.0	80.3	75.0	70.9	75.5	65.8	74.5	74.6	0.33	0.32	0.36	0.37
SAM-R1 (Huang et al., 2025)	–	79.2	–	–	74.7	–	–	73.1	–	–	–	–
VisionReasoner (Liu et al., 2026)	–	78.9	–	–	74.9	–	72.6	71.3	**0.48**	**0.51**	0.55	**0.62**
SAM-VETERAN (Du et al., 2026)	–	80.8	–	–	76.6	–	–	73.4	–	–	–	–
LatentRefine-SegZero	**80.0**	**82.8**	**76.4**	**74.8**	**79.8**	**68.3**	**76.1**	**76.9**	0.47	0.50	**0.56**	0.61

Comparison with pretrained baselines. As shown in Table 1, LatentRefine-SegZero consistently improves upon the Qwen2.5-VL-7B + SAM2 pipeline on both in-domain and out-of-domain benchmarks. SAM 3 alone performs less well on the referring expression benchmarks, suggesting that concept-prompt segmentation by itself is insufficient for these queries. Combining Qwen3-VL-8B with SAM 3 improves the pretrained pipeline on all three in-domain splits, but still trails LatentRefine-SegZero and the strongest supervised methods. This gap highlights the value of task-specific optimization for strong in-domain segmentation performance. On RefAdv and PR-Bench, however, Qwen3-VL-8B + SAM 3 outperforms our method, underscoring the value of pretrained capabilities for out-of-domain referring expressions. Together, these results motivate our goal of improving in-domain segmentation while maintaining out-of-domain generalization.

4.3 Ablation Studies

Effect of Routing Guided by Reflective Reasoning. Figure 4 compares LatentRefine-SegZero with variants that always select the semantic alignment pathway, always select the semantic relocalization pathway, or randomly route samples between the two pathways. Our reliability-aware routing consistently achieves the best performance, outperforming the stronger single-pathway variant by 0.5–0.8 cIoU points and random routing by 1.7–2.4 points. Although random routing uses the same overall pathway proportions as our method, it assigns samples without considering the reliability of their initial target proposals. This comparison demonstrates that the gain primarily arises from the proposal reliability assessment enabled by reflective reasoning, rather than from simply combining the two refinement pathways.

Table 2: Ablation study of the two-stage latent prompt connector on RefCOCOg val. Results are reported in cIoU.

Variant	Alignment	Relocalization
w/o Point Anchors	72.4	60.2
MLP Connector	70.5	65.6
One-Stage Connector	72.6	67.5
Two-Stage Connector (Ours)	**72.6**	**68.0**

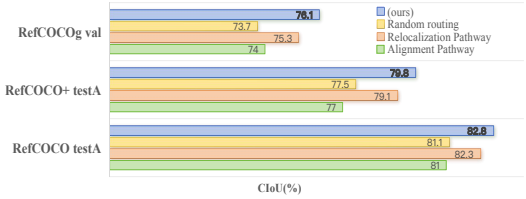

Figure 4: Ablation study of routing guided by reflective reasoning on in-domain benchmarks.

Table 3: Ablation study of joint connector pre-alignment. Results are reported in cIoU.

Joint Pre-Alignment	Alignment Pathway		Relocalization Pathway	
	RefCOCO+ testA	RefCOCOg val	RefCOCO+ testA	RefCOCOg val
✗	76.7	73.6	78.3	74.2
✓	**77.0**	**74.0**	**79.1**	**75.3**

Table 4: Ablation study of routing guided by reflective reasoning on out-of-domain benchmarks.

Dataset	Metric	Relocalization Only	Ours
RefAdv	BBox IoU	0.33	**0.47**
	Box Acc@0.5	0.30	**0.50**
PR-Bench	BBox IoU	0.38	**0.56**
	Box Acc@0.5	0.39	**0.61**

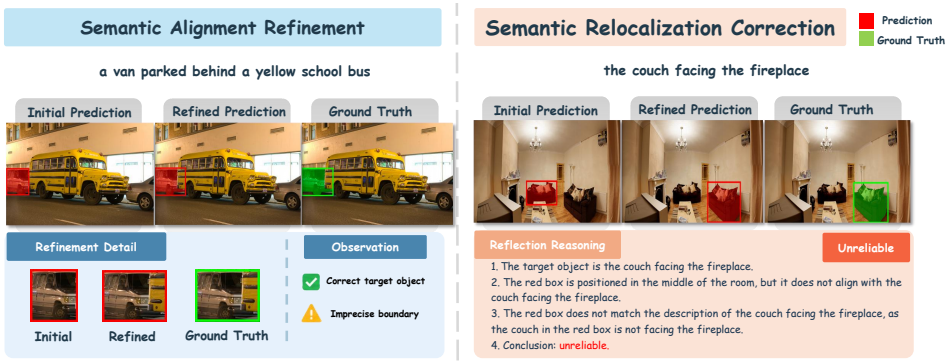

Figure 5: Qualitative examples of LatentRefine-SegZero: semantic alignment refines a reliable proposal (left), while semantic relocalization corrects an unreliable one (right).

Table 4 shows the benefit of reliability-aware routing on out-of-domain benchmarks. Routing all samples through semantic relocalization discards even reliable initial proposal embeddings, reducing Box Acc@0.5 from 0.50 to 0.30 on RefAdv and from 0.61 to 0.39 on PR-Bench. By preserving reliable spatial constraints through semantic alignment and relocalizing only unreliable proposals, our method achieves better out-of-domain performance.

Effect of Connector Design. Table 2 compares connector designs on RefCOCOg val after joint pre-alignment but before pathway specialization (Appendix A.4). Removing point anchors reduces cIoU by 0.2 points for semantic alignment and 7.8 points for semantic relocalization. The two-stage connector outperforms the MLP by 2.1 and 2.4 points, respectively; compared with the one-stage connector, it matches alignment performance and improves relocalization by another 0.5 points.

Effect of Joint Connector Pre-Alignment. Joint pre-alignment establishes a shared mapping to the SAM prompt space before pathway specialization. Across the three benchmarks in Table 3 and Appendix Table 7, it improves semantic alignment by 0.2–0.4 cIoU and semantic relocalization by 0.6–1.1 cIoU, with larger gains for target recovery.

4.4 Visualizations

Figure 5 illustrates both refinement pathways. Semantic alignment refines mask boundaries for reliable proposals, while reflective reasoning guides semantic relocalization to recover the correct target from unreliable proposals.

5 Conclusion

We present LatentRefine-SegZero, a reliability-aware latent-space refinement framework for referring segmentation. The method uses reflective reinforcement learning to assess initial proposal reliability and dynamically selects between semantic alignment and semantic relocalization. Experiments on RefCOCO, RefCOCO+, RefCOCOg, RefAdv, and PR-Bench demonstrate that LatentRefine-SegZero effectively balances in-domain segmentation accuracy with generalization to out-of-domain referring expressions.

AI USE STATEMENT

We used OpenAI Codex to assist with drafting and organizing the appendix, editing its language, and aligning its terminology and notation with the main text. Codex also assisted with inspecting the existing training and evaluation scripts to check the appendix's descriptions of reward functions, model components, training configurations, and evaluation protocols, and with preparing numerical checks of the reward functions. The authors take responsibility for the final content of this work, including text, claims, and artifacts prepared with AI assistance.

ETHICS STATEMENT

This work uses publicly available datasets under their established licenses and guidelines. Our method improves visual grounding and segmentation in MLLMs without introducing additional risks beyond existing models. It may inherit biases from pretrained models and training data. Incorrect localization or segmentation may pose risks in safety-critical applications. Responsible use, transparent evaluation, and appropriate human oversight are therefore essential. This work adheres to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

All datasets used in our experiments are publicly available. Section 4.1 describes the training and evaluation datasets, metrics, backbones, and principal optimization settings. The model architecture and training objectives are specified in Sections 3 and 3.4. The appendix details proposal assessment and latent prompt construction in Appendices A.1 and A.2, training data construction and pathway optimization in Appendices A.3 and A.4, and hyperparameters and evaluation metrics in Appendices B.1 and B.3. The complete joint pre-alignment ablation is reported in Table 7, and inference efficiency is documented in Appendix B.4. Our GRPO implementation is based on the open-source veRL framework, with task-specific modifications for proposal generation, reliability assessment, reward calculation, and mixed-task sampling. Together, these materials document the data selection, reward components, trainable parameters, optimization settings, and evaluation procedures used in the reported experiments.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026.

Guanhua Chen, Yutong Yao, Shenghe Sun, Ci-Jun Gao, Shudong Liu, Lidia S Chao, Feng Wan, and Derek F Wong. Chain-of-Procedure: Hierarchical visual-language reasoning for procedural qa. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17207–17224, 2026.

Yi-Chia Chen, Wei-Hua Li, Cheng Sun, Yu-Chiang Frank Wang, and Chu-Song Chen. SAM4MLLM: Enhance multi-modal large language model for referring expression segmentation. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10818–10838. PMLR, 2025.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-Adv: Exploring MLLM visual reasoning in referring expression tasks. In *International Conference on Learning Representations*, 2026.

Tianyuan Du, Haopeng Li, Zhen Fan, Jiarui Zhang, Panwang Pan, and Yang Zhang. SAM-Veteran: An MLLM-based human-like SAM agent for reasoning segmentation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Tianyi Gao, Han Fang, Tianyi Ding, Hao Li, Xin Wei, Hongbo Sun, Xiaodong Dong, Ye Yuan, Jinglin Xu, Kongming Liang, et al. Mixture-of-Thought-Tokens: Unifying perception and reasoning for free-form multimodal grounding. *arXiv preprint arXiv:2607.24407*, 2026.

Sandesh Hegde, Jaison Saji Chacko, Debarshi Banerjee, and Uma Mahesh. GenSeg-R1: RL-driven vision-language grounding for fine-grained referring segmentation. *arXiv preprint arXiv:2602.09701*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Rui Hu, Lianghui Zhu, Yuxuan Zhang, Tianheng Cheng, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. GroundingSuite: Measuring complex multi-granular pixel grounding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23105–23114. IEEE, 2025.

Jiaqi Huang, Zunnan Xu, Jun Zhou, Ting Liu, Yicheng Xiao, Mingwen Ou, Bowen Ji, Xiu Li, and Kehong Yuan. SAM-R1: Leveraging SAM for reward feedback in multimodal segmentation via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 138362–138383, 2025.

Shiu-hong Kao, Chak Ho Huang, Huaiqian Liu, Yu-Wing Tai, and Chi-Keung Tang. CoT-Seg: Rethinking segmentation with chain-of-thought reasoning and self-correction. *arXiv preprint arXiv:2601.17420*, 2026.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 787–798, 2014.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *Advances in Neural Information Processing Systems*, volume 36, pp. 29914–29934, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pp. 19730–19742. PMLR, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023a.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

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

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

Yuqi Liu, Tianyuan Qu, Zhisheng Zhong, Bohao Peng, Shu Liu, Bei Yu, and Jiaya Jia. VisionReasoner: Unified reasoning-integrated visual perception via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 94069–94086, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Zhenyu Lu, Liupeng Li, Jinpeng Wang, Yan Feng, Bin Chen, Ke Chen, and Yaowei Wang. CoPRS: Learning positional prior from chain-of-thought for reasoning segmentation. In *International Conference on Learning Representations*, volume 2026, pp. 37940–37959, 2026.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20, 2016.

Rui Qian, Chuanhang Deng, Qiang Huang, Jian Xiong, Mingxuan Li, Yingbo Zhou, Wei Zhai, Jintao Chen, and Dejing Dou. AnchorSeg: Language grounded query banks for reasoning segmentation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20490–20505, 2026.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. GLaMM: Pixel grounding large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13009–13018. IEEE, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024a.

Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. PixelLM: Pixel reasoning with large multimodal model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26364–26373. IEEE, 2024b.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. CRIS: CLIP-driven referring image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11676–11685. IEEE, 2022.

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

Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. VISA: Reasoning video object segmentation via large language models. In *European Conference on Computer Vision*, pp. 98–115. Springer, 2024.

Ziang Yan, Zhilin Li, Yinan He, Chenting Wang, Kunchang Li, Xinhao Li, Xiangyu Zeng, Zilei Wang, Yali Wang, Yu Qiao, et al. Task preference optimization: Improving multimodal large language models with vision task alignment. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29880–29892. IEEE, 2025.

Rui Yang, Lin Song, Yicheng Xiao, Runhui Huang, Yixiao Ge, Ying Shan, and Hengshuang Zhao. HaploVL: A single-transformer baseline for multi-modal understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 71073–71091. PMLR, 2025.

Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. LISA++: An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023.

Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. LAVT: Language-aware vision transformer for referring image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18134–18144. IEEE, 2022.

Zuyao You and Zuxuan Wu. Seg-R1: Segmentation can be surprisingly simple with reinforcement learning. *arXiv preprint arXiv:2506.22624*, 2025.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pp. 69–85. Springer, 2016.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. MAttNet: Modular attention network for referring expression comprehension. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1307–1315. IEEE, 2018.

Seokju Yun, Dongheon Lee, Noori Bae, Jaesung Jun, Chanseul Cho, and Youngmin Ro. StAR: Segment anything reasoner. *arXiv preprint arXiv:2603.14382*, 2026.

Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 71737–71767, 2024a.

Yuxuan Zhang, Tianheng Cheng, Lianghui Zhu, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. EVF-SAM: Early vision-language fusion for text-prompted segment anything model. *Image and Vision Computing*, 173:106028, 2026.

Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. PSALM: Pixelwise segmentation with large multi-modal model. In *European Conference on Computer Vision*, pp. 74–91. Springer, 2024b.

Lianghui Zhu, Bin Ouyang, Yuxuan Zhang, Tianheng Cheng, Rui Hu, Haocheng Shen, Longjin Ran, Xiaoxin Chen, Li Yu, Wenyu Liu, et al. LENS: Learning to segment anything with unified reinforced reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13952–13960, 2026.

Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen. SegAgent: Exploring pixel understanding capabilities in MLLMs by imitating human annotator trajectories. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3686–3696. IEEE, 2025.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15116–15127. IEEE, 2023a.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *Advances in Neural Information Processing Systems*, volume 36, pp. 19769–19782, 2023b.

APPENDIX CONTENTS

A Supplementary Method Details

A.1 Proposal Assessment Details
- Proposal reasoning
- Reflective reasoning

A.2 Latent Prompt Connector Details
- Connector configuration
- Reliability-Aware Latent Prompt Refinement

A.3 Training Data Construction
- Proposal Reasoning Data
- Reflective Reasoning Data
- Two-Stage Latent Prompt Connector Data
  - Joint connector pre-alignment
  - Semantic alignment data
  - Semantic relocalization data

A.4 Training Strategy
- Stage 1: mixed GRPO training
- Reward definitions
- Stage 2: Two-Stage Latent Prompt Connector Training
  - Joint connector pre-alignment
  - Pathway specialization
  - Alignment pathway training
  - Relocalization pathway training

B Supplementary Experimental Details

B.1 Implementation Details
- Complete joint pre-alignment ablation

B.2 Grouping of Comparison Methods

B.3 Evaluation Metrics

B.4 Inference Efficiency

B.5 Additional Visualizations

A SUPPLEMENTARY METHOD DETAILS

This appendix provides additional details on reliability assessment, latent prompt construction, and pathway-specific optimization. We follow the notation in Section 3, with $\xi \in \{\text{align}, \text{reloc}\}$ denoting the two refinement pathways.

A.1 PROPOSAL ASSESSMENT DETAILS

Proposal assessment contains two consecutive MLLM generations: proposal reasoning and reflective reasoning. The former generates an initial geometric proposal, while the latter assesses its reliability and selects the corresponding refinement pathway.

Proposal reasoning. Given an image I and a referring expression q, the proposal prompt u_{prop} asks the MLLM to predict one bounding box and two interior points. The proposal prompt u_{prop} is shown below, where $\{\texttt{Question}\}$ is replaced by q:

```
<image>
Please find "{Question}" with a bounding box and two points. Compare
nearby objects carefully and select the object that best matches the
referring expression.
Output the reasoning process inside <think>...</think> and the final
answer inside <answer>...</answer>. The final answer must be valid JSON
containing one bounding box and two interior points:
{
  "bbox": [10, 100, 200, 210],
  "points_1": [30, 110],
  "points_2": [35, 180]
}
```

The predicted coordinates are defined on the resized MLLM input, whose longest side is 840 pixels, and are mapped back to the original image resolution before being passed to SAM. The parsed proposal is

$$g^{(0)} = \left(b^{(0)}, p_1^{(0)}, p_2^{(0)}\right), \tag{14}$$

where $b^{(0)}$ denotes the bounding box and $p_1^{(0)}, p_2^{(0)}$ denote the two interior points.

Let $s^{(0)}$ denote the proposal-generation input, consisting of the image and task prompt. As described in Section 3.3, the proposal queries Q_{prop} are appended to this sequence to extract the representation used by semantic alignment:

$$h^{\text{align}} = \mathcal{H}_\theta\left(s^{(0)}, Q_{\text{prop}}\right). \tag{15}$$

Reflective reasoning. For reliability assessment, we render $b^{(0)}$ on the original image as a red box with a line width of four pixels, obtaining

$$I^{(r)} = \text{Render}\left(I, b^{(0)}\right). \tag{16}$$

The two predicted points are not rendered because this stage assesses whether the box involves the intended semantic target, rather than its precise geometric alignment.

The reflective prompt u_{refl} takes $I^{(r)}$ and q as input. The reflective prompt u_{refl} is shown below:

```
<image>
Question: "{Question}"

You are an expert annotator of referred object localization. For an image and a referring
question, you are given a red box and asked to evaluate whether the red box involves the
referred target object.

Reason 4 steps:
```

```
1. Identify the target object described in the question.
2. Zoom in on the red-box region and inspect it carefully. Identify all objects visible within
the red box, including foreground, background, and partially visible objects.
3. Check whether the important details in the question, such as relation to other objects,
color, position, size, and attribute, are consistent. We do not require the red box to be
perfectly aligned with the object. However, the box should involve the referred target object.
4. Give the final conclusion: "reliable" or "unreliable". Do not include any explanation.
Answer "reliable" if no other visible object in the image matches the description of the
target object better than the boxed candidate. Answer "unreliable" only if you find a better
candidate in your reasoning.

Output exactly 4 reasoning steps in <think>...</think> and the final answer in
<answer>...</answer>.
The final answer must be valid JSON with exactly one key: "decision".

Examples:
Example 1:
<think>
1. The target object is the dog lying on the sofa.
2. The red box covers the dog, including its visible body region.
3. Some details are partially ambiguous, but no other visible object in the image matches the
question better than the boxed dog.
4. Conclusion: reliable
</think>
<answer>{"decision":"reliable"}</answer>

Example 2:
<think>
1. The target object is the man on the left.
2. The red box covers a different person.
3. The position detail does not match because the boxed person is on the right side, while the
question refers to the man on the left.
4. Conclusion: unreliable
</think>
<answer>{"decision":"unreliable"}</answer>

Now begin to decide whether the red box involves the referred target object in the question.
```

A proposal is considered reliable if the red box involves the intended target and no visible object provides a better match. Reliable and unreliable proposals are routed to semantic alignment and semantic relocalization, respectively. An unparsable decision is treated as unreliable.

For semantic relocalization, we retain the diagnostic reasoning generated before the explicit reliability conclusion. Specifically, we remove the fourth reasoning step and the final `<answer>` block, and denote the retained reasoning by r. The resulting reflection sequence is

$$s^{(r)} = \left(I^{(r)}, u_{\text{refl}}, r\right). \tag{17}$$

This removes the explicit `reliable`/`unreliable` result while preserving the evidence needed for target recovery. The reflective queries Q_{refl} are then appended to $s^{(r)}$ to extract the semantic relocalization representation:

$$h^{\text{reloc}} = \mathcal{H}_{\theta,\Delta\theta_{\text{reloc}}}\left(s^{(r)}, Q_{\text{refl}}\right). \tag{18}$$

A.2 Latent Prompt Connector Details

Connector configuration. Each pathway appends 64 learnable query tokens specific to that pathway to the MLLM sequence to obtain h^{ξ}, where $\xi \in \{\text{align}, \text{reloc}\}$. These MLLM queries are distinct from the 12 learned connector queries used by QFormer_1. The two connector stages are instantiated as

$$\begin{aligned} Z_{\xi} &= \text{QFormer}_1^{\xi}(Q_{\xi}, h^{\xi}) \in \mathbb{R}^{12\times 1024}, \\ \mathcal{C}_{\xi}(h^{\xi}, P_{\text{pt}}) &= \text{QFormer}_2^{\xi}(P_{\text{pt}}, Z_{\xi}) \in \mathbb{R}^{2\times 256}. \end{aligned} \tag{19}$$

Here, $P_{\text{pt}} \in \mathbb{R}^{2\times 256}$ contains the SAM embeddings of the two initial proposal points. Each QFormer uses cross-attention with eight heads, followed by a Transformer encoder with two layers and GELU activation. Its feed-forward dimension is four times the width of the corresponding stage. The two pathways use independent connector parameters.

Reliability-Aware Latent Prompt Refinement. As formulated in Eq. 4, both pathways use P_{pt} as internal spatial anchors for QFormer_2. For a reliable proposal, the original prompt embedding

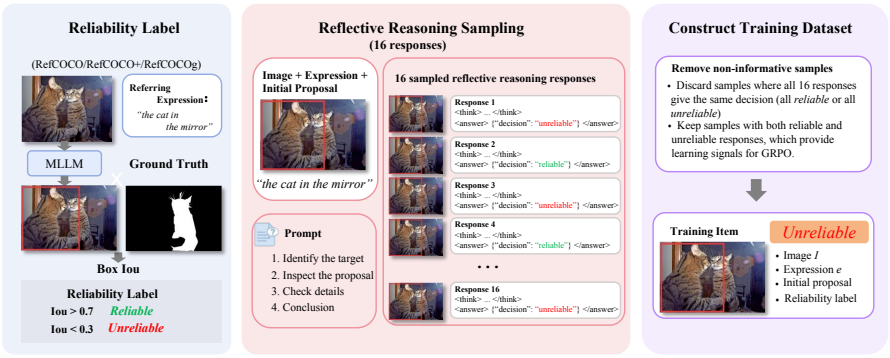

Figure 6: Construction of the reflective reasoning training data.

$z^{(0)}_{\text{prop}}$, which contains the initial box and point tokens, is concatenated with the two tokens produced by $\mathcal{C}_{\text{align}}$. For an unreliable proposal, the sparse prompt input to the mask decoder contains only the two tokens produced by $\mathcal{C}_{\text{reloc}}$; the original proposal embedding is not directly passed to the decoder. Thus, the initial points condition latent prompt construction in both pathways, while only semantic alignment preserves the original geometric prompts as direct decoding constraints.

A.3 Training Data Construction

We construct separate training data for proposal reasoning, reflective reasoning, and the two-stage latent prompt connectors. All samples are drawn from the training splits of RefCOCO, RefCOCO+, and RefCOCOg (Yu et al., 2016; Mao et al., 2016).

Proposal Reasoning Data. Each source example contains an image I, a referring expression, and its target mask M^{GT}. Following Seg-Zero (Liu et al., 2025), we derive the supervision tuple $(b^{\text{GT}}, p_1^{\text{GT}}, p_2^{\text{GT}})$ from the mask. The box b^{GT} is the tight extent of the foreground, and p_1^{GT} and p_2^{GT} are the centers of the two largest inscribed circles in M^{GT}. We randomly select 9,000 examples to train proposal reasoning with GRPO.

Reflective Reasoning Data. We generate initial proposals for the source examples using Qwen2.5-VL. For example i, let $b_i^{(0)}$ be the predicted box and let $v_i^{(0)} = \text{IoU}(b_i^{(0)}, b_i^{\text{GT}})$. We assign the reliability target as

$$\rho_i^{\text{GT}} = \begin{cases} \text{reliable}, & v_i^{(0)} > 0.7, \\ \text{unreliable}, & v_i^{(0)} < 0.3. \end{cases} \tag{20}$$

Examples with $0.3 \le v_i^{(0)} \le 0.7$ are excluded from reliability assessment training. We render the predicted box on the original image to provide the visual input for reflection.

Before GRPO training, we sample 16 responses from the reflection policy for each candidate example using temperature 1.2 and top-$p = 1.0$. After parsing the reliability decision from each response, we count the correct decisions:

$$C_i = \sum_{j=1}^{16} \mathbf{1}\left[\hat{\rho}_i^{(j)} = \rho_i^{\text{GT}}\right]. \tag{21}$$

We exclude examples with $C_i = 0$ or $C_i = 16$ and select 9,000 examples with $0 < C_i < 16$ for reflective reasoning training. This selection favors examples that can provide variation in the group comparisons used by GRPO. Each selected example contains the referring expression, the image with its predicted box rendered, the initial proposal, and the reliability target. Figure 6 illustrates this process.

Two-Stage Latent Prompt Connector Data. Connector training uses approximately 300,000 examples from the training splits of RefCOCO, RefCOCO+, and RefCOCOg. Each example provides a referring expression and a ground truth mask for pixel-level supervision.

Joint connector pre-alignment. We use the approximately 300,000 examples to jointly pre-align the two connectors with the SAM prompt space before pathway specialization. The ground truth masks supervise the predicted masks in both pathways.

Semantic alignment data. We select approximately 100,000 examples whose initial predicted boxes have an IoU greater than 0.7 with the ground truth boxes. Their initial proposals provide geometric prompts, which are perturbed during training as described in Appendix A.4. The ground truth masks provide pixel-level supervision.

Semantic relocalization data. We select approximately 50,000 examples whose initial predicted boxes have an IoU below 0.3 with the ground truth boxes. For each example, the image with its predicted box rendered and the reflective reasoning provide the context for constructing new latent prompts. The ground truth mask provides pixel-level supervision.

A.4 Training Strategy

Our training procedure contains a mixed-GRPO stage followed by supervised connector training.

Stage 1: mixed GRPO training. Stage 1 applies Group Relative Policy Optimization (GRPO) to jointly optimize initial proposal generation and reliability assessment. Each prompt produces eight rollouts. Proposal rollouts use the format, box-IoU, box-L1, and point rewards in Eq. 12. Reliability assessment rollouts use format, reasoning-decision consistency, and decision rewards. Training batches mix proposal-generation, reliable-assessment, and unreliable-assessment examples at a 7:3:4 ratio. The 7:3:4 proportion is maintained at the sampler level over successive batches, rather than enforced exactly within each batch. This stage updates the Qwen2.5-VL policy.

Reward definitions. Let $\widehat{g} = (\widehat{b}, \widehat{p}_1, \widehat{p}_2)$ denote a parsed proposal and $g^{\mathrm{GT}} = (b^{\mathrm{GT}}, p_1^{\mathrm{GT}}, p_2^{\mathrm{GT}})$ its ground truth target. Let W and H be the image width and height. For proposal generation, $R_{\mathrm{format}} = 1$ if the response follows the required structure and contains one valid box and two valid points; otherwise, $R_{\mathrm{format}} = 0$. The geometric reward components are

$$
\begin{aligned}
R_{\mathrm{boxIoU}} &= \mathrm{IoU}\Big(\mathrm{round}(\widehat{b}), b^{\mathrm{GT}}\Big), \\
R_{\mathrm{boxL1}} &= \max\left(0,\, 1 - \frac{\|\widehat{b} - b^{\mathrm{GT}}\|_1}{W + H}\right), \\
\pi^\star &= \arg\min_{\pi \in S_2} \sum_{k=1}^{2} \|\widehat{p}_k - p_{\pi(k)}^{\mathrm{GT}}\|_1, \\
R_{\mathrm{point}} &= \mathbf{1}\Big[\widehat{p}_1 \in \widehat{b} \wedge \widehat{p}_2 \in \widehat{b}\Big] \\
&\quad \times \frac{1}{2} \sum_{k=1}^{2} \max\left(0,\, 1 - \frac{\|\widehat{p}_k - p_{\pi^\star(k)}^{\mathrm{GT}}\|_1}{W + H}\right).
\end{aligned}
\tag{22}
$$

Here, IoU computes box IoU after rounding the predicted coordinates and uses inclusive pixel areas. The set S_2 contains the two possible permutations of the points. We select the matching that minimizes the total L1 distance between the predicted and ground-truth points. The point reward averages the matched distance scores, normalized by $W + H$, and is set to zero if either predicted point lies outside the predicted box.

For reliability assessment, $R_{\mathrm{format}} = 1$ if the response contains valid `<think>` and `<answer>` blocks and a binary `decision`; otherwise, it is zero. Let ρ^{think} be the conclusion expressed in the reasoning, $\widehat{\rho}$ the parsed final decision, and ρ^{GT} the reliability target. We define

$$
\begin{aligned}
R_{\mathrm{cons}} &= \mathbf{1}\big[\rho^{\mathrm{think}} = \widehat{\rho}\big], \\
R_{\mathrm{decision}} &= \mathbf{1}\big[\widehat{\rho} = \rho^{\mathrm{GT}}\big].
\end{aligned}
\tag{23}
$$

For the proposal generation reward in Eq. 12, we set the weights of the format, box IoU, box L1, and point rewards to 0.1, 0.3, 0.3, and 0.3, respectively. For the reliability assessment reward, the weights of the format, consistency, and decision rewards are set to 0.1, 1.0, and 1.0, respectively.

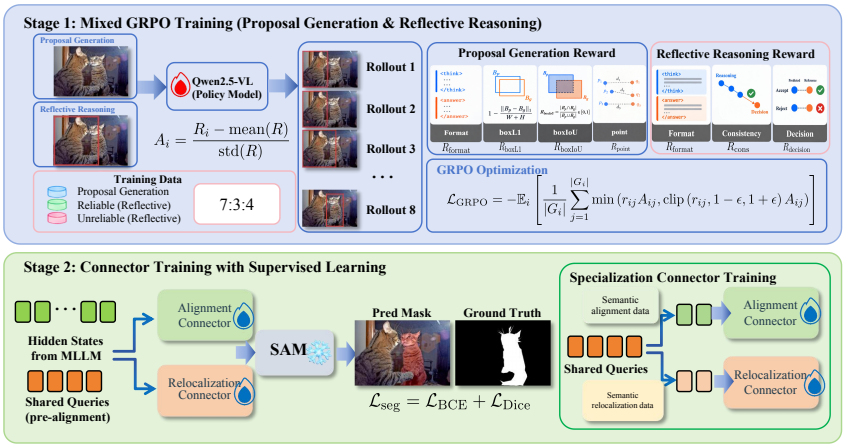

Figure 7: Overview of the two-stage training procedure.

Stage 2: Two-Stage Latent Prompt Connector Training. Stage 2 trains the latent prompt connectors with pixel-level mask supervision, as illustrated in Figure 7. It consists of joint connector pre-alignment followed by separate optimization of the two refinement pathways.

Joint connector pre-alignment. We use the approximately 300,000 training examples described in Appendix A.3 to pre-align both connectors with the SAM prompt space. The two pathways initially share one set of learnable query parameters. The MLLM backbone and SAM remain frozen during pixel-level optimization.

Pathway specialization. The shared learnable query parameters are copied into separate sets for the two pathways. Semantic alignment is optimized on the approximately 100,000 reliable examples in the semantic alignment data, whose initial box IoU exceeds 0.7. Semantic relocalization is optimized on the approximately 50,000 unreliable examples in the semantic relocalization data, whose initial box IoU is below 0.3. Each pathway updates its own connector using ground truth masks as pixel-level targets. Table 5 summarizes the trainable components in Stage 2.

Table 5: Trainable components during Stage 2.

Component	Pre-alignment	Alignment	Relocalization
MLLM backbone and SAM	Frozen	Frozen	Frozen
Alignment connector	Updated	Updated	Frozen
Relocalization connector	Updated	Frozen	Updated
Learnable query token	Shared	Q_{prop}	Q_{refl}
Relocalization LoRA	Disabled	Disabled	Updated

Alignment pathway training. Alignment specialization uses the box and point prompts described in Appendix A.3. We apply progressively stronger geometric perturbations to the box coordinates and point locations over three training phases. Box jitter ratios are 0.10, 0.20, and 0.30. Point jitter ratios are 0.05, 0.10, and 0.15. Phase boundaries occur at 30%, 60%, and 100% of the configured training progress. Coordinates are clipped to the image bounds. Perturbed points are projected onto foreground pixels. This curriculum exposes the alignment connector to varying geometric precision during mask training.

Relocalization pathway training. Relocalization specialization uses the same approximately 50,000 unreliable examples defined as semantic relocalization data in Appendix A.3, whose initial box IoU is below 0.3. Relocalization also updates its pathway-specific LoRA parameters. The LoRA parameters $\Delta\theta_{\text{reloc}}$ use rank 16, scaling 64, and dropout 0.05. They adapt the attention query, key, value, and output projections.

B Supplementary Experimental Details

B.1 Implementation Details

The implementation details are summarized in Table 6. The implementation code is included in the supplementary material submitted with this paper. We group initial proposal generation and reliability assessment under RL, and joint pre-alignment and the two pathway-specific training runs under SFT.

Table 6: Implementation details of LatentRefine-SegZero.

Category	Item	Setting
General	Backbones	Qwen2.5-VL-7B / SAM2-Large
	Hardware	8× NVIDIA A800
RL	Training runs	Initial proposal generation; reliability assessment
	Algorithm	GRPO
	Rollouts per prompt	8
	Learning rate	1×10^{-6}
	Epochs	1
	Rollout temperature	1.0
	Maximum prompt / response length	1,500 / 256
	KL coefficient	0.01
	Sampling ratio (proposal / reliable / unreliable)	$7:3:4$
SFT	Training runs	Joint pre-alignment; alignment; relocalization
	Learning rate	3×10^{-5}
	Epochs	3
	Schedule after warm-up	Cosine
	Optimizer / weight decay	AdamW / 0.01
	Batch size	64

B.2 Grouping of Comparison Methods

The two groups in Table 1 describe how pixel-level supervision is used to update the pretrained MLLM, rather than whether a method uses supervised learning or reinforcement learning at any stage. We place a method in the *pixel-loss updates to pretrained MLLM* group when gradients from a mask-level segmentation loss update the pretrained MLLM parameters. Methods that do not make such updates are placed in the second group, even if they use other forms of supervision to train the MLLM.

LISA, VISA, PixelLM, and GLaMM use supervised segmentation objectives to adapt their MLLM-based segmentation pipelines (Lai et al., 2024; Yan et al., 2024; Ren et al., 2024b; Rasheed et al., 2024). LENS is included in the same group despite also using reinforcement learning: its joint training objective combines GRPO with a supervised segmentation loss, and its joint-training stage unfreezes the MLLM (Zhu et al., 2026). Thus, the presence of reinforcement learning alone does not determine its placement.

The second group collects methods whose reported training does not use a mask-level loss to directly update the pretrained MLLM parameters. SAM-R1, VisionReasoner, and SAM-VETERAN use reward-based optimization to improve reasoning or the generation of spatial prompts for segmentation (Huang et al., 2025; Liu et al., 2026; Du et al., 2026). SegAgent is a distinct case: it fine-tunes the MLLM on simulated annotation trajectories and uses policy-improvement and search-based techniques, so we include it here according to the pixel-loss criterion, not because it is a purely RL-trained or training-free method (Zhu et al., 2025).

LatentRefine-SegZero uses both GRPO and pixel-level supervision. Its pixel-level loss updates the connectors, learnable queries, and pathway-specific LoRA parameters, while the pretrained MLLM backbone weights are kept fixed during this stage. Accordingly, it satisfies the second group's criterion. We repeat its results in both groups solely to facilitate within-group comparisons; the two rows report the same model and checkpoint. This grouping describes the optimization pathways of the compared methods. It does not, by itself, establish that updating pretrained weights causes a loss of general capabilities or poorer out-of-domain performance.

Table 7: Complete ablation of joint connector pre-alignment. Results are reported in cIoU.

	Semantic Alignment			Semantic Relocalization		
Joint Pre-Alignment	**RefCOCO testA**	**RefCOCO+ testA**	**RefCOCOg val**	**RefCOCO testA**	**RefCOCO+ testA**	**RefCOCOg val**
✗	80.8	76.7	73.6	81.7	78.3	74.2
✓	**81.0**	**77.0**	**74.0**	**82.3**	**79.1**	**75.3**

Complete joint connector pre-alignment ablation. Table 7 reports the complete three-benchmark comparison, including RefCOCO testA in addition to the two benchmarks shown in Table 3. Joint pre-alignment improves both pathways on every split, with larger gains for semantic relocalization.

B.3 Evaluation Metrics

Let $\widehat{M}_i$ be the binarized final prediction M_i and let M_i^{GT} be its reference mask. Define $a_i = |\widehat{M}_i \cap M_i^{\mathrm{GT}}|$ and $u_i = |\widehat{M}_i \cup M_i^{\mathrm{GT}}|$. We compute

$$\mathrm{cIoU} = \frac{\sum_i a_i}{\sum_i u_i}. \tag{24}$$

This metric aggregates intersection and union areas across the evaluation set. We assign cIoU a value of zero when the total union is zero.

Predicted masks are binarized using foreground probability strictly greater than 0.5. The final box is the tight extent of foreground pixels; an empty prediction returns $[0, 0, 0, 0]$. The RefAdv and PR-Bench grounding protocols evaluate this mask-derived box against the reference box, measuring localization after latent refinement. Let N_b denote the number of evaluated examples with reference boxes. For box IoU $v_i = \mathrm{IoU}(\widehat{b}_i, b_i^{\mathrm{GT}})$, where $\widehat{b}_i$ encloses $\widehat{M}_i$, the metrics are

$$\mathrm{BBoxIoU} = \frac{1}{N_b}\sum_i v_i, \qquad \mathrm{BoxAcc@0.5} = \frac{1}{N_b}\sum_i \mathbf{1}[v_i > 0.5]. \tag{25}$$

Box Acc@0.5 uses a strict IoU threshold, and box overlap is computed with inclusive pixel areas.

B.4 Inference Efficiency

We compare inference efficiency on RefCOCO testA using the average numbers of MLLM and SAM calls per example and the wall-clock latency under the Transformers and vLLM backends. The results are reported in Table 8.

Table 8: Inference efficiency on RefCOCO testA. Calls are averaged per example. † denotes results reported by the source papers.

Method	Avg. MLLM Calls	Avg. SAM Calls	Transformers Time (s)	vLLM Time (s)
Qwen2.5-VL+SAM2†	1	1	3.43	1.54
SegAgent† (Zhu et al., 2025)	7	7	8.95	–
SAM-Veteran† (Du et al., 2026)	2.21	–	5.09	2.47
LatentRefine-SegZero	2	1	5.83	2.72

Qwen2.5-VL+SAM2 completes inference with one MLLM call and one SAM call, whereas SegAgent uses a fixed seven-step MLLM inference pipeline. SAM-Veteran reduces this cost through adaptive termination, requiring 2.21 MLLM calls on average under vLLM. LatentRefine-SegZero uses exactly two MLLM generations, one for the initial proposal and one for reliability assessment, followed by a single SAM decoding using the prompt produced by the selected pathway. This structure supports reliability assessment and adaptive refinement with slightly fewer MLLM calls than SAM-Veteran. Replacing the Transformers backend with vLLM reduces our latency from 5.83 to

2.72 seconds per example, bringing it close to SAM-Veteran. Overall, our method incurs comparable inference cost to prior refinement methods while achieving an effective balance between in-domain referring segmentation performance and out-of-domain generalization.

B.5 Additional Visualizations

Figure 8 presents additional examples of the semantic alignment and semantic relocalization pathways.

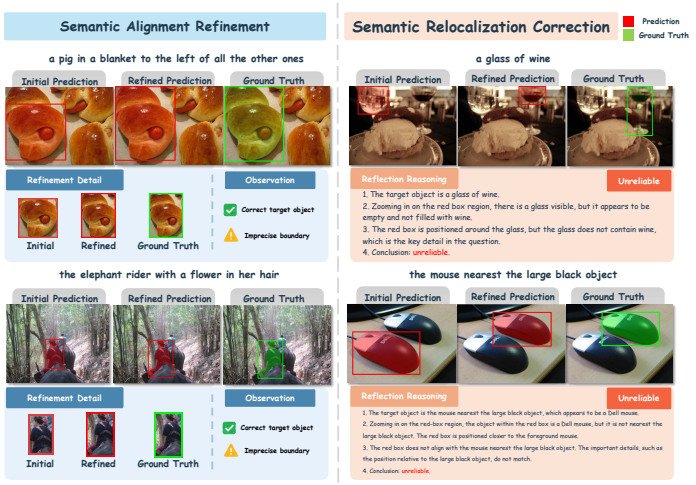

Figure 8: Additional qualitative examples of the semantic alignment pathway (left) and semantic relocalization pathway (right).