RECAGENT: REINFORCING REFERRING EXPRESSION COMPREHENSION THROUGH AGENTIC TOOL-AUGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Referring expression comprehension (REC) requires localizing the image region described by a free-form expression, and multimodal large language models (MLLMs) provide a flexible interface for this task while remaining prone to three recurring failures: they bind compositional constraints inconsistently, adopt a candidate before sufficient visual evidence is gathered, and translate a correct semantic decision into an imprecise box. We present **RecAgent**, an agentic framework that recasts REC as active evidence acquisition and optimizes it with tool-augmented reinforcement learning. RecAgent is trained in two stages. A supervised cold start on *RefCoT-4K*, a self-generated corpus of executable chain-of-thought trajectories, equips the policy to decompose an expression, hypothesize candidate referents, and verify each hypothesis. At inference, the policy queries a diagnostic tool library comprising an Expression Decomposer, a Visual Grounder, an Attribute Matcher, and a Spatial Reasoner, selecting which tools to call and in what order according to the ambiguity at hand. To make this behavior learnable, we propose a verification-gated composite reward that couples continuous IoU-weighted localization feedback with format validity, tool-usage quality, and a trajectory-efficiency term activated on verified-correct answers, rewarding trajectories that are short, correct, and well formed. Evaluations across standard (RefCOCO/+/g) and challenging generalization (gRefCOCO, Ref-L4) benchmarks validates our SOTA performance and the efficiency of active visual reasoning.

1 INTRODUCTION

Referring Expression Comprehension (REC) aims to localize the visual instance specified by a natural-language referring expression, typically by predicting its corresponding bounding box in an image (Chen et al., 2025a; Park et al., 2026; Yu et al., 2025). Unlike conventional object detection that identifies instances from predefined semantic categories, REC requires distinguishing a particular referent from potentially similar objects according to free-form linguistic descriptions involving appearance, attributes, spatial configurations, and inter-object relations. Early approaches typically rank candidate regions through cross-modal matching (Mao et al., 2016; Yu et al., 2018), while subsequent approaches progressively simplify this pipeline by jointly modeling image-text correspondence and directly predicting spatial locations (Deng et al., 2021; Zhu et al., 2022a). With the emergence of multimodal foundation models, referring and grounding have further been incorporated into general-purpose vision-language architectures capable of processing open-ended descriptions and spatial coordinates within a unified interface (Han et al., 2024; Wang et al., 2025c).

Recent advances in Vision-Language Models (VLMs) (Radford et al., 2021; Hurst et al., 2024; Liu et al., 2024a; Comanici et al., 2025; Wang et al., 2025a) provide a promising new paradigm for REC. By combining powerful visual representations with large-scale language knowledge and multimodal reasoning capabilities, modern VLMs like the Qwen-VL series (Bai et al., 2025b;a) can interpret complex referring expressions, recognize diverse visual concepts, and directly produce spatial predictions without relying on specialized grounding pipelines. Their broad semantic knowledge further enables flexible understanding of targets specified through combinations of object categories, visual attributes, and relational descriptions. These capabilities make VLMs particularly appealing as general-purpose models for referring expression grounding.

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.

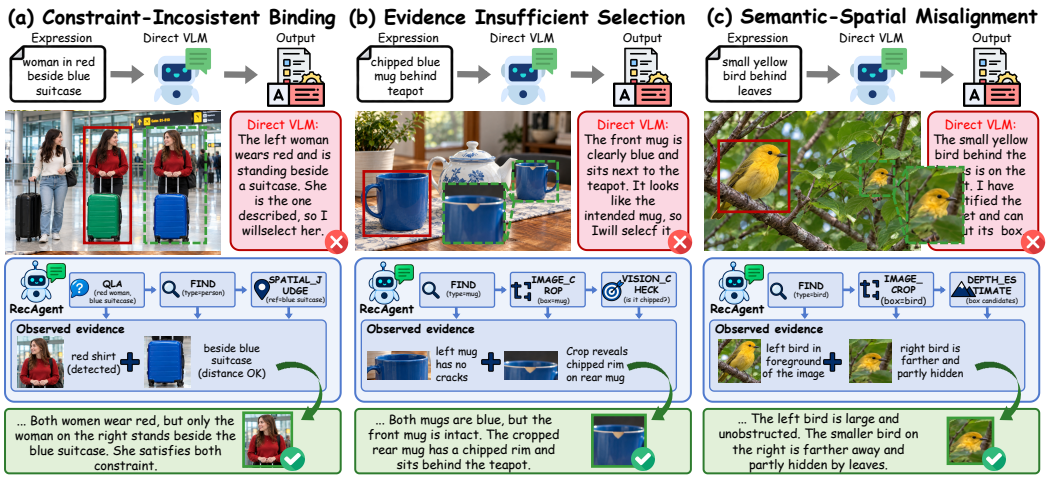


Figure 1: Three failure modes of direct vision-language model (VLM) grounding in referring expression comprehension and how RecAgent resolves them. In each illustrative trajectory, RecAgent calls tools to acquire discriminative evidence and correct the localization. Red boxes show direct VLM predictions; green dashed boxes show the corrected targets.

Despite their capabilities, directly applying VLMs to REC still exposes three fundamental limitations as shown in Fig.1. ❶ **Constraint-Inconsistent Binding.** Although VLMs can recognize candidate objects and understand individual linguistic concepts, they may fail to correctly associate multiple attributes, entities, and relational modifiers with their intended referents. This issue becomes particularly pronounced for expressions involving nested relations or multiple visually similar instances, where understanding individual concepts does not necessarily imply correctly resolving their compositional dependencies. ❷ **Evidence-Insufficient Selection.** Even when the referring expression is correctly understood, VLMs may prematurely commit to visually salient or semantically plausible instances without verifying whether all discriminative constraints are satisfied. Fine-grained appearance, local attributes, spatial relations, and relative depth can therefore be overlooked, causing the model to select a plausible candidate rather than the visually verified referent. ❸ **Semantic-Spatial Misalignment.** Correctly identifying the intended referent at the semantic level does not necessarily translate into accurate spatial localization. This discrepancy becomes particularly pronounced for small, partially occluded, or densely distributed instances, where subtle local and relational evidence is required to precisely determine the target region.

To address these challenges, we argue that an effective REC agent requires three synergistic capabilities. ❶ **Constraint-Guided Analysis**, which explicitly organizes complex expressions into target entities, attributes, quantities, and relational constraints, enabling the model to establish a structured representation of the conditions that the intended referent should satisfy. ❷ **Evidence-Oriented Verification**, which actively acquires complementary evidence to retrieve, compare, and verify competing referents through local appearance, attribute compatibility, spatial relations, and relative depth, rather than committing to a candidate from a single-pass observation. ❸ **Verification-Guided Grounding**, which integrates the accumulated verification results to eliminate inconsistent candidates, refine the target region, and determine when sufficient evidence has been obtained for reliable localization. Together, these capabilities reformulate REC from passive coordinate prediction into an active process of referential analysis, candidate verification, and spatial grounding.

Therefore, we introduce **RecAgent**, a novel framework that integrates active evidence acquisition, **tool-augmented reinforcement learning (RL)** with a verification-gated reward for robust referring expression comprehension. Specifically, to bridge the constraint-inconsistent gap, our method first constructs the ***RefCoT-4K* dataset** for **supervised fine-tuning (SFT)**. *RefCoT-4K* is a high-quality chain-of-thought (CoT) dataset that explicitly simulates the active evidence acquisition process by structuring the logical progression from expression decomposition to candidate verification, teaching the model to identify missing discriminative constraints and execute purposeful tool calls rather than relying on superficial semantic correlations.

Subsequently, we address evidence-insufficient selection and semantic-spatial misalignment with a tool-augmented GRPO stage, in which the agent dynamically calls a diagnostic tool library to acquire the missing visual evidence. We further propose a verification-gated composite reward over fine-grained localization, tool-flow quality, and trajectory efficiency, guiding the agent to decide *how*, *when*, and *why* to use external tools for genuinely grounded predictions. Our contributions are:

- **Tool-Augmented Agentic RL.** We introduce a novel framework that unifies active evidence acquisition with tool-augmentation RL. By enforcing a structured tool-augmented paradigm in SFT stage and a verification-gated reward in RL stage, it enables reliable reasoning for robust REC.
- **High-Quality CoT Dataset.** We construct *RefCoT-4K*, a large-scale reasoning dataset that explicitly structures the logical progression from linguistic parsing to candidate verification, effectively bridging the gap between passive perception and active grounding.
- **Multi-Tool Integration.** We design a multimodal tool library integrating expression decomposition, candidate retrieval, attribute matching, and spatial reasoning, which dynamically augments reasoning to resolve both fine-grained attribute ambiguity and semantic-spatial misalignment.
- **Empirical Effectiveness.** Experiments on five standard and challenging benchmarks evaluate localization accuracy, generalization, component contributions, and the accuracy–efficiency trade-off of our agentic, tool-augmented approach.

2 Related Work (Extended Ver. in Appx. C)

Referring Expression Comprehension. REC aims to localize the specific image region described by a natural language expression (Mao et al., 2016; Yu et al., 2016). Existing methods can be broadly divided into two-stage (Yu et al., 2018; Liu et al., 2019; Yang et al., 2019) and one-stage (Yang et al., 2022; Zhu et al., 2022b; Deng et al., 2023) paradigms, the latter enabling end-to-end box regression conditioned on language. Transformer-based grounding models such as MDETR (Kamath et al., 2021) further unify detection and language understanding via cross-modal attention. With vision-language pre-training, open-set detectors (Li et al., 2022a; Liu et al., 2024b) and MLLM-based grounding models (You et al., 2024; Lai et al., 2024; Xia et al., 2024) have substantially improved zero-shot generalization. In contrast, RecAgent introduces an interactive decision-making architecture with tool-augmented RL for robust referring expression comprehension.

Multimodal LLMs Reasoning. RL-based post-training, as exemplified by OpenAI-o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), has significantly enhanced LLM reasoning. These paradigms are extended to MLLMs for mathematical VQA (Peng et al., 2025), image segmentation (Liu et al., 2025), and video understanding (Feng et al., 2025b). However, when applied to REC, models often compress localization into a single-pass mapping, suffering from evidence-insufficient selection and long-sequence hallucination (Chen et al., 2025b) due to a lack of explicit verification. To address this, we propose multimodal CoT reasoning via tool-augmented RL, explicitly reformulating REC as an active visual evidence acquisition process to reduce spatial hallucinations.

Tool-Augmented Agentic System. External tools have proven effective in enhancing multimodal reasoning. Early works introduce visual evidence into CoT (Sun et al., 2025), while LLaVA-Plus (Liu et al., 2023b) and VPD (Hu et al., 2024) pioneer training strategies and program-derived data for tool use. Recent methods like TACO (Ma et al., 2024) and PyVision (Zhao et al., 2025) further expand tool use with RL. Despite the success of general agent frameworks like ReAct (Yao et al., 2022) in language tasks, existing visual approaches primarily rely on static tool pipelines, failing to dynamically adapt to complex, open-vocabulary spatial constraints in REC. Our framework addresses this by modeling REC as a tool-conditioned framework, explicitly acquiring discriminative evidence and dynamically balancing tool efficiency through a verification-gated reward.

3 Methodology

Overview. We propose RecAgent, a novel framework that integrates active evidence acquisition with tool-augmented RL for robust REC, as shown in Fig. 2. Section 3.1 presents the problem formulation and our two-stage workflow. Section 3.2 details the **Training Data Construction**, which synthesizes structured reasoning chains essential for executable tool invocation and evidence verification. Section 3.3 describes the **Agentic Supervised Fine-tuning**, where the Qwen3.5-VL base model is fine-tuned to initialize the agent's policy for tool invocation. Section 3.4 outlines the **Agen-**

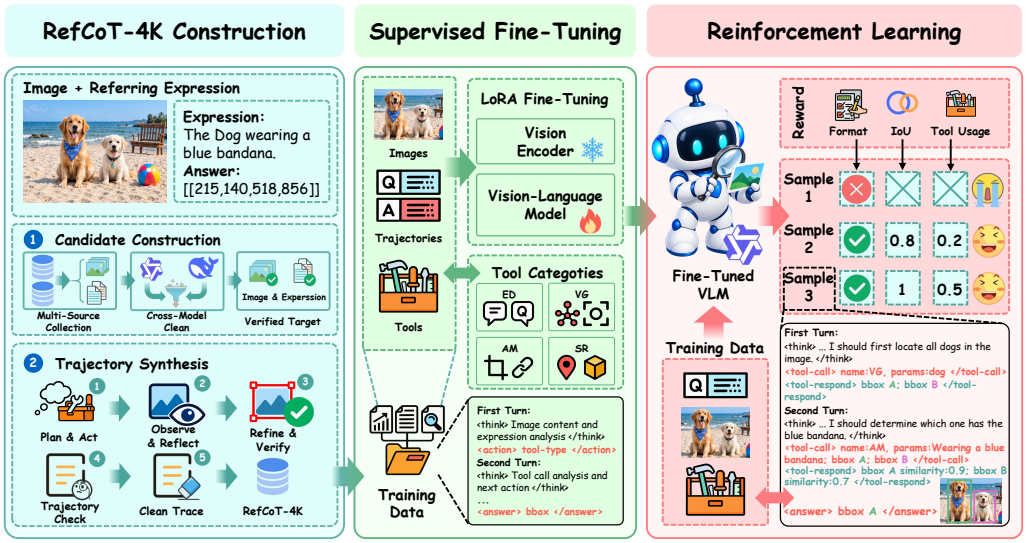


Figure 2: Overview of the proposed **RecAgent** framework. RecAgent integrates active evidence acquisition with tool-augmented reinforcement learning for robust referring expression comprehension. The pipeline consists of three stages: (i) **Training Data Construction**, which synthesizes structured reasoning chains for executable tool invocation and evidence verification; (ii) **Agentic Supervised Fine-tuning**, which initializes the agent's tool-invocation policy; and (iii) **Agentic Reinforcement Learning**, which optimizes the policy via GRPO with a verification-gated composite reward balancing fine-grained localization and trajectory efficiency.

tic Reinforcement Learning, which employs the GRPO algorithm to optimize agent policy via a verification-gated composite reward that balances fine-grained localization and trajectory efficiency.

3.1 Problem Formulation & Workflow Design

The goal in referring expression comprehension is to accurately localize the visual instance specified by a natural-language expression. Given an image I and a referring expression Q, the model outputs a set of target bounding boxes $\mathcal{B} = \{b_k\}_{k=1}^{K}$ that correspond to the intended referents, where b_k denotes the k-th predicted box and K is the number of referents described by Q. Unlike traditional methods that rely on a passive single-pass mapping $f_\theta(I, Q) \mapsto \hat{\mathcal{B}}$, REC demands active evidence acquisition and constraint verification. Formally, the model must accurately parse the complex linguistic constraints from Q, identify potential referents in I, and logically verify their fine-grained attributes and spatial relations before projecting the final localization. To achieve this, we formulate REC as a multi-turn decision process in which the agent interleaves reasoning, tool invocation, and evidence verification, instead of following a fixed tool sequence.

Adaptive Tool-Augmented Reasoning. Given an image I and expression Q, the agent maintains an evolving reasoning context that accumulates the constraints parsed so far, the candidate boxes under consideration, and the evidence returned by previous tool calls. Let H_t denote this context at turn t, with $H_0 = (I, Q)$. At each turn, the policy π_θ generates a response $a_t \sim \pi_\theta(\cdot \mid H_{t-1})$, which is either a tool request specifying a set of calls $C_t \subseteq \mathcal{T}$ or the final answer. A tool request is executed and its outputs are returned as an observation o_t and appended to the context:

$$o_t = \mathrm{Exec}(C_t, I, H_{t-1}), \qquad H_t = H_{t-1} \oplus (a_t, o_t), \tag{1}$$

where $\oplus$ denotes concatenation. When a_t is the final answer, the episode terminates at turn $T = t$ and the box set emitted in a_T defines the prediction $\hat{\mathcal{B}}$. Since only the responses a_t are produced by the policy while the observations o_t are determined by the environment, a trajectory is the sequence of responses $\tau = (a_1, o_1, \ldots, a_T)$, whose likelihood factorizes as:

$$\pi_\theta(\tau \mid I, Q) = \prod_{t=1}^{T} \pi_\theta(a_t \mid H_{t-1}). \tag{2}$$

Crucially, neither the number nor the order of tool calls is pre-specified. The agent decides which constraints remain unresolved and which tool can resolve them, may issue several calls within the same turn, and can re-invoke a tool when earlier evidence is insufficient or conflicting, yielding a reflect and retry behavior. Expression decomposition, candidate discovery, attribute verification, and final selection therefore emerge as functional phases rather than a hard-coded template.

Tool Categories. As shown in Fig. 2, our framework integrates a diverse diagnostic tool library to facilitate a multi-stage reasoning process for accurate REC. ❶ **Expression Decomposer (ED)**: to resolve constraint-inconsistent binding in complex queries, this tool decomposes lengthy or compound expressions into structured entities, attributes, quantities, and relational constraints for focused location. ❷ **Visual Grounder (VG)**: to address evidence-insufficient selection, this tool acts as an open-vocabulary detector that retrieves text-conditioned candidate bounding boxes with confidence scores for initial recall. ❸ **Attribute Matcher (AM)**: to mitigate cross-modal hallucinations, this tool crops and magnifies specific candidate sub-regions to perform fine-grained visual verification (e.g., texture, color, or object states). ❹ **Spatial Reasoner (SR)**: to address semantic-spatial misalignment, this tool returns verifiable geometric or relative depth relationships between candidate pairs, ensuring spatial consistency with the referring expression. Details are provided in Appendix D.

3.2 Training Data Construction

A fundamental challenge when applying existing MLLMs to REC tasks is the absence of explicit reasoning chains. Standard pipelines provide static labels but lack intermediate cognitive steps: **what** tool to use, **which** evidence to acquire, and **how** plausible targets are verified. Moreover, recent re-evaluations of REC benchmarks have shown that existing annotations can contain ambiguous expressions, incomplete target sets, and incorrect correspondences between bounding boxes and expressions (Chen et al., 2025a). Such noise is particularly harmful for reasoning supervision. If a model is asked to explain an erroneous annotation, it may produce a coherent-looking but factually unsupported tool-use trace, thereby turning annotation errors into spurious decision patterns. To address these problems, we devise a process for synthesizing high-fidelity structured reasoning chains for SFT, thereby bridging the gap between perception and grounding during inference. which creates explicit examples to subtly resolve the above-mentioned cognitive steps, thus effectively bridge the gap between perception and grounding during inference.

Candidates Construction. We construct *RefCoT-4K* in two independent stages, each with a single responsibility. In the first stage, we draw a source-balanced pool of 4,000 REC examples that spans concise appearance-based expressions, long compositional descriptions, small objects, and generalized cases with multiple or no targets. We then verify each proposed box with two independent vision-language models: Qwen3.8-27B performs a first pass, and DeepSeek-v4.1-flash re-examines every case it rejects. An annotation is discarded only when both models reject it, so this screening retains 3,579 (image,expression,b-box) triples whose targets can be trusted as reasoning endpoint.

Trajectories Synthesis. In the second stage, we let Qwen3.8-27B (Bai et al., 2025a) solve each verified prompt with the diagnostic tools under soft tool-use guidance. Each resulting trajectory follows the five phases of the Fig. 2. **(1) Plan & Act**: the model first states which evidence is still missing and issues a tool call to obtain it. **(2) Observe & Reflect**: it then inspects the returned observation, and whenever the evidence is insufficient or conflicting, reflects on the unmet constraint. **(3) Refine & Verify**: guided by this reflection, it retries with a more targeted call and checks the accumulated evidence against the constraint. **(4) Trajectory Check**: Qwen-VL-Max then acts as an expert validator, reading the produced trajectory, checking that each step is supported by the tool observations, and verifying whether the final boxes agree with the grounding target, so that trajectories which are unfaithful or incorrect are rejected. **(5) Clean Trace**: for the retained trajectories, we remove the guidance turns and keep only the model's own decisions and the real tool observations, converting them into SFT conversations that form *RefCoT-4K*. Source statistics, filtering criteria, split details, and generation settings are given in Appendix A.

3.3 Agentic Supervised Fine-Tuning

After constructing the training data, we introduce a **cold-start phase** that performs supervised fine-tuning (SFT) before reinforcement learning (RL). Motivated by the R1-Zero recipe (Guo et al., 2025), we first attempt to train our method with direct RL from the uninitialized policy. However,

the rollouts show a progressive decline in the frequency of tool invocation, and the agent ultimately collapses into a no-tool, single-pass predictor. This behavior likely arises from a known pitfall of cold-start RL. Without any supervised signal, the policy rarely produces valid tool calls and instead emits unparseable or empty trajectories, which keeps the verifiable reward sparse and the group-normalized advantages too weak to reinforce tool use (Dong et al., 2025). Since prior work shows that an executable tool-use format must be established by SFT before RL refine it (Feng et al., 2025a), we adopt a supervised cold start to provide an effective initialization for the RL stage.

Specifically, we formalize each training instance as $\mathcal{W} = (\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{O})$, where $\mathcal{X}$ denotes the input modality, $\mathcal{Y}$ represents task instructions, $\mathcal{Z} = \{z_1, \ldots, z_T\}$ captures reasoning steps, and $\mathcal{O}$ is the target output. The objective minimizes the negative log-likelihood of the reasoning process:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{\mathcal{W}\sim\mathcal{D}}\left[\sum_{t=1}^{T} \log p_\theta(z_t \mid \mathcal{X}, \mathcal{Y}, z_{<t})\right], \tag{3}$$

where $p_\theta(\cdot)$ denotes conditional probability distribution. This formulation aligns the generated reasoning chains with both the task instructions and ground-truth outputs during the cold-start stage.

3.4 Agentic Reinforcement Learning

Group Relative Policy Optimization (GRPO). We adopt the GRPO algorithm (Shao et al., 2024) for policy optimization. GRPO innovatively employs a groupwise comparison framework to evaluate candidate responses. Specifically, for each query q paired with its ground-truth solution a from dataset D, the algorithm generates a set of rollout trajectories $\{o_1, o_2, \ldots, o_G\}$ based on the previous policy $\pi_{\theta_{\text{old}}}$. The policy π_θ is then refined through the optimization of this objective function:

$$\begin{aligned}\mathcal{L}_{GRPO}(\theta) = -\mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G} \\ \left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon\right)A_i\right) - \beta\mathbb{D}_{KL}(\pi_\theta\|\pi_{ref})\right)\Bigg],\end{aligned} \tag{4}$$

$$\mathbb{D}_{KL}(\pi_\theta||\pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \tag{5}$$

where β is adopted to balance the trade-off between exploration and stability during optimization. Then the advantage estimator A_i is calculated using normalized rewards from the trajectory group:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}. \tag{6}$$

Each trajectory o_i receives a composite reward through rule-based verification designed as follows.

Reward Design. Effective reward functions for agentic REC must balance fine-grained localization, structural executability, tool-flow quality, and trajectory efficiency, while preventing common RL shortcuts such as empty answers or premature stopping. We therefore construct a composite, verifiable reward with four components: localization reward R_{loca}, format-validity reward R_{format}, tool-usage quality reward R_{tool}, and verification-gated efficiency reward R_{verf}.

The **format-validity reward** checks whether the nested tags, `<think>`, `<tool-call>`, and `<answer>`, are correctly matched, so that the reasoning, the tool call, and the final answer can all be parsed. It also validates that the content inside `<answer>` is well-formed JSON, and assigns a reward of 1 only when both checks pass and 0 otherwise.

For the **tool-usage reward**, each GRPO group contains a no-tool baseline τ_0. Let $\mathcal{S}$ denote the set of exploratory trajectories whose tool calls are all valid and successful, and let $M = |\mathcal{S}|$. A trajectory containing any invalid or failed tool call is excluded from $\mathcal{S}$ and receives zero tool-usage reward.For each $\tau_i \in \mathcal{S}$, we define its localization improvement over the no-tool baseline as:

$$\Delta_i = R_{\text{loca}}(\tau_i) - R_{\text{loca}}(\tau_0). \tag{7}$$

To encourage diverse tool-use strategies, let U_i denote the set of distinct tools that provide valid evidence in trajectory τ_i. We measure the diversity of τ_i relative to the other successful exploratory

trajectories using the average Jaccard distance:

$$D_i = \begin{cases} \dfrac{1}{M-1} \displaystyle\sum_{\substack{\tau_j \in \mathcal{S} \\ j \neq i}} \left(1 - \dfrac{|U_i \cap U_j|}{|U_i \cup U_j|}\right), & M > 1, \\ 0, & M \leq 1. \end{cases} \tag{8}$$

The resulting tool-usage reward is:

$$R_{\text{tool}}(\tau_i) = \begin{cases} \Delta_i + \eta\, \mathbb{I}[\Delta_i > \delta]\, \min(\Delta_i, \Delta_{\max}) D_i, & \tau_i \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Here, η (default 0.2) controls the diversity bonus, δ (default 0.1) filters out small improvements that may be caused by rollout noise, and $\Delta_{\max}$ (default 0.6) limits the influence of unusually large localization gains. Therefore, tool-use diversity is rewarded only when the trajectory improves localization over the shared no-tool baseline.

Directly penalizing trajectory length globally can incentivize empty answers or premature stopping. We therefore introduce a **verification-gated efficiency reward**. Let L_i denote the number of model-generated tokens, excluding tool observations, in trajectory τ_i. We activate the length reward only when the final answer is format-valid and correctly grounded:

$$R_{\text{verf}}(\tau_i) = \begin{cases} \text{RankScore}(-L_i), & \text{if } R_{\text{format}}(\tau_i) = 1 \wedge \text{Acc@0.5}(\tau_i) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The ranking is computed only among correct trajectories in the same rollout group. This verification-gated design prefers "short and correct" trajectories over "verbose but correct" ones, without rewarding invalid, empty, or incorrectly terminated outputs. The final reward is formally defined as:

$$R(\tau, B^*) = R_{\text{loca}} + \lambda_f R_{\text{format}} + \lambda_t R_{\text{tool}} + \lambda_v R_{\text{verf}}, \tag{11}$$

where $\lambda_f, \lambda_t, \lambda_v$ respectively balance format validity, tool-flow quality, verified trajectory efficiency.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate RecAgent on three standard REC benchmarks and two challenging extensions. RefCOCO (Yu et al., 2016) contains descriptions of people and objects; RefCOCO+ (Yu et al., 2016) excludes absolute location words, placing greater emphasis on appearance and relations. RefCOCOg (Mao et al., 2016) features longer, more compositional expressions. We report val, testA, and testB for RefCOCO and RefCOCO+, and the Google validation (*val-g*) and UMD validation/test (*val-u*/*test-u*) splits for RefCOCOg. To assess grounding beyond a single referent, gRefCOCO (He et al., 2023) includes single-, multi-, and no-target expressions, evaluated on testA and testB. Ref-L4 (Chen et al., 2025a) evaluates long descriptions and precise localization on its val and test splits. SFT and RL use only designated training or adaptation splits; evaluated test examples are excluded from optimization.

Evaluation metrics. For RefCOCO, RefCOCO+, and RefCOCOg, we report Acc@0.5: the percentage of expressions whose predicted box matches the ground truth at IoU ≥ 0.5. Following the gRefCOCO protocol, we report target-present precision and no-target accuracy (N-acc) separately. A target-present prediction is correct only when all and only the annotated targets are matched one-to-one at IoU ≥ 0.5; a no-target prediction is correct only when the output is empty. For Ref-L4, we additionally report Acc@0.75 and Acc@0.9 to assess tighter box alignment, together with mAcc averaged over IoU thresholds from 0.50 to 0.95 in steps of 0.05.

Implementation details. We initialize RecAgent from Qwen3.5-VL-4B. In the cold-start stage, we train a LoRA adapter for three epochs with rank 32, scaling factor 64, dropout 0.05, and a learning rate of 5×10^{-5}. The per-device batch size is 16. After merging the adapter, we perform full-parameter GRPO for two epochs while freezing the vision tower. The actor learning rate is 1×10^{-6}, the KL coefficient is 0.01, and the entropy coefficient is 0.001. Each prompt uses a group of eight rollouts sampled with temperature 1.2 and top-$p = 1.0$; an agent may take at most eight assistant turns. Training is conducted on 4 NVIDIA H100 (80GB) GPUs. More details are in Appendix B.1.

Table 1: Comparison (%) on standard REC benchmarks (RefCOCO, RefCOCO+ and RefCOCOg). All values are reported as Acc@0.5 (%). Note that best and second best results are highlighted.

Method	Venue & Year	RefCOCO			RefCOCO+			RefCOCOg		
		val	testA	testB	val	testA	testB	val-g	val-u	test-u
TransVG (Deng et al., 2021)	ICCV 21	81.02	82.72	78.35	64.82	70.70	56.94	67.02	68.67	67.73
MDETR (Kamath et al., 2021)	ICCV 21	86.57	89.58	81.41	79.52	84.09	70.62	—	81.64	80.89
VLTVG (Li et al., 2022b)	CVPR 22	84.77	87.24	80.49	74.19	78.93	65.17	72.98	76.04	74.18
QRNet (Ye et al., 2022)	CVPR 22	84.01	85.85	82.34	72.94	76.17	63.81	71.89	73.03	72.52
OFA-base (Wang et al., 2022)	ICML 22	88.48	90.67	83.30	81.39	87.15	74.29	—	82.29	82.31
MaPPER (Liu et al., 2024c)	EMNLP 24	86.03	88.90	81.19	74.92	81.12	65.68	74.60	76.32	75.81
MDETR + VIE-DM (Ke et al., 2025)	ICLR 25	90.53	92.15	85.95	82.84	86.83	73.35	—	83.80	83.56
OFA-base + VIE-DM (Ke et al., 2025)	ICLR 25	90.53	91.75	85.63	84.08	89.65	76.20	—	84.27	83.98
RecAgent (Ours)	-	91.53	92.73	84.04	85.26	88.24	76.92	85.59	85.47	86.53

Table 2: Evaluation (%) in challenging settings: generalized grounding (gRefCOCO) and long-expression, strict-IoU grounding (Ref-L4 val+test). Pr denotes Pr@(F1$=1$, IoU≥ 0.5) computed over all expressions, N-acc denotes no-target accuracy. The best and second-best are highlighted.

Method	Venue & Year	gRefCOCO				Ref-L4			
		Pr (A)	N-acc (A)	Pr (B)	N-acc (B)	Acc@0.5	Acc@0.75	Acc@0.9	mAcc
OFA-Large (Wang et al., 2022)	ICML 22	-	-	-	-	72.53	62.31	45.02	59.17
KOSMOS-2 (Peng et al., 2023)	arXiv 23	22.06	-	15.96	-	48.53	38.34	17.54	34.72
Qwen-VL-Chat (Bai et al., 2023)	arXiv 23	-	-	-	-	73.80	58.05	37.16	55.94
SPHINX-v2-1k (Lin et al., 2023)	arXiv 23	-	-	-	-	81.31	70.49	46.59	65.39
UNINEXT-R50 (Yan et al., 2023)	CVPR 23	46.40	49.30	42.90	48.20	-	-	-	-
CogVLM-Grounding (Wang et al., 2024)	ACL 24	-	-	-	-	81.70	70.77	48.35	66.09
Ferret-7B (You et al., 2024)	ICLR 24	49.50	45.20	43.50	43.80	57.54	42.44	21.01	40.29
SimVG-DB (Dai et al., 2024)	EMNLP 24	64.60	57.20	54.80	57.20	-	-	-	-
HieA2G-R101 (Wang et al., 2025b)	AAAI 25	66.00	60.10	56.50	56.00	-	-	-	-
RecAgent (Ours)	-	68.53	72.45	63.26	57.62	85.38	71.76	47.50	67.24

4.2 Comparison with SOTA Methods

Standard REC Benchmarks. Table 1 compares RecAgent with specialized grounding models on RefCOCO, RefCOCO+, and RefCOCOg. Averaging the reported splits, RecAgent obtains **89.43**, **83.47**, and **85.86** accuracy, respectively. On matched splits, it is competitive with the strongest RefCOCO baseline (-0.11 points) and improves the best RefCOCO+ and RefCOCOg results by **0.16** and **1.88** points, respectively. The largest gain is on **RefCOCOg**, whose longer, compositional expressions place greater demands on attribute and relational reasoning.

Generalized Scenarios. To assess our generalization, we evaluate our method on complex scenario benchmarks including gRefCOCO and Ref-L4 in Tab. 2. Traditional methods typically struggle in these scenarios due to their inability to handle variable target quantities, long-tail descriptive constraints, and cross-image instance discrimination. On the gRefCOCO dataset, which requires multi-target and no-target grounding, RecAgent consistently surpasses all previous methods, achieving **68.53%** and **63.26%** on testA and testB. This outperformance validates our *constraint-limited analysis*, allowing the agent to systematically parse quantities and relational dependencies rather than prematurely committing to a single salient object. Furthermore, on the Ref-L4 benchmark characterized by long expressions and strict high-IoU thresholds, our method achieves a **85.38%** Acc@0.5 and a substantial lead in the mAcc metric (**67.24%**). More qualitative examples are in Appendix B.2.

4.3 Ablation Studies

Training Stages. Rows *(a)-(d)* of Tab. 3 compare the pretrained model, SFT, SFT followed by RL without tool access, and the full RecAgent. SFT improves average accuracy by **7.64** points, showing that the cold start establishes an executable policy. RL provides a further **1.48**-point gain, while enabling tools during both optimization and inference adds **2.99** points. Together, these results separate the benefit of domain-specific training, and test the need for the proposed two-stage optimization.

Tool-Usage. In rows *(e)-(h)* of Tab. 3, we remove one tool at a time from the full framework. Removing ED, VG, AM, and SR reduces average accuracy by **0.72**, **1.71**, **1.38**, and **1.01** points, respectively. More importantly, the drops differ across benchmarks: ED and SR are most useful

Table 3: Ablations (%) on RefCOCO/+/g datasets. We adopt the mean Acc@0.5 in Table 1.

Method	RefCOCO	RefCOCO+	RefCOCOg	Tools	RefCOCO	RefCOCO+	RefCOCOg
(a) Qwen3.5-4B	76.96	71.84	73.62	*(e)* w/o. ED	**88.92**	**82.96**	**84.71**
(b) 4B+SFT	85.74	78.26	81.35	*(f)* w/o. VG	87.81	81.62	84.19
(c) 4B+SFT+RL	87.02	80.11	82.67	*(g)* w/o. AM	88.21	81.74	84.67
(d) 4B+SFT+RL+Tools	**89.43**	**83.47**	**85.86**	*(h)* w/o. SR	88.74	82.62	84.38
Reward Term	RefCOCO	RefCOCO+	RefCOCOg	Num of Gen.	RefCOCO	RefCOCO+	RefCOCOg
(i) $R_{\text{loca}} + R_{\text{format}}$	87.56	81.12	83.72	*(m) Num = 2*	87.94	81.56	84.02
(j) (i) $+R_{\text{tool}}$	88.71	82.48	84.92	*(n) Num = 4*	88.55	82.14	84.73
(k) (i) $+R_{\text{verf}}$	88.34	81.95	84.53	*(o) Num = 6*	89.01	82.93	85.28
(l) (i) $+R_{\text{tool}} + R_{\text{verf}}$	**89.43**	**83.47**	**85.86**	*(p) Num = 8*	**89.43**	**83.47**	**85.86**

Table 4: Inference efficiency on the **RefCOCOg val-g** split. Token counts exclude tool observations.

Method	Acc@0.5	Calls	Success (%)	Tokens
SFT agent	81.18	1.64	92.1	286
SFT+RL w/o. R_{verf}	84.61	1.93	97.0	318
RecAgent	85.59	1.21	98.4	221

for long or relational descriptions, whereas VG and AM are most useful when multiple visually similar candidates are present. The full model in row *(d)* performs best overall, showing that the tools provide complementary rather than interchangeable evidence.

Reward Functions. In rows *(i)-(l)*, we analyze the reward terms used during RL. The localization-plus-format baseline in row *(i)* provides a strong task-level signal but does not distinguish purposeful evidence acquisition from redundant interaction. Adding R_{tool} in row *(j)* improves accuracy by **1.24** points and increases the successful-call rate by **5.8** points. Adding R_{verf} in row *(k)* improves accuracy by **0.81** points while reducing assistant-generated tokens by **27.4%**. Combining both terms in row *(l)* gives the best accuracy–efficiency trade-off. Because the efficiency term is activated only for correctly grounded, format-valid trajectories, shorter incorrect answers receive no benefit.

Number of Generation. Rows *(m)-(p)* vary the number of rollouts in each GRPO group. Larger groups expose the policy to more diverse candidate trajectories and provide a more stable within-group ranking, but increase rollout cost almost linearly. Average accuracy increases from **84.51** with two generations to **86.25** with eight. We use eight generations in the final model because it provides the best validation accuracy; the corresponding compute trade-off is reported in Appendix **??**.

4.4 Efficiency and Agent Behavior

Accuracy alone does not establish that an agent acquires evidence selectively. We therefore report the average number of tool calls, the successful-call rate, and assistant-generated tokens in Tab. 4. All variants use the same hardware, decoding configuration, and evaluation examples. Compared with the ungated variant, the full reward reduces tool calls by **37.3%**, while improving Acc@0.5 by **0.98** points. We also stratify gRefCOCO into no-target, single-target, and multi-target subsets. The gains of **2.8**, **0.7**, and **1.9** points, respectively, test whether the improvement comes from cardinality-aware verification rather than only conventional single-object grounding.

5 Conclusion

In this work, we introduced **RecAgent**, a novel agentic framework designed to enhance REC through active evidence acquisition and tool-augmented RL. Our approach leverages three components: (i) The *RefBench-4K* CoT dataset for SFT, bridging the perception-grounding gap by structuring the logical progression from expression decomposition to candidate verification; (ii) A diagnostic tool library integrating expression decomposition, visual grounding, attribute matching, spatial reasoning, enabling dynamic reasoning with external tools to acquire missing discriminative evidence and resolve referential ambiguities; (iii) A verification-gated composite reward mechanism that jointly optimizes fine-grained localization, tool-flow quality, and trajectory efficiency. Extensive evaluations on five benchmarks validate our SOTA performance, demonstrating the empirical effectiveness of our paradigm. Future work will explore expanding the diagnostic tool library and applying verification-guided grounding to broader complex visual reasoning applications.

AI USE STATEMENT

Generative AI was used to generate tool-interleaved reasoning trajectories for RefCoT-4K, screen source annotations, and assess the correctness and faithfulness of generated trajectories. Vision-language models also provided responses for selected tools during agent training and inference. AI tools assisted with language editing and manuscript assessment and revision. The authors are responsible for verifying all scientific claims, experimental results, citations, and the final manuscript.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, et al. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

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, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Jierun Chen, Fangyun Wei, Jinjing Zhao, Sizhe Song, Bohuai Wu, Zhuoxuan Peng, S-H Gary Chan, and Hongyang Zhang. Revisiting referring expression comprehension evaluation in the era of large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 513–524. IEEE, 2025a.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023.

Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025b.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ming Dai, Lingfeng Yang, Yihao Xu, Zhenhua Feng, et al. Simvg: A simple framework for visual grounding with decoupled multi-modal fusion. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1749–1759. IEEE, 2021.

Jiajun Deng, Zhengyuan Yang, Daqing Liu, Tianlang Chen, Wengang Zhou, Yanyong Zhang, Houqiang Li, and Wanli Ouyang. Transvg++: End-to-end visual grounding with language conditioned vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45 (11):13636–13652, 2023.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiafang Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.16410.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms, 2025a. URL https://arxiv.org/abs/2504.11536.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025b.

Kanoko Goto, Takumi Hirose, Mahiro Ukai, Shuhei Kurita, and Nakamasa Inoue. Referring expression comprehension for small objects. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21231–21242. IEEE, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Zeyu Han, Fangrui Zhu, Qianru Lao, and Huaizu Jiang. Zero-shot referring expression comprehension via structural similarity between images and captions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14364–14375. IEEE, 2024.

Xiangzhao Hao, Kuan Zhu, Hongyu Guo, Haiyun Guo, Ning Jiang, Quan Lu, Ming Tang, and Jinqiao Wang. Referring expression instance retrieval and a strong end-to-end baseline. *arXiv preprint arXiv:2506.18246*, 2025.

Shuting He, Henghui Ding, Chang Liu, and Xudong Jiang. Grec: Generalized referring expression comprehension. *arXiv preprint arXiv:2308.16182*, 2023.

Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4418–4427. IEEE, 2017.

Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9590–9601, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 1760–1770. IEEE, 2021.

Jingcheng Ke, Jun-Cheng Chen, I-Hong Jhuo, Chia-Wen Lin, and Yen-Yu Lin. Generation and comprehension hand-in-hand: Vision-guided expression diffusion for boosting referring expression generation and comprehension. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022a.

Yang Li, Yan Xu, Chunfeng Yuan, Wei Liu, et al. Improving visual grounding with visual-linguistic verification and iterative reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12479–12488, 2022b.

Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10877–10886. IEEE, 2020.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models, 2023.

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

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024a.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang, Jianfeng Gao, and Chunyuan Li. Llava-plus: Learning to use tools for creating multimodal agents, 2023b. URL https://arxiv.org/abs/2311.05437.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, 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, 2024b.

Ting Liu, Zunnan Xu, Yue Hu, Liangtao Shi, et al. Mapper: Multimodal prior-guided parameter efficient tuning for referring expression comprehension. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024c.

Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1950–1959. IEEE, 2019.

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.

Zixian Ma, Jianguo Zhang, Zhiwei Liu, Jieyu Zhang, Juntao Tan, Manli Shu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Caiming Xiong, et al. Taco: Learning multi-modal action models with synthetic chains-of-thought-and-action. *arXiv preprint arXiv:2412.05479*, 2024.

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.

Hyejin Park, Junhyuk Kwon, Suha Kwak, and Jungseul Ok. Viro: Robust and efficient neuro-symbolic reasoning with verification for referring expression comprehension. *arXiv preprint arXiv:2601.12781*, 2026.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*, 2025.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, et al. Kosmos-2: Grounding multimodal large language models to the world, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangyan Sun, Mingyu Jin, Zhenting Wang, Cheng-Long Wang, Siqi Ma, Qifan Wang, Tong Geng, Ying Nian Wu, Yongfeng Zhang, and Dongfang Liu. Visual agents as fast and slow thinkers. In *The Thirteenth International Conference on Learning Representations*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

Peng Wang, An Yang, Rui Men, Junyang Lin, et al. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pp. 23318–23340. PMLR, 2022.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, et al. Cogvlm: Visual expert for pretrained language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl 3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Yaxian Wang, Henghui Ding, Shuting He, Xudong Jiang, Bifan Wei, and Jun Liu. Hierarchical alignment-enhanced adaptive grounding network for generalized referring expression comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8042–8050, 2025b.

Yaxian Wang, Henghui Ding, Shuting He, Xudong Jiang, Bifan Wei, and Jun Liu. Hierarchical alignment-enhanced adaptive grounding network for generalized referring expression comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8042–8050, 2025c.

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.

Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, et al. Universal instance perception as object discovery and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15325–15336, 2023.

Li Yang, Yan Xu, Chunfeng Yuan, Wei Liu, Bing Li, and Weiming Hu. Improving visual grounding with visual-linguistic verification and iterative reasoning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9489–9498. IEEE, 2022.

Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4643–4652. IEEE, 2019.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jiabo Ye, Junfeng Tian, Ming Yan, Xiaoshan Yang, et al. Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15502–15512, 2022.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. In *International Conference on Learning Representations*, volume 2024, pp. 57153–57180, 2024.

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.

Seonghoon Yu, Joonbeom Hong, Joonseok Lee, and Jeany Son. Latent expression generation for referring image segmentation and grounding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21374–21383. IEEE, 2025.

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025.

Yiyi Zhou, Rongrong Ji, Gen Luo, Xiaoshuai Sun, Jinsong Su, Xinghao Ding, Chia-Wen Lin, and Qi Tian. A real-time global inference network for one-stage referring expression comprehension. *IEEE Transactions on Neural Networks and Learning Systems*, 34(1):134–143, 2021.

Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *European Conference on Computer Vision*, pp. 598–615. Springer, 2022a.

Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *European Conference on Computer Vision*, pp. 598–615. Springer, 2022b.

APPENDIX

In this appendix, we provide more experimental details, related work, tool library, and discussions for a comprehensive evaluation and understanding of our method. Detailed contents are as follows:

A RefCoT-4K Construction Details **15**

A.1 Source Datasets and Candidate Sampling 15

A.2 Cross-Model Annotation Verification 16

A.3 Composition and Query Statistics 16

A.4 Image-Disjoint Split and CoT Generation 17

B Experiment **18**

B.1 More Experimental Details 18

B.2 More Case Studies 18

C Related Work **19**

C.1 Referring Expression Comprehension. 19

C.2 Multimodal LLMs Reasoning. 19

D Tool Library Details **19**

D.1 Expression Decomposer (ED) 20

D.2 Visual Grounder (VG) 21

D.3 Attribute Matcher (AM) 21

D.4 Spatial Reasoner (SR) 21

E Prompt, Function-Call, and Data-Conversion Specification **23**

E.1 Shared output and function-call contract 23

E.2 Collection prompt before SFT 23

E.3 Compact SFT and RL prompt 25

E.4 Candidate cleaning and SFT-record cleaning 25

E.5 DeepSeek expert trajectory review 26

A REFCOT-4K CONSTRUCTION DETAILS

This section documents the data path implemented in the released v1 code. We use *RefCoT-4K* to denote the construction procedure initialized from 4,000 candidates. The screened annotation pool contains 3,579 examples. This number is deliberately distinguished from the number of exported SFT records: the latter depends on the particular trajectory-generation run and its input selection. In particular, the conversion script performs structural cleaning, not a hidden semantic re-annotation of every trajectory. We report this distinction because otherwise candidate, verified annotation, generated trajectory, and SFT-record counts can be inadvertently conflated.

A.1 SOURCE DATASETS AND CANDIDATE SAMPLING

Sources. We draw complementary supervision from five datasets. RefCOCO+ (Yu et al., 2016) emphasizes appearance and relational descriptions by excluding absolute location words. RefCOCOg (Mao et al., 2016) provides longer and more compositional expressions. gRefCOCO (He et al., 2023) extends REC to single-target, multi-target, and no-target cases. REIRCOCO (Hao et al., 2025) contributes retrieval-oriented expressions, including many small targets. Ref-L4 (Chen et al.,

Table 5: Cross-model filtering of the 4,000 collected candidates. "Keep" includes both accept and uncertain decisions at the corresponding stage.

Stage	Input	Output	Routing decision
Qwen3.8-27B verification	4,000	2,674	keep accept/uncertain
Second-pass review queue	1,326	1,326	recheck first-pass rejects
DeepSeek Flash Vision	1,326	683	accept and keep
DeepSeek Flash Vision	1,326	222	uncertain and keep
DeepSeek Flash Vision	1,326	421	reject and discard
Final screened pool	4,000	3,579	$2,674 + 683 + 222$

2025a) supplies long, fine-grained descriptions. We use the training splits of RefCOCO+, RefCOCOg, gRefCOCO, and REIRCOCO. For Ref-L4, we use its designated validation data only as an adaptation source and evaluate on the disjoint test split; we therefore do not treat this adaptation subset as an uncontaminated validation set. No evaluated test example is used for either SFT or RL.

Canonicalization and validity checks. Every source record is converted to a common tuple consisting of an image, a referring expression, and zero or more pixel-space $xyxy$ boxes. We normalize whitespace and retain expressions containing alphabetic text and at most 48 words. A non-empty box must satisfy $0 \leq x_1 < x_2 \leq W$ and $0 \leq y_1 < y_2 \leq H$. We additionally exclude boxes whose area occupies less than 0.015% or more than 98% of the image, which removes nearly invisible or degenerate full-image annotations. An example is deduplicated using the image basename, lower-cased normalized expression, and the complete target-box set; different valid expressions for the same image remain distinct examples.

Balanced sampling. We allocate 800 candidates to each source, producing the 4,000-example input pool. Within each source, we stratify candidates by the Cartesian product of target mode (none/single/multiple), query length (short: at most 4 words; medium: 5–10 words; long: more than 10 words), and target area (small: less than 3% of the image; medium: 3–18%; large: at least 18%), and sample proportionally across the resulting strata. For a multi-target example, target area is the sum of its annotated box areas. To preserve the abstention behavior that ordinary REC corpora lack, 200 of the 800 initial gRefCOCO candidates are explicitly allocated to no-target queries. Sampling uses seed 20260908.

A.2 Cross-Model Annotation Verification

Verification is performed on the source annotation before CoT generation. For each candidate, we render every ground-truth box in red and assign it a numeric identifier. A vision-language verifier receives the marked image and expression and judges whether the proposed boxes cover *all and only* the described objects. For a no-target example, it instead judges whether the scene truly contains no matching object. The output schema is a decision in {accept, reject, uncertain} and a confidence level in {high, medium, low}.

Qwen3.8-27B performs the first verification pass. Its accept and uncertain decisions form an automatic keep set of 2,674 examples, whereas all 1,326 rejected examples are forwarded to DeepSeek Flash Vision for an independent second judgment. The second verifier accepts 683, marks 222 as uncertain, and rejects 421. We retain the first two groups and discard only the 421 examples rejected again. This conservative policy is intentional: the second pass shows that 905/1,326 (68.3%) first-pass rejections are recoverable, so deleting every uncertain or singly rejected case would disproportionately remove short expressions and visually difficult examples. The exact verifier prompt and output contract are given in Appendix E; the flow is summarized in Table 5.

A.3 Composition and Query Statistics

Table 6 reports statistics recomputed from the final annotation file. Query length is measured after whitespace normalization using space-separated words. The complete pool averages 10.18 words per query, but its sources deliberately span very different regimes: RefCOCO+ averages only 3.60 words, whereas Ref-L4 averages 24.04. The final length distribution is 1,067 short (29.8%), 1,280

Table 6: Source composition of the screened RefCoT-4K annotation pool. The S/M/N columns denote single-target, multi-target, and no-target examples.

Source	Initial	Retained	Mean query words	S	M	N
RefCOCO+ (Yu et al., 2016)	800	721	3.60	721	0	0
RefCOCOg (Mao et al., 2016)	800	723	8.41	723	0	0
gRefCOCO (He et al., 2023)	800	753	4.66	351	211	191
REIRCOCO (Hao et al., 2025)	800	643	10.10	643	0	0
Ref-L4 (Chen et al., 2025a)	800	739	24.04	739	0	0
Total	4,000	3,579	10.18	3,177	211	191

medium (35.8%), and 1,232 long (34.4%) queries. This mixture prevents the CoT generator from seeing only verbose expressions for which explicit decomposition is obviously useful.

The target-mode distribution is therefore 3,177 single-target (88.8%), 211 multi-target (5.9%), and 191 no-target (5.3%) examples. Among the 3,388 target-present examples, 429 (12.7%) are small, 1,699 (50.1%) are medium, and 1,260 (37.2%) are large under the area thresholds above. Together, query length, target scale, and target cardinality define the diversity axes shown in the collection block of our data figure.

A.4 Image-Disjoint Split and CoT Generation

Partitioning. We group examples by image filename before splitting, which also keeps a COCO image shared across different source datasets on the same side. With seed 42, 10% of the 3,323 unique images are assigned to an internal holdout. This yields 2,991 optimization images containing 3,215 examples and 332 holdout images containing 364 examples. The holdout is used for internal model selection only; it is not an official benchmark split. SFT and RL optimization consume the same 3,215-example side of this partition, while all official benchmark test splits remain external to the construction pipeline.

Trajectory synthesis and conversion. For every selected prompt, the collection program supplies the image, expression, and the registered OpenAI-style function schemas to the generator. There is no prescribed tool order: the model chooses a tool only when it judges that the corresponding evidence is missing, observes the actual return, and then either makes another call or answers. The collection prompt, XML call syntax, optional in-context examples, decoding defaults, and all control messages are listed in Appendix E. The default collection configuration uses temperature 0.2, Qwen thinking enabled, at most six agent turns, and no artificial per-turn output cap (the command line can override all of these settings).

Before conversion, DeepSeek Flash Vision performs the expert trajectory review: it receives the original image, screened target boxes, deterministic final metrics, and the complete executed transcript, and accepts only trajectories whose final answer is correct and whose material reasoning claims are supported by real preceding observations. The implementation and its exact prompt are in Appendix E. The conversion program then keeps a generated record only when it has a non-empty query and image field, has at least one model-side turn, and ends on an assistant/function-call turn; malformed input JSON is skipped. It replaces the collection-time system and initial user messages with the compact SFT prompt, removes collection-control messages (including "plan then act," "use more tools," and "force answer" nudges), preserves assistant decisions and actual tool observations, canonicalizes thought delimiters, and serializes calls as structured function calls. Consecutive tool observations are merged. Role alternation can additionally be enforced with `--strict-roles`; it is a warning rather than a rejection criterion by default. Thus claims of trajectory-level factual validation and accuracy filtering are performed by the DeepSeek review gate, not by this mechanical conversion step. All boxes in tool arguments, observations, and final answers are Qwen normalized $xyxy$ coordinates in $[0, 1000]$. The resulting supervision teaches the full loop *analyze ambiguity* $\rightarrow$ *acquire evidence* $\rightarrow$ *update the hypothesis* $\rightarrow$ *verify and answer*.

B EXPERIMENT

B.1 MORE EXPERIMENTAL DETAILS

Evaluation splits and statistics. We evaluate every referring expression as one example, including cases in which several expressions refer to the same image region. The official testA and testB splits contain 5,657 and 5,095 expressions for RefCOCO, and 5,726 and 4,889 for RefCOCO+, respectively. RefCOCOg val-g contains 14,432 expressions. The gRefCOCO testA and testB splits contain 19,200 and 16,063 expressions and retain the benchmark's no-target and multi-target cases. Ref-L4 test contains 31,921 expressions. Final results are computed on these complete splits rather than on subsets used for intermediate checkpoint selection. The separate construction-time split and its leakage controls are detailed in Appendix A.

Coordinate and matching protocol. The agent emits boxes as $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ in Qwen's normalized $[0, 1000]$ coordinate space. We convert both predictions and annotations to pixel coordinates before computing IoU. In the multi-target setting, a prediction can match at most one ground-truth box. We greedily match boxes in a fixed deterministic order and require every ground-truth target to exceed the specified IoU threshold. On gRefCOCO, target-present precision additionally requires the number of predicted boxes to equal the number of targets, and N-acc is computed over no-target expressions. An empty output is correct only for a no-target expression. Malformed answers are assigned zero accuracy. For mAcc, we average sample accuracy at thresholds $\{0.50, 0.55, \ldots, 0.95\}$. For each final result, we report the mean over **3** random seeds.

Training and inference configuration. SFT uses bfloat16, LoRA rank 32, $\alpha = 64$, dropout 0.05, three epochs, and a learning rate of 5×10^{-5}. The maximum sequence length is 16,384. GRPO uses a prompt batch size of 16 and eight trajectories per prompt. The vision tower is frozen, while the language-side parameters are updated with a learning rate of 10^{-6} for two epochs. We use a KL coefficient of 0.01, an entropy coefficient of 0.001, temperature 1.2, top-$p = 1.0$, and at most eight assistant turns. We set $(\lambda_f, \lambda_t, \lambda_v)$ to $(0.02, 0.10, 0.01)$ and η to 0.20. The maximum image edge is 2,048 pixels. Training runs on four NVIDIA H100 80GB GPUs. For evaluation, we set temperature to zero and generate one trajectory per example. We will release prompts, tool schemas, data-processing scripts, and evaluation code.

B.2 MORE CASE STUDIES

Figure 3 traces four representative grounding trajectories and contrasts RecAgent with direct Qwen3.5-4B predictions. The upper-left example shows compositional target cardinality: the conjunction "the left picture and right guy" denotes two entities, rather than one region. RecAgent first extracts the two entity types and their left–right relation, retrieves a candidate for each, and verifies the assignment before returning both boxes. In contrast, the direct model collapses the expression into a single-person prediction.

The upper-right example illustrates why verification is also important when the correct answer is empty. Although several people and cup-like regions are retrieved, no candidate jointly satisfies the white-suit, right-side, standing, and holding-a-cup constraints. RecAgent therefore rejects the partial matches and returns no box, whereas the direct model selects a plausible-looking but unsupported person. The lower-left example combines a carrier–object relation with a cardinality constraint. By separately retrieving the left-side carrier and nearby luggage candidates, then checking which luggage is shouldered by that person, RecAgent identifies the two requested suitcases and excludes a subordinate part region that should not be counted independently.

Finally, the lower-right example highlights fine-grained multi-target recall. For "all mobiles," broad retrieval proposes several small regions; a crop inspection rejects one table object, and a final visual check verifies the remaining three phones before they are returned. Across these cases, the decisive evidence is made explicit in the trajectory: constraint decomposition sets the search space, while targeted verification corrects premature semantic commitments and prevents unsupported boxes or counts. These qualitative examples complement aggregate results by showing how the agent's intermediate observations lead to its final decision.

C Related Work

C.1 Referring Expression Comprehension.

REC aims to localize the unique image region described by a natural language expression (Mao et al., 2016; Yu et al., 2016). Existing methods can be broadly divided into two-stage (Yu et al., 2018; Hu et al., 2017; Liu et al., 2019; Yang et al., 2019) and one-stage (Liao et al., 2020; Yang et al., 2022; Zhou et al., 2021; Zhu et al., 2022b; Deng et al., 2023) paradigms, the latter enabling end-to-end box regression conditioned on language. Transformer-based grounding models such as MDETR (Kamath et al., 2021) further unify detection and language understanding via cross-modal attention. With large-scale vision-language pre-training, open-set detectors (Li et al., 2022a; Liu et al., 2024b) and MLLM-based grounding models (Chen et al., 2023; You et al., 2024; Lai et al., 2024; Xia et al., 2024) have substantially improved zero-shot generalization, yet most still assume a single target exists per expression.

To relax this assumption, Generalized REC (GREC) (He et al., 2023) and GRES (Liu et al., 2023a) extend REC to no-target and multi-target expressions, with follow-up work introducing adaptive counting and hierarchical alignment mechanisms to determine output cardinality (Wang et al., 2025b). Other efforts target extreme scale variation via zooming-based fine-tuning for small-object localization (Goto et al., 2025). A parallel line of work tackles REC through LLM-driven compositional reasoning, decomposing expressions into executable perception programs (Surís et al., 2023); however, these pipelines largely assume correct intermediate outputs, motivating recent verification-integrated designs to suppress cascading errors and hallucinated predictions (Park et al., 2026).

C.2 Multimodal LLMs Reasoning.

RL-based post-training, as exemplified by OpenAI-o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), has significantly enhanced LLM reasoning. These paradigms are extended to MLLMs for mathematical VQA (Peng et al., 2025), image segmentation (Liu et al., 2025), and video understanding (Feng et al., 2025b). However, when applied to Referring Expression Comprehension (REC), models often compress localization into a single-pass mapping, suffering from evidence-insufficient selection and long-sequence hallucination (Chen et al., 2025b) due to a lack of explicit verification on fine-grained attributes and spatial relations. To address this, we propose multimodal CoT reasoning via tool-augmented RL, explicitly reformulating REC as an active visual evidence acquisition process to reduce spatial hallucinations.

Tool-Augmented Agentic System. External tools have proven effective in enhancing multimodal reasoning. Early works introduce visual evidence into CoT (Sun et al., 2025), while LLaVA-Plus (Liu et al., 2023b) and VPD (Hu et al., 2024) pioneer training strategies and program-derived data for tool use. Recent methods like TACO (Ma et al., 2024) and PyVision (Zhao et al., 2025) further expand tool use with RL. Despite the success of general agent frameworks like ReAct (Yao et al., 2022) in language tasks, existing visual approaches primarily rely on static tool pipelines, failing to dynamically adapt to complex, open-vocabulary spatial constraints in REC. Our framework addresses this by modeling REC as a tool-conditioned framework, explicitly acquiring discriminative evidence and dynamically balancing tool efficiency through a verification-gated reward.

D Tool Library Details

The main paper uses four *conceptual* families (language analysis, candidate grounding, attribute verification, and spatial reasoning). The v1 implementation exposes nine concrete functions, not four callable tools. All nine are registered by default and supplied as OpenAI-style function schemas; an allow/deny list is the only way to restrict them. Each schema has `additionalProperties=false`: unknown arguments, missing required arguments, and type errors yield a failed tool result rather than silent repair. All boxes are Qwen-normalized $xyxy$ coordinates in $[0, 1000]$.

The category labels indicate the primary evidence supplied by a function, not a fixed execution graph: the policy may call none, one, or several tools. The runtime logs tool name, arguments, status, error, and metadata, but exposes only the return value or error to the model. An omitted `image` is

Table 7: Concrete tool definitions. The category column follows the four conceptual families in the main paper. Successful returns are serialized as `{"ok": true, "value": ...}`, and failures as `{"ok": false, "error": ...}`.

Tool	Category	Arguments	Function and return
`QLA`	ED	`query`: expression to parse; `source`: analysis engine(s).	Decomposes entities, attributes, target flags, and relations; multi-source output retains source-specific referent/relation blocks.
`REPHRASE`	ED	`query`: expression to clarify.	Returns a clearer expression and atomic target phrases for QLA/FIND.
`FIND`	VG	`objects`: atomic object cues; `top_k`: candidate limit; `box_threshold`, `text_threshold`: detection cutoffs; `image`: source image.	GroundingDINO candidate detection with cached-candidate merging and IoU de-duplication; returns labels, boxes, scores, and source.
`VISUAL_GROUNDING`	VG	`query`: expression to ground; `image`: source image.	Direct contextual vision-LLM grounding; returns boxes, count, query, and an annotated-image reference when available.
`CLIP_MATCH`	AM	`text`: discriminative attribute cue; `bboxes`: candidates to compare; `image`: source image.	Ranks candidate crops by CLIP similarity and returns descending candidate indices, scores, and boxes.
`IMAGE_CROP`	AM	`bbox`: region to magnify; `image`: source image/crop; `save_path`: output location.	Produces a concrete crop reference and its box.
`VISION_SELF_CHECK`	AM	`bboxes`: candidates; `query`: matching criterion; `image`: source image.	Visually verifies numbered candidates and returns matches, counts, and an annotated-image reference.
`SPATIAL_JUDGE`	SR	`bbox_a`: subject box; `bbox_b`: reference box.	Computes the direction-and-distance relation of A relative to B; containment is not considered.
`DEPTH_ESTIMATE`	SR	`bbox_a`/`bbox_b`: regions to compare; `image`: source image.	Returns global depth statistics, a near/mid/far grid, regional statistics, and relative ordering when regions are supplied.

bound to the task image. Tool-returned boxes are copied without rescaling in both prompts and final answers.

D.1 Expression Decomposer (ED)

ED resolves linguistic ambiguity before or alongside grounding. Its primary function, `QLA`, offers two independently executable parsing paths. With `source=llm`, it makes an *isolated*, tool-free call to the current Qwen backend (thinking disabled; 512-token limit), asking for JSON with `referents` and `relations`. Each referent has a canonical `T=type.instance` identifier, optional attribute list `V`, optional quantity, and a target flag; each relation has a subject, relation type, scope, and reference entity or frame. Thus this path is a structured LLM semantic parse rather than a hand-written subject–verb–object parser.

With `source=spacy`, QLA uses `en_core_web_sm` lazily at runtime. It splits coordinated clauses, extracts noun chunks, uses the rightmost noun/proper noun as the entity head, and collects dependency modifiers (`amod`, `compound`, `nmod`, `nummod`, and `acl`) as attributes. A small spatial lexicon maps terms such as "left", "behind", "next to", and "in front of" to frame- or entity-relative relations. Entity instances are numbered deterministically. The `all` option returns both parses rather than forcing a heuristic merge, allowing the agent to compare them. `REPHRASE` is

another isolated call to the same backend; it returns a shorter disambiguated expression and atomic attribute–class phrases for the downstream QLA/FIND calls.

D.2 VISUAL GROUNDER (VG)

VG proposes where a described object may be. `FIND` uses **GroundingDINO-Tiny** (`IDEA-Research/grounding-dino-tiny`) through the Transformers zero-shot object-detection interface. For each atomic text cue, it runs text-conditioned detection, converts pixel boxes to Qwen $[0, 1000]$ coordinates, and returns the highest-scoring candidates. If cached GroundingDINO or grounding results are available in the preprocessing store, they are merged with online detections; boxes overlapping an online result above the IoU de-duplication threshold are removed. The default detector and text thresholds are 0.20 and 0.25, respectively.

`VISUAL_GROUNDING` is complementary: it sends the image and the full referring query to an isolated call of the contextual Qwen vision model, with a JSON-only box prompt and thinking enabled. The returned JSON is parsed from the answer/text, coordinates are normalized or corrected when necessary, clamped to $[0, 1000]$, and degenerate boxes are discarded. Both VG functions additionally write a numbered annotated image when they return boxes. Hence FIND prioritizes candidate recall, whereas VISUAL_GROUNDING supplies an independent direct grounding hypothesis.

D.3 ATTRIBUTE MATCHER (AM)

AM resolves ambiguity among visually similar candidate regions. `CLIP_MATCH` uses a process-wide OpenCLIP **ViT-L/14** instance with Apple **DFN2B-CLIP-ViT-L-14** weights. It crops each supplied candidate box from the image, encodes the crop and the short attribute cue, and ranks candidates by image–text similarity (the DFN2B sigmoid/logit-scale score when that checkpoint is active). It is intentionally given an attribute cue, such as color, clothing, accessory, texture, or pose, rather than the whole expression.

`IMAGE_CROP` converts Qwen coordinates back to pixels, crops the PIL image, and saves the result under `/crops/`; a returned crop reference can be supplied to later tools. `VISION_SELF_CHECK` converts candidate boxes to pixels, draws persistent numbered boxes on a copy of the image, and makes an isolated call to the contextual Qwen vision model asking for the matching indices. Its parser accepts JSON/list answers and common textual "box 1" forms, maps valid one-based indices back to Qwen boxes, and saves the annotated image. These tools are therefore used after VG produces competing instances, rather than for re-scoring a long global description.

D.4 SPATIAL REASONER (SR)

SR tests relational constraints that cannot be resolved from an object label alone. `SPATIAL_JUDGE` is deterministic and model-free: it computes candidate centers on the fixed 1000×1000 canvas, selects a cardinal or diagonal direction from the center offset, and categorizes Euclidean separation relative to the image diagonal as nearby, somewhat distant, or far. It returns a single relation such as "far on the left"; containment is deliberately not inferred.

`DEPTH_ESTIMATE` uses **Depth Anything V2 Small** through the Transformers depth-estimation interface. Predicted inverse depth is inverted so that smaller values mean nearer regions, resized to the source image, and summarized as an 8×8 grid. Each cell and requested box is classified near/mid/far relative to the global mean (cutoffs 0.8 and 1.2), and a supplied pair receives a relative front/back ordering. Depth is auxiliary evidence only: the policy combines it with grounded boxes and visual attributes before selecting the final box set.



Figure 3: Qualitative case studies comparing direct Qwen3.5-4B predictions (blue) with RecAgent trajectories (teal). From upper left to lower right, the examples demonstrate: (i) decomposing a conjunctive expression into two targets and verifying their left–right relation; (ii) rejecting partial attribute matches and correctly returning no target; (iii) grounding a carrier–luggage relation while enforcing the requested count; and (iv) retrieving, crop-verifying, and counting small mobile phones. Red text marks the critical predictions, constraints, or observations that determine each outcome.

E PROMPT, FUNCTION-CALL, AND DATA-CONVERSION SPECIFICATION

We distinguish *trajectory collection* from *SFT/RL training*. Collection can append placeholder in-context examples and recovery messages to obtain a complete trace. The final SFT converter removes those controls and replaces the collection prompt with the compact training prompt. Thus the prompt found in a raw collection trajectory is not necessarily the one that optimizes the released SFT adapter.

E.1 SHARED OUTPUT AND FUNCTION-CALL CONTRACT

The first user turn is `<image>` followed by the expression and a format footer. The model reasons in `<think>` and then emits either one tool call or the final `<answer>`, never both in one turn. The collector parses the following canonical XML call form (it also accepts a JSON call list inside `<tool_call>`). A tool observation is appended by the runtime before the next assistant turn.

```
<think>State the missing evidence and the next action.</think>
<tool_call>
<function=FIND>
<parameter=objects>
red chair
</parameter>
<parameter=top_k>
5
</parameter>
</function>
</tool_call>

<think>Evidence is sufficient.</think>
<answer>{"boxes": [[x1,y1,x2,y2]]}</answer>
```

The converter stores calls as Qwen structured function calls, `[{"name":"FIND", "arguments":{...}}]`, and tool returns as `observation` turns. It normalizes nonempty reasoning to `<think></think>`. A one-target answer is one four-number list, multiple targets are a list of such lists, and no target is `{"boxes": []}`. All use Qwen $[0, 1000]$ $xyxy$ coordinates.

E.2 COLLECTION PROMPT BEFORE SFT

The following is the base system prompt.

```
You are a visual grounding agent for referring expressions. Locate every
object satisfying the expression and return its bounding box or boxes.

# Core Decision Principle
Choose tools adaptively according to the expression and current visual
evidence. There is no mandatory pipeline or fixed tool order. A simple,
unambiguous expression may be answered directly or with one useful tool.
For ambiguous, nested, relational, or visually uncertain cases, combine
complementary tools only when they provide useful evidence. Never call a
tool merely to increase the call count.

# Reasoning Guidance
Keep reasoning concise, accurate, and decision-oriented. First identify
only the constraints needed to distinguish the target: entity,
attributes, quantity, spatial relations, relative depth, and reference
objects. Form an initial hypothesis, acquire missing evidence, eliminate
inconsistent candidates, and stop once the evidence is sufficient. QLA
and REPHRASE are optional aids for genuinely long or ambiguous
expressions, not mandatory first steps. VISION_SELF_CHECK is a
near-final verification tool when real candidate boxes already exist.
VISUAL_GROUNDING may be used as an independent grounding check or
fallback.
```

```
# Evidence Discipline
Never invent visual evidence, tool results, or candidate boxes. A tool
response with ok=false, an error, zero valid detections, or no valid
bbox is not positive evidence. If a tool fails, use a relevant
alternative or rely on trustworthy direct visual evidence. Use only tool
names and parameters exposed by the tool schemas supplied with the
current request.

# Unified Coordinate Convention
All boxes in tool arguments, tool returns, and the final answer use
Qwen-normalized xyxy coordinates in [0,1000]. Tool boxes are already
normalized: copy them as-is and never rescale them or treat them as
pixels.

# Strict Turn Protocol
Every assistant turn must have exactly one of these two complete forms:
<think>brief reasoning</think>
<tool_call><function=TOOL_NAME>...</function></tool_call>

or:
<think>brief reasoning</think>
<answer>{"boxes": ...}</answer>

Call exactly one tool in a tool turn. Never emit a bare or unfinished
<think>. Never place <function=...> outside <tool_call>. Every
think/tool_call/function/parameter/answer opening tag must have exactly
one matching closing tag in the correct nesting order. Use only schema
parameters with their declared JSON types. Do not combine a tool call
and final answer in one turn, and do not append text after </tool_call>
or </answer>.

# Final Answer
Use JSON inside <answer>: one target is [x1,y1,x2,y2], multiple targets
are [[x1,y1,x2,y2],...], and no valid target is []."""
```

The collection user message is `Locate every object described by the referring expression: "{expression}". Return their bounding boxes.`, then the literal footer:

```
Protocol:
- Think concisely, then either call tool(s) or give the final answer
  (not both in the same turn).
- Freely choose tools; prefer the minimal useful set.
- ALL boxes (tool args, tool returns, final answer) are already
  Qwen-normalized xyxy in [0,1000]. Copy tool-returned boxes as-is;
  do not convert or re-normalize.
Final format:
<think>...concise plan...</think>
<answer>{"boxes": ...}</answer>
Answer JSON: one box [x1,y1,x2,y2]; multi [[...],...]; none [].
```

During trajectory collection, we use a sampling temperature of 0.2, enable the model's reasoning mode, and allow up to six agent turns without imposing an additional per-turn token limit. When a rollout fails to produce an answer or exhausts its turn budget, we may issue a collection-only recovery instruction that asks the agent to state a concise plan, seek additional complementary evidence, or stop tool use and return a numeric prediction. These recovery instructions are removed before supervised fine-tuning, so they do not become part of the training data. They are also disabled during evaluation: incomplete or invalid rollouts are recorded as failures rather than being regenerated.

E.3 COMPACT SFT AND RL PROMPT

The supplied SFT configuration and both RL parquet builders import the compact v1 prompt in `configs/agent/sft_prompts.py`; this, rather than the longer optional v2 experiment in the same file, is the training system prompt:

```
You are a visual grounding agent for referring expressions.
Use tools when needed to gather evidence, then answer with boxes.
Coordinate convention: ALL boxes in tool arguments, tool returns
(bbox_xyxy / bboxes), and the final <answer> are already
Qwen-normalized xyxy in [0,1000]. Copy tool boxes as-is; never
rescale or treat them as pixels.
Each turn: either call tool(s) or give the final answer, not both.
```

Its literal user footer is:

```
Protocol:
- Think briefly, then call tool(s) or answer (not both).
- Tool-returned boxes are already Qwen xyxy [0,1000]; copy as-is.
Final format:
 thinking...concise...
<answer>{"boxes": ...}</answer>
Answer JSON: one [x1,y1,x2,y2]; multi [[...],...]; none [].
```

SFT starts with `<image>` and `Locate every object described by the referring expression: "{expression}". Return their bounding boxes.` plus this footer. RL uses the same two prompt turns but executes live tools, rather than replaying an SFT trajectory. The optional v2 prompt enforces English-only reasoning and embeds a nine-tool name list plus two placeholder demonstrations; it is selected only by an explicit v2 caller and is not the template imported by the provided SFT YAML or RL parquet builders.

E.4 CANDIDATE CLEANING AND SFT-RECORD CLEANING

Candidate construction precedes any generated CoT. Each source record is canonicalized to image, expression, dimensions, and pixel-space target boxes. The code normalizes expression whitespace and rejects an empty expression, more than 48 whitespace-separated words, or no alphabetic character. It also rejects absent image names, non-positive dimensions, and any target-present record with an out-of-image/degenerate box, a box below 0.015% of image area, or a box above 98%. Explicit no-target records retain an empty box set. Cross-source duplicates are removed using SHA-1 over image basename, lower-cased normalized expression, and complete box set; distinct expressions for one image are retained. Sampling draws 800 examples from each of five sources, stratified by target mode, expression length, and aggregate target area, reserves 200 gRefCOCO no-target examples, and uses seed 20260908.

For annotation verification, ground-truth boxes are drawn red and numbered. Qwen3.8-27B first pass and DeepSeek-v4.1-flash second pass receive:

```
Expression: "{query}"
The image contains {n} red numbered proposed target box(es).
Do these boxes jointly contain ALL and ONLY the objects described by the
expression? Judge the expression-to-box alignment, not whether the boxes
are
aesthetically tight. Return ONLY
{"decision":"accept|reject|uncertain","confidence":"high|medium|low"}.
```

For $n = 0$, the middle text instead asks whether the image truly contains no object matching the expression. The parser maps invalid/unparseable output to `uncertain, low`. First-pass accept and uncertain cases are kept; only first-pass rejects are rechecked, and only a second reject is discarded.

E.5 DEEPSEEK EXPERT TRAJECTORY REVIEW

After tool-interleaved generation and before conversion to the training format, an independent vision-language reviewer receives the original image, screened pixel-space ground truth, deterministic answer metrics, and the bounded executed trajectory. The trajectory contains the assistant's reasoning, structured tool calls, and actual tool observations, and is treated as untrusted evidence during review. The reviewer must return exactly:

```
{"decision":"accept|reject", "faithful":true|false,
 "correct":true|false, "reason":"brief evidence-based reason"}
```

It accepts only when `decision=accept`, `faithful=true`, and `correct=true`; malformed replies, uncertainty, request failures, unsupported reasoning claims, invented observations/boxes, and incorrect/missing final answers are rejected. Concise trajectories and flexible tool order are not penalized. The exact decision instruction is:

```
Accept only if BOTH conditions hold:
1. The final answer identifies all and only the intended targets (for an
empty target set, it must correctly abstain).
2. Every material reasoning claim and final selection is supported by
the image or by an actual preceding tool observation. Reject
hallucinated observations, invented boxes/evidence, malformed or missing
final answers, and calls whose stated result is not present in the
transcript. Do not require a fixed tool order and do not reject a
concise trajectory merely for using few tools.
```

We retain the complete set of reviewed trajectories for audit and convert only those that satisfy all review criteria into training examples. During conversion, malformed or incomplete records are removed, collection-only recovery instructions are discarded, and prompts are replaced with the training prompt. Tool calls and reasoning delimiters are canonicalized, consecutive tool observations are merged, and each trajectory is required to end with an agent response. The resulting examples pair a system instruction and tool specifications with the multi-turn interaction and its associated image. We optionally enforce strict role alternation; otherwise, deviations are recorded for inspection. These normalization steps are applied only after the expert faithfulness and correctness filter.