# FROM FINE-GRAINED RECOGNITION TO FINE-GRAINED MANIPULATION: SEMANTIC VISUAL PRIMITIVES FOR GROUNDED MULTIMODAL REASONING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Multimodal large language models (MLLMs) can often describe fine-grained visual details, such as part attributes or functional object regions, but may fail to faithfully ground these cues or use them for recognition and action. This reveals a fine-grained semantic grounding gap, where intermediate evidence remains plausible in language or localized by geometry-only primitives, yet is not structured as grounded evidence for reasoning. In this paper, we propose Semantic Visual Primitive Reasoning (SVP-R), a unified framework that turns visual primitives into structured evidence that binds geometry, semantics, and action cues. Each Semantic Visual Primitive binds *where* a part is, *what* property it exhibits, and *how* it can support an action, forming an evidence graph for fine-grained recognition and manipulation. To prevent such primitives from becoming post-hoc rationales, we introduce graph binding, a shared objective that verifies the evidence graph at the node, edge, and graph levels, instantiated as Reinforced Evidence Binding (REB) for recognition and Affordance Evidence Binding (AEB) for manipulation. Extensive experiments on fine-grained recognition and manipulation show that SVP-R consistently outperforms existing grounded reasoning and VLA methods, and interventions confirm that its decisions follow the grounded evidence. These results suggest that fine-grained multimodal reasoning requires binding visual details into structured evidence for perception and action.

## 1 INTRODUCTION

Multimodal large language models (MLLMs) have made remarkable progress in visual understanding, multimodal reasoning, and vision-language-action modeling for robotic manipulation (Liu et al., 2025a; Black et al., 2025b). Their capabilities are now being extended to fine-grained tasks, such as fine-grained recognition (Xie et al., 2025; He et al., 2026a) and fine-grained manipulation (Yin et al., 2025), where success depends on subtle part-level evidence. In fine-grained recognition (Wei et al., 2022), distinguishing a subordinate bird species may hinge on subtle part-level evidence, such as crown color or bill shape. As for fine-grained manipulation (Xu et al., 2026), precise peg-in-hole insertion requires localizing the relevant parts and satisfying tight alignment constraints. Both tasks require models not only to perceive fine-grained visual details, but also to ground them and use them to support recognition or action.

Although MLLMs can often verbalize intermediate visual cues (Bai et al., 2025), these descriptions do not necessarily imply faithful reasoning. A model may state plausible attributes or functional regions while failing to bind them to the corresponding image evidence, and its final decision may still rely on holistic shortcuts, language priors, or memorized category associations (Yu et al., 2026a), as illustrated in Fig. 1(a). We refer to this limitation as the *fine-grained semantic grounding gap*: discriminative cues may be described in language, yet remain weakly connected to the visual evidence that should support the final prediction or action.

Recent visual reasoning methods partially address this problem by interleaving boxes, points, or other spatial markers into the reasoning trace (Sarch et al., 2025; Liu et al., 2025b). These geometry primitives provide explicit references for *where* the model is looking, as shown in Fig. 1(b), but remain incomplete for fine-grained reasoning. A localized region alone does not specify *what* subtle



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.

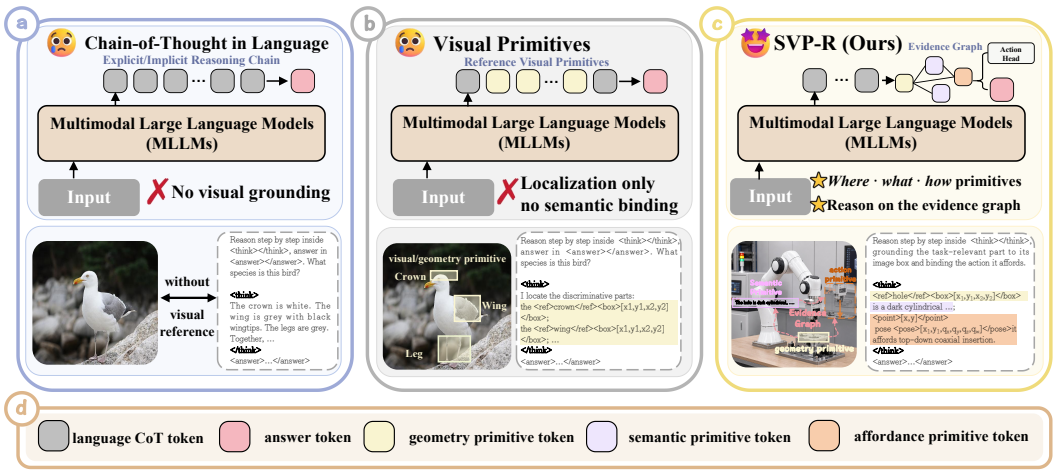



Figure 1: **From language-level reasoning to grounded, faithful reasoning.** Unlike (a) language-only chains with no visual grounding and (b) geometry-only visual primitives that localize but neither carry semantics nor bind to the decision, (c) our SVP-R emits complete *where*/*what*/*how* primitives and reasons over a bound evidence graph, yielding grounded and faithful predictions and actions. (d) presents the legend.

property is observed or *how* that region supports a category decision or executable action. Geometry-only primitives may therefore function as post-hoc pointers rather than decision-bearing evidence.

To bridge the fine-grained semantic grounding gap, we propose **Semantic Visual Primitive Reasoning** (SVP-R), a unified framework that turns visual primitives into structured, grounded evidence. As shown in Fig. 1(c), each **Semantic Visual Primitive** (SVP) binds a shared part to its geometry, semantic property, and action affordance, specifying *where* the part is, *what* it exhibits, and *how* it can support action. SVP-R organizes these primitives into an evidence graph that explicitly connects localized regions, part-level attributes, and functional cues. The graph provides a common interface for fine-grained recognition and manipulation, where predictions are read from grounded semantic evidence and actions are conditioned on grounded functional evidence.

To ensure that the primitives are actually used rather than merely produced, we introduce graph binding, an objective over the evidence graph shared by both tasks, which constrains the node, edge, and graph levels to compel the model to reason over verified evidence. For fine-grained recognition, graph binding is instantiated as **Reinforced Evidence Binding** (REB), a verifiable, correctness-gated reinforcement objective that rewards reasoning traces whose primitives accurately localize the relevant parts, faithfully describe their visual attributes, and support the correct prediction through an explicit class–attribute readout. Its binding rewards are activated only when the final answer is correct, preventing the model from obtaining high rewards through plausible but task-irrelevant evidence or explanations of incorrect predictions. For fine-grained manipulation, the primitives, including the grasp pose the referred part affords, are derived from a hidden state of the policy, and the same objective is instantiated as **Affordance Evidence Binding** (AEB), which forces them to land on the referred part rather than a distractor. SVP-R thereby learns to reason and act over a bound evidence graph rather than simply generate grounded-looking intermediate cues.

We evaluate SVP-R on fine-grained recognition and manipulation. On CUB (Wah et al., 2011), Stanford Dogs (Khosla et al., 2011), and Stanford Cars (Krause et al., 2013), SVP-R improves average accuracy by 9.8% over the comparative method and by 6.0% on held-out categories. We further construct embodied fine-grained localization tasks based on MetaFine (Xu et al., 2026), where SVP-R reduces the part-localization error from 12.06 to 0.54 pixels. For fine-grained manipulation, we evaluate on five simulated tasks from RoboTwin (Chen et al., 2025) and MetaFine and three real-robot tasks, where SVP-R outperforms competing policies. Ablations further confirm that graph binding makes the model reason over the evidence it grounds, which also improves robustness under transfer. Our contributions are summarized as follows:

- We identify the *fine-grained semantic grounding gap* in MLLMs and propose Semantic Visual Primitives, evidence units that bind *where* a part is, *what* it exhibits, and *how* it supports action within a unified evidence graph.

- We introduce graph binding, a shared objective that verifies the evidence graph at the node, edge, and graph levels, instantiated as Reinforced Evidence Binding for recognition and Affordance Evidence Binding for manipulation.
- We demonstrate that SVP-R improves fine-grained recognition, held-out generalization, part localization, and manipulation success over grounded reasoning methods and VLA policies, with interventions confirming that its decisions follow the grounded evidence.

## 2 Related Work

**Fine-grained visual recognition** Fine-grained recognition requires perceiving local discriminative cues and reading them as semantically meaningful attributes. Prior work makes such cues explicit through part localization, attribute prediction, and concept bottlenecks (Wei et al., 2022; Yang et al., 2023), and recent MLLMs fold attributes and reasoning into the task (He et al., 2025; 2026b). Yet attributes alone are not verifiable evidence: models still fall back on language priors or holistic shortcuts while producing fine-grained explanations (Yu et al., 2026a; Tong et al., 2024; Kim & Ji, 2024). Reinforcement learning helps, but its reward targets answer accuracy (He et al., 2026a), leaving intermediate attributes unconstrained. We instead bind each attribute to its discriminative part and make that binding a precondition for reward rather than a byproduct of description.

**Grounded chain-of-thought reasoning** Chain-of-thought with verifiable rewards is a common recipe for improving MLLM reasoning, yet purely textual chains leave intermediate steps ungrounded. Recent work grounds them in image regions, by supervised localize-then-answer pipelines (Wu & Xie, 2024; Shao et al., 2024) or by rewarding emitted boxes (Liu et al., 2025b; Fan et al., 2025; Cao et al., 2025; Sarch et al., 2025). Such rewards verify where the model looked, not what is there or whether it drove the decision (Yu et al., 2026b; Xiao et al., 2026). We instead reward complete evidence: a primitive must localize a part, name a property holding there, and support the correct class before earning credit.

**Visual intermediates for robotic manipulation** Vision-language-action models map observations and instructions to actions (Black et al., 2025b;a; Liu et al., 2025a), leaving the intermediate visual evidence unused. Recent methods insert an explicit intermediate: subgoal images (Zhao et al., 2025) or reasoning tokens conditioning a flow-matching expert (Deng et al., 2025; Chen et al., 2026), with reinforcement learning beginning to verify them by rewarding keypoint accuracy or affordance localization (Kim et al., 2025; Wang et al., 2026a). Yet these intermediates carry geometry without part-level affordances. Our primitives bind each part to its location and affordance, and are decoded from an intermediate hidden state rather than emitted as tokens.

## 3 Methodology

In this section, we first present an overview of the proposed Semantic Visual Primitive Reasoning (SVP-R) and how it empowers both fine-grained recognition and fine-grained manipulation within a unified interface. We then detail its semantic visual primitives and how they ground reasoning through the binding evidence graph. Finally, we delve into the two tasks and show how the binding objective is optimized for them.

### 3.1 Overview: From Fine-grained Recognition to Fine-grained Manipulation

Both fine-grained recognition and manipulation require the model to ground its reasoning in fine-grained visual evidence, and we propose SVP-R to empower both tasks.

As shown in Fig. 2, given an image $I$ and an instruction, the proposed SVP-R structures such evidence into a set of *semantic visual primitives*. For a referenced part $p$, the primitive is the triple

$$g_p = (\mathbf{b}_p,\ a_p,\ c_p), \tag{1}$$

where $\mathbf{b}_p$ localizes the part, $a_p$ is the attribute predicated of it, and $c_p$ is the affordance it supports. For recognition, a primitive may locate the crown of a bird and state that it is white. For manipulation, given the instruction to grasp the mug by its handle, it may locate the handle, identify it as

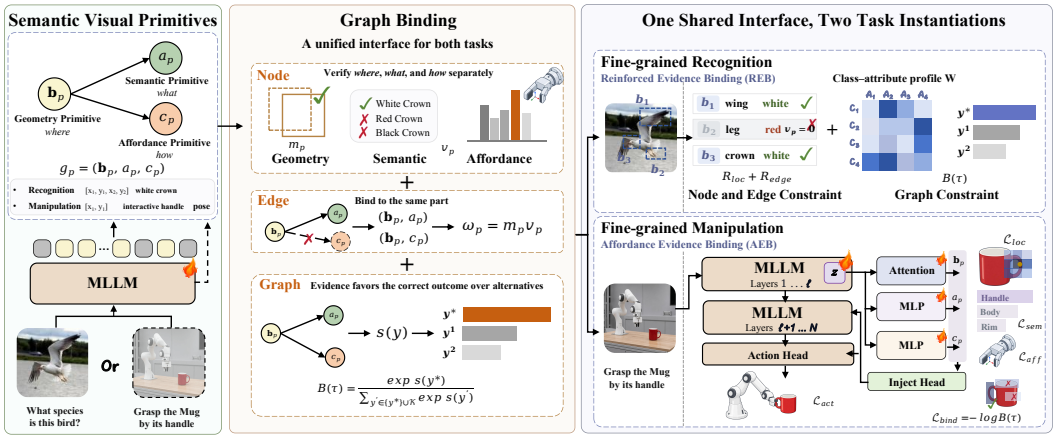



Figure 2: **Overview of the proposed Semantic Visual Primitive Reasoning (SVP-R) method.** The MLLM produces semantic visual primitives $g_p = (\mathbf{b}_p, a_p, c_p)$ that bind *where*, *what*, and *how* for a part. Graph binding verifies each node against annotation, keeps an edge only when both endpoints hold, and requires the verified edges to favor $y^\star$ over hard alternatives via $B(\tau)$. The same objective becomes the REB reward for recognition and the AEB objective for manipulation, where primitives decoded from a token $\mathbf{z}$ condition the action expert.

the referred part, and give the pose that affords grasping. The shared index $p$ turns the triple into a single unit of evidence rather than three separate claims, and the two tasks consume it through the same interface. Recognition reasons out a category from the localized attributes, while manipulation further reads the affordance to derive an action.

We then propose graph binding to ensure that the primitives support the final decision. The objective credits a primitive only when its location is right, its claim holds in the image, and the evidence it contributes separates the correct outcome from its hard alternatives, so that the decision is derived from the evidence rather than accompanied by it. Under the same objective, recognition and manipulation share one interface and differ only in how it is instantiated.

### 3.2 Semantic Visual Primitives with Graph Binding

In this section, we detail the form of the semantic visual primitives and the graph binding that verifies them for reasoning.

**Semantic visual primitives** A primitive is written as a single expression that localizes and describes a discriminative part, <ref>$p$</ref><box>$\mathbf{b}_p$</box> is $a_p$, followed by the affordance $c_p$ when the primitive drives an action for manipulation. Binding the three components into one expression is what makes a primitive checkable, since the model cannot claim a property without saying where it holds. A part named without a box, or a box given without a claim, is not a primitive and carries no evidence. A trace $\tau$ contains primitives over a referenced part set $\mathcal{P}(\tau)$.

**Graph binding** The primitives form a graph whose nodes are the components $\mathbf{b}_p$, $a_p$, and $c_p$, and whose edges $(\mathbf{b}_p, a_p)$ and $(\mathbf{b}_p, c_p)$ attach the attribute and affordance to the part. Graph binding constrains the graph at three levels.

***Node level*** Each component is checked against annotation. Let $m_p \in [0, 1]$ measure how well $\mathbf{b}_p$ matches the annotated location of part $p$, and $v_p$ whether the claim $a_p$ or $c_p$ is correctly described. A node is credited only when its own check passes.

***Edge level*** An edge holds only when both of its endpoints hold, so its weight is the product

$$\omega_p = m_p \, v_p. \tag{2}$$

A correct claim on a mislocalized part, or a hallucinated one on a correct part, earns nothing.

***Graph level*** Edges that hold must also support the decision. Let $y^\star$ be the correct outcome, $\mathcal{K}$ a set of hard alternatives, and $s(y)$ the evidence score that the surviving edges lend to candidate $y$. The

binding objective is the probability of the correct outcome among the candidates,

$$B(\tau) = \frac{\exp s(y^\star)}{\sum_{y' \in \{y^\star\} \cup \mathcal{K}} \exp s(y')}. \tag{3}$$

$B(\tau)$ is high only when the verified evidence picks out $y^\star$ over its alternatives, so credit flows to edges that are correct *and* discriminative. This is what binds the graph to the decision rather than merely checking its nodes.

Furthermore, $B(\tau)$ serves as a unified objective shared by both tasks, while $m_p$, $v_p$, and $s(y)$ are instantiated per task. It becomes a reward for fine-grained recognition, **Reinforced Evidence Binding** (REB), and the training objective for fine-grained manipulation, **Affordance Evidence Binding** (AEB). Section 3.3 gives both instantiations.

### 3.3 SVP-R for Fine-grained Recognition and Manipulation

In this section, we detail SVP-R for fine-grained recognition and fine-grained manipulation.

**Fine-grained Recognition** The outcome $y^\star$ is the subordinate category, $\mathcal{K}$ its hard confusers, and $\mathcal{A}(I)$ the part-attribute pairs that hold in image $I$. Primitives are emitted as tokens in the reasoning text. We first fine-tune a base MLLM on structured traces to teach the schema, then optimize with GRPO using REB.

***Node level*** Localization is a Gaussian in the distance between the predicted box center $\mathbf{c}(\mathbf{b}_p)$ and the annotated part center $\boldsymbol{\mu}_p$,

$$m_p = \exp\Big( -\tfrac{1}{2} \|\mathbf{c}(\mathbf{b}_p) - \boldsymbol{\mu}_p\|_2^2 / \sigma^2 \Big), \tag{4}$$

which ignores box extent and so cannot be inflated by shrinking the box. The semantic primitive is checked by $v_p = \mathbb{1}[(p, a_p) \in \mathcal{A}(I)]$, whether the stated attribute holds in the image.

***Edge level*** The edge weight is $\omega_p = m_p v_p$, so a primitive earns credit only when it both points at the right part and states a true attribute of it.

***Graph level*** A profile $\mathbf{W}$ records how characteristic each part-attribute pair is of each category, and surviving edges are aggregated through it,

$$s(y) = \sum_{p \in \mathcal{P}(\tau)} \omega_p \, \mathbf{W}[\, y, (p, a_p)\,], \tag{5}$$

and $B(\tau)$ is computed from $s(y)$ by Eq. 3. Since $\omega_p$ gates every term, an attribute typical of $y^\star$ but absent from the image contributes nothing.

***Reward*** REB takes one statistic per level, the mean node score $R_{\text{loc}}$, the mean edge weight $R_{\text{edge}}$, and the graph-level $B(\tau)$, and combines them with answer correctness $R_{\text{acc}}$,

$$R(\tau) = R_{\text{acc}} + R_{\text{loc}} + R_{\text{acc}}\big(R_{\text{edge}} + B(\tau)\big). \tag{6}$$

The binding terms are released only when the answer is correct, so a trace cannot earn credit for evidence that explains a wrong prediction, and malformed traces receive zero reward.

**Fine-grained Manipulation** The outcome $y^\star$ is the part the instruction refers to and $\mathcal{K}$ the other candidate parts in the scene. Here the primitive is read out from the MLLM inside the policy rather than emitted as tokens, and the action expert conditions on the same primitive that is scored, so that recognition writes what control reads.

As shown in Fig. 2, a learnable primitive token $\mathbf{z}$ is appended to the input, and the primitive is read from its hidden state at an intermediate layer. The geometry $\mathbf{b}_p$ becomes a heatmap $\hat{\mathbf{h}}$ over the image patches, obtained as a softmax of the attention from $\mathbf{z}$ to the image tokens, and its expected location $\hat{\mathbf{p}}$ gives the target point. The attribute $a_p$ becomes a distribution $\hat{\mathbf{q}}$ over the candidate parts, naming which part the primitive refers to, and the affordance $c_p$ becomes an end-effector pose $\hat{\mathbf{T}}$, both read from $\mathbf{z}$ by linear heads.

The decoded primitive is then injected back at the same layer. Its three components are fused into a goal vector:

$$\mathbf{g} = \mathrm{MLP}\big([\hat{\mathbf{p}};\ \hat{\mathbf{q}};\ \hat{\mathbf{T}}]\big), \tag{7}$$

which is added to the state token of the action expert, while the heatmap additionally marks the target patches on the image tokens.

***Node level*** Each component is supervised against its annotation. $\mathcal{L}_{\mathrm{loc}}$ combines a cross-entropy on the patch containing the annotated target with a regression on the predicted point, $\mathcal{L}_{\mathrm{sem}}$ is a cross-entropy over the candidate parts, and $\mathcal{L}_{\mathrm{aff}}$ a Huber loss on the end-effector pose.

***Graph level*** The heatmap should concentrate on the part the instruction refers to rather than on the other candidate parts. Let $M_y$ be the heatmap mass inside the annotated region of candidate $y$. Taking $s(y) = \log M_y$, Eq. (3) measures the share of mass on the referred part, and we minimize $\mathcal{L}_{\mathrm{bind}} = -\log B(\tau)$. With $\mathcal{L}_{\mathrm{node}}$ for the node-level supervision on the heatmap, the part identity, and the pose, AEB adds it and the graph term to the flow-matching loss of the policy,

$$\mathcal{L}_{\mathrm{AEB}} = \mathcal{L}_{\mathrm{act}} + \mathcal{L}_{\mathrm{node}} + \mathcal{L}_{\mathrm{bind}}. \tag{8}$$

# 4 Experiments

## 4.1 Datasets and Empirical Settings

**Datasets** We conduct experiments on five benchmarks across fine-grained recognition and manipulation. For fine-grained recognition, we evaluate SVP-R on CUB (Wah et al., 2011), Stanford Dogs (Khosla et al., 2011), and Stanford Cars (Krause et al., 2013), which are widely used for the task. We further construct a spatial localization split from MetaFine (Xu et al., 2026), an evaluation platform for fine-grained manipulation with part-annotated articulated assets and affordance labels. For fine-grained manipulation, we evaluate two tasks from MetaFine, three tasks from RoboTwin 2.0 (Chen et al., 2025) that impose explicit part-level or attribute-level constraints, and three real-robot tasks, all of which require satisfying fine-grained constraints to succeed. RoboTwin tasks are evaluated under the Hard setting with strong domain randomization (cf. App. A).

**Empirical Settings** Fine-grained recognition is measured through both image classification and robotic spatial localization. Both are evaluated under the *Within* setting, with the official train/test splits, and the *Held-out* setting, where the model is trained on half of the classes and tested on the other half for classification, and under out-of-domain perturbations for localization. Manipulation follows the MetaFine protocol (Xu et al., 2026) with 100 trials per task, and the real-robot tasks report the success rate over 25 trials. We compare SVP-R with a suite of MLLMs and VLAs. For recognition, SVP-R and all trained baselines are built on the same suite of Qwen2.5-VL backbone, 2 epochs of supervised fine-tuning with AdamW, and the same reinforcement learning schedule on 8 NVIDIA H100 GPUs. For manipulation, SVP-R is built on $\pi_0$ (Black et al., 2025b). Further implementation details are given in App. B.

## 4.2 Main Results

**Fine-grained Recognition** We evaluate SVP-R for fine-grained recognition across the *Within* and *Held-out* settings in Tab. 1. Under *Within*, SVP-R reaches 90.41% at 7B and 85.27% at 3B, where the 3B model already surpasses all competing methods. On localization it cuts the part error to 0.54 pixels, over an order of magnitude below the strongest competitor at 12.06. The gains hold under *Held-out*, where no baselines exceed 33% while SVP-R reaches 39.01%, over 6% higher, and keeps its localization error near 3 pixels, showing that the bound evidence transfers to unseen categories.

**Fine-grained Manipulation** As shown in Tab. 2, we evaluate SVP-R for fine-grained manipulation on both simulation benchmarks and a real robot. SVP-R consistently outperforms the compared methods across both settings. On MetaFine, all baselines fail almost entirely on the Peg task, whereas SVP-R reaches $29\%$. SVP-R grounds not only the target part but also the grasp pose it affords as an affordance primitive, which is what makes such tightly constrained tasks feasible. On RoboTwin 2.0 under the *Hard* setting, SVP-R improves over the compared methods by $13\%$ on Beat

Table 1: **Fine-grained recognition results.** Recognition accuracy (%, ↑) on CUB, Stanford Dogs, and Stanford Cars, and robotic spatial localization on the MetaFine *grasp* and *peg* tasks reported as localization error in pixels (↓). The **yellow blocks** are tested under the *Within* setting and the **blue blocks** under the *Held-out* setting. "Avg" averages the columns within each block. The best result in each block is marked in **bold**.

| METHOD | PARAMS | **Recognition** (acc, ↑) | | | | **Localization** (px err, ↓) | | |
|---|---|---|---|---|---|---|---|---|
| | | *CUB* | *Dogs* | *Cars* | *Avg* | *Grasp* | *Peg* | *Avg* |
| Visual-CoT (Shao et al., 2024) | 3B | 62.56 | 79.30 | 80.77 | 74.21 | 18.00 | 6.12 | 12.06 |
| ViGoRL (Sarch et al., 2025) | 3B | 79.45 | 80.50 | 81.80 | 80.58 | 27.60 | 7.40 | 17.50 |
| GRIT (Fan et al., 2025) | 7B | 82.65 | 87.23 | 61.47 | 77.12 | 17.05 | 14.87 | 15.96 |
| Visual-RFT (Liu et al., 2025b) | 7B | 89.23 | 86.09 | 63.30 | 79.54 | 17.98 | 7.09 | 12.54 |
| **SVP-R (Ours, 3B)** | 3B | 84.87 | 81.60 | 89.34 | 85.27 | 1.31 | **0.40** | 0.85 |
| **SVP-R (Ours, 7B)** | 7B | **91.29** | **88.52** | **91.42** | **90.41** | **0.47** | 0.61 | **0.54** |
| Visual-CoT (Shao et al., 2024) | 3B | 32.23 | 22.77 | 22.65 | 25.88 | 23.20 | 19.07 | 21.13 |
| ViGoRL (Sarch et al., 2025) | 3B | 45.02 | 29.14 | 24.83 | 33.00 | 29.51 | 23.68 | 26.59 |
| GRIT (Fan et al., 2025) | 7B | 47.54 | 16.63 | 18.79 | 27.65 | 21.77 | 17.80 | 19.79 |
| Visual-RFT (Liu et al., 2025b) | 7B | 42.27 | 14.50 | 20.30 | 25.69 | 23.27 | 15.31 | 19.29 |
| **SVP-R (Ours, 3B)** | 3B | **51.53** | 35.42 | **30.07** | **39.01** | **3.06** | **3.51** | **3.29** |
| **SVP-R (Ours, 7B)** | 7B | 49.09 | **38.12** | 28.31 | 38.51 | 8.57 | 3.93 | 6.25 |

Table 2: **Fine-grained manipulation results.** Success rate (%, ↑) on two MetaFine tasks, three RoboTwin 2.0 tasks, and three real-robot tasks. **Simulation benchmarks** are shaded in yellow and **real-robot tasks** in blue. RoboTwin follows the official single-task protocol with 50 clean demonstrations per task, evaluated over 100 trials under the *Hard* (domain-randomized) setting; real-robot results are averaged over 25 trials per task. The best result in each column is in **bold**.

| METHOD | **MetaFine** | | **RoboTwin 2.0** | | | **Real Robot** | | |
|---|---|---|---|---|---|---|---|---|
| | *Grasp* | *Peg* | *Beat Block* | *Click Bell* | *Blocks Ranking RGB* | *Grasp Handle* | *Close Box* | *Grasp Cube* |
| RDT (Liu et al., 2025a) | 42 | 0 | 37 | 9 | 0 | 64 | 88 | 68 |
| $\pi_0$ (Black et al., 2025b) | 75 | 0 | 21 | 3 | 5 | 76 | 92 | 72 |
| $\pi_{0.5}$ (Black et al., 2025a) | 80 | 0 | 41 | 15 | 9 | 74 | 92 | 77 |
| **SVP-R (Ours, 3B)** | **83** | **29** | **54** | **26** | **17** | **84** | **96** | **80** |

Block, and is the only policy that makes meaningful progress on Blocks Ranking RGB, a task requiring the robot to discriminate blocks by color before placing them. The advantage further transfers to the real robot, where SVP-R attains the best success rate on all three tasks.

### 4.3 ABLATION STUDIES AND DISCUSSIONS

We hereby analyze and discuss our SVP-R by answering the following questions, based on ablation studies on fine-grained recognition with CUB and manipulation with MetaFine.

**Can SVP-R ground the fine-grained evidence within the reasoning?** SVP-R provides a unified interface that empowers both tasks with evidence grounding. As shown in Tab. 1, the localization precision directly reflects such grounding ability, reaching an error of 0.54 pixels under the *Within* setting and only 3.29 pixels under the severe perturbations of the *Held-out* setting. As reported in Tab. 3, SVP-R lifts the grounding hit rate from 18.8% to 84.0% at the 3B scale, while retaining 82.1% on the held-out categories with only a marginal drop, indicating that the grounding ability generalizes beyond the categories seen during training.

Table 3: Grounding hit rate (%, ↑) of the referred parts on CUB under *Within* and *Held-out* settings.

| Method | *Within* | *Held-out* |
|---|---|---|
| Qwen2.5-VL-3B | 18.8 | 18.8 |
| **SVP-R (Ours, 3B)** | **84.0** | **82.1** |
| Qwen2.5-VL-7B | 27.8 | 26.3 |
| **SVP-R (Ours, 7B)** | **85.7** | **83.9** |

Furthermore, we visualize the grounding results for both tasks (cf. Fig. 3). Within the trace of the reasoning, SVP-R grounds the interactive part as a geometry primitive and the grasp pose as an affordance primitive for fine-grained manipulation, and the discriminative part as a geometry primitive for fine-grained recognition. As visualized, SVP-R predicts the grasp pose and the localization pre-




Figure 3: **Grounding within the SVP-R reasoning trace.** For fine-grained manipulation, SVP-R localizes the interactive part and predicts its grasp pose, with predictions drawn as a solid line and green triangle and the ground truth as a dashed line and red cross. For fine-grained recognition, the predicted box of each referred part is shown in green and the ground truth in red.

cisely enough to satisfy the part-level constraints, which translates into substantially higher success rates on fine-grained manipulation, while for recognition the grounded discriminative parts provide the subtle cues on which the final category prediction is conditioned.

**Can SVP-R bind the reasoning to the grounded evidence?** To evaluate whether SVP-R narrows the fine-grained semantic grounding gap, we intervene on the grounded evidence in both tasks, where a truly bound prediction should degrade once the evidence is corrupted. For fine-grained recognition, we consider three interventions on the attributes (cf. Tab. 4), *i.e.*, removing them entirely (None), replacing them with those of a confusing category such as giving a Cardinal the attributes of a Pyrrhuloxia (Swap), and shuffling them across parts such as attaching the wing color to the bill (Shuffle). All three degrade the accuracy substantially, from 84.87% to 32.03%, 53.37%, and 41.04% respectively. Notably, shuffling hurts more than swapping, indicating that the prediction is bound not only to the attribute content but also to its association with the corresponding part.

Table 4: Binding on **fine-grained recognition**. Accuracy (%, ↑) on CUB.

| Intervention | Acc. |
|---|---|
| None | 32.03 |
| Shuffle | 41.04 |
| Swap | 53.37 |
| **SVP-R (Ours)** | **84.87** |

We further ask which level of graph binding produces this sensitivity. With a reward that supervises only the nodes, first the geometry alone and then the geometry together with the attribute, removing the elicited evidence entirely costs just 3.77% and 4.64% respectively, so the trace is decorative and the prediction hardly registers that the evidence is gone. Adding the edge-level term raises that cost to 30.61%, and the full reward to 52.84 points, so graph binding is what turns the trace from a description of the image into the evidence the decision rests on.

For fine-grained manipulation, we apply the counterpart interventions on the predicted primitives (cf. Tab. 5), *i.e.*, removing them (None), replacing them with the valid primitives of another object (Swap), and displacing them by a small offset (Jitter). Success collapses from 83% to 0% under None and remains at 7% under Swap, showing that the policy reads the content of the evidence rather than the presence of a well-formed interface, while Jitter leaves 52%, completing a monotonic dose-response over the intervention strength. Together, both tasks show the same pattern, that corrupting the grounded evidence consistently degrades the prediction, confirming that SVP-R genuinely conditions its prediction on what it grounds.

Table 5: Binding on **fine-grained manipulation**. Success rate (%, ↑) on MetaFine.

| Intervention | *Grasp* | *Peg* |
|---|---|---|
| None | 0 | 0 |
| Swap | 7 | 0 |
| Jitter | 52 | 15 |
| **SVP-R (Ours)** | **83** | **29** |

**Which layer should bridge the reasoning to the action?** For fine-grained manipulation, SVP-R reads the primitive from the hidden state of the primitive token at an intermediate layer $L$ and injects it back at the same layer, so $L$ controls both how much evidence the token has accumulated and how much depth remains for the expert to use it. To verify this design, we vary $L$ while keeping the others fixed (cf. Fig. 4). The success rate declines steadily as the interface moves deeper, dropping from 83% to 46% on Grasp. More tellingly, the gap between the model's own primitive and the jittered one shrinks along the same axis, from 31% to 4% on *Grasp* and from 11% to 2% on *Peg*,

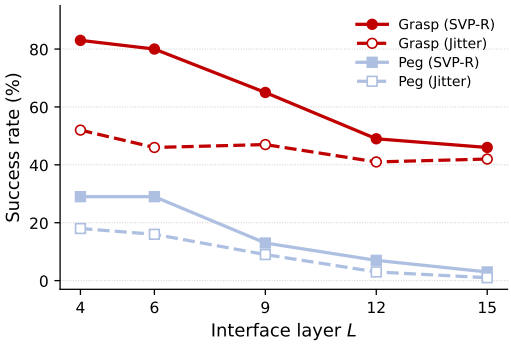



Figure 4: **Effect of the interface layer.** Success rate (%) on *Grasp* and *Peg* with SVP-R (solid) and under the *Jitter* intervention (dashed).

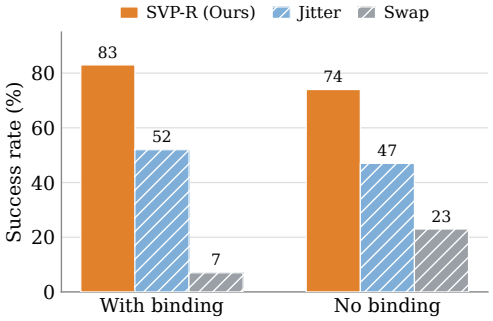



Figure 5: **Primitive interventions on Grasp-Part.** *Jitter* perturbs the predicted location, *Swap* replaces the referent identity. The AEB loss leaves tolerance to *Jitter* unchanged but cuts tolerance to *Swap* by 3.7×, binding identity rather than location.

meaning that perturbing the primitive barely changes the behavior once the interface sits deep in the network. The expert has stopped reading the primitive and fallen back to a shortcut. We therefore place the interface at $L{=}4$, where the reasoning trace still carries decodable evidence and the expert still has enough depth to consume it.

**Does the AEB loss drive the binding?** For fine-grained manipulation, we intervene on the primitive at inference and measure how the policy responds. *Jitter* perturbs the predicted location while keeping the referent, and *Swap* replaces the identity with that of a distractor part. As shown in Fig. 5, the AEB loss raises success on the MetaFine *Grasp* from 74% to 83%, and changes what the policy is sensitive to. Under *Jitter* both settings retain the same fraction of their performance, 62.7% with the loss and 63.5% without, so binding does not simply make the policy robust to spatial noise. Under *Swap* the gap is 8.4% against 31.1%, a 3.7× difference in tolerance to a wrong referent. Without the AEB loss the policy operates in a presence-gated regime, where the interface must be populated but its content is largely interchangeable; with it, the action follows the part the primitive names.

**Does REB actually instill the binding?** To verify the graph binding of SVP-R, we ablate its two binding terms in the recognition setting, where the REB reward combines answer correctness and localization with the edge-level $R_{\text{edge}}$ and the graph-level $B(\tau)$. Removing $R_{\text{edge}}$, which requires a claimed attribute to hold at the part the model localized, degrades accuracy by 1.88% on average. Removing $B(\tau)$, which requires the surviving evidence to separate the class from its confusers, degrades it by 1.39%. Both terms are needed on both datasets, and neither alone recovers the full reward, confirming that accuracy comes from evidence that is correctly bound and discriminative rather than from either property on its own.

Table 6: **REB reward ablation.** Recognition accuracy (%, ↑) when the edge-level and graph-level terms are removed. All arms use the 3B backbone.

| REWARD | *CUB* | *Cars* | *Avg* |
|---|---|---|---|
| w/o $R_{\text{edge}}$ | 83.47 | 86.99 | 85.23 |
| w/o $B(\tau)$ | 83.94 | 87.51 | 85.72 |
| **Full (REB)** | 84.87 | 89.34 | 87.11 |

## 5 CONCLUSIONS

Discriminative cues may be described in language by MLLMs, yet remain weakly connected to the visual evidence that should support the final prediction or action. We identified this as the fine-grained semantic grounding gap and presented SVP-R, a unified interface that lets MLLMs reason with semantic visual primitives binding part location, attributes, and action affordances. The primitives form an evidence graph that graph binding verifies at the node, edge, and graph levels, crediting only evidence that is localized, present, and discriminative. Experiments show improved fine-grained recognition, held-out generalization, localization precision, and manipulation success, and interventions on the primitive confirm that decisions follow the evidence rather than accompany it. Future work will use the primitives as grounded demonstrations for in-context learning of manipulation, and extend them to temporal and 3D grounding for longer-horizon interaction.

## AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure:

- *Help develop theoretical models or conceptual frameworks.* The tools helped formalize the method.
- *Design or provide feedback on research methodology or experiments.* We used them to discuss the design of the experiments and how to validate the method more comprehensively.
- *Implement methods.* The framework and the key components were written by the authors; the tools were used to debug the code and improve its readability.
- *Clean and reformat datasets.* An open-vocabulary detector and an MLLM assist in constructing the part-level annotations of Stanford Dogs and Stanford Cars (cf. App. A).
- *Interpret results.* The tools helped analyze the experimental results.

We have not used generative AI tools to generate synthetic data sets, to propose or refine hypotheses, or to assist with translation, and the remaining required-disclosure tasks, namely formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, and supporting qualitative and thematic data analysis, are not applicable.

Additionally, we used generative AI tools for the following tasks with recommended disclosure:

- *Edit a research paper to improve readability.* The initial drafts were written by the authors, and the tools were used to polish the wording and to shorten passages under the page limit.
- *Identify relevant literature.* We used them to search for related work, and every returned paper was checked by the authors before being cited.
- *Suggest a structure for a research paper.* We discussed the organization of the method and experiment sections with the tools.

We have reviewed all AI-assisted work, including verifying every AI-suggested reference against the original publication, and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REPRODUCIBILITY STATEMENT

The datasets, evaluation protocols, and the construction of part-level annotations are described in App. A. The adaptation and training configuration of each compared method is given in App. B, and the implementation of SVP-R for both tasks, including the reward terms, the primitive interface, and the training recipe, in App. C. All reported results follow the protocols stated in Sec. 4.1. Code is available at `https://anonymous.4open.science/r/SVP-R_MLLM`.

## REFERENCES

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

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proc. Conf. Robot Learn.*, pp. 17–40, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control. In *Proc. Robot. Sci. Syst.*, 2025b.

Meng Cao, Haoze Zhao, Can Zhang, Xiaojun Chang, Ian Reid, and Xiaodan Liang. GroundR1: Incentivizing grounded visual reasoning via reinforcement learning. *arXiv preprint arXiv:2505.20272*, 2025.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Xiwen Chen, Zelin Li, Zhiruo Zhou, Huiming Chen, Chenwei Wang, and Xiaojun Zhu. PointingVLA: Typed spatial grounding interfaces for vision-language-action manipulation. *arXiv preprint arXiv:2608.23138*, 2026.

Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, Zhizheng Zhang, and He Wang. GraspVLA: a grasping foundation model pre-trained on billion-scale synthetic action data. In *Proc. Conf. Robot Learn.*, pp. 1004–1029, 2025.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. GRIT: Teaching MLLMs to think with images. In *Advances in Neural Inf. Process. Syst.*, pp. 116522–116543, 2025.

Hulingxiao He, Geng Li, Zijun Geng, Jinglin Xu, and Yuxin Peng. Analyzing and boosting the power of fine-grained visual recognition for multi-modal large language models. In *Proc. Int. Conf. Learn. Representations*, pp. 89927–89944, 2025.

Hulingxiao He, Zijun Geng, and Yuxin Peng. Fine-R1: Make multi-modal LLMs excel in fine-grained visual recognition by chain-of-thought reasoning. In *Proc. Int. Conf. Learn. Representations*, pp. 68140–68158, 2026a.

Hulingxiao He, Zhi Tan, and Yuxin Peng. Taxonomy-aware representation alignment for hierarchical visual recognition with large multimodal models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 31124–31134, 2026b.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. Workshops*, 2011.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-R1: Reinforcement learning for enhanced embodied reasoning in robotics. In *Advances in Neural Inf. Process. Syst.*, pp. 161472–161507, 2025.

Jeonghwan Kim and Heng Ji. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 6187–6207, 2024.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proc. IEEE Int. Conf. Comp. Vis. Workshops*, pp. 554–561, 2013.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a diffusion foundation model for bimanual manipulation. In *Proc. Int. Conf. Learn. Representations*, pp. 29982–30009, 2025a.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. In *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 2034–2044, 2025b.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning. In *Advances in Neural Inf. Process. Syst.*, pp. 150977–151013, 2025.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *Advances in Neural Inf. Process. Syst.*, pp. 8612–8642, 2024.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes Wide Shut? exploring the visual shortcomings of multimodal LLMs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 9568–9578, 2024.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Hanqing Wang, Shaoyang Wang, Yiming Zhong, Zemin Yang, Jiamin Wang, Zhiqing Cui, Jiahao Yuan, Yifan Han, Mingyu Liu, and Yuexin Ma. Affordance-R1: Reinforcement learning for generalizable affordance reasoning in multimodal large language models. In *Proc. Conf. AAAI*, pp. 9738–9746, 2026a.

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

Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):8927–8948, 2022.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 13084–13094, 2024.

Xi Xiao, Chen Liu, Chih-Ting Liao, Yunbei Zhang, Qizhen Lan, Yuxiang Wei, Lin Zhao, Janet Wang, Jianyang Gu, Muchao Ye, Tianyang Wang, and Hao Xu. Staying VIGILant: Mitigating visual laziness via counterfactual visual alignment in MLLMs. In *Proc. Eur. Conf. Comp. Vis.*, pp. 287–305, 2026.

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Gengshen Zhang, Dawei Leng, and Yuhui Yin. FG-CLIP: Fine-grained visual and textual alignment. In *Proc. Int. Conf. Mach. Learn.*, pp. 68777–68793, 2025.

He-Yang Xu, Pengyuan Zhang, Zongyuan Ge, Xiaoshuai Hao, Serge Belongie, Xin Geng, Yuxin Peng, and Xiu-Shen Wei. Beyond binary success: A diagnostic meta-evaluation framework for fine-grained manipulation. *arXiv preprint arXiv:2605.19986*, 2026.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 19187–19197, 2023.

Yifan Yin, Zhengtao Han, Shivam Aarya, Shuhang Xu, Jianxin Wang, Jiawei Peng, Angtian Wang, Alan Yuille, and Tianmin Shu. PartInstruct: Part-level instruction following for fine-grained robot manipulation. In *Proc. Robot. Sci. Syst.*, 2025.

Hong-Tao Yu, Yuxin Peng, Serge Belongie, and Xiu-Shen Wei. Benchmarking large vision-language models on fine-grained image tasks: A comprehensive evaluation. In *Proc. Int. Conf. Learn. Representations*, pp. 44677–44700, 2026a.

Zhangyuan Yu, Wanran Sun, Guangjing Yang, Xiaohu Wu, and Qicheng Lao. Cfpo: Counterfactual policy optimization for multimodal reasoning. *arXiv preprint arXiv:2606.23206*, 2026b.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 1702–1713, 2025.

## APPENDIX OVERVIEW

This appendix provides details on the datasets, the annotation pipeline, the compared methods, and the implementation of SVP-R that are omitted from the main body due to space limitations:

- Section A describes the datasets, the evaluation protocols, and the construction of part-level annotations, covering the automatic pipeline for Stanford Dogs and Stanford Cars, and the simulator-derived supervision.
- Section B describes the compared methods and how each is adapted to fine-grained recognition and manipulation.
- Section C provides implementation details of SVP-R for both tasks, including the training recipe and the design of the primitive interface.

# A Datasets and Annotation Construction

## A.1 Datasets

We evaluate SVP-R on a diverse set of benchmarks spanning fine-grained visual recognition, robotic spatial localization, and fine-grained manipulation. The evaluation suite contains three standard fine-grained recognition datasets, two simulation platforms for robotic manipulation, and three real-robot tasks. Below we provide additional details of the datasets and task settings used in our experiments.

**CUB** CUB (Wah et al., 2011) contains 11,788 images of 200 bird species, split into 5,994 training and 5,794 test images. Every image is annotated with 15 part landmarks together with their visibility, and with 312 binary attributes organized as part–value pairs such as *crown color* or *bill shape*.

**Stanford Dogs** Stanford Dogs (Khosla et al., 2011) contains 20,580 images of 120 breeds sourced from ImageNet, with 12,000 training and 8,580 test images. The dataset provides only image-level class labels and object bounding boxes, so both the part-level boxes and the per-part attributes are constructed by the pipeline described in Sec. A.2. Breeds of the same group differ mainly in coat texture, ear shape, and muzzle length, and the large intra-class variation in pose and background makes part-level evidence particularly important.

**Stanford Cars** Stanford Cars (Krause et al., 2013) contains 16,185 images of 196 models, split into 8,144 training and 8,041 test images, with each class defined at the granularity of make, model, and year. As with Stanford Dogs, part-level annotations are constructed automatically. Discriminative evidence is concentrated in a few structural parts such as the grille, headlights, and roofline, which differ subtly across model years of the same make.

**MetaFine** MetaFine (Xu et al., 2026) is an evaluation platform designed for fine-grained manipulation, built on part-annotated articulated assets with a closed-set affordance vocabulary. In simulation, the pose, local 3D bounding box, segmentation mask, affordance label, and candidate grasp poses of every part are available as ground truth. We use MetaFine for two fine-grained tasks. For embodied spatial localization, we build a split in which a task-specific prompt elicits the box of the referred part and its contact point, described in Sec. A.2, and add perturbations at test time to form the *Held-out* setting. For fine-grained manipulation, we use *Grasp* (grasp part) and *Peg* (peg-in-hole): *Grasp* requires localizing the referred functional part and grasping it from above, while *Peg* additionally requires inserting a peg into a hole under tight tolerances, imposing a fine-grained constraint on the predicted pose.

**RoboTwin 2.0** RoboTwin 2.0 (Chen et al., 2025) is a bimanual manipulation benchmark with a scalable data generator and strong domain randomization. We select three tasks whose success depends on fine-grained part- or attribute-level discrimination: *Beat Block* requires striking the correct block with a hammer, *Click Bell* requires pressing a small functional part, and *Blocks Ranking RGB* requires ordering blocks by color, which demands fine-grained understanding of each block before reasoning about the placement order. We follow the official single-task protocol, in which each task is trained on 50 clean demonstrations and evaluated under the randomized (*Hard*) setting.

**Real-robot tasks** We use a single Franka arm with a parallel gripper, observed by a third-person and an in-hand Intel RealSense D435 camera. We evaluate three tasks that impose part-level constraints (cf. Fig. 6): *Grasp Handle* requires grasping an object by its handle rather than its body, so that the policy must resolve which part the instruction refers to before contact, while *Close Box* and *Grasp Cube* require acting on a specified region of the target object. Each task is trained on 100 teleoperated demonstrations.

**Evaluation protocols** For fine-grained recognition and spatial localization, the *Within* setting uses the official train/test splits over all classes. The *Held-out* setting trains on the first half of the classes and tests on the remaining half for recognition, so that memorized class identity is unavailable, and evaluates localization under out-of-domain lighting perturbations. Recognition is scored by whether the ground-truth class name appears in the answer span, and localization by the pixel error between the predicted and annotated targets. For manipulation, each simulated task is evaluated over 100

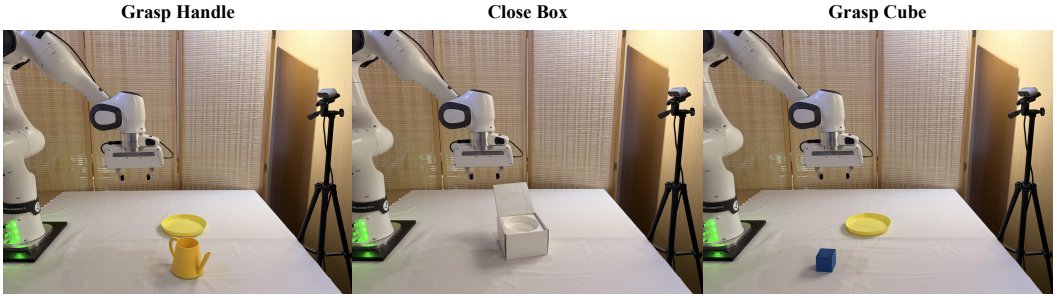



Figure 6: **Real-robot tasks.** Example observations and rollouts for *Grasp Handle*, *Close Box*, and *Grasp Cube*, each requiring the policy to act on a specified part of the target object.

trials, following the MetaFine protocol on MetaFine and the *Hard* setting on RoboTwin 2.0. Each real-robot task is evaluated over 25 trials, and success rates are reported per task.

### A.2 Construction of Part-level Annotations

Part-level supervision comes from three different sources. CUB provides part keypoints, which we turn into fixed-size windows; Stanford Dogs and Stanford Cars have no part labels and are annotated automatically; MetaFine and RoboTwin 2.0 expose exact part geometry from the simulator.

CUB provides 15 part keypoints per image with a visibility flag. We merge left–right pairs into a single semantic type, giving 12 part types, and turn each visible keypoint into a square window. The window carries no information about the true extent of the part, and only its centre is meaningful, which is why the node-level score $m_p$ is a Gaussian on the box centre rather than an IoU.

**Automatic annotation for Stanford Dogs and Stanford Cars** Neither dataset provides part-level labels, so we annotate them automatically with no human-corrected labels. The pipeline pairs a grounder, LocateAnything-3B (Wang et al., 2026b), with a judge, Qwen2.5-VL-32B-Instruct (Bai et al., 2025), and all decoding is greedy. Dogs are annotated with parts, *e.g.*, ear, head, eye, leg, and muzzle, and cars with wheel, door, mirror, headlight, and grille.

The grounder takes a short referring phrase for each part and emits a box in the same format used by SVP-R, `<ref>part</ref><box>···</box>`, with integer coordinates in $[0, 1000]$. For each part we compare two or three phrasings on a small sample and keep the best, for example `muzzle` against `dog snout`. The part *tail* was dropped from the dog vocabulary at this stage, as it was rarely localized correctly.

Each grounded box is then cropped with 8% padding, resized to 336 pixels, and verified by the judge in a single constrained call that also elicits its dominant colour:

```
This is a cropped region from a {obj} photo.
(1) Does it show the {obj}'s {part} (the whole part OR
a clear portion of it is fine)?  Answer 'no' ONLY if it
clearly shows a different body part, a person, or no {obj}
at all.
(2) Its main color, choose ONE from [black, white, gray,
silver, brown, tan, red, orange, yellow, green, blue, gold,
cream, pink, multicolor].
Reply EXACTLY: verify=<yes|no>; color=<one color>.
```

The reply is parsed by a strict regular expression, and a colour outside the closed palette is mapped to *uncertain*. A separate whole-image call separates a part that is genuinely absent from one the grounder missed:

```
Is the {obj}'s {part} clearly visible in this image?  Answer
yes or no.
```

A negative answer marks the part absent, while a positive answer without a usable box marks it for re-grounding, which retries every phrasing variant at full resolution under autoregressive decoding

and is followed by a second verification pass. Each box is then verified by the judge and filtered by simple geometric rules that reject degenerate or overly large boxes and duplicate detections.

Finally, one whole-image call per image asks the judge for closed-vocabulary attributes, such as ear shape, muzzle length and coat texture for dogs, and body type, grille shape and headlight shape for cars. Examples of the resulting annotations are shown in Fig. 7. The class–attribute profile $\mathbf{W}$ is obtained by aggregating these per-image attributes within each training class, and is built from training classes only under the *Held-out* protocol.

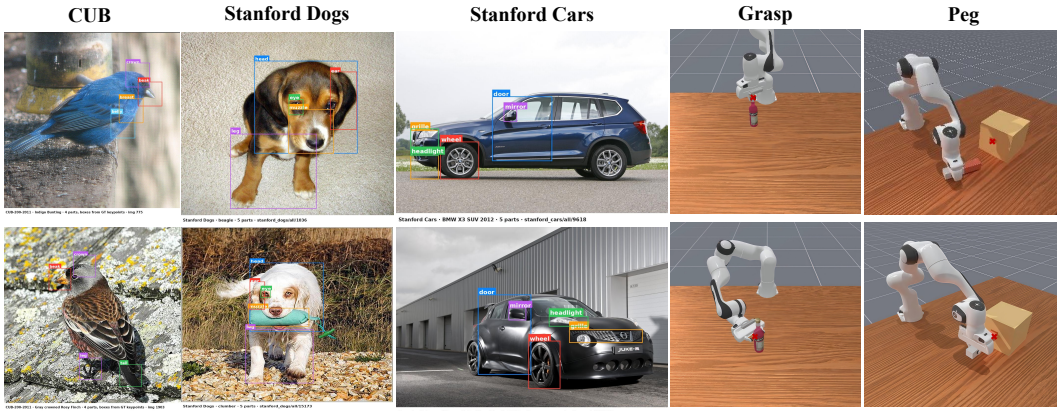



Figure 7: **Examples of part-level annotations.** Boxes on CUB are windows centred on the official keypoints; those on Stanford Dogs and Stanford Cars are produced by the automatic pipeline; those on the MetaFine *Grasp* and *Peg* tasks are projected from the simulator.

**Simulator-derived annotations** For MetaFine, the interactive part is read directly from the simulator and projected into the image, as visualized in Fig. 7. For manipulation, we replay each demonstration to recover the part location and the pre-grasp pose at gripper closure, which serve as the primitive supervision. The same procedure applies to RoboTwin 2.0, where the referred object or part and its interaction pose are obtained from the simulator state.

# B Implementation of Compared Methods

## B.1 Fine-grained Recognition

All recognition baselines are trained on the same data as SVP-R and evaluated under the same protocol. Methods with a supervised stage use 2 epochs with learning rate $1 \times 10^{-4}$, a cosine schedule and 3% warmup, and RL-based methods follow the same reinforcement learning protocol as SVP-R. Note that Visual-RFT and GRIT apply reinforcement learning without a supervised warm-up, following their original settings, which limits them on Stanford Cars, where model names cannot be recovered from the base model's prior alone. All methods are built on Qwen2.5-VL (Bai et al., 2025) with 3B and 7B backbones. For spatial localization, each method is prompted to output the location of the referred part, and the error is measured in pixels against the annotated target.

**Visual-CoT** Visual-CoT (Shao et al., 2024) first localizes the question-relevant region and then answers from a cropped view. We adapt it with a two-turn supervised format: the first turn predicts a bounding box of the object from the full image, and the second turn predicts the class label from the crop. At test time, the model crops with its own predicted box and falls back to the full image when no valid box is parsed. The method therefore performs object-level zooming, without a part–attribute structure or any check that the localized region supports the prediction.

**Visual-RFT** Visual-RFT (Liu et al., 2025b) applies verifiable-reward reinforcement learning directly to a base model. Following the original setting, we train it with GRPO without supervised fine-tuning, using answer correctness and output format as rewards. The reasoning content is not supervised and carries no explicit grounding.

**ViGoRL** ViGoRL (Sarch et al., 2025) anchors each reasoning step to image coordinates. We follow its SFT-then-RL pipeline: the supervised traces mention discriminative parts together with coordinates, and GRPO then optimizes answer correctness, output format, and a grounding term that checks whether the trace contains at least one visual anchor. The grounding signal thus rewards the presence of coordinates rather than their accuracy or their binding to attributes.

**GRIT** GRIT (Fan et al., 2025) interleaves bounding boxes into the reasoning trace, with the output format `<think>⋯ </think><rethink>⋯ </rethink><answer>⋯ </answer>` and boxes given as JSON coordinates normalized to $[0, 1000]$. We use the released system prompt verbatim and port its GRPO-GR reward, which sums five equally weighted terms: answer correctness, answer format, reasoning format, a grounding term granted when at least one box appears before `<rethink>`, and a repetition penalty. The only deviation is that the original answer score from GPT-4o is replaced by the ground-truth-in-answer check shared by all methods, which is verifiable and consistent across baselines; the counting term specific to TallyQA is removed. As in the original work, GRIT is trained with reinforcement learning from the base model without supervised fine-tuning.

### B.2 Fine-grained Manipulation

On MetaFine and the real-robot tasks, RDT (Liu et al., 2025a), $\pi_0$ (Black et al., 2025b), and $\pi_{0.5}$ (Black et al., 2025a) are trained by us under the same data and schedule as SVP-R. On RoboTwin 2.0, the results of RDT and $\pi_0$ are taken from the official leaderboard under the single-task protocol and the *Hard* evaluation setting, while $\pi_{0.5}$ is trained by us.

## C Implementation of SVP-R

### C.1 Fine-grained Recognition

Both the 3B and 7B variants share the same recipe. We first fine-tune on grounded traces for 2 epochs with learning rate $1 \times 10^{-4}$, a cosine schedule and 3% warmup, keeping the vision encoder frozen, then optimize REB with GRPO. Box coordinates are expressed on a $[0, 1000]$ grid, the localization bandwidth is $\sigma = 0.08$ in normalized coordinates. A trace passes the format gate only if it contains the reasoning and answer spans and at least two well-formed primitives. Training uses 8 NVIDIA H100 GPUs.

### C.2 Fine-grained Manipulation

SVP-R builds on $\pi_0$ (Black et al., 2025b) initialized from the released base weights, in which a PaliGemma VLM and an action expert share a transformer with joint attention and actions are predicted in chunks of 50 steps. The primitive token is appended to the prefix and read at layer $L = 4$ of the 18-layer VLM: the heatmap comes from the attention of the primitive token to the $16 \times 16$ base-camera image tokens, and the part identity and the end-effector pose are read by single linear heads. The decoded primitive is encoded into a goal vector by a two-layer MLP and injected back at the same layer, as a residual on the state token of the action expert and as a marker on the image tokens, both scaled relative to the norm of the token they modify. The policy is fully fine-tuned with AdamW at learning rate $1 \times 10^{-4}$ for 30k steps, with an effective batch size of 32 on 4 NVIDIA H100 GPUs.